



Beta

EDB Postgres™ Advanced Server Installation Guide

EDB Postgres™ Advanced Server 11

August 1, 2018

EDB Postgres™ Advanced Server Installation Guide
by EnterpriseDB® Corporation
Copyright © 2009 - 2018 EnterpriseDB Corporation. All rights reserved.

Beta

EnterpriseDB Corporation, 34 Crosby Drive Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

Table of Contents

1	Introduction.....	5
1.1	Typographical Conventions Used in this Guide.....	6
2	Requirements Overview.....	7
2.1	Supported Platforms.....	7
2.2	Linux Installation Prerequisites.....	8
3	Using a Package Manager to Install Advanced Server.....	10
3.1	Installing an RPM Package.....	11
3.1.1	Advanced Server RPM Installers.....	13
3.1.2	Performing a Minor Version Update of an RPM Installation.....	16
3.2	Configuring a Package Installation.....	17
3.2.1	Creating a Database Cluster and Starting the Service.....	17
3.2.2	Using a Service Configuration File on CentOS or Redhat 6.x.....	18
3.2.2.1	Specifying INITDBOPTS Options.....	19
3.2.3	Modifying the Data Directory Location on CentOS or Redhat 7.x.....	21
3.3	Starting Multiple Postmasters with Different Clusters.....	22
3.4	Creating an Advanced Server Repository on an Isolated Network.....	24
4	Installing Advanced Server with the Interactive Installer.....	26
4.1	Windows Installation Prerequisites.....	28
4.2	Performing a Graphical Installation on Windows.....	29
4.3	Invoking the Graphical Installer from the Command Line.....	47
4.3.1	Performing an Unattended Installation.....	48
4.3.2	Performing an Installation with Limited Privileges.....	49
4.3.3	Reference - Command Line Options.....	57
4.4	Using StackBuilder Plus.....	64
5	Managing an Advanced Server Installation.....	72
5.1	Starting and Stopping Advanced Server and Supporting Components.....	72
5.2	Controlling a Service on Linux.....	73
5.2.1	Controlling a Service on CentOS or RHEL 7.x.....	73
5.2.2	Controlling a Service on CentOS or RHEL 6.x.....	74
5.2.3	Using pg_ctl to Control Advanced Server.....	75
5.2.4	Configuring Component Services to AutoStart at System Reboot.....	76
5.3	Controlling a Service on Windows.....	77
5.3.1	Controlling Server Startup Behavior on Windows.....	78
6	Configuring Advanced Server.....	80
6.1	Modifying the postgresql.conf File.....	81
6.2	Modifying the pg_hba.conf File.....	82
6.3	Setting Advanced Server Environment Variables.....	83

6.4	Connecting to Advanced Server with psql	84
6.5	Connecting to Advanced Server with the pgAdmin 4 Client	85
7	Upgrading an Installation With pg_upgrade	87
7.1	Performing an Upgrade - Overview	88
7.1.1	Linking versus Copying.....	89
7.2	Invoking pg_upgrade	90
7.2.1	Command Line Options - Reference	92
7.3	Upgrading to Advanced Server 11 – Step-by-Step	95
7.4	Upgrading a pgAgent Installation	104
7.5	pg_upgrade Troubleshooting.....	105
7.5.1	Upgrade Error - There seems to be a postmaster servicing the cluster	105
7.5.2	Upgrade Error - fe_sendauth: no password supplied	105
7.5.3	Upgrade Error - New cluster is not empty; exiting	105
7.5.4	Upgrade Error - Failed to load library	105
7.6	Reverting to the Old Cluster.....	106
8	Uninstalling Advanced Server	107
8.1	Uninstalling an RPM Package.....	107
8.2	Using Advanced Server Uninstallers at the Command Line	108

1 Introduction

The EDB Postgres Advanced Server Installation Guide is a comprehensive guide to installing EDB Postgres Advanced Server (Advanced Server). In this guide you will find detailed information about:

- Software prerequisites for Advanced Server 11 installation.
- Using a package manager to install and update Advanced Server and its supporting components or utilities on a Linux host.
- Graphical installation options available through the interactive setup wizard on Windows.
- Managing an Advanced Server installation.
- Configuring an Advanced Server package installation.
- Using `pg_upgrade` to upgrade from an earlier version of Advanced Server to Advanced Server 11.
- Uninstalling Advanced Server and its components.

1.1 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions, a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically in the sentence that defines it for the first time.
- Fixed-width (mono-spaced) font is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *Italic fixed-width font* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM table_name;`
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [] denote that one or none of the enclosed terms may be substituted. For example, [a | b] means choose one of “a” or “b” or neither of the two.
- Braces { } denote that exactly one of the enclosed alternatives must be specified. For example, { a | b } means exactly one of “a” or “b” must be specified.
- Ellipses ... denote that the preceding term may be repeated. For example, [a | b] ... means that you may have the sequence, “b a a b a”.

2 Requirements Overview

The following sections detail the supported platforms and installation requirements for EDB Postgres Advanced Server 11.

2.1 Supported Platforms

The Advanced Server 11 RPM packages are supported on the following platforms:

64 bit Linux:

- Red Hat Enterprise Linux (x86_64) 6.x and 7.x
- CentOS (x86_64) 6.x and 7.x
- OEL Linux 6.x and 7.x
- PPC-LE 8 running RHEL or CentOS 7.x

The Advanced Server 11 graphical installers are supported on the 64 bit Windows platforms:

- Windows Server 2016
- Windows Server 2012 R2

2.2 Linux Installation Prerequisites

You can use an RPM package to install Advanced Server and supporting components on a Linux host.

Installing EPEL

Before installing Advanced Server, you may be required to install the `EPEL` (Extra Packages for Enterprise Linux) repository. You can use `yum` to install the package:

```
yum install epel-release
```

If `yum` cannot access a repository that contains `epel-release`, you will get an error message:

```
No package epel available.  
Error: Nothing to do
```

If you receive this error, you can download the `EPEL rpm` package, and install it manually. To manually install `EPEL`, download the `rpm` package, assume superuser privileges, navigate into the directory that contains the package, and install `EPEL` with the command:

```
yum install epel-release
```

For more information about installing `EPEL`, visit:

https://fedoraproject.org/wiki/EPEL#How_can_I_use_these_extra_packages.3F

Linux-specific Software Requirements

You must install `xterm`, `konsole`, or `gnome-terminal` before executing any console-based program installed by the Advanced Server installer. Without a console program, you will not be able to access Advanced Server configuration files through menu selections.

SELinux Permissions

Before invoking an installer on a system that is running SELinux, you must set SELinux to `permissive` mode.

The following example works on Redhat Enterprise Linux, Fedora Core or CentOS distributions. Use comparable commands that are compatible with your Linux distribution to set SELinux to `permissive` mode during installation and return it to `enforcing` mode when installation is complete.

Before installing Advanced Server, set SELinux to `permissive` mode with the command:

```
# setenforce Permissive
```

When the installation is complete, return SELinux to `enforcing` mode with the command:

```
# setenforce Enforcing
```

Migration Toolkit or EDB*Plus Installation Pre-requisites

Before using an RPM or StackBuilder Plus to install Migration Toolkit or EDB*Plus, you must first install Java (version 1.8 or later). On a Linux system, you can use the yum package manager to install Java. Open a terminal window, assume superuser privileges, and enter:

```
# yum install java
```

Follow the onscreen instructions to complete the installation.

3 Using a Package Manager to Install Advanced Server

You can use the yum package manager to install Advanced Server or Advanced Server supporting components. yum will attempt to satisfy package dependencies as it installs a package, but requires access to the Advanced Server repositories. If your system does not have access to a repository via the Internet, you can use RPM to install a package or create a local repository, but you may be required to manually satisfy package dependencies.

You must have credentials for the EnterpriseDB repository to install Advanced Server; to submit a request for credentials, click an Access Repository button on the following page:

<https://www.enterprisedb.com/advanced-downloads>

The Advanced Server RPM installs Advanced Server and the core components of the database server. For a list of the RPM installers available for Advanced Server and supporting components, see Section [3.1](#).

Installing the server package creates a database superuser named `enterprisedb`. The user is assigned a user ID (UID) and a group ID (GID) of 26. The user has no default password; use the `passwd` command to assign a password for the user. The default shell for the user is `bash`, and the user's home directory is `/var/lib/edb/as11`.

By default, Advanced Server logging is configured to write files to the `log` subdirectory of the `data` directory, rotating the files each day and retaining one week of log entries. You can customize the logging behavior of the server by modifying the `postgresql.conf` file; for more information about modifying the `postgresql.conf` file, please see Section [6.1](#).

The RPM installers place Advanced Server components in the directories listed in the table below:

EDBAS Component	Path to Installation Directory
Executables	<code>/usr/edb/as11/bin</code>
Libraries	<code>/usr/edb/as11/lib</code> or <code>/usr/edb/as11/lib64</code>
Documentation	<code>/usr/edb/as11/share/doc</code>
Contrib	<code>/usr/edb/as11/share/contrib</code>
Data	<code>/var/lib/edb/as11/data</code>
Backup area	<code>/var/lib/edb/as11/backups</code>
Templates	<code>/usr/edb/as11/share</code>
Procedural Languages	<code>/usr/edb/as11/lib</code> or <code>/usr/edb/as11/lib64</code>
Development Headers	<code>/usr/edb/as11/include</code>
Shared data	<code>/usr/edb/as11/share</code>
Regression tests	<code>/usr/edb/as11/lib/pgxs/src/test/regress</code>
SGML Documentation	<code>/usr/edb/as11/share/doc</code>

3.1 Installing an RPM Package

Before installing Advanced Server or a supporting component via an RPM package over the web, you must create the repository configuration file (`edb-repo`). The repository configuration file contains connection and authentication information for the EnterpriseDB repository. To prepare your system to perform an RPM installation:

1. Assume superuser privileges and use either `rpm` or `yum` to create the repository configuration file:

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

or

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

2. Use your choice of editor to modify the repository configuration file, enabling each repository from which you will install packages, and providing your credentials. The repository configuration file is named `edb.repo`; it resides in `/etc/yum.repos.d`.

To enable a repository, change the value of the `enabled` parameter to 1 and replace the user name and password placeholders in the `baseurl` specification with your user name and the repository password.

Installing Advanced Server

To install Advanced Server 11, enable and provide connection credentials for the `edbas11` repository and the `enterprisedb-dependencies` repository:

```
[edbas11]
name=EnterpriseDB Advanced Server 11 $releasever -
$basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/
11/redhat/rhel-$releasever-$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

```
[enterprisedb-dependencies]
name=EnterpriseDB Dependencies $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/
dependencies/redhat/rhel-$releasever-$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Installing supporting components

The repository configuration file also contains an entry for the `enterprisedb-tools` repository. Enable the `[enterprisedb-tools]` and the `[enterprisedb-dependencies]` entries in the `edb.repo` file when installing Advanced Server supporting components:

```
[enterprisedb-tools]
name=EnterpriseDB Tools $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/
tools/redhat/rhel-$releasever-$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

3. After modifying applicable entries in the repository configuration file, save the configuration file and exit the editor.

Then, you can use the `yum install` command to install Advanced Server or supporting components. For example, to install the server and its core components, use the command:

```
yum install edb-as11-server
```

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 a `y`, and press `Return` to continue.

After installing Advanced Server, you must configure the installation; see Section [3.2](#), *Configuring a Package Installation*, for details.

For information about the available packages, see Section [3.1.1](#).

During the installation, `yum` may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

3.1.1 Advanced Server RPM Installers

The tables that follow list the packages that are available from EnterpriseDB. Please note that you can also use the `yum search` command to access a list of the packages that are currently available from your configured repository. To use the `yum search` command, open a command line, assume root privileges, and enter:

```
yum search package
```

Where *package* is the search term that specifies the name (or partial name) of a package. The repository search will return a list of available packages that include the specified search term.

Please note: the available package list is subject to change.

Package Name	Package Installs
edb-as11-server	This package installs core components of the Advanced Server database server.
edb-as11-server-client	The <code>edb-as11-server-client</code> package contains client programs and utilities that you can use to access and manage Advanced Server.
edb-as11-server-contrib	The <code>edb-as11-contrib</code> package installs contributed tools and utilities that are distributed with Advanced Server. Files for these modules are installed in: Documentation: <code>/usr/edb/as11/share/doc</code> Loadable modules: <code>/usr/edb/as11/lib</code> Binaries: <code>/usr/edb/as11/bin</code>
edb-as11-server-core	The <code>edb-as11-server-core</code> package includes the programs needed to create the core functionality behind the Advanced Server database.
edb-as11-server-devel	The <code>edb-as11-server-devel</code> package contains the header files and libraries needed to compile C or C++ applications that directly interact with an Advanced Server server and the <code>ecpg</code> or <code>ecpgPlus C</code> preprocessor.
edb-as11-server-docs	The <code>edb-as11-server-docs</code> package installs the readme file.
edb-as11-server-edb-modules	This package installs supporting modules for Advanced Server
edb-as11-server-indexadvisor	This package installs Advanced Server's Index Advisor feature. The Index Advisor utility helps determine which columns you should index to improve performance in a given workload.
edb-as11-server-libs	The <code>edb-as11-server-libs</code> package provides the essential shared libraries for any Advanced Server client program or interface.
edb-as11-server-llvmjit	This package installs PostgreSQL supporting modules.
edb-as11-server-pldebugger	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
edb-as11-server-plperl	The <code>edb-as11-server-plperl</code> package installs the PL/Perl procedural language for Advanced Server. Please note that the <code>edb-as11-server-plperl</code> package is dependent on the platform-supplied version of Perl.
edb-as11-server-plpython	The <code>edb-as11-server-plpython</code> package installs the PL/Python

Package Name	Package Installs
	procedural language for Advanced Server. Please note that the <code>edb-as11-server-plpython</code> package is dependent on the platform-supplied version of Python.
<code>edb-as11-server-pltcl</code>	The <code>edb-as11-pltcl</code> package installs the PL/Tcl procedural language for Advanced Server. Please note that the <code>edb-as11-pltcl</code> package is dependent on the platform-supplied version of TCL.
<code>edb-as11-server-sqlprofiler</code>	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
<code>edb-as11-server-sqlprotect</code>	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
<code>edb-as11-server-sslutils</code>	This package installs functionality that provides SSL support.
<code>edb-as11-server-cloneschema</code>	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
<code>edb-as11-server-parallel-clone</code>	This package installs functionality that supports the EDB Clone Schema extension.
<code>edb-as11-edbplus</code>	The <code>edb-edbplus</code> package contains the files required to install the EDB*Plus command line client. EDB*Plus commands are compatible with Oracle's SQL*Plus.
<code>edb-as11-pgagent</code>	This package installs pgAgent; pgAgent is a job scheduler for Advanced Server. Before installing this package, you must install EPEL; for detailed information about installing EPEL, see Section 2.2 .
<code>edb-icache</code>	This package installs InfiniteCache.
<code>edb-icache-devel</code>	This is a supporting package for InfiniteCache.
<code>edb-as11-pgsnmpd</code>	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
<code>edb-as11-pljava</code>	This package installs PL/Java, providing access to Java stored procedures, triggers and functions via the JDBC interface.
<code>edb-as11-pgpool35-extensions</code>	This package creates pgPool extensions required by the server.
<code>libevent-edb</code> <code>libiconv-edb</code> <code>libicu-edb</code>	These packages contain supporting library files.

The following table lists the packages for Advanced Server 11 supporting components that are stored in the `Tools` repository:

Package Name	Package Installs
<code>edb-pgpool35</code>	This package contains the pgPool-II installer. pgPool provides connection pooling for Advanced Server installations.
<code>edb-pgpool35-devel</code>	This package contains the pgPool-II headers and libraries.
<code>edb-jdbc</code>	The <code>edb-jdbc</code> package includes the .jar files needed for Java programs to access an Advanced Server database.
<code>edb-migrationtoolkit</code>	The <code>edb-migrationtoolkit</code> package installs Migration Toolkit, facilitating migration to an Advanced Server database from Oracle, PostgreSQL, MySQL, Sybase and SQL Server.
<code>edb-oci</code>	The <code>edb-oci</code> package installs the EnterpriseDB Open Client library, allowing applications that use the Oracle Call Interface API to connect to an Advanced Server database.
<code>edb-oci-devel</code>	This package installs the OCI include files; install this package if you are developing C/C++ applications that require these files.

Package Name	Package Installs
edb-odbc	This package installs the driver needed for applications to access an Advanced Server system via ODBC.
edb-odbc-devel	This package installs the ODBC include files; install this package if you are developing C/C++ applications that require these files.
edb-pgbouncer17	This package contains PgBouncer (a lightweight connection pooler). This package requires the libevent package.
edb-xdb	This package contains the xDB installer; xDB provides asynchronous cross-database replication. For more information, visit http://www.enterprisedb.com/faq-xdb-multi-master
edb-xdb-console	This package provides support for xDB.
edb-xdb-libs	This package provides support for xDB.
edb-xdb-publisher	This package provides support for xDB.
edb-xdb-subscriber	This package provides support for xDB.

Please Note: Available packages are subject to change.

3.1.2 Performing a Minor Version Update of an RPM Installation

If you used an RPM package to install Advanced Server or its supporting components, you can use yum to perform a minor version upgrade to a more recent version. To review a list of the package updates that are available for your system, open a command line, assume root privileges, and enter the command:

```
yum check-update package_name
```

Where *package_name* is the search term for which you wish to search for updates. Please note that you can include wild-card values in the search term. To use yum update to install an updated package, use the command:

```
yum update package_name
```

Where *package_name* is the name of the package you wish to update. Include wild-card values in the update command to update multiple related packages with a single command. For example, use the following command:

```
yum update edb*
```

To update all packages whose names include the expression `edb`.

Please note that the `yum update` command will only perform an update between minor releases; to update between major releases, you should use `pg_upgrade`. For more information about using `pg_upgrade`, see Section 7.

For more information about using yum commands and options, enter `yum --help` on your command line, or visit:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-yum.html

3.2 Configuring a Package Installation

The packages that install the database server component create a service configuration file (on version 6.x hosts) or unit file (on version 7.x hosts), and service startup scripts.

The PostgreSQL `initdb` command creates a database cluster. If you are using an RPM package to install Advanced Server, you must manually configure the service and invoke `initdb` to create your cluster.

When invoking `initdb`, you can:

- Specify environment options on the command line.
- Include the `service` command on RHEL or CentOS 6.x, and use service configuration file to configure the environment.
- Include the `systemd` service manager on RHEL or CentOS 7.x use the service configuration file to configure the environment.

3.2.1 Creating a Database Cluster and Starting the Service

After specifying any options in the service configuration file, you must create the database cluster and start the service; these steps are platform specific.

On RHEL or CentOS 6.x

To create a database cluster in the `PGDATA` directory that listens on the port specified by the `PGPORT` specified in the service configuration file described in [Section 3.2.2](#), assume root privileges, and invoke the `service` script:

```
service edb-as-11 initdb
```

You can also assign a locale to the cluster when invoking `initdb`. By default, `initdb` will use the value specified by the `$LANG` operating system variable, but if you append a preferred locale when invoking the script, the cluster will use the alternate value. For example, to create a database cluster that uses simplified Chinese, invoke the command:

```
service edb-as-11 initdb zh_CH.UTF-8
```

After creating a database cluster, start the database server with the command:

```
service edb-as-11 start
```

On RHEL or CentOS 7.x

To invoke `initdb` on a RHEL or CentOS 7.x system, with the options specified in the service configuration file, assume the identity of the operating system superuser:

```
su - root
```

Then, invoke `initdb`:

```
/usr/edb/as11/bin/edb-as-11-setup initdb
```

After creating the cluster, use `systemctl` to start, stop, or restart the service:

```
systemctl { start | stop | restart } edb-as-11
```

For more information about using the service command, please see Section [5.2](#).

3.2.2 Using a Service Configuration File on CentOS or Redhat 6.x

On a CentOS or RedHat version 6.x host, the RPM installer creates a service configuration file named `edb-as-11.sysconfig` in `/etc/sysconfig/edb/as11` (see Figure 3.1). Please note that options specified in the service configuration file are only enforced if `initdb` is invoked via the `service` command; if you manually invoke `initdb` (at the command line), you must specify the other options (such as the location of the data directory and installation mode) on the command line.

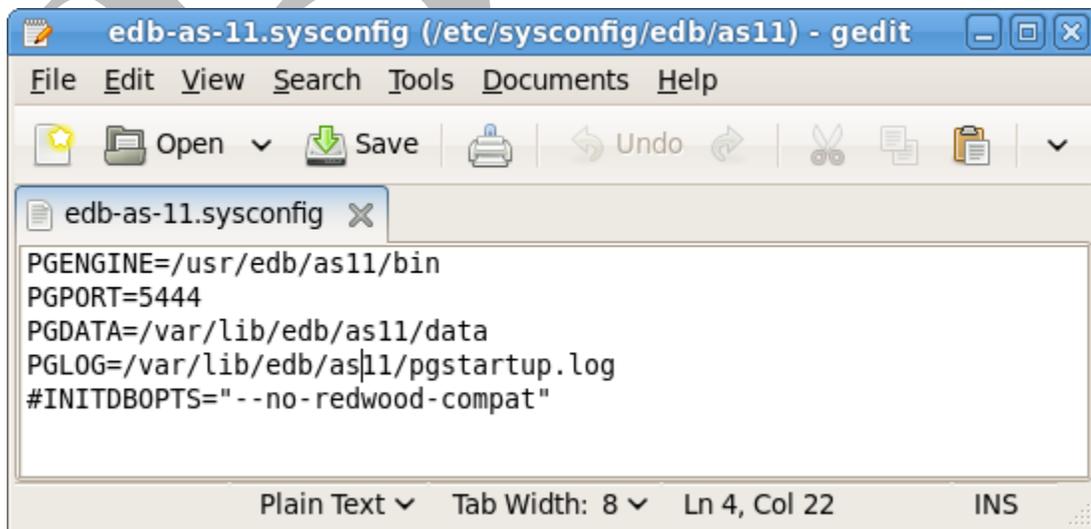


Figure 3.1 -The Advanced Server service configuration file.

The file contains the following environment variables:

- `PGENGINE` specifies the location of the engine and utility executable files.
- `PGPORT` specifies the listener port for the database server.
- `PGDATA` specifies the path to the data directory.
- `PGLOG` specifies the location of the log file to which the server writes startup information.
- Use `INITDBOPTS` to specify any `initdb` option or options that you wish to apply to the new cluster.

You can modify the `edb-as-11.sysconfig` file before using the service command to invoke the `startup` script to change the listener port, data directory location, startup log location or installation mode. If you plan to create more than one instance on the same system, you may wish to copy the `edb-as-11.sysconfig` file (and the associated `edb-as-11` startup script) and modify the file contents for each additional instance that resides on the same host.

3.2.2.1 Specifying INITDBOPTS Options

You can use the `INITDBOPTS` variable to specify your cluster configuration preferences. By default, the `INITDBOPTS` variable is commented out in the service configuration file; unless modified, when you run the service startup script, the new cluster will be created in a mode compatible with Oracle databases. Clusters created in this mode will contain a database named `edb`, and have a database superuser named `enterprisedb`.

To create a new cluster in PostgreSQL mode, remove the pound sign (`#`) in front of the `INITDBOPTS` variable, enabling the `"--no-redwood-compat"` option. Clusters created in PostgreSQL mode will contain a database named `postgres`, and have a database superuser named `postgres`.

You may also specify multiple `initdb` options. For example, the following statement:

```
INITDBOPTS="--no-redwood-compat -U alice --locale=en_US.UTF-8"
```

Creates a database cluster (without compatibility features for Oracle) that contains a database named `postgres` that is owned by a user named `alice`; the cluster uses UTF-8 encoding.

In addition to the cluster configuration options documented in the PostgreSQL core documentation, Advanced Server supports the following `initdb` options:

```
--no-redwood-compat
```

Include the `--no-redwood-compat` keywords to instruct the server to create the cluster in PostgreSQL mode. When the cluster is created in PostgreSQL

mode, the name of the database superuser will be `postgres`, the name of the default database will be `postgres`, and Advanced Server's features compatible with Oracle databases will not be available to the cluster.

`--redwood-like`

Include the `--redwood-like` keywords to instruct the server to use an escape character (an empty string (' ')) following the `LIKE` (or PostgreSQL-compatible `ILIKE`) operator in a SQL statement that is compatible with Oracle syntax.

`--icu-short-form`

Include the `--icu-short-form` keywords to create a cluster that uses a default ICU (International Components for Unicode) collation for all databases in the cluster. For more information about Unicode collations, please refer to the *EDB Postgres Advanced Server Guide* available at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

For more information about using `initdb`, and the available cluster configuration options, see the PostgreSQL Core Documentation available at:

<https://www.postgresql.org/docs/11/static/app-initdb.html>

You can also view online help for `initdb` by assuming superuser privileges and entering:

```
/path_to_initdb_installation_directory/initdb --help
```

Where `path_to_initdb_installation_directory` specifies the location of the `initdb` binary file.

3.2.3 Modifying the Data Directory Location on CentOS or Redhat 7.x

On a CentOS or RedHat version 7.x host, the unit file is named `edb-as-11.service` and resides in `/usr/lib/systemd/system`. The unit file contains references to the location of the Advanced Server data directory. You should avoid making any modifications directly to the unit file because it may be overwritten during package upgrades.

By default, data files reside under `/var/lib/edb/as11/data` directory. To use a data directory that resides in a non-default location, create a copy of the unit file under the `/etc` directory:

```
cp /usr/lib/systemd/system/edb-as-11.service
   /etc/systemd/system/
```

After copying the unit file to the new location, modify the service file (`/etc/systemd/system/edb-as-11.service`) with your editor of choice, correcting any required paths.

Then, use the following command to reload `systemd`, updating the modified service scripts:

```
systemctl daemon-reload
```

Then, start the Advanced Server service with the following command:

```
systemctl start edb-as-11
```

3.3 Starting Multiple Postmasters with Different Clusters

You can configure Advanced Server to use multiple postmasters, each with its own database cluster. The steps required are version specific to the Linux host.

On RHEL or CentOS 6.x

The `edb-as11-server-core` RPM contains a script that starts the Advanced Server instance. The script can be copied, allowing you to run multiple services, with unique data directories and that monitor different ports. You must have `root` access to invoke or modify the script.

The example that follows creates a second instance on an Advanced Server host; the secondary instance is named `secondary`:

1. Create a hard link in `/etc/rc.d/init.d` (or equivalent location) to the `edb-as-11` service (named `secondary-edb-as-11`):

```
ln edb-as-11 secondary-edb-as-11
```

Be sure to pick a name that is not already used in `/etc/rc.d/init.d`.

2. Create a file in `/etc/sysconfig/edb/as11/` named `secondary-edb-as-11`. This file is where you would typically define `PGDATA` and `PGOPTS`. Since `$PGDATA/postgresql.conf` will override many of these settings (except `PGDATA`) you might notice unexpected results on startup.
3. Create the target `PGDATA` directory.
4. Assume the identity of the Advanced Server database superuser (`enterprisedb`) and invoke `initdb` on the target `PGDATA`. For information about using `initdb`, please see the PostgreSQL Core Documentation available at:

<https://www.postgresql.org/docs/11/static/app-initdb.html>

5. Edit the `postgresql.conf` file to specify the port, address, TCP/IP settings, etc. for the `secondary` instance.
6. Start the postmaster with the following command:

```
service secondary-edb-as-11 start
```

On RHEL or CentOS 7.x

The `edb-as11-server-core` RPM for version 7.x contains a unit file that starts the Advanced Server instance. The file allows you to start multiple services, with unique data directories and that monitor different ports. You must have `root` access to invoke or modify the script.

The example that follows creates an Advanced Server installation with two instances; the secondary instance is named `secondary`:

1. Make a copy of the default file with the new name. As noted at the top of the file, all modifications must reside under `/etc`. You must pick a name that is not already used in `/etc/systemd/system`.

```
cp /usr/lib/systemd/system/edb-as-11.service
/etc/systemd/system/secondary-edb-as-11.service
```

2. Edit the file, changing `PGDATA` to point to the new data directory that you will create the cluster against.
3. Create the target `PGDATA` with user `enterprisedb`.
4. Run `initdb`, specifying the setup script:

```
/usr/edb/as11/bin/edb-as-11-setup initdb secondary-edb-as-11
```

5. Edit the `postgresql.conf` file for the new instance, specifying the port, the IP address, TCP/IP settings, etc.
6. Make sure that new cluster runs after a reboot:

```
systemctl enable secondary-edb-as-11
```

7. Start the second cluster with the following command:

```
systemctl start secondary-edb-as-11
```

3.4 **Creating an Advanced Server Repository on an Isolated Network**

You can create a local repository to act as a host for the Advanced Server RPM packages if the server on which you wish to install Advanced Server (or supporting components) cannot directly access the EnterpriseDB repository. Please note that this is a high-level listing of the steps requires; you will need to modify the process for your individual network.

To create and use a local repository, you must:

1. Use yum to install the `epel-release`, `yum-utils`, and `createrepo` packages:

```
yum install epel-release
yum install yum-utils
yum install createrepo
```

2. Create a directory in which to store the repository:

```
mkdir /srv/repos
```

3. Copy the RPM installation packages to your local repository. You can download the individual packages or use a tarball to populate the repository.
4. Sync the RPM packages and create the repository.

```
reposync -r edbas11 -p /srv/repos
createrepo /srv/repos
```

5. Install your preferred webserver on the host that will act as your local repository, and ensure that the repository directory is accessible to the other servers on your network.
6. On each isolated database server, configure yum to pull updates from the mirrored repository on your local network. For example, you might create a repository configuration file called `/etc/yum.repos.d/edb-repo` with connection information that specifies:

```
[edbas11]
name=EnterpriseDB Advanced Server 11
baseurl=https://yum.your_domain.com/edbas11
enabled=1
gpgcheck=0
```

After specifying the location and connection information for your local repository, you can use yum commands to install Advanced Server and its supporting components on the isolated servers. For example:

```
yum install edb-as11-server
```

For more information about creating a local yum repository, visit:

<https://wiki.centos.org/HowTos/CreateLocalRepos>

Beta

4 Installing Advanced Server with the Interactive Installer

You can use the Advanced Server interactive installer to install Advanced Server on Windows. The interactive installer is available from the EnterpriseDB website at:

<https://www.enterprisedb.com/software-downloads-postgres>

You can invoke the graphical installer in different installation modes to perform an Advanced Server installation:

- For information about using the graphical installer, see Section [4.2](#).
- For information about performing an unattended installation, see Section [4.3.1](#).
- For information about performing an installation with limited privileges, see Section [4.3.2](#).
- For information about the command line options you can include when invoking the installer, see Section [4.3.3](#).

During the installation, the graphical installer copies a number of temporary files to the location specified by the `TEMP` environment variable. You can optionally specify an alternate location for the temporary files by modifying the `TEMP` environment variable.

If invoking the installer from the command line, you can set the value of the variable on the command line. Use the command:

```
SET TEMP=temp_file_location
```

Where `temp_file_location` specifies the alternate location for the temporary files.

Please Note: If you are invoking the installer to perform a system upgrade, the installer will preserve the configuration options specified during the previous installation.

Setting Cluster Preferences with the Graphical Installer

During an installation, the graphical installer invokes the PostgreSQL `initdb` utility to initialize a cluster. If you are using the graphical installer, you can use the `INITDBOPTS` environment variable to specify your `initdb` preferences. Before invoking the graphical installer, set the value of `INITDBOPTS` at the command line, specifying one or more cluster options. For example:

```
SET INITDBOPTS= -k -E=UTF-8
```

If you specify values in `INITDBOPTS` that are also provided by the installer (such as the `-D` option, which specifies the installation directory), the value specified in the graphical installer will supersede the value if specified in `INITDBOPTS`.

For more information about using `initdb` cluster configuration options, see the PostgreSQL Core Documentation available at:

<https://www.postgresql.org/docs/11/static/app-initdb.html>

In addition to the cluster configuration options documented in the PostgreSQL core documentation, Advanced Server supports the following `initdb` options:

`--no-redwood-compat`

Include the `--no-redwood-compat` keywords to instruct the server to create the cluster in PostgreSQL mode. When the cluster is created in PostgreSQL mode, the name of the database superuser will be `postgres`, the name of the default database will be `postgres`, and Advanced Server's features compatible with Oracle databases will not be available to the cluster.

`--redwood-like`

Include the `--redwood-like` keywords to instruct the server to use an escape character (an empty string (`'`)) following the `LIKE` (or PostgreSQL-compatible `ILIKE`) operator in a SQL statement that is compatible with Oracle syntax.

`--icu-short-form`

Include the `--icu-short-form` keywords to create a cluster that uses a default ICU (International Components for Unicode) collation for all databases in the cluster. For more information about Unicode collations, please refer to the *EDB Postgres Advanced Server Guide* available at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

4.1 Windows Installation Prerequisites

User Privileges

To perform an Advanced Server installation on a Windows system, you must have administrator privileges. If you are installing Advanced Server into a Windows system that is configured with `User Account Control` enabled, you can assume sufficient privileges to invoke the graphical installer by right clicking on the name of the installer and selecting `Run as administrator` from the context menu.

Windows-specific Software Requirements

You should apply Windows operating system updates before invoking the Advanced Server installer. If (during the installation process) the installer encounters errors, exit the installation, and ensure that your version of Windows is up-to-date before restarting the installer.

Migration Toolkit or EDB*Plus Installation Pre-requisites

Before using an RPM or StackBuilder Plus to install Migration Toolkit or EDB*Plus, you must first install Java (version 1.8 or later). If you are using Windows, Java installers and instructions are available online at:

<http://www.java.com/en/download/manual.jsp>

4.2 Performing a Graphical Installation on Windows

A graphical installation is a quick and easy way to install Advanced Server 11 on a Windows system. Use the wizard's dialogs to specify information about your system and system usage; when you have completed the dialogs, the installer performs an installation based on the selections made during the setup process.

To invoke the wizard, you must have administrator privileges. Assume administrator privileges, and double-click the `edb-as11-server-11.x.x-x-windows-x64` executable file.

To install Advanced Server on some versions of Windows, you may be required to right click on the file icon and select `Run as Administrator` from the context menu to invoke the installer with `Administrator` privileges.

When the Language Selection popup opens, select an installation language and click OK to continue to the Setup window (shown in Figure 4.1).

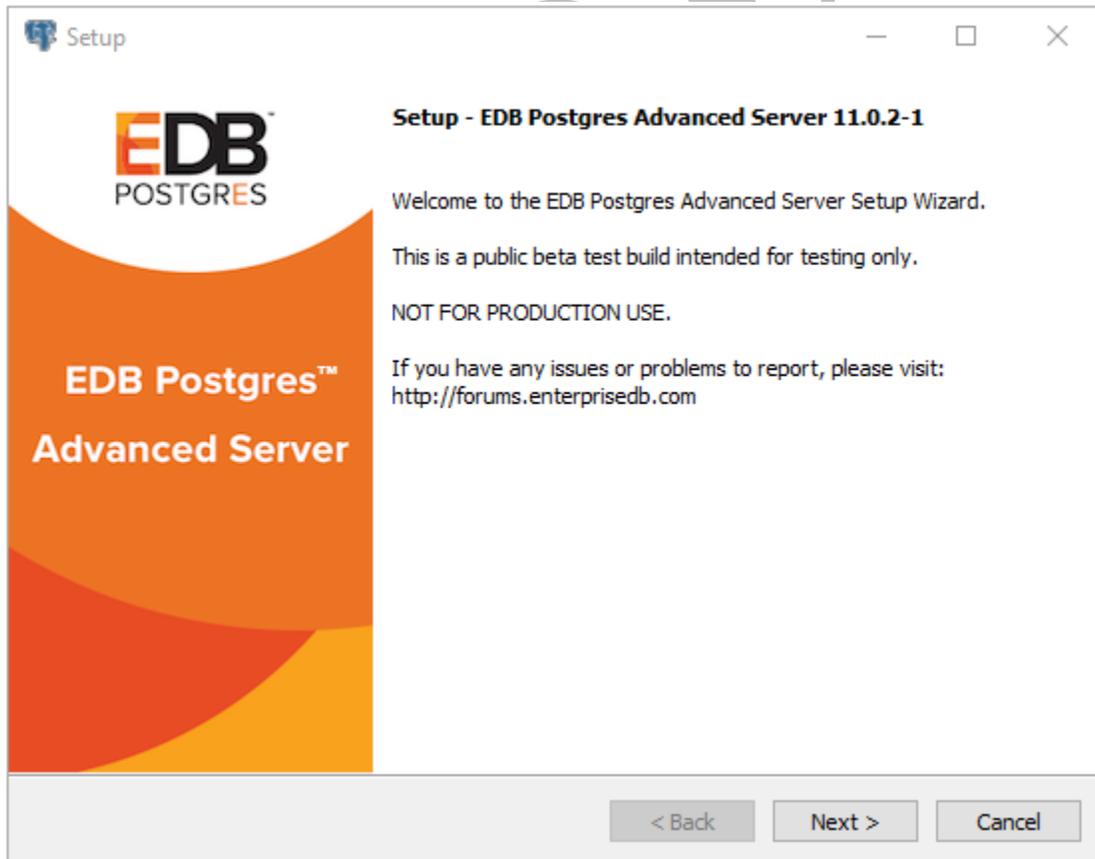


Figure 4.1 -The Advanced Server installer Welcome window

Click `Next` to continue.

The EnterpriseDB License Agreement (see Figure 4.2) opens.



Figure 4.2 -The EnterpriseDB License Agreement

Carefully review the license agreement before highlighting the appropriate radio button; click `Next` to continue.

The Installation Directory window opens, as shown in Figure 4.3.

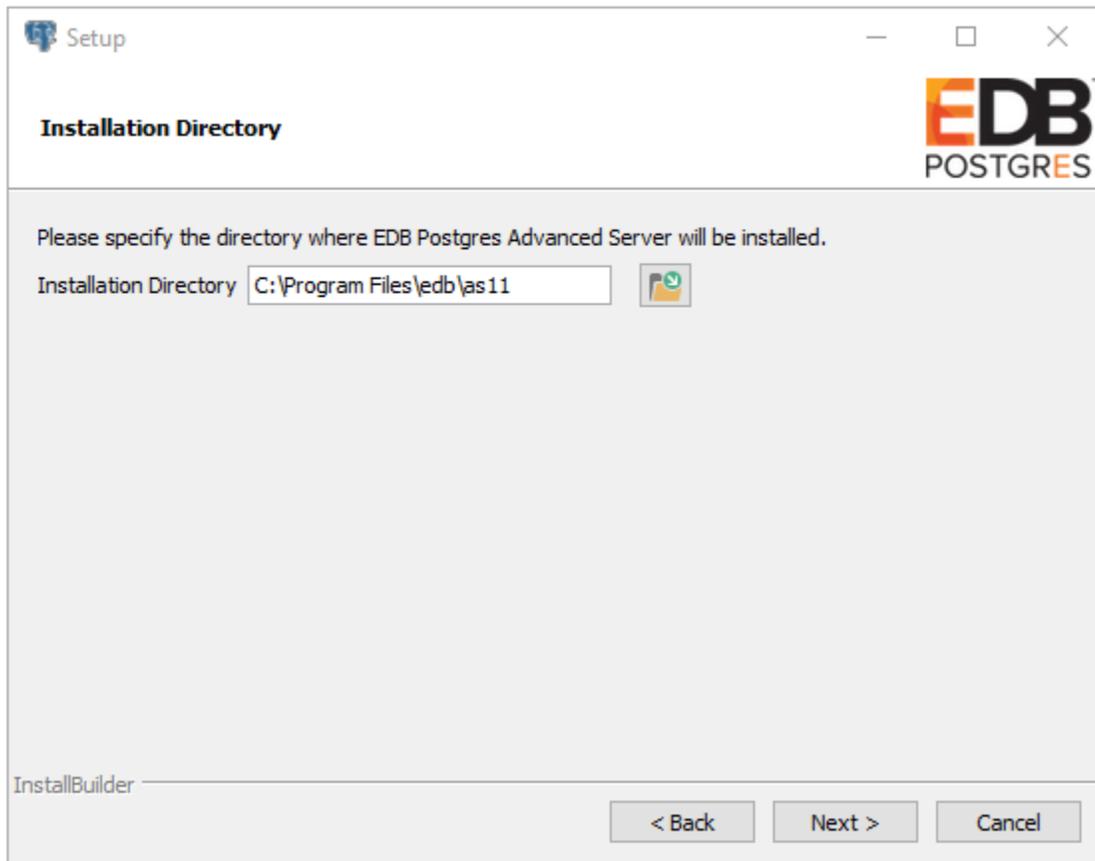


Figure 4.3 -The Installation Directory window.

By default, the Advanced Server installation directory is:

```
C:\Program Files\edb\as11
```

You can accept the default installation location, and click **Next** to continue, or optionally click the **File Browser** icon to open the **Browse For Folder** dialog to choose an alternate installation directory.

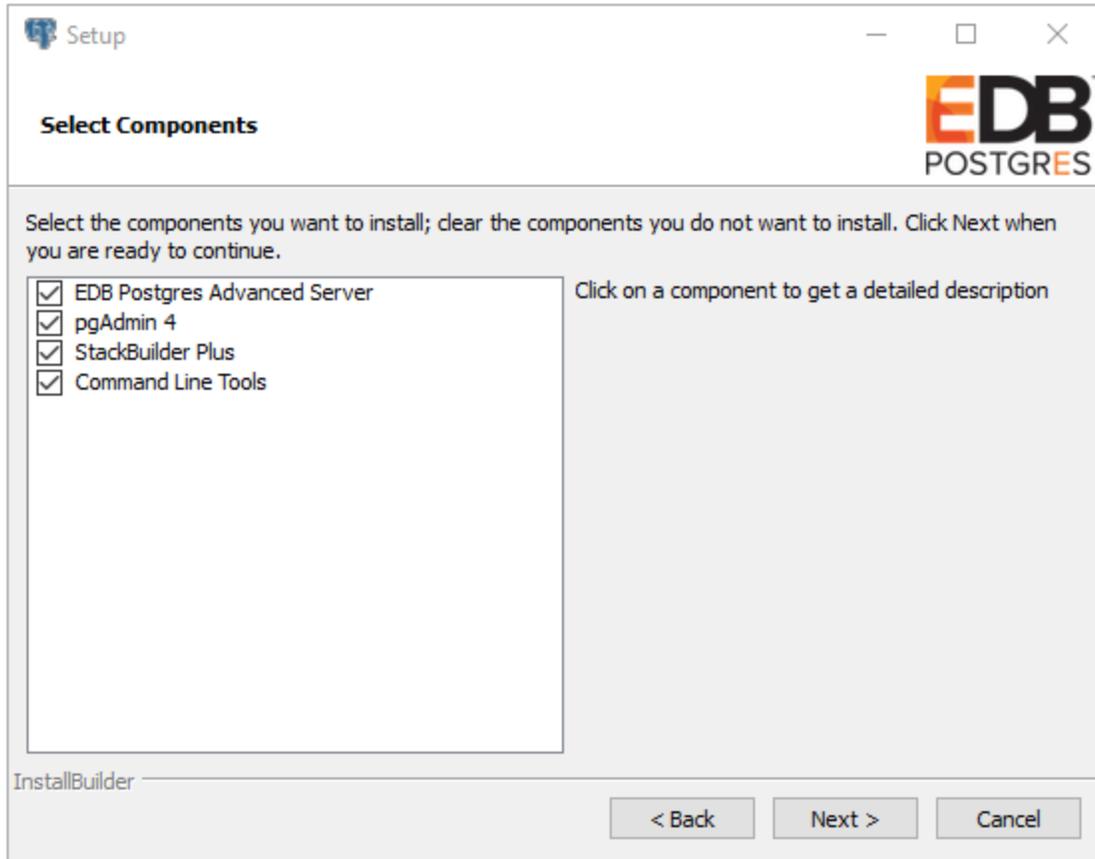


Figure 4.4 -The Select Components window

The `Select Components` window (shown in Figure 4.4) contains a list of optional components that you can install with the Advanced Server Setup wizard. You can omit a module from the Advanced Server installation by deselecting the box next to the components name.

The Setup wizard can install the following components while installing Advanced Server 11:

EDB Postgres Advanced Server

Select the `EDB Postgres Advanced Server` option to install Advanced Server 11.

pgAdmin 4

Select the `EDB Postgres pgAdmin 4` option to install the pgAdmin 4 client. pgAdmin 4 provides a powerful graphical interface for database management and monitoring.

StackBuilder Plus

The StackBuilder Plus utility is a graphical tool that can update installed products, or download and add supporting modules (and the resulting dependencies) after your Advanced Server setup and installation completes. See Section [4.4](#) for more information about StackBuilder Plus.

Command Line Tools

The Command Line Tools option installs command line tools and supporting client libraries including:

- libpq
- psql
- EDB*Loader
- ecpgPlus
- pg_basebackup, pg_dump, and pg_restore
- pg_bench
- and more.

Please note: the Command Line Tools are required if you are installing Advanced Server or pgAdmin 4.

After selecting the components you wish to install, click `Next` to open the `Additional Directories` window (shown in Figure 4.5).

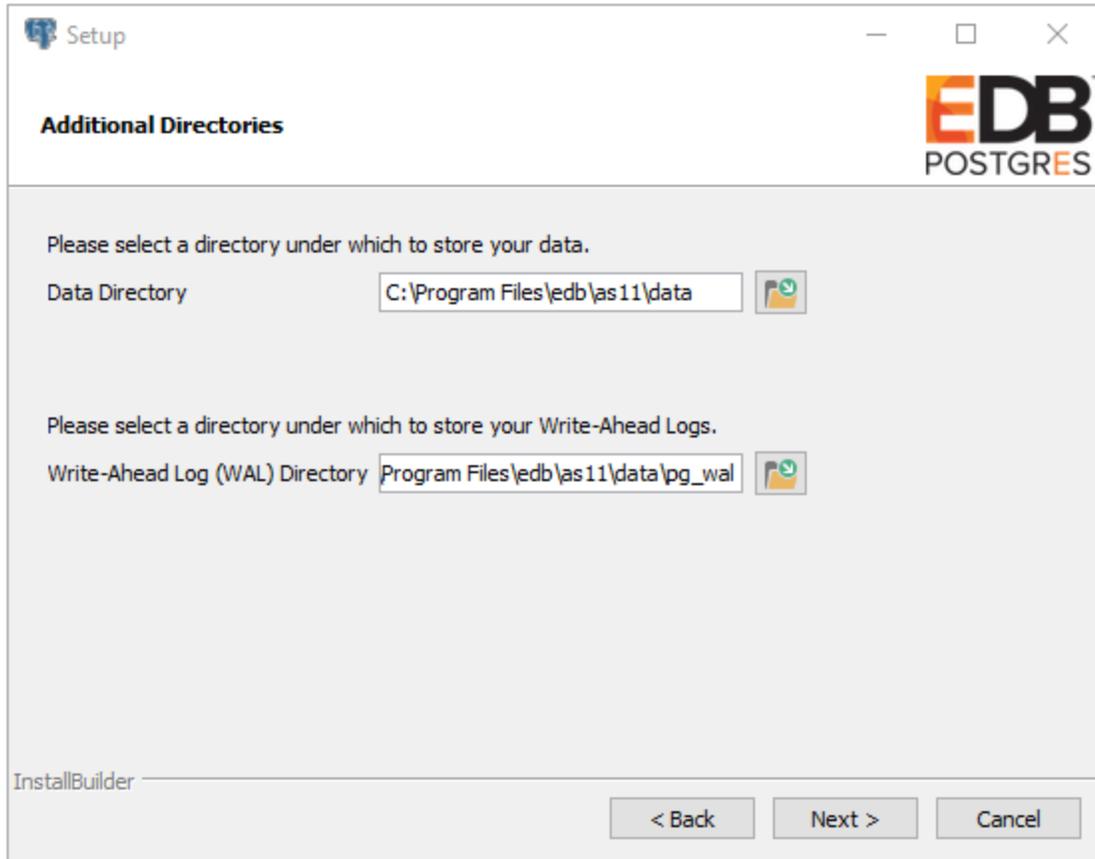


Figure 4.5 -The Additional Directories window.

By default, the Advanced Server data files are saved to:

```
C:\Program Files\edb\as11\data
```

The default location of the Advanced Server Write-Ahead Log (WAL) Directory is:

```
C:\Program Files\edb\as11\data\pg_wal
```

Advanced Server uses write-ahead logs to promote transaction safety and speed transaction processing; when you make a change to a table, the change is stored in shared memory and a record of the change is written to the write-ahead log. When you perform a COMMIT, Advance Server writes contents of the write-ahead log to disk.

Accept the default file locations, or use the File Browser icon to select an alternate location; click Next to continue to the Advanced Server Dialect window.

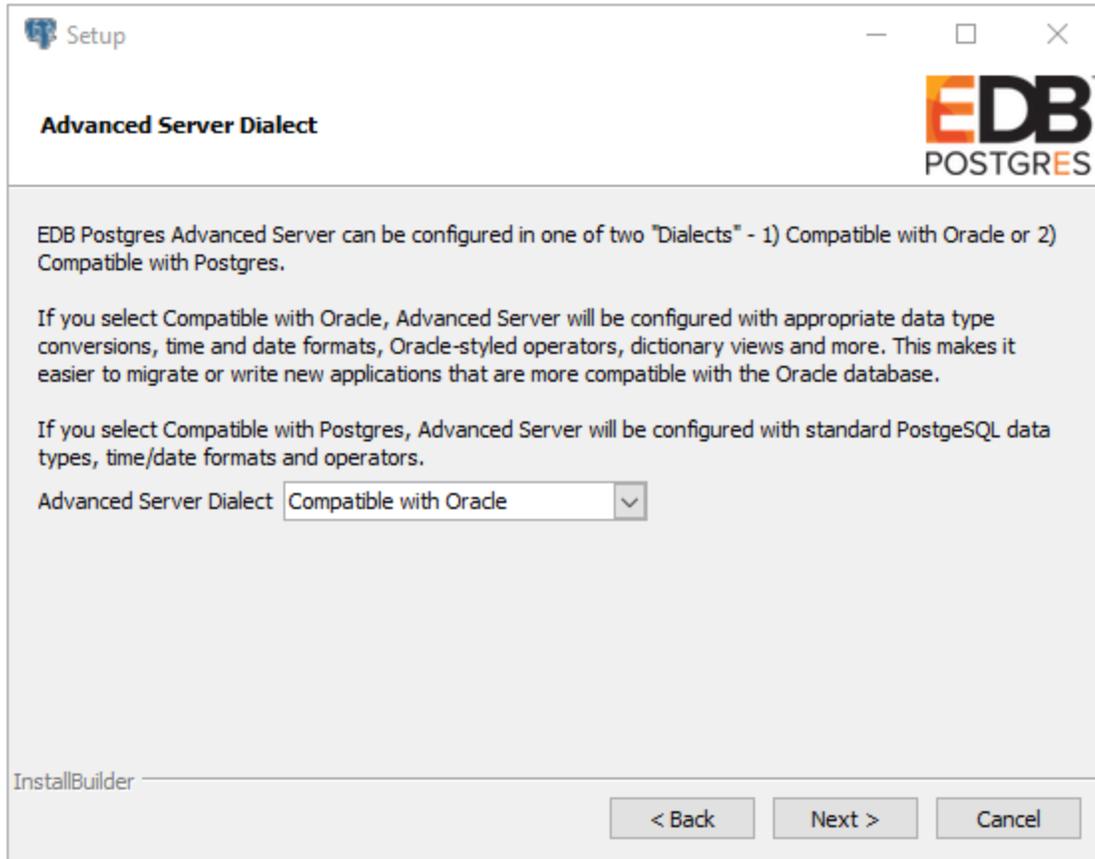


Figure 4.6 -The Advanced Server Dialect window.

Use the drop-down listbox on the Advanced Server Dialect window (see Figure 4.6) to choose a server dialect. The server dialect specifies the compatibility features supported by Advanced Server.

By default, Advance Server installs in Compatible with Oracle mode; you can choose between Compatible with Oracle and Compatible with PostgreSQL installation modes.

Compatible with Oracle

If you select Compatible with Oracle on the Configuration Mode dialog, the installation will include the following features:

- Data dictionary views compatible with Oracle databases.
- Oracle data type conversions.
- Date values displayed in a format compatible with Oracle syntax.
- Support for Oracle-styled concatenation rules (if you concatenate a string value with a NULL value, the returned value is the value of the string).

- Schemas (`dbo` and `sys`) compatible with Oracle databases added to the `SEARCH_PATH`.
- Support for the following Oracle built-in packages:

Package	Functionality Compatible with Oracle Databases
<code>dbms_alert</code>	Provides the ability to register for, send and receive alerts.
<code>dbms_aq</code>	Provides queueing functionality for Advanced Server.
<code>dbms_aqadm</code>	Provides supporting functionality for <code>dbms_aq</code> .
<code>dbms_crypto</code>	Provides a way to encrypt or decrypt RAW, BLOB or CLOB data.
<code>dbms_job</code>	Implements job-scheduling functionality.
<code>dbms_lob</code>	Provides the ability to manage large objects.
<code>dbms_lock</code>	Provides support for the <code>DBMS_LOCK.SLEEP</code> procedure.
<code>dbms_mview</code>	Provides a way to manage and refresh materialized views.
<code>dbms_output</code>	Provides the ability to display a message on the client.
<code>dbms_pipe</code>	Provides the ability to send a message from one session and read it in another session.
<code>dbms_profiler</code>	Collects and stores performance data about PL/pgSQL and SPL statements.
<code>dbms_random</code>	Provides a way to generate random numbers.
<code>dbms_rls</code>	Implements row level security.
<code>dbms_scheduler</code>	Provides a way to create and manage Oracle-style jobs.
<code>dbms_session</code>	A partial implementation that provides support for <code>DBMS_SESSION.SET ROLE</code> .
<code>dbms_sql</code>	Implements use of Dynamic SQL
<code>dbms_utility</code>	Provides a collection of misc functions and procedures.
<code>utl_encode</code>	Provides a way to encode or decode data.
<code>utl_file</code>	Provides a way for a function, procedure or anonymous block to interact with files stored in the server's file system.
<code>utl_http</code>	Provides a way to use HTTP or HTTPS to retrieve information found at a URL.
<code>utl_mail</code>	Provides a simplified interface for sending email and attachments.
<code>utl_raw</code>	Provides a way to manipulate or retrieve the length of raw data types.
<code>utl_smtp</code>	Implements smtp email functions.
<code>utl_url</code>	Provides a way to escape illegal and reserved characters in a URL.

This is not a comprehensive list of the compatibility features for Oracle included when Advanced Server is installed in `Compatible with Oracle` mode; more information about see the *Database Compatibility for Oracle Developer's Guide* available from the EnterpriseDB website at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

If you choose to install in `Compatible with Oracle` mode, the Advanced Server superuser name is `enterprisedb`.

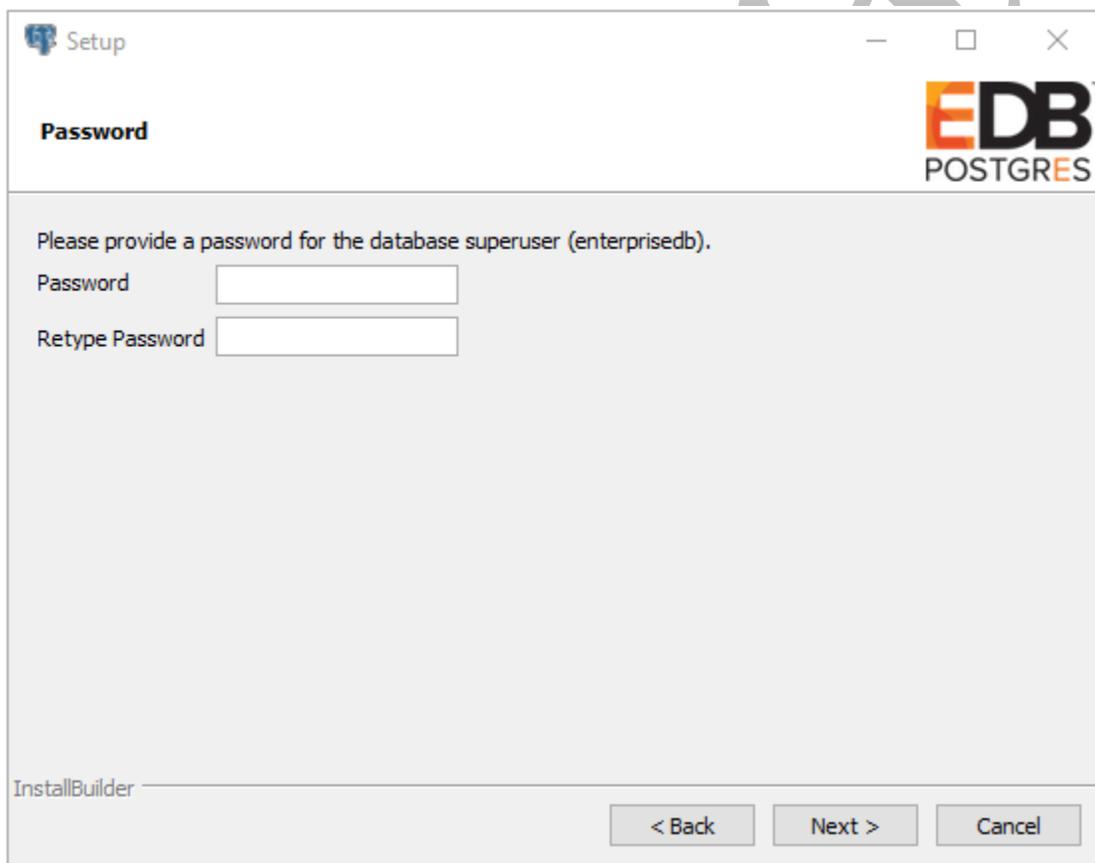
Compatible with PostgreSQL

If you select `Compatible with PostgreSQL`, Advanced Server will exhibit compatibility with PostgreSQL version 11. If you choose to install in `Compatible with PostgreSQL` mode, the default Advanced Server superuser name is `postgres`.

For detailed information about PostgreSQL functionality, visit the official PostgreSQL website at:

<http://www.postgresql.org>

After specifying a configuration mode, click `Next` to continue to the `Password` window (shown in Figure 4.7).



The screenshot shows a Windows-style window titled "Setup" with the EDB PostgreSQL logo in the top right corner. The window is titled "Password" and contains the following text: "Please provide a password for the database superuser (enterisedb)." Below this text are two input fields: "Password" and "Retype Password". At the bottom of the window, there are three buttons: "< Back", "Next >", and "Cancel". The text "InstallBuilder" is visible in the bottom left corner of the window.

Figure 4.7 -The Password window.

Advanced Server uses the password specified on the `Password` window for the database superuser. The specified password must conform to any security policies existing on the Advanced Server host.

After you enter a password in the `Password` field, confirm the password in the `Retype Password` field, and click `Next` to continue.

The `Additional Configuration` window opens (shown in Figure 4.8).

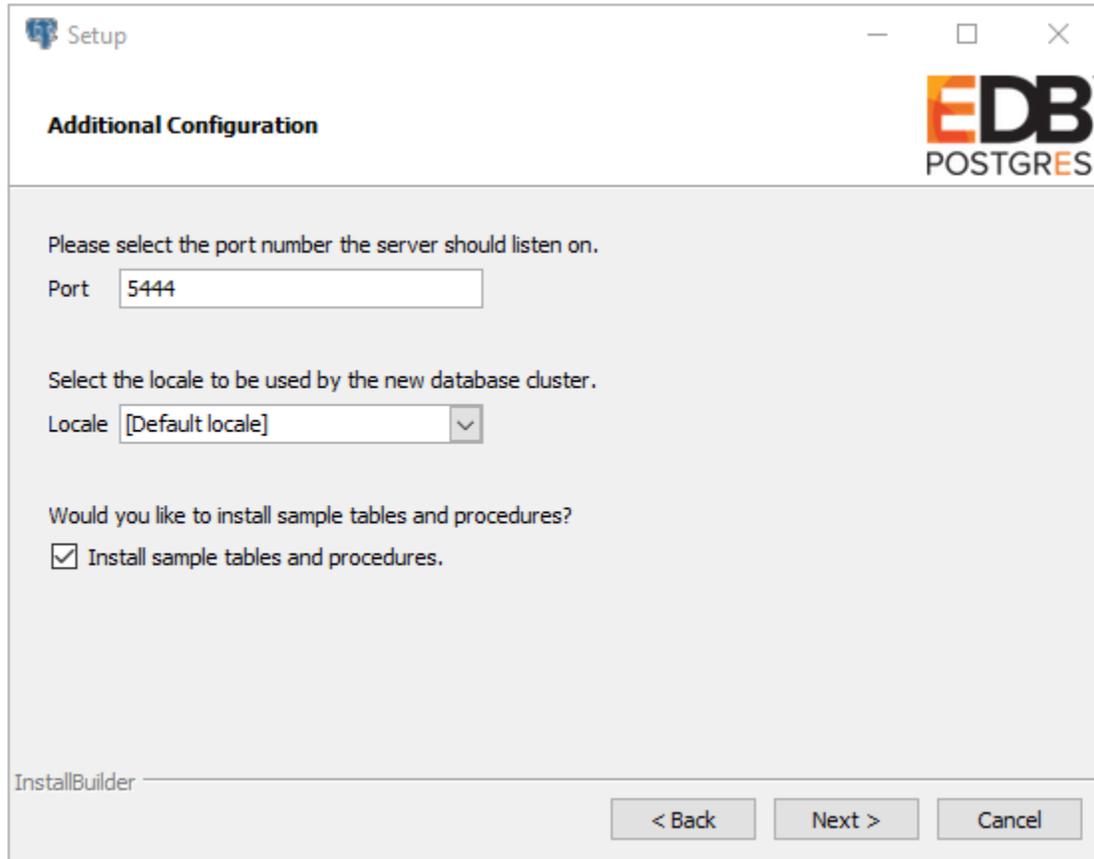


Figure 4.8 -The Additional Configuration window.

Use the fields on the `Additional Configuration` window to specify installation details:

- Use the `Port` field to specify the port number that Advanced Server should listen to for connection requests from client applications. The default is 5444.
- If the `Locale` field is set to `[Default locale]`, Advanced Server uses the system locale as the working locale. Use the drop-down listbox next to `Locale` to specify an alternate locale for Advanced Server.
- By default, the `Setup` wizard installs corresponding sample data for the server dialect specified by the compatibility mode (`Oracle` or `PostgreSQL`). Clear the checkbox next to `Install sample tables and procedures` if you do not wish to have sample data installed.

After verifying the information on the `Additional Configuration` window, click `Next` to open the `Dynatune Dynamic Tuning: Server Utilization` window (shown in Figure 4.9).

The graphical `Setup` wizard facilitates performance tuning via the `Dynatune Dynamic Tuning` feature. `Dynatune` functionality allows `Advanced Server` to make optimal usage of the system resources available on the host machine on which it is installed.

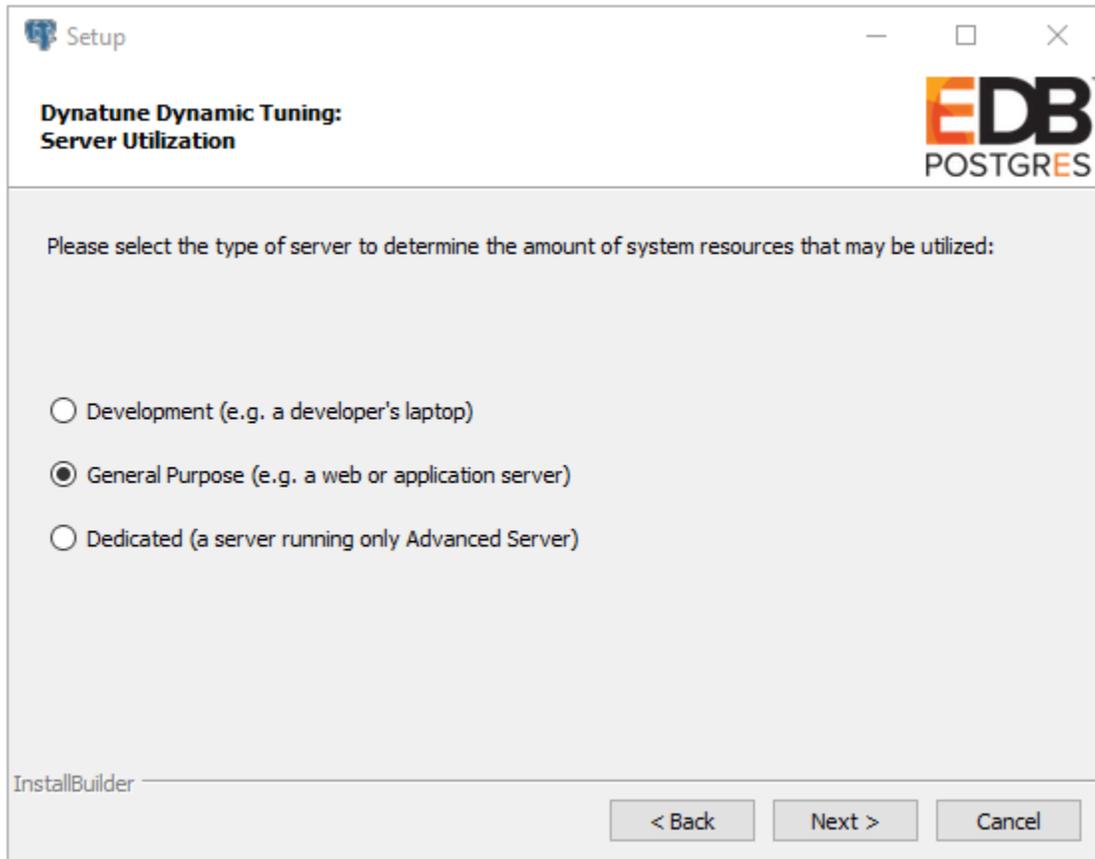


Figure 4.9 -The Dynatune Dynamic Tuning: Server Utilization window.

The `edb_dynatune` configuration parameter determines how `Advanced Server` allocates system resources. Use the radio buttons on the `Server Utilization` window to set the initial value of the `edb_dynatune` configuration parameter:

- Select `Development` to set the value of `edb_dynatune` to 33. A low value dedicates the least amount of the host machine's resources to the database server. This is a good choice for a development machine.
- Select `General Purpose` to set the value of `edb_dynatune` to 66. A mid-range value dedicates a moderate amount of system resources to the database server. This would be a good setting for an application server with a fixed number of applications running on the same host as `Advanced Server`.

- Select `Dedicated` to set the value of `edb_dynatune` to 100. A high value dedicates most of the system resources to the database server. This is a good choice for a dedicated server host.

After the installation is complete, you can adjust the value of `edb_dynatune` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for your changes to take effect.

Select the appropriate setting for your system, and click `Next` to continue to the `Dynatune Dynamic Tuning: Workload Profile` window (shown in Figure 4.10).

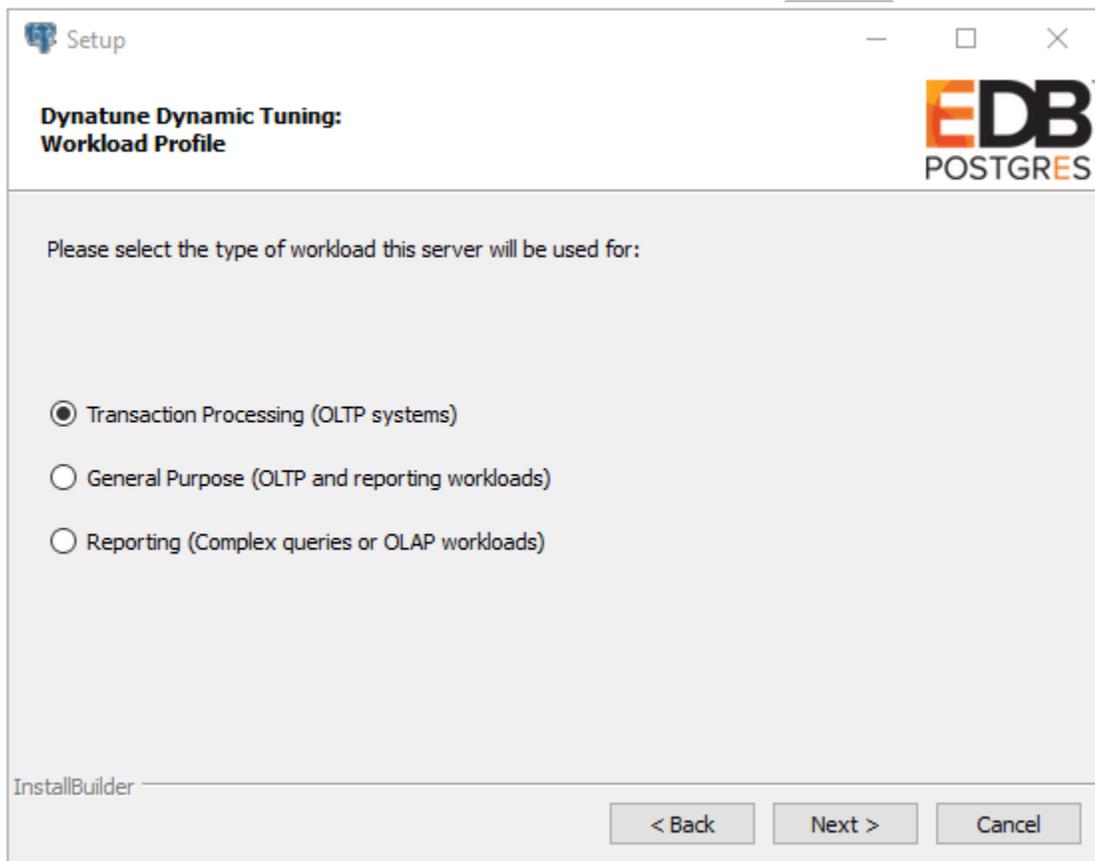


Figure 4.10 -The Dynatune Dynamic Tuning: Workload Profile window.

Use the radio buttons on the `Workload Profile` window to specify the initial value of the `edb_dynatune_profile` configuration parameter. The `edb_dynatune_profile` parameter controls performance-tuning aspects based on the type of work that the server performs.

- **Select Transaction Processing (OLTP systems) to specify an `edb_dynatune_profile` value of `oltp`. Recommended when Advanced Server is supporting heavy online transaction processing.**
- **Select General Purpose (OLTP and reporting workloads) to specify an `edb_dynatune_profile` value of `mixed`. Recommended for servers that provide a mix of transaction processing and data reporting.**
- **Select Reporting (Complex queries or OLAP workloads) to specify an `edb_dynatune_profile` value of `reporting`. Recommended for database servers used for heavy data reporting.**

After the installation is complete, you can adjust the value of `edb_dynatune_profile` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for your changes to take effect.

For more information about `edb_dynatune` and other performance-related topics, see the *EDB Postgres Advanced Server Guide* available from the EnterpriseDB website at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

Click **Next** to continue. The **Update Notification Service** window (shown in Figure 4.11) opens.

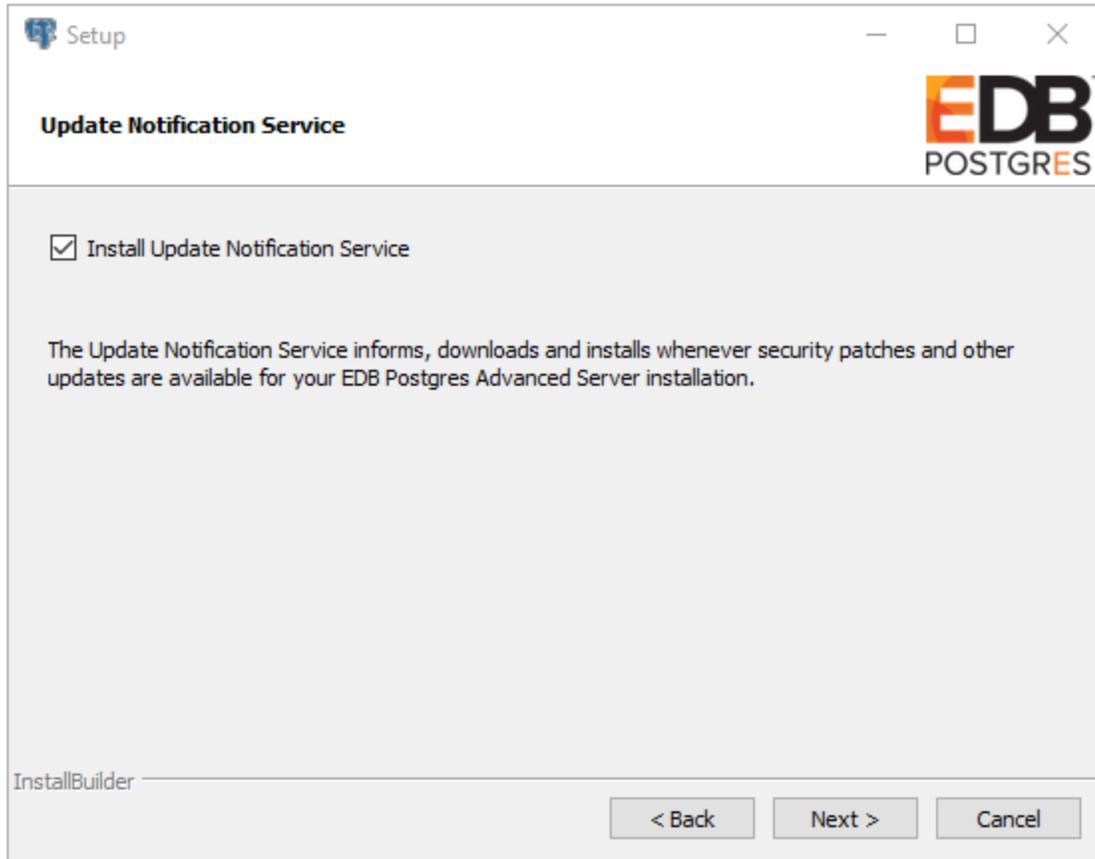


Figure 4.11 -The Update Notification Service window.

When enabled, the update notification service notifies you of any new updates and security patches available for your installation of Advanced Server.

By default, Advanced Server is configured to start the service when the system boots; clear the `Install Update Notification Service` checkbox, or accept the default, and click `Next` to continue.

The `Pre Installation Summary` opens as shown in Figure 4.12.

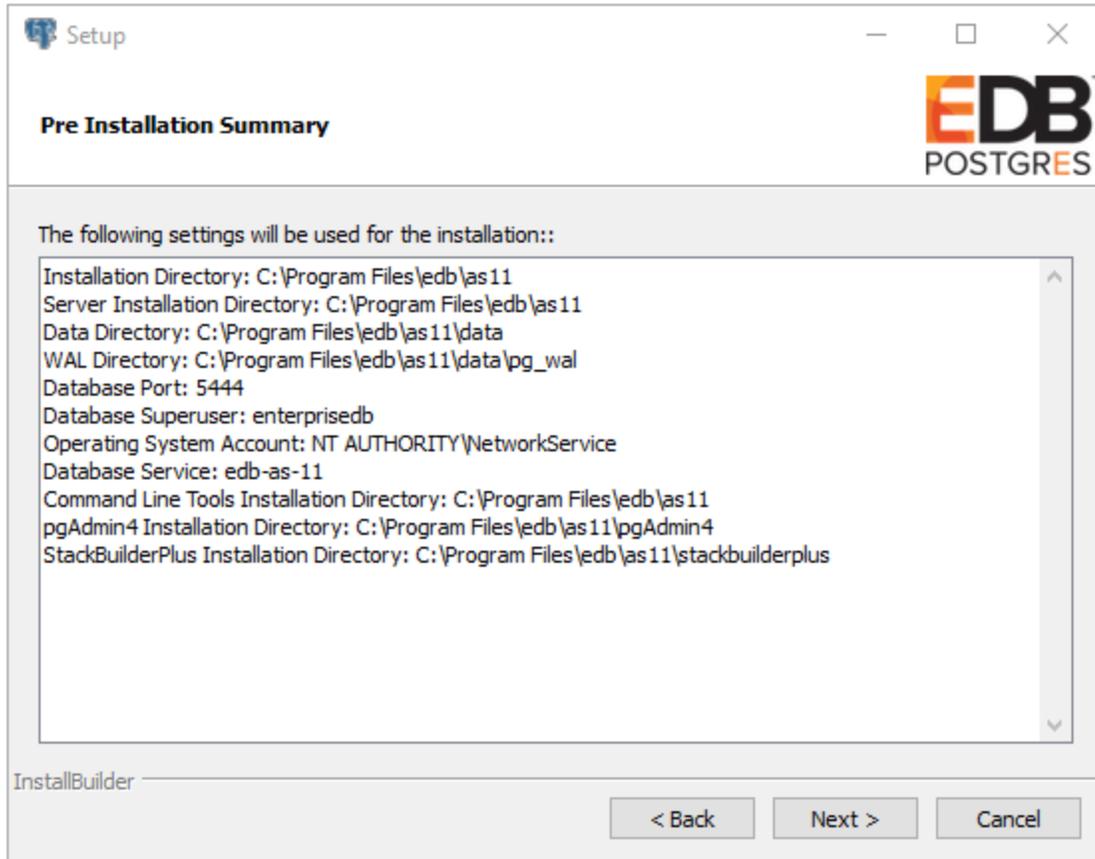


Figure 4.12 -The Pre Installation Summary.

The Pre Installation Summary provides an overview of the options specified during the Setup process. Review the options before clicking Next; use the Back button to navigate back through the dialogs and update any options.

The `Ready to Install` window (see Figure 4.13) confirms that the installer has the information it needs about your configuration preferences to install Advanced Server. Click `Next` to continue.

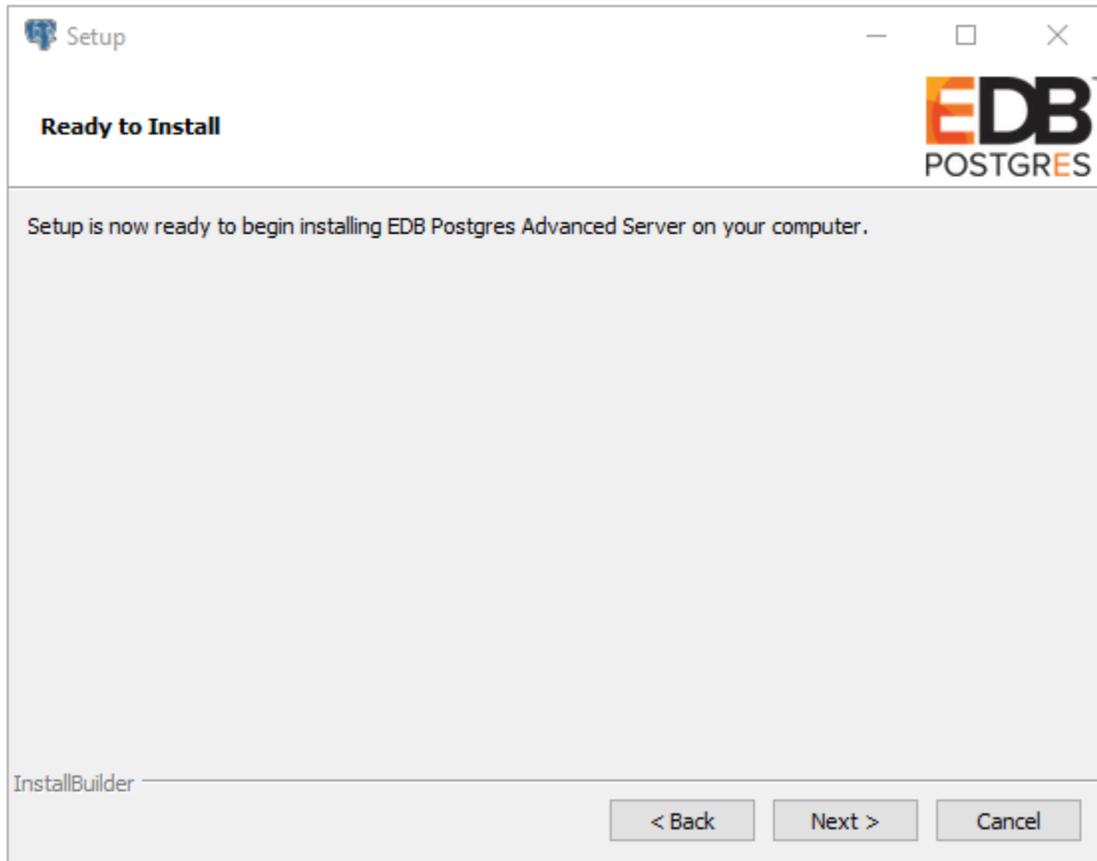


Figure 4.13 -The Ready to Install window.

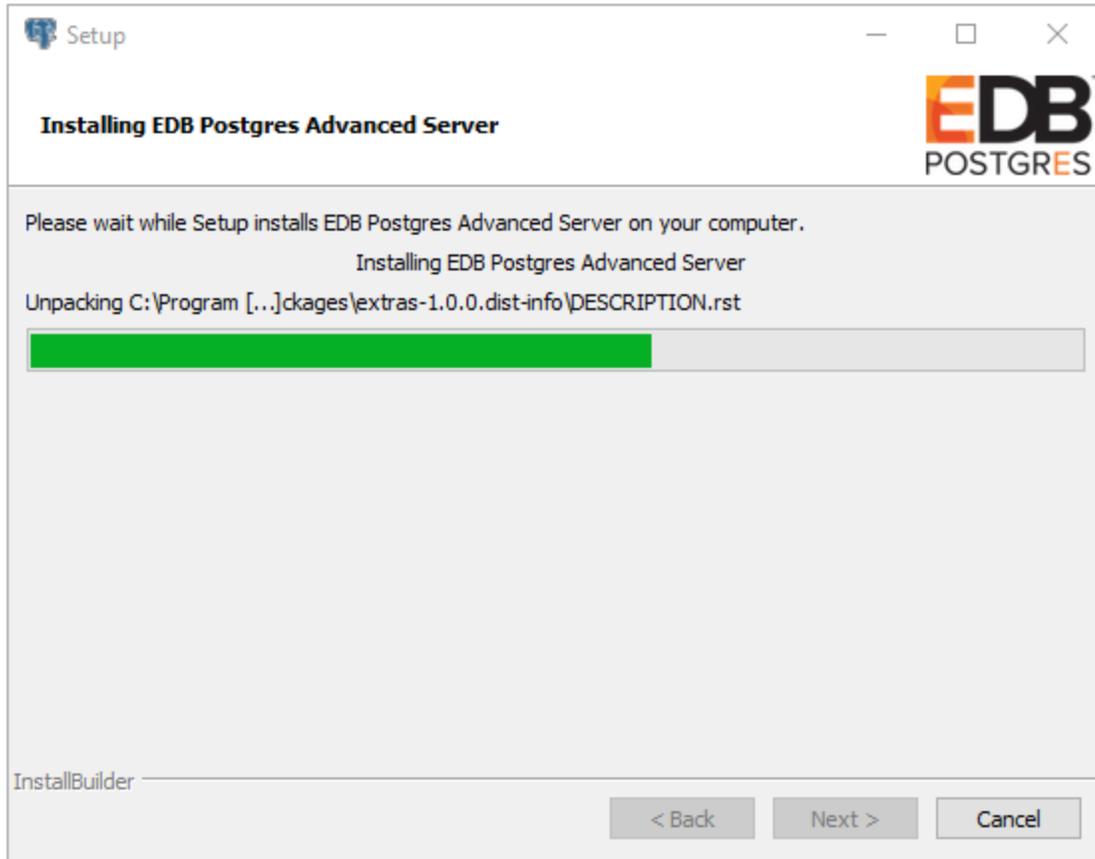


Figure 4.14 -Installing Advanced Server.

As each supporting module is unpacked and installed, the module's installation is confirmed with a progress bar (see Figure 4.14).

Before the Setup wizard completes the Advanced Server installation, it offers to Launch StackBuilder Plus at exit? (see Figure 4.15).

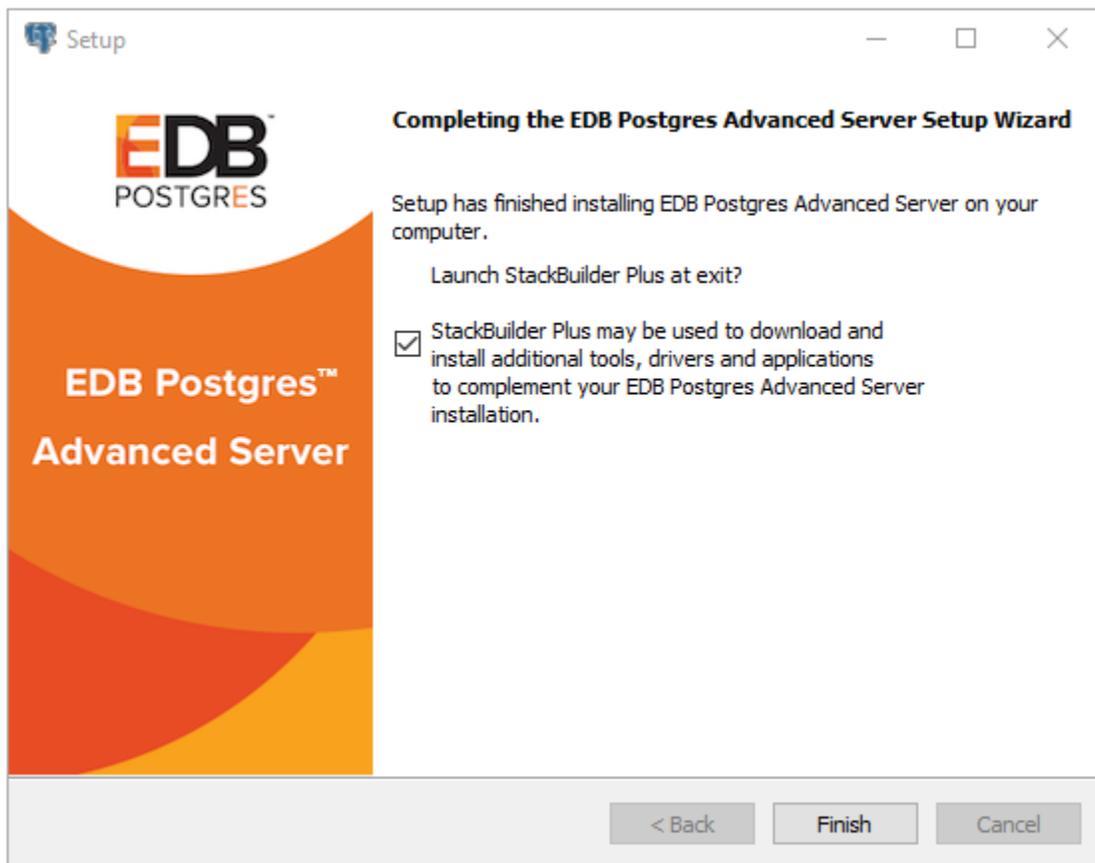


Figure 4.15 -The Setup wizard offers to Launch StackBuilder Plus at exit.

You can clear the StackBuilder Plus checkbox and click Finish to complete the Advanced Server installation, or accept the default and proceed to StackBuilder Plus.

EDB Postgres StackBuilder Plus is included with the installation of Advanced Server and its core supporting components. StackBuilder Plus is a graphical tool that can update installed products, or download and add supporting modules (and the resulting dependencies) after your Advanced Server setup and installation completes. See [Section 4.4](#) for more information about StackBuilder Plus.

4.3 Invoking the Graphical Installer from the Command Line

The command line options of the Advanced Server installer offer functionality for Windows systems that reside in situations where a graphical installation may not work because of limited resources or privileges. You can:

- Include the `--mode unattended` option when invoking the installer to perform an installation without user input.
- Invoke the installer with the `--extract-only` option to perform a minimal installation when you don't hold the privileges required to perform a complete installation.

Not all command line options are suitable for all situations. For a complete reference guide to the command line options, see Section [4.3.3, Reference - Command Line Options](#).

Please note: If you are invoking the installer from the command line to perform a system upgrade, the installer will ignore command line options, and preserve the configuration of the previous installation.

4.3.1 Performing an Unattended Installation

To specify that the installer should run without user interaction, include the `--mode unattended` command line option. In unattended mode, the installer uses one of the following sources for configuration parameters:

- command line options (specified when invoking the installer)
- parameters specified in an option file
- Advanced Server installation defaults

You can embed the non-interactive Advanced Server installer within another application installer; during the installation process, a progress bar allows the user to view the progression of the installation.

You must have administrative privileges to install Advanced Server using the `--mode unattended` option. If you are using the `--mode unattended` option to install Advanced Server with a client, the calling client must be invoked with superuser or administrative privileges.

To start the installer in unattended mode, navigate to the directory that contains the executable file, and enter:

```
edb-as11-server-11.x.x-x-windows-x64.exe --mode unattended
--superpassword database_superuser_password --
servicepassword system_password
```

When invoking the installer, include the `--servicepassword` option to specify an operating system password for the user installing Advanced Server.

Use the `--superpassword` option to specify a password that conforms to the password security policies defined on the host; enforced password policies on your system may not accept the default password (`enterprisedb`).

4.3.2 Performing an Installation with Limited Privileges

To perform an abbreviated installation of Advanced Server without access to administrative privileges, invoke the installer from the command line and include the `--extract-only` option. The `--extract-only` option extracts the binary files in an unaltered form, allowing you to experiment with a minimal installation of Advanced Server.

If you invoke the installer with the `--extract-only` options, you can either manually create a cluster and start the service, or run the installation script. To manually create the cluster, you must:

- Initialize the cluster
- Configure the cluster
- Start the service with `pg_ctl`

For more information about the `initdb` and `pg_ctl` commands, please see the PostgreSQL Core Documentation at:

<https://www.postgresql.org/docs/11/static/app-initdb.html>

<https://www.postgresql.org/docs/11/static/app-pg-ctl.html>

If you include the `--extract-only` option, the installer steps through a shortened form of the `Setup` wizard. During the brief installation process, the installer generates an installation script that can be later used to complete a more complete installation. You must have administrative privileges to invoke the installation script.

The installation script:

- Initializes the database cluster if the cluster is empty.
- Configures the server to start at boot-time.
- Establishes initial values for Dynatune (dynamic tuning) variables.

The scripted Advanced Server installation does not create menu shortcuts or access to EDB Postgres StackBuilder Plus, and no modifications are made to registry files. The Advanced Server Update Monitor will not detect components installed by the scripted installation, and will not issue alerts for available updates to those components.

To perform a limited installation and generate an installation script, download and unpack the Advanced Server installer. Navigate into the directory that contains the installer, and invoke the installer with the command:

```
edb-as11-server-11.x.x-x-windows.exe --extract-only yes
```

A dialog opens, prompting you to choose an installation language. Select a language for the installation from the drop-down listbox, and click **OK** to continue. The **Setup Wizard** opens (shown in Figure 4.16).

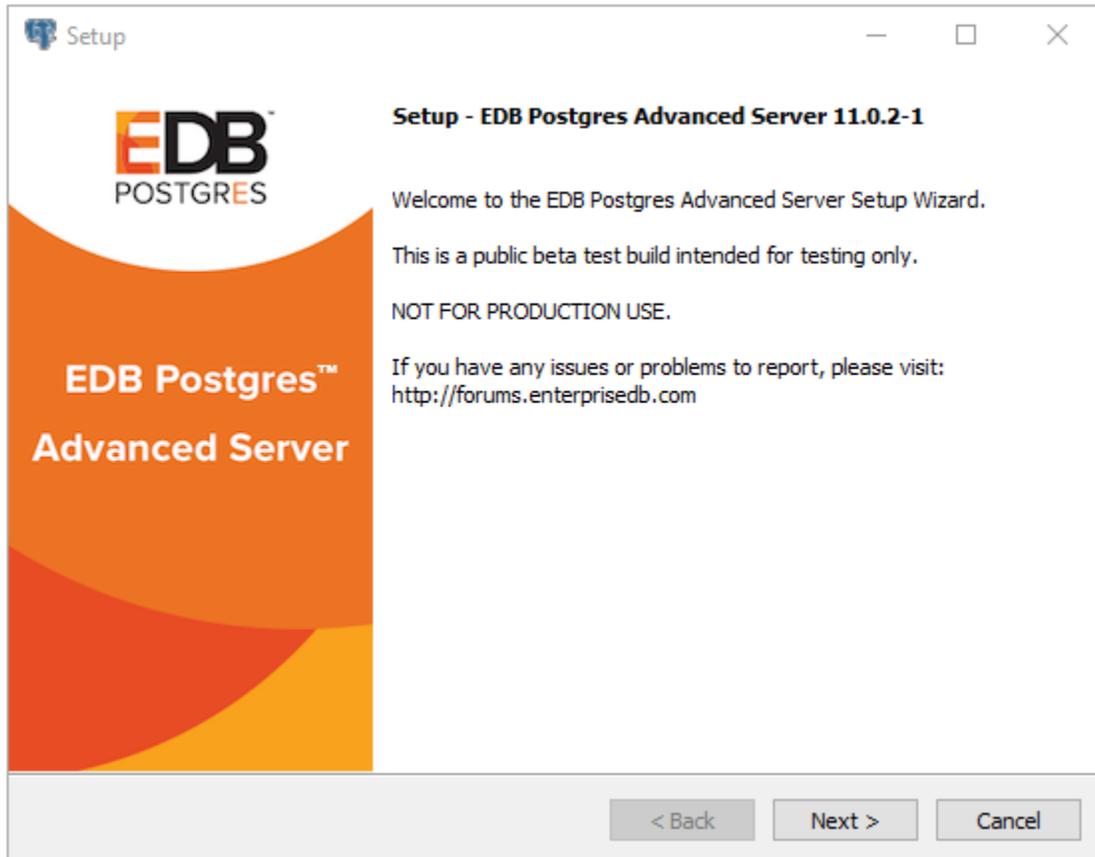


Figure 4.16 -The Welcome window.

Click **Next** to continue.

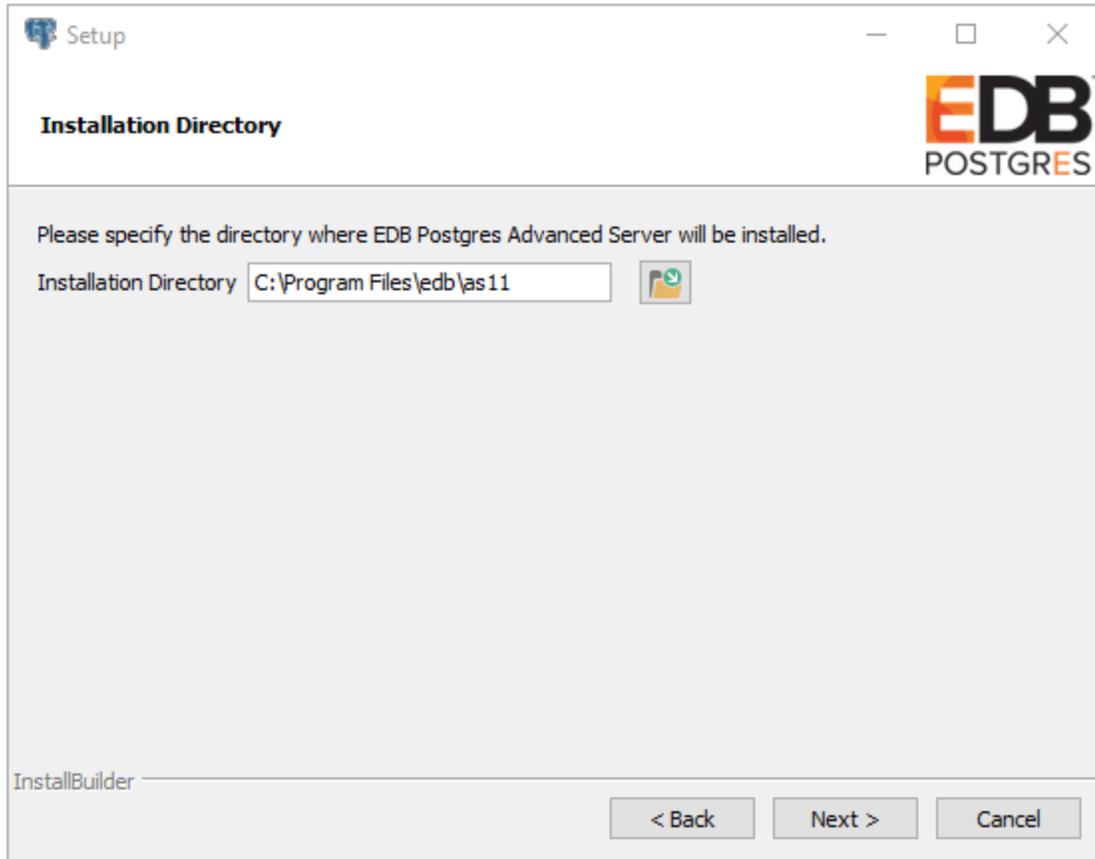


Figure 4.17 -Specify an installation directory.

On Windows, the default Advanced Server installation directory is:

```
C:\Program Files\edb\as11
```

You can accept the default installation location (see Figure 4.17), and click `Next` to continue to the `Ready to Install` window (shown in Figure 4.18), or optionally click the `File Browser` icon to choose an alternate installation directory.

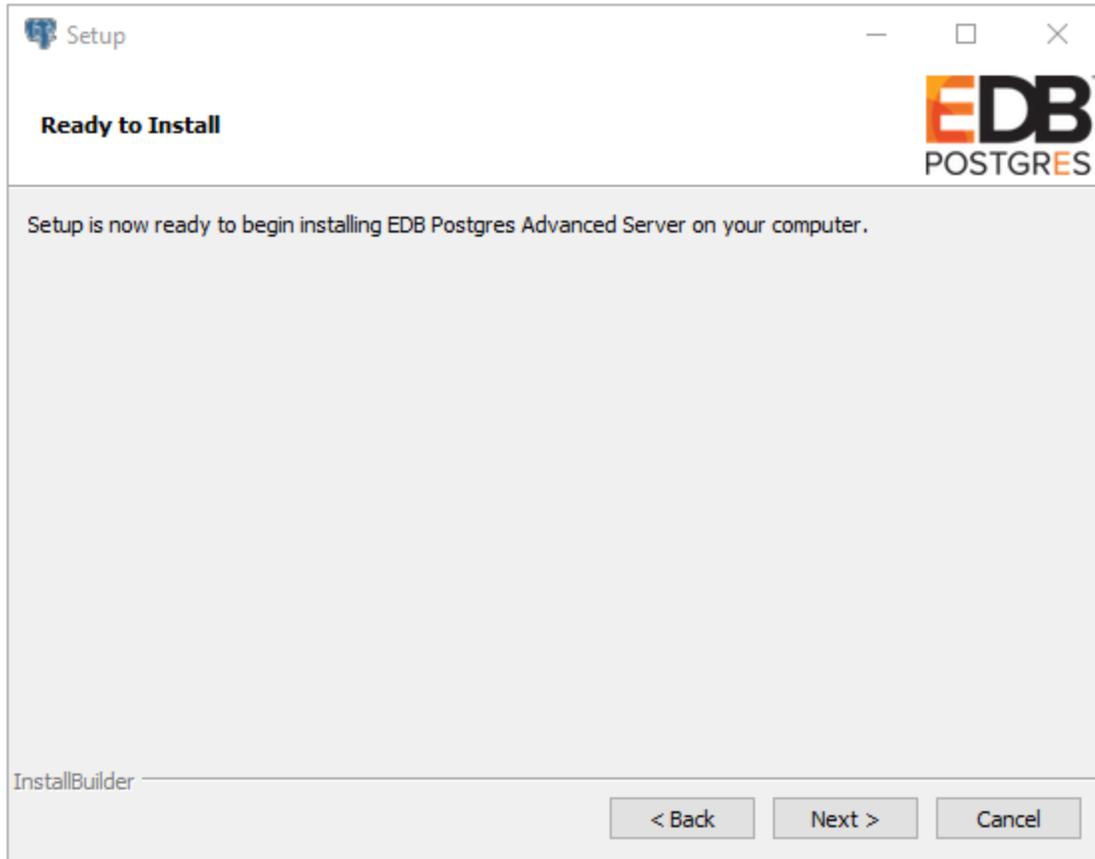


Figure 4.18 -The Setup wizard is ready to install Advanced Server.

Click `Next` to proceed with the Advanced Server installation. During the installation, progress bars and popups mark the installation progress. The installer notifies you when the installation is complete (see Figure 4.19).

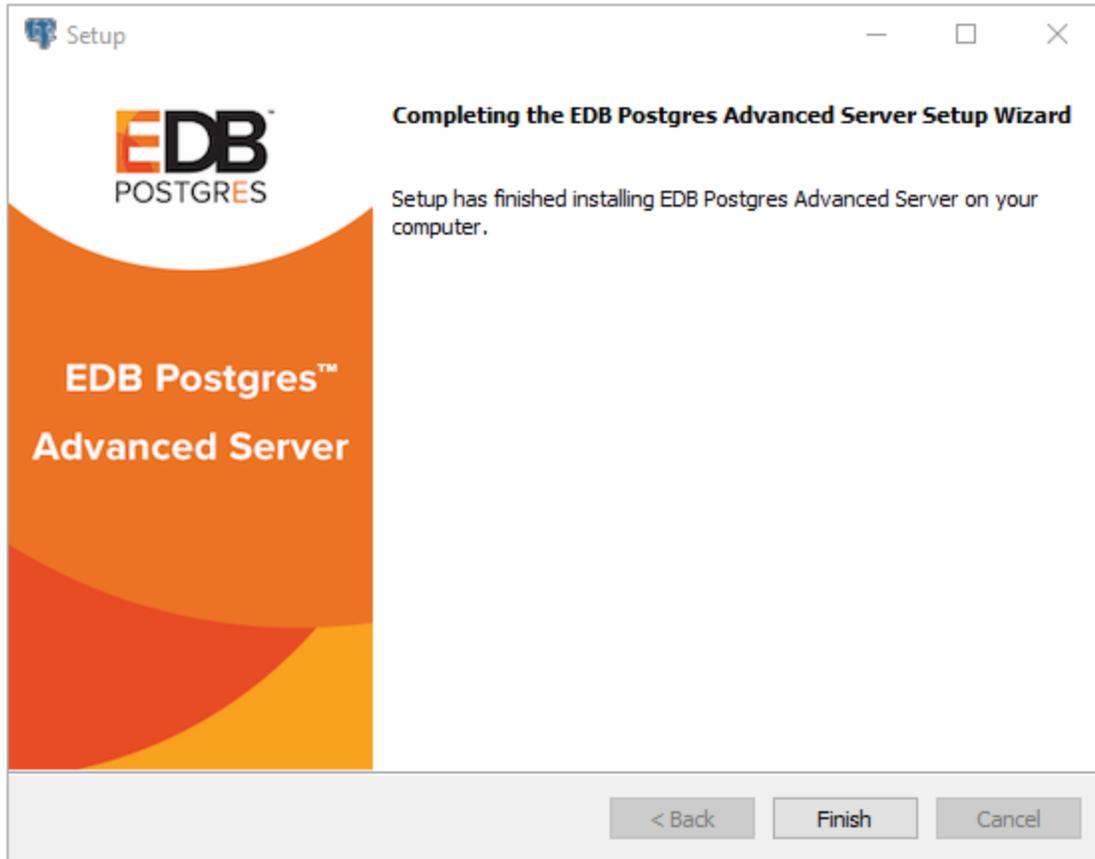


Figure 4.19 -The Advanced Server installation is complete.

After completing the minimal installation, you can execute a script to initialize a cluster and start the service. The script is (by default) located in:

```
C:\Program Files\edb
```

To execute the installation script, open a command line and assume administrative privileges. Navigate to the directory that contains the script, and execute the command:

```
cscript runAsAdmin.vbs
```

The installation script executes at the command line, prompting you for installation configuration information. The default configuration value is displayed in square braces immediately before each prompt; update the default value or press `Enter` to accept the default value and continue.

Example

The following dialog is an example of a scripted installation. The actual installation dialog may vary, and will reflect the options specified during the installation.

Specify the installation directory is the directory where Advanced Server is installed:

```
Please enter the installation directory [ C:\Program
Files\edb\as11 ] :
```

Specify the directory in which Advanced Server data will be stored:

```
Please enter the data directory path: [C:\Program
Files\edb\as11\data ] :
```

Specify the WAL directory (where the write-ahead log will be written):

```
Please enter the Write-Ahead Log (WAL) directory path:
[C:\Program Files\edb\as11\data\pg_wal ] :
```

The database mode specifies the database dialect with which the Advanced Server installation is compatible. The optional values are `oracle` or `postgresql`.

```
Please enter database mode: [ oracle ] :
```

Compatible with Oracle Mode

Specify `oracle` mode to include the following functionality:

- Data dictionary views and data type conversions compatible with Oracle databases.
- Date values displayed in a format compatible with Oracle syntax.
- Oracle-styled concatenation rules (if you concatenate a string value with a `NULL` value, the returned value is the value of the string).
- Schemas (`dbo` and `sys`) compatible with Oracle databases added to the `SEARCH_PATH`.
- Support for the following Oracle built-in packages.

This is not a comprehensive list of the compatibility features for Oracle included when Advanced Server is installed in `Compatible with Oracle` mode; more information about Advanced Server is available in the *Database Compatibility for Oracle Developer's Guide* available at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

If you choose to install in `Compatible with Oracle` mode, the Advanced Server superuser name is `enterprisedb`.

Compatible with PostgreSQL Mode

Specify `postgres` to install Advanced Server with complete compatibility with Postgres version 11.

For more information about PostgreSQL functionality, see the PostgreSQL Core Documentation available at:

<http://www.enterprisedb.com/products-services-training/products/documentation>

If you choose to install in `Compatible with PostgreSQL` mode, the Advanced Server superuser name is `postgres`.

Specify a port number for the Advanced Server listener to listen on:

NOTE: We will not be able to examine, if port is currently used by other application.

Please enter port: [5444] :

Specify a locale for the Advanced Server installation. If you accept the `DEFAULT` value, the locale defaults to the locale of the host system.

Please enter the locale: [DEFAULT] :

You can optionally install sample tables and procedures. Press `Return`, or enter `Y` to accept the default and install the sample tables and procedures; enter an `n` and press `Return` to skip this step.

Install sample tables and procedures? (Y/n): [Y] :

Specify a password for the database superuser. By default, the database superuser is named `enterprisedb`.

Please enter the password for the SuperUser(enterprisedb): [] :

Specify a password for the service account user.

Please enter the password for the ServiceAccount(enterprisedb): [] :

The server utilization value is used as an initial value for the `edb_dynatune` configuration parameter. `edb_dynatune` determines how Advanced Server allocates system resources.

- A low value dedicates the least amount of the host machine's resources to the database server; a low value is a good choice for a development machine.

- A mid-range value dedicates a moderate amount of system resources to the database server. A mid-range value is a good setting for an application server with a fixed number of applications running on the same host as Advanced Server.
- A high value dedicates most of the system resources to the database server. This is a good choice for a dedicated server host.

Specify a value between 1 and 100:

```
Please enter the Server Utilization: [ 66 ] :
```

After the installation is complete, you can adjust the value of `edb_dynatune` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for the changes to take effect.

The workload profile value is used as an initial value for the `edb_dynatune_profile` configuration parameter. `edb_dynatune_profile` controls performance-tuning based on the type of work that the server performs.

- Specify `oltp` if the server will be supporting heavy online transaction workloads.
- Specify `mixed` if the server will provide a mix of transaction processing and data reporting.
- Specify `reporting` if the database server will be used for heavy data reporting.

Specify a value between 1 and 100:

```
Please enter the Workload Profile: [ oltp ] :
```

After the installation is complete, you can adjust the value of `edb_dynatune_profile` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation, and restarting the server.

After processing, the Advanced Server installation is complete.

4.3.3 Reference - Command Line Options

You can optionally include the following parameters for an Advanced Server installation on the command line, or in a configuration file when invoking the Advanced Server installer.

```
--create_samples { yes | no }
```

Use the `--create_samples` option to specify whether the installer should create the sample tables and procedures for the database dialect specified with the `--databasemode` parameter. The default is `yes`.

```
--databasemode { oracle | postgresql }
```

Use the `--databasemode` parameter to specify a database dialect. The default is `oracle`.

```
--datadir data_directory
```

Use the `--datadir` parameter to specify a location for the cluster's data directory. *data_directory* is the name of the directory; include the complete path to the desired directory.

```
--debuglevel { 0 | 1 | 2 | 3 | 4 }
```

Use the `--debuglevel` parameter to set the level of detail written to the *debug_log* file (see `--debugtrace`). Higher values produce more detail in a longer trace file. The default is `2`.

```
--debugtrace debug_log
```

Use the `--debugtrace` parameter to troubleshoot installation problems. *debug_log* is the name of the file that contains installation troubleshooting details.

```
--disable-components component_list
```

Use the `--disable-components` parameter to specify a list of Advanced Server components to exclude from the installation. By default, *component_list* contains `'` (the empty string). *component_list* is a comma-separated list containing one or more of the following components:

```
dbserver
```

EDB Postgres Advanced Server 11.

pgadmin4

The EDB Postgres pgAdmin 4 provides a powerful graphical interface for database management and monitoring.

```
--enable_acledit { 1 | 0 }
```

The `--enable_acledit 1` option instructs the installer to grant permission to the user specified by the `--serviceaccount` option to access the Advanced Server binaries and data directory. By default, this option is disabled if `--enable_acledit 0` is specified or if the `--enable_acledit` option is completely omitted.

Note: Specification of this option is valid only when installing on Windows. The `--enable_acledit 1` option should be specified when a *discretionary access control list* (DACL) needs to be set for allowing access to objects on which Advanced Server is to be installed. See the following for information on a DACL:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa446597\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa446597(v=vs.85).aspx)

In order to perform future operations such as upgrading Advanced Server, access to the data directory must exist for the service account user specified by the `--serviceaccount` option. By specifying the `--enable_acledit 1` option, access to the data directory by the service account user is provided.

```
--enable-components component_list
```

Although this option is listed when you run the installer with the `--help` option, the `--enable-components` parameter has absolutely no effect on which components are installed. All components will be installed regardless of what is specified in *component_list*. In order to install only specific selected components, you must use the `--disable-components` parameter previously described in this section to list the components you do not want to install.

```
--extract-only { yes | no }
```

Include the `--extract-only` parameter to indicate that the installer should extract the Advanced Server binaries without performing a complete installation. Superuser privileges are not required for the `--extract-only` option. The default value is `no`.

```
--help
```

Include the `--help` parameter to view a list of the optional parameters.

```
--installer-language { en | ja | zh_CN | zh_TW | ko }
```

Use the `--installer-language` parameter to specify an installation language for Advanced Server. The default is `en`.

`en` specifies English.

`ja` specifies Japanese

`zh_CN` specifies Chinese Simplified.

`zh_TW` specifies Traditional Chinese.

`ko` specifies Korean.

```
--install_runtimes { yes | no } (Windows only.)
```

Include `--install_runtimes` to specify whether the installer should install the Microsoft Visual C++ runtime libraries. Default is `yes`.

```
--locale locale
```

Specifies the locale for the Advanced Server cluster. By default, the installer will use to the locale detected by `initdb`.

```
--mode {text | unattended}
```

Use the `--mode` parameter to specify an installation mode. The following modes are supported:

`text` - Specify `text` to perform a text mode installation in a console window. This is a Linux-only option.

`unattended` - Specify `unattended` to specify that the installer should perform an installation that requires no user input during the installation process.

```
--optionfile config_file
```

Use the `--optionfile` parameter to specify the name of a file that contains the installation configuration parameters. `config_file` must specify the complete path to the configuration parameter file.

```
--prefix installation_dir/as11.x
```

Use the `--prefix` parameter to specify an installation directory for Advanced Server. The installer will append a version-specific sub-directory (i.e. `as11`) to the specified directory. The default installation directory is:

```
C:\Program Files\edb\as11
```

```
--serverport port_number
```

Use the `--serverport` parameter to specify a listener port number for Advanced Server.

If you are installing Advanced Server in unattended mode, and do not specify a value using the `--serverport` parameter, the installer will use port 5444, or the first available port after port 5444 as the default listener port.

```
--server_utilization {33 | 66 | 100}
```

Use the `--server_utilization` parameter to specify a value for the `edb_dynatune` configuration parameter. The `edb_dynatune` configuration parameter determines how Advanced Server allocates system resources.

- A value of 33 is appropriate for a system used for development. A low value dedicates the least amount of the host machine's resources to the database server.
- A value of 66 is appropriate for an application server with a fixed number of applications. A mid-range value dedicates a moderate amount of system resources to the database server. The default value is 66.
- A value of 100 is appropriate for a host machine that is dedicated to running Advanced Server. A high value dedicates most of the system resources to the database server.

When the installation is complete, you can adjust the value of `edb_dynatune` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for the changes to take effect.

```
--serviceaccount user_account_name
```

Use the `--serviceaccount` parameter to specify the name of the user account that owns the server process.

- If `--databasemode` is set to `oracle` (the default), the default value of `--serviceaccount` is `enterprisedb`.

- If `--databasemode` is `postgresql`, the default value of `--serviceaccount` is set to `postgres`.

Please note that for security reasons, the `--serviceaccount` parameter must specify the name of an account that does not hold administrator privileges.

If you specify both the `--serviceaccount` option and the `--enable_acledit 1` option when invoking the installer, the database service and pgAgent will use the same service account, thereby having the required permissions to access the Advanced Server binaries and data directory.

Note: If you are installing on a Windows host, see the `--enable_acledit` option in this section for additional information. Specification of the `--enable_acledit` option is permitted only when installing on Windows.

Please note that on Windows hosts, if you do not include the `--serviceaccount` option when invoking the installer, the `NetworkService` account will own the database service, and the pgAgent service will be owned by either `enterprisedb` or `postgres` (depending on the installation mode).

`--servicename service_name`

Use the `--servicename` parameter to specify the name of the Advanced Server service. The default is `edb-as-11`.

`--servicepassword user_password`

Use `--servicepassword` to specify the OS system password. If unspecified, the value of `--servicepassword` defaults to the value of `--superpassword`.

`--superaccount super_user_name`

Use the `--superaccount` parameter to specify the user name of the database superuser.

- If `--databasemode` is set to `oracle` (the default), the default value of `--superaccount` is `enterprisedb`.
- If `--databasemode` is set to `postgresql`, the default value of `--superaccount` is set to `postgres`

`--superpassword superuser_password`

Use `--superpassword` to specify the database superuser password. If you are installing in non-interactive mode, `--superpassword` defaults to `enterprisedb`.

```
--unattendedmodeui { none | minimal | minimalWithDialogs }
```

Use the `--unattendedmodeui` parameter to specify installer behavior during an unattended installation.

Include `--unattendedmodeui none` to specify that the installer should not display progress bars during the Advanced Server installation.

Include `--unattendedmodeui minimal` to specify that the installer should display progress bars during the installation process. This is the default behavior.

Include `--unattendedmodeui minimalWithDialogs` to specify that the installer should display progress bars and report any errors encountered during the installation process (in additional dialogs).

```
--version
```

Include the `--version` parameter to retrieve version information about the installer:

```
EDB Postgres Advanced Server 11 --- Built on 2018-03-15
00:04:00 IB: 15.11.1-201511121057
```

```
--workload_profile {oltp | mixed | reporting}
```

Use the `--workload_profile` parameter to specify an initial value for the `edb_dynatune_profile` configuration parameter. `edb_dynatune_profile` controls aspects of performance-tuning based on the type of work that the server performs.

- Specify `oltp` if the Advanced Server installation will be used to support heavy online transaction processing workloads.
- The default value is `oltp`.
- Specify `mixed` if Advanced Server will provide a mix of transaction processing and data reporting.
- Specify `reporting` if Advanced Server will be used for heavy data reporting.

After the installation is complete, you can adjust the value of `edb_dynatune_profile` by editing the `postgresql.conf` file, located in the `data` directory of your Advanced Server installation. After editing the `postgresql.conf` file, you must restart the server for the changes to take effect.

For more information about `edb_dynatune` and other performance-related topics, see the *EDB Postgres Advanced Server Guide* available at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

Beta

4.4 Using StackBuilder Plus

Please note: StackBuilder Plus is supported only on Windows systems.

The StackBuilder Plus utility provides a graphical interface that simplifies the process of updating, downloading and installing modules that complement your Advanced Server installation. When you install a module with StackBuilder Plus, StackBuilder Plus automatically resolves any software dependencies.

You can invoke StackBuilder Plus at any time after the installation has completed by selecting the StackBuilder Plus menu option from the Apps menu. Enter your system password (if prompted), and the StackBuilder Plus welcome window opens (shown in Figure 4.20).

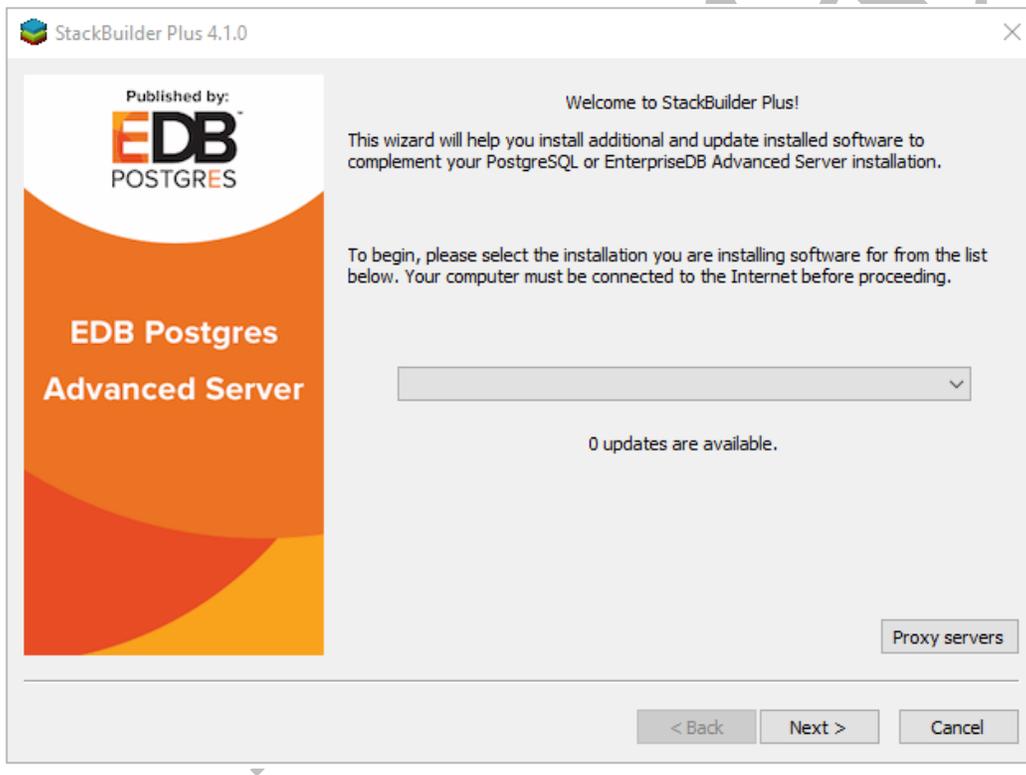


Figure 4.20 -The StackBuilder Plus welcome window.

Use the drop-down listbox on the welcome window to select your Advanced Server installation.

StackBuilder Plus requires Internet access; if your installation of Advanced Server resides behind a firewall (with restricted Internet access), StackBuilder Plus can download program installers through a proxy server. The module provider determines if the module can be accessed through an HTTP proxy or an FTP proxy; currently, all updates are transferred via an HTTP proxy and the FTP proxy information is not used.

If the selected Advanced Server installation has restricted Internet access, use the `Proxy Servers` button on the `Welcome` window to open the `Proxy servers` dialog (shown in Figure 4.21).

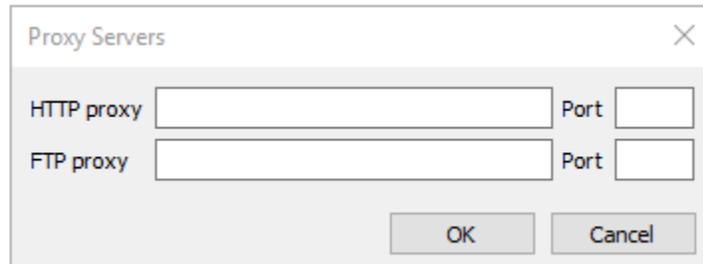


Figure 4.21 –The Proxy servers dialog.

Enter the IP address and port number of the proxy server in the `HTTP proxy` on the `Proxy servers` dialog. Currently, all StackBuilder Plus modules are distributed via HTTP proxy (FTP proxy information is ignored). Click `OK` to continue.

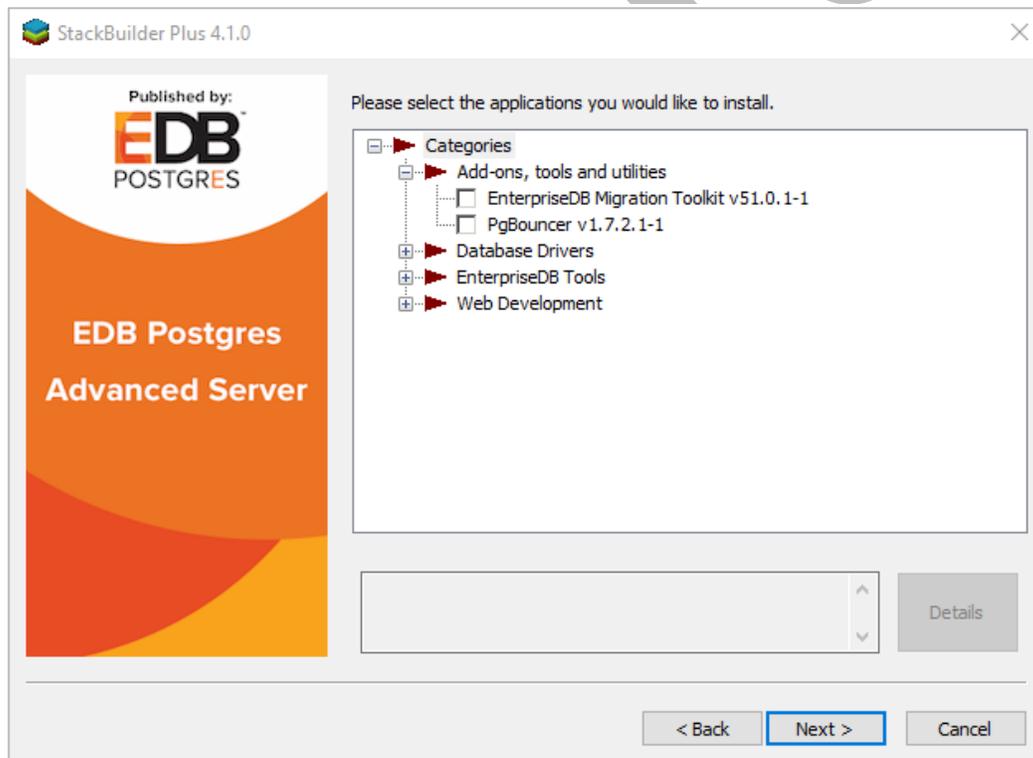


Figure 4.22 –The StackBuilder Plus module selection window.

The tree control on the StackBuilder Plus module selection window (shown in Figure 4.22) displays a node for each module category.

Expand a module, and highlight a component name in the tree control to review a detailed description of the component. To add one or more components to the installation or to upgrade a component, check the box to the left of a module name and click `Next`.

StackBuilder Plus confirms the packages selected (Figure 4.23).

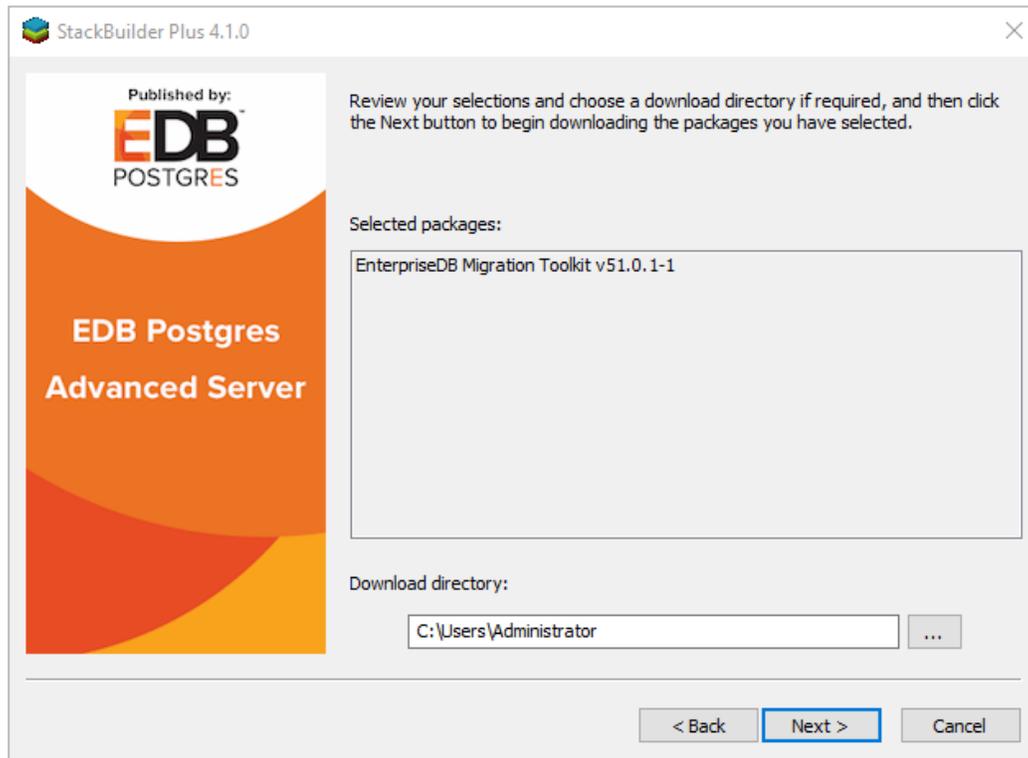


Figure 4.23 -A summary window displays a list of selected packages.

Use the browse button (...) to the right of the `Download directory` field to open a file selector, and choose an alternate location to store the downloaded installers. Click `Next` to connect to the server and download the required installation files.

When the download completes, a window opens that confirms the installation files have been downloaded and are ready for installation (see Figure 4.24).

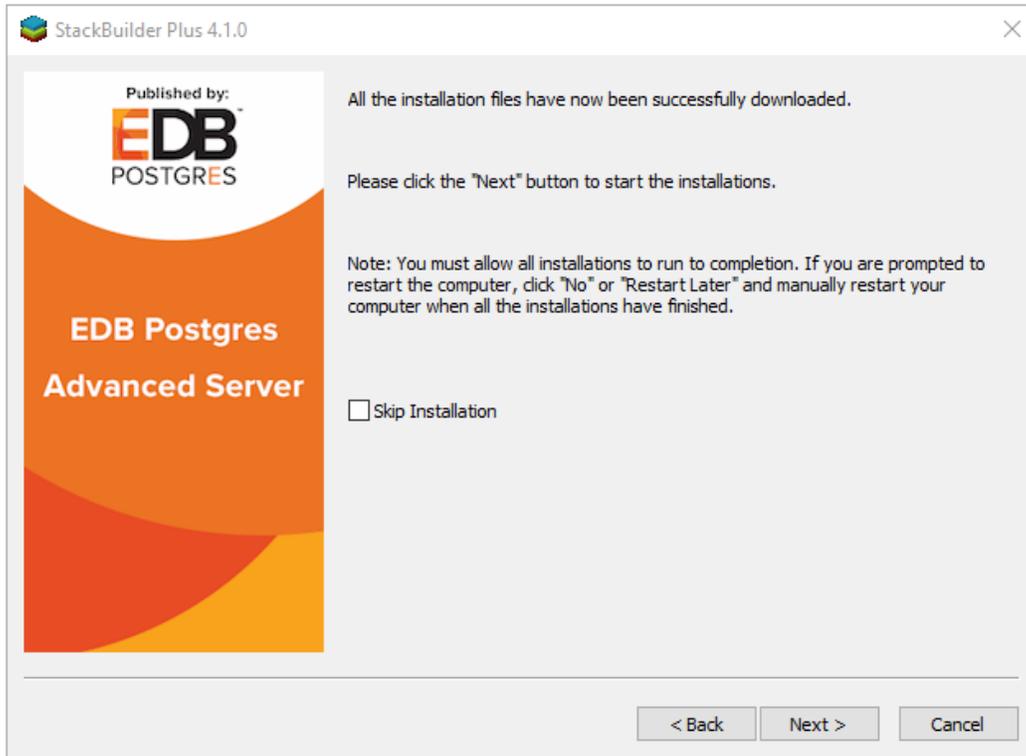


Figure 4.24 -Confirmation that the download process is complete.

You can check the box next to `Skip Installation`, and select `Next` to exit StackBuilder Plus without installing the downloaded files, or leave the box unchecked and click `Next` to start the installation process.

Each downloaded installer has different requirements. As the installers execute, they may prompt you to confirm acceptance of license agreements, to enter passwords, and provide configuration information.

During the installation process, you may be prompted by one (or more) of the installers to restart your system. Select `No` or `Restart Later` until all installations are completed. When the last installation has completed, reboot the system to apply all of the updates.

You may occasionally encounter packages that don't install successfully. If a package fails to install, StackBuilder Plus will alert you to the installation error with a popup dialog, and write a message to the log file at `%TEMP%`.

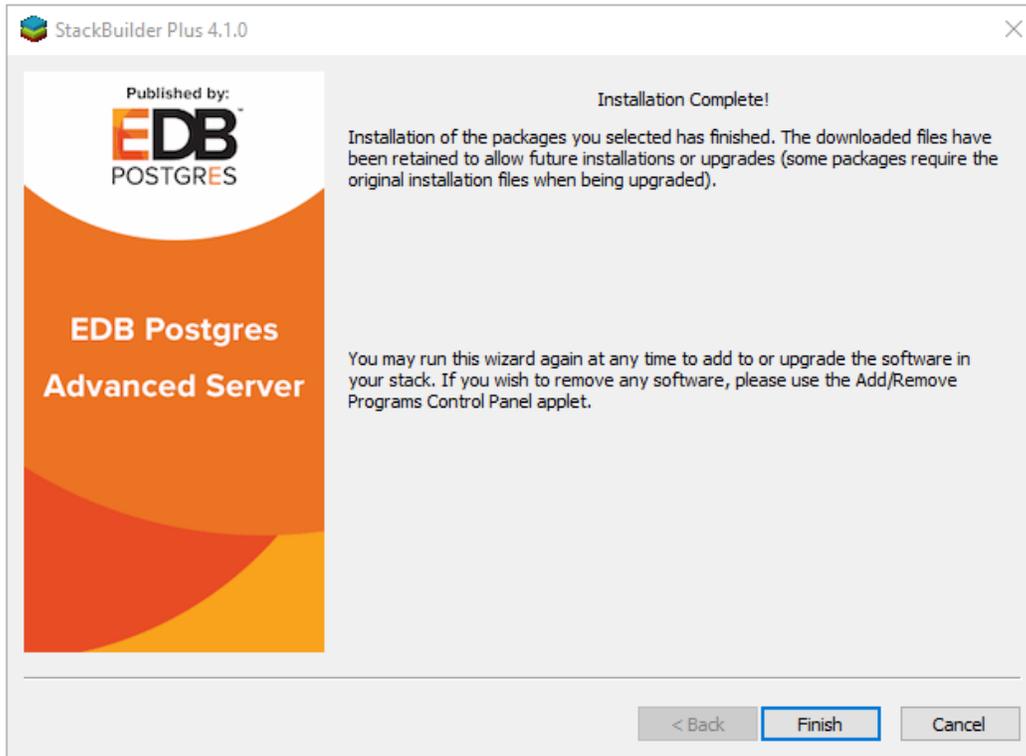


Figure 4.25 -StackBuilder Plus confirms the completed installation.

When the installation is complete, StackBuilder Plus will alert you to the success or failure of the installations of the requested packages (see Figure 4.25). If you were prompted by an installer to restart your computer, reboot now.

4.5 Using the Update Monitor

The Update Monitor utility polls the EnterpriseDB website and alerts you to security updates and enhancements for Windows hosts as they become available. Update Monitor is automatically installed by the graphical installer.

When Update Monitor is actively monitoring, the Postgres elephant icon is displayed in the system tray (see Figure 4.26).



Figure 4.26 -The Update Monitor icon.

If you have installed more than one version of Advanced Server, Update Monitor watches for updates and alerts for all installed versions. When Update Monitor finds an update or alert, it displays an alert symbol to let you know that an update or alert is available for one of the Advanced Server installations (see Figure 4.27).



Figure 4.27 -The Update Monitor icon displays an alert.

Right click on the symbol to access the Update Monitor context menu (shown in Figure 4.28).

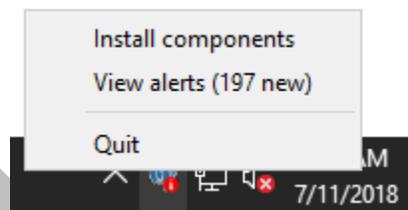


Figure 4.28 -The Update Monitor context menu.

Click `Install components` to open StackBuilder Plus, and check for component updates. See Section [4.4 Using StackBuilder Plus](#) for detailed information about the update process.

If one or more alerts are available for your Advanced Server installation, the context menu displays the alert count. Select `View alerts` to display the EnterpriseDB Advanced Server Alerts window (see Figure 4.29).

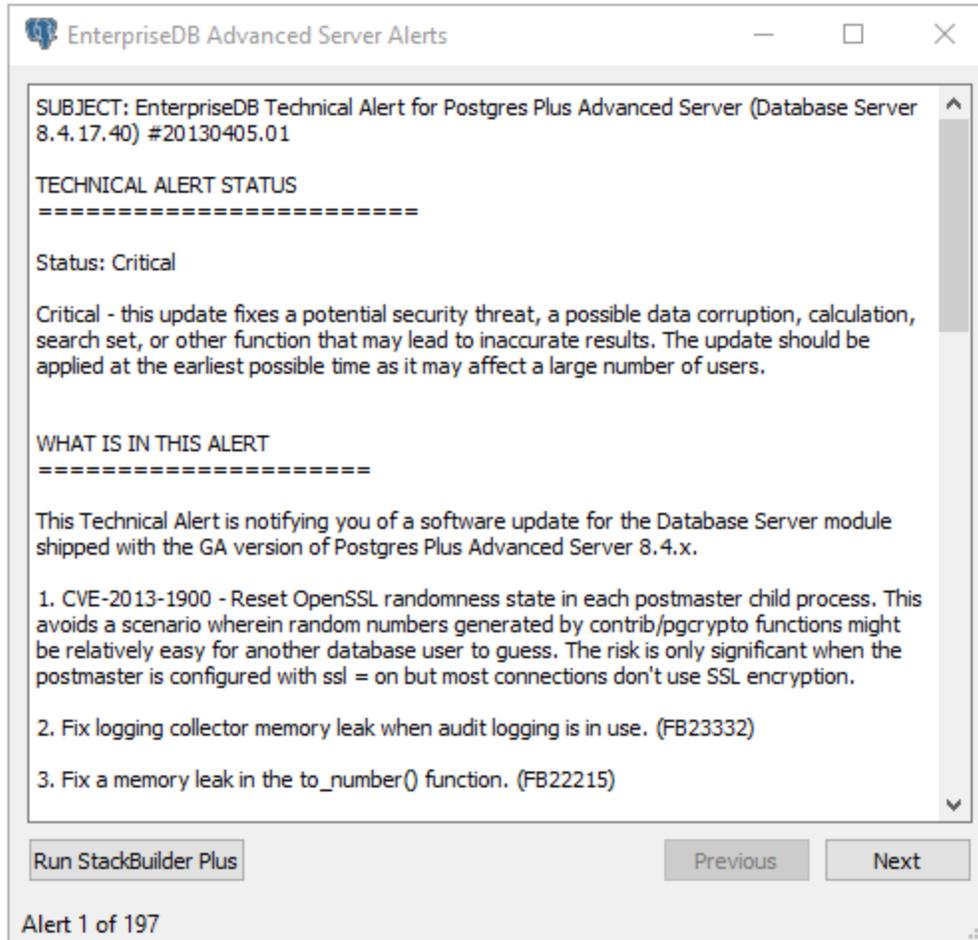


Figure 4.29 - An EnterpriseDB Technical alert.

The technical alert will provide information about updates and issues that might impact your installation.

- Click the `Run StackBuilder Plus` button to open StackBuilder Plus and install relevant software updates.
- Use the `Next` button to access each successive alert.

When you've reviewed all of the alerts, the icon no longer displays the alert symbol; use the X icon in the upper-right corner of the dialog to close Update Monitor.

4.6 Installation Troubleshooting

Difficulty Displaying Java-based Applications

If you encounter difficulty displaying Java-based server features (controls or text not being displayed correctly, or blank windows), upgrading to the latest `libxcb-xlib` libraries should correct the problem on most distributions. Please visit the following link for other possible work-arounds:

http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6532373

--mode unattended Authentication Errors

Authentication errors from component modules during unattended installations may indicate that the specified values of `--servicepassword` or `--superpassword` may be incorrect.

Errors During an Advanced Server Installation on Windows

If you encounter an error during the installation process on a Windows system, exit the installation, and ensure that your version of Windows is up-to-date. After applying any outstanding operating system updates, re-invoke the Advanced Server installer.

The Installation Fails to Complete Due to Existing data Directory Contents

If an installation fails to complete due to existing content in the data directory, the server will write an error message to the server logs:

```
A data directory is neither empty, or a recognisable data
directory.
```

If you encounter a similar message, you should confirm that the data directory is empty; the presence of files (including the system-generated `lost+found` folder) will prevent the installation from completing. Either remove the files from the data directory, or specify a different location for the data directory before re-invoking the installer to complete the installation.

5 Managing an Advanced Server Installation

Unless otherwise noted, the commands and paths noted in the following section assume that you have performed an installation with the interactive installer.

5.1 Starting and Stopping Advanced Server and Supporting Components

A service is a program that runs in the background and requires no user interaction (in fact, a service provides no user interface); a service can be configured to start at boot time, or manually on demand. Services are best controlled using the platform-specific operating system service control utility. Many of the Advanced Server supporting components are services.

The following table lists the names of the services that control Advanced Server and services that control Advanced Server supporting components:

Advanced Server Component Name	Linux Service Name	Windows Service Name
Advanced Server	edb-as-11	edb-as-11
pgAgent	edb-pgagent-11	EDB Postgres Advanced Server 11 Scheduling Agent
PgBouncer	edb-pgbouncer-1.7	edb-pgbouncer-1.7
pgPool-II	edb-pgpool-3.5	N/A
Slony	edb-slony-replication-11	edb-slony-replication-11

You can use the command line or an operating system specific graphical interface to control Advanced Server's database server and the services of Advanced Server's supporting components.

5.2 Controlling a Service on Linux

The commands that control the Advanced Server service on a Linux platform are version specific.

5.2.1 Controlling a Service on CentOS or RHEL 7.x

If your installation of Advanced Server resides on version 7.x of RHEL and CentOS, you must use the `systemctl` command to control the Advanced Server service and supporting components.

The `systemctl` command must be in your search path and must be invoked with superuser privileges. To use the command, open a command line, and enter:

```
systemctl action service_name
```

Where:

action

action specifies the action taken by the service command. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `status` to discover the current status of the service.

service_name

service_name specifies the name of the service.

5.2.2 Controlling a Service on CentOS or RHEL 6.x

On version 6.x of RHEL or CentOS Linux, you can control a service at the command line with the `service` command. The `service` command can be used to manage an Advanced Server cluster, as well as the services of component software installed with Advanced Server.

Using the `service` command to change the status of a service allows the Linux service controller to keep track of the server status (the `pg_ctl` command does not alert the service controller to changes in the status of a server). The command must be in your search path and must be invoked with superuser privileges. Open a command line, and issue the command:

```
service service_name action
```

The Linux `service` command invokes a script (with the same name as the service) that resides in `/etc/init.d`. If your Linux distribution does not support the `service` command, you can call the script directly by entering:

```
/etc/init.d/service_name action
```

Where:

`service_name`

`service_name` specifies the name of the service.

`action`

`action` specifies the action taken by the service command. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `condstop` to stop the service without displaying a notice if the server is already stopped.
- `restart` to stop and then start the service.
- `condrestart` to restart the service without displaying a notice if the server is already stopped.
- `try-restart` to restart the service without displaying a notice if the server is already stopped.
- `status` to discover the current status of the service.

5.2.3 Using `pg_ctl` to Control Advanced Server

You can use the `pg_ctl` utility to control an Advanced Server service from the command line on any platform. `pg_ctl` allows you to start, stop, or restart the Advanced Server database server, reload the configuration parameters, or display the status of a running server. To invoke the utility, assume the identity of the cluster owner, navigate into the home directory of Advanced Server, and issue the command:

```
./bin/pg_ctl -D data_directory action
```

data_directory

data_directory is the location of the data controlled by the Advanced Server cluster.

action

action specifies the action taken by the `pg_ctl` utility. Specify:

- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.
- `reload` sends the server a `SIGHUP` signal, reloading configuration parameters
- `status` to discover the current status of the service.

For more information about using the `pg_ctl` utility, or the command line options available, please see the official PostgreSQL Core Documentation available at:

<https://www.postgresql.org/docs/11/static/app-pg-ctl.html>

Choosing Between `pg_ctl` and the `service` Command

You can use the `pg_ctl` utility to manage the status of an Advanced Server cluster, but it is important to note that `pg_ctl` does not alert the operating system service controller to changes in the status of a server, so it is beneficial to use the `service` command whenever possible.

5.2.4 Configuring Component Services to AutoStart at System Reboot

After installing, configuring and starting the services of Advanced Server supporting components on a Linux system, you must manually configure your system to autostart the service when your system reboots. To configure a service to autostart on a Linux system, open a command line, assume superuser privileges, and enter the following command.

On a Redhat-compatible Linux system:

```
/sbin/chkconfig service_name on
```

Where *service_name* specifies the name of the service.

On Windows, you can use the `Service Properties` dialog to control the service startup type. For more information about controlling startup behavior on Windows, see Section [5.3.1](#).

5.3 Controlling a Service on Windows

The Windows operating system includes a graphical service controller that offers control of Advanced Server and the services associated with Advanced Server components. The `Services` utility can be accessed through the `Administrative Tools` section of the `Windows Control Panel`.

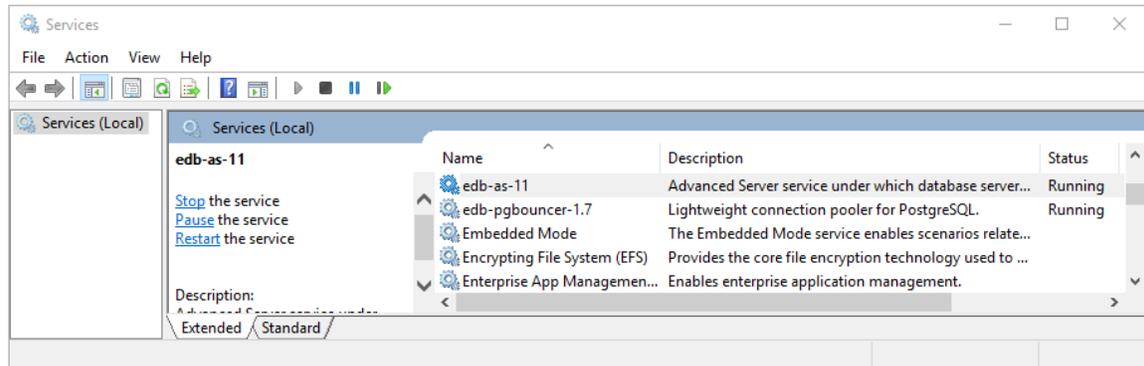


Figure 5.1 - The Advanced Server service in the Windows Services window.

The `Services` window displays an alphabetized list of services; the `edb-as-11` service controls Advanced Server (see Figure 5.1).

- Use the `Stop the service` option to stop the instance of Advanced Server. Please note that any user (or client application) connected to the Advanced Server instance will be abruptly disconnected if you stop the service.
- Use the `Start the service` option to start the Advanced Server service.
- Use the `Pause the service` option to tell Advanced Server to reload the server configuration parameters without disrupting user sessions for many of the configuration parameters. See Section 6, *Configuring Advanced Server* for more information about the parameters that can be updated with a server reload.
- Use the `Restart the service` option to stop and then start the Advanced Server. Please note that any user sessions will be terminated when you stop the service. This option is useful to reset server parameters that only take effect on server start.

5.3.1 Controlling Server Startup Behavior on Windows

You can use the Windows Services utility to control the startup behavior of the server. Right click on the name of the service you wish to update, and select `Properties` from the context menu to open the `Properties` dialog.

Use the drop-down listbox in the `Startup type` field (shown in Figure 5.2) to specify how the Advanced Server service will behave when the host starts.

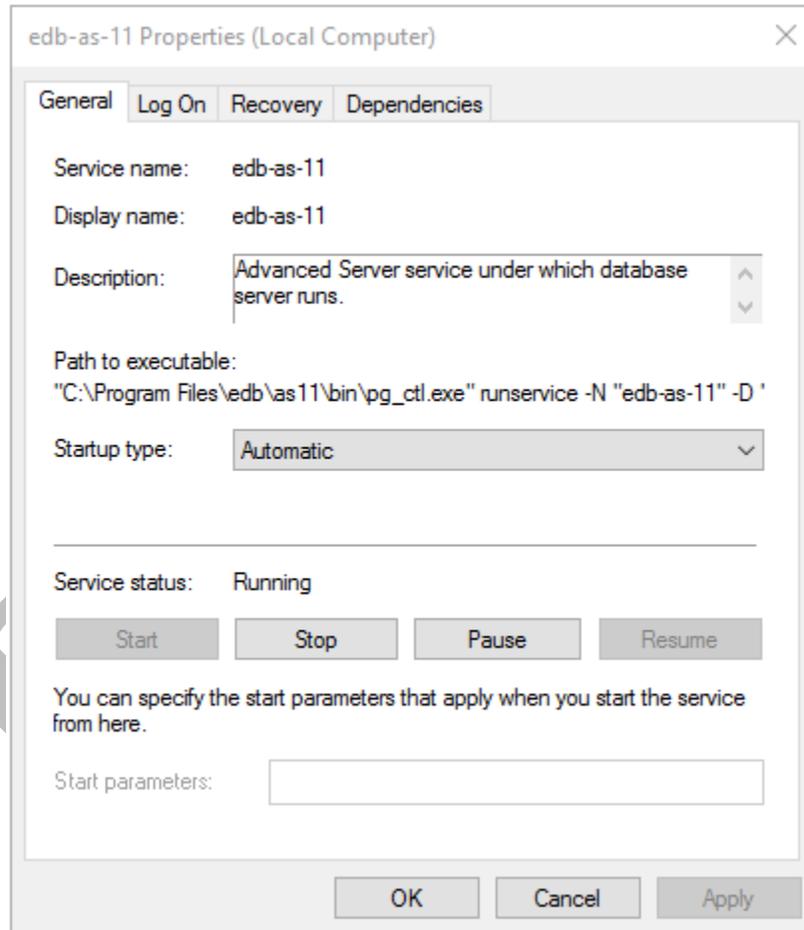


Figure 5.2 - Specifying Advanced Server's startup behavior.

- Specify `Automatic (Delayed Start)` to instruct the service controller to start after boot.
- Specify `Automatic` to instruct the service controller to start and stop the server whenever the system starts or stops.
- Specify `Manual` to instruct the service controller that the server must be started manually.

- Specify `Disabled` to instruct the service controller to disable the service; after disabling the service, you must stop the service or restart the server to make the change take effect. Once disabled, the server's status cannot be changed until `Startup type` is reset to `Automatic (Delayed Start)`, `Automatic` or `Manual`.

Beta

6 Configuring Advanced Server

You can easily update parameters that determine the behavior of Advanced Server and supporting components by modifying the following configuration files:

- The `postgresql.conf` file determines the initial values of Advanced Server configuration parameters.
- The `pg_hba.conf` file specifies your preferences for network authentication and authorization.
- The `pg_ident.conf` file maps operating system identities (user names) to Advanced Server identities (roles) when using `ident`-based authentication.

You can use your editor of choice to open a configuration file, or on Windows, navigate through the EDB Postgres menu to open a file (see Figure 6.1).

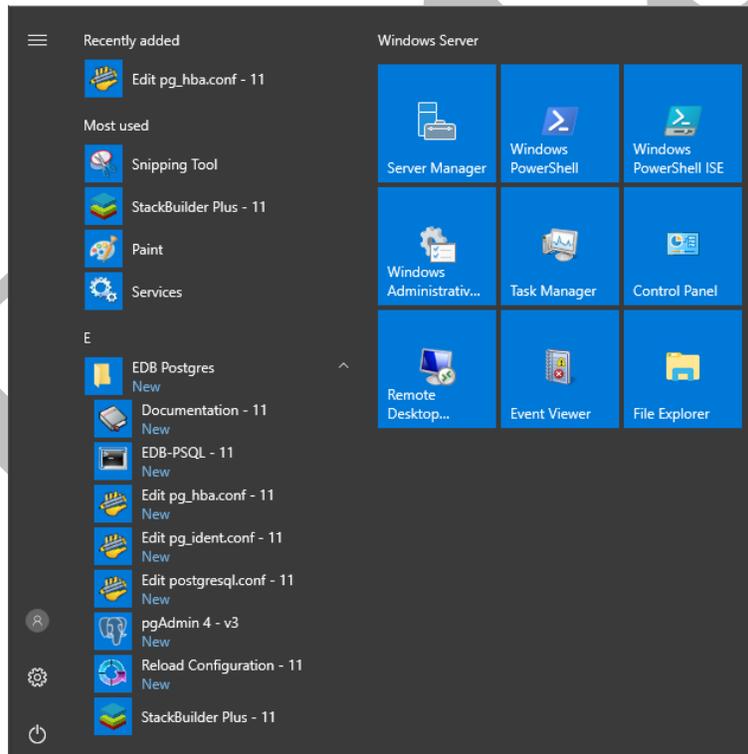
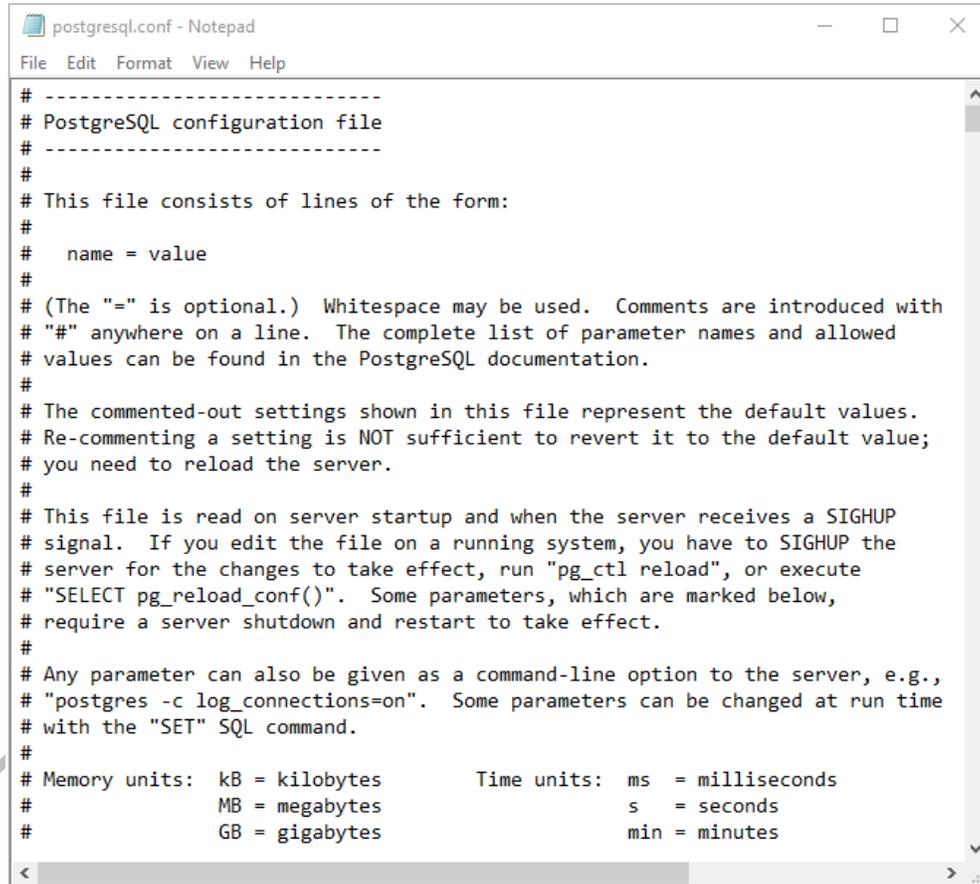


Figure 6.1 - Accessing the configuration files through the Windows system menu.

6.1 Modifying the `postgresql.conf` File

Configuration parameters in the `postgresql.conf` file specify server behavior with regards to auditing, authentication, encryption, and other behaviors. The `postgresql.conf` file resides in the `data` directory under your Advanced Server installation.



```

postgresql.conf - Notepad
File Edit Format View Help
# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.) Whitespace may be used. Comments are introduced with
# "#" anywhere on a line. The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal. If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, run "pg_ctl reload", or execute
# "SELECT pg_reload_conf()". Some parameters, which are marked below,
# require a server shutdown and restart to take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on". Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units: kB = kilobytes      Time units: ms = milliseconds
#                MB = megabytes      s = seconds
#                GB = gigabytes      min = minutes

```

Figure 6.2 - The `postgresql.conf` file.

Parameters that are preceded by a pound sign (`#`) are set to their default value (as shown in the parameter setting). To change a parameter value, remove the pound sign and enter a new value. After setting or changing a parameter, you must either *reload* or *restart* the server for the new parameter value to take effect.

Within the `postgresql.conf` file, some parameters contain comments that indicate change requires restart (see Figure 6.2). To view a list of the parameters that require a server restart, execute the following query at the `psql` command line:

```
SELECT name FROM pg_settings WHERE context = 'postmaster';
```

6.2 Modifying the `pg_hba.conf` File

Appropriate authentication methods provide protection and security. Entries in the `pg_hba.conf` file specify the authentication method or methods that the server will use when authenticating connecting clients. Before connecting to the server, you may be required to modify the authentication properties specified in the `pg_hba.conf` file.

When you invoke the `initdb` utility to create a cluster, `initdb` creates a `pg_hba.conf` file for that cluster that specifies the type of authentication required from connecting clients.

The default authentication configuration specified in the `pg_hba.conf` file is:

```
# TYPE      DATABASE         USER                ADDRESS             METHOD
# "local" is for Unix domain socket connections only
local      all              all                 md5
# IPv4 local connections:
host       all              all                 127.0.0.1/32       md5
# IPv6 local connections:
host       all              all                 ::1/128             md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local     replication     enterisedb          md5
#host      replication     enterisedb          127.0.0.1/32       md5
#host      replication     enterisedb          ::1/128             md5
```

To modify the `pg_hba.conf` file, open the file with your choice of editor. After modifying the authentication settings in the `pg_hba.conf` file, use the Windows services utility or the Linux command line to restart the server and apply the changes.

For more information about authentication, and modifying the `pg_hba.conf` file, see the PostgreSQL Core Documentation at:

<https://www.postgresql.org/docs/11/static/auth-pg-hba-conf.html>

6.3 Setting Advanced Server Environment Variables

The graphical installer provides a script that simplifies the task of setting environment variables for Windows users. The script sets the environment variables for your current shell session; when your shell session ends, the environment variables are destroyed. You may wish to invoke `pgplus_env.bat` from your system-wide shell startup script, so that environment variables are automatically defined for each shell session.

The `pgplus_env` script is created during the Advanced Server installation process and reflects the choices made during installation. To invoke the script, open a command line and enter:

```
C:\Program Files\edb\as11\pgplus_env.bat
```

As the `pgplus_env.bat` script executes, it sets the following environment variables:

```
PATH="C:\Program Files\edb\as11\bin";%PATH%
EDBHOME=C:\Program Files\edb\as11
PGDATA=C:\Program Files\edb\as11\data
PGDATABASE=edb
REM @SET PGUSER=enterprisedb
PGPORT=5444
PGLOCALEDIR=C:\Program Files\edb\as11\share\locale
```

If you have used an installer created by EnterpriseDB to install PostgreSQL, the `pg_env` script performs the same function:

```
C:\Progra~1\PostgreSQL\11\pg_env.bat
```

As the `pg_env.bat` script executes on PostgreSQL, it sets the following environment variables:

```
PATH="C:\Program Files\PostgreSQL\11\bin";%PATH%
PGDATA=C:\Program Files\PostgreSQL\11\data
PGDATABASE=postgres
PGUSER=postgres
PGPORT=5432
PGLOCALEDIR=C:\Program Files\PostgreSQL\11\share\locale
```

6.4 Connecting to Advanced Server with psql

`psql` is a command line client application that allows you to execute SQL commands and view the results. To open the `psql` client, the client must be in your search path. The executable resides in the `bin` directory, under your Advanced Server installation.

Use the following command and options to start the `psql` client (see Figure 6.3):

```
psql -d edb -U enterprisedb
```

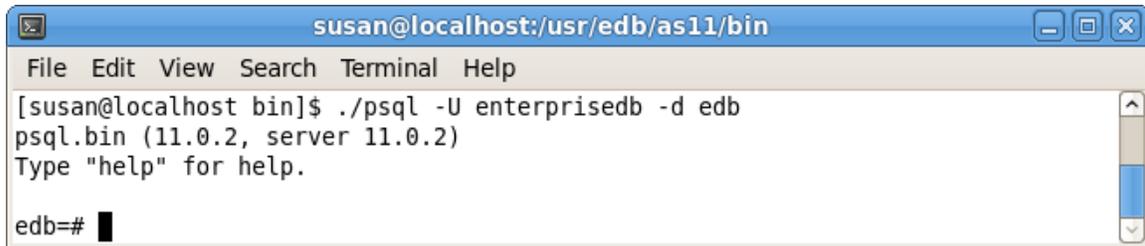


Figure 6.3 - Connecting to the server.

Where:

- d specifies the database to which `psql` will connect;
- U specifies the identity of the database user that will be used for the session.

If you have performed an installation with the interactive installer, you can access the `psql` client by selecting `EDB-PSQL` from the `EDB Postgres` menu. When the client opens, provide connection information for your session.

For more information about using the command line client, please refer to the PostgreSQL Core Documentation at:

<https://www.postgresql.org/docs/11/static/app-psql.html>

6.5 Connecting to Advanced Server with the pgAdmin 4 Client

pgAdmin 4 provides an interactive graphical interface that you can use to manage your database and database objects. Easy-to-use dialogs and online help simplify tasks such as object creation, role management, and granting or revoking privileges. The tabbed browser panel provides quick access to information about the object currently selected in the pgAdmin tree control.

The client is distributed with the graphical installer (for Windows). To open pgAdmin, select pgAdmin4 from the EDB Postgres menu. The client opens in your default browser (see Figure 6.4).

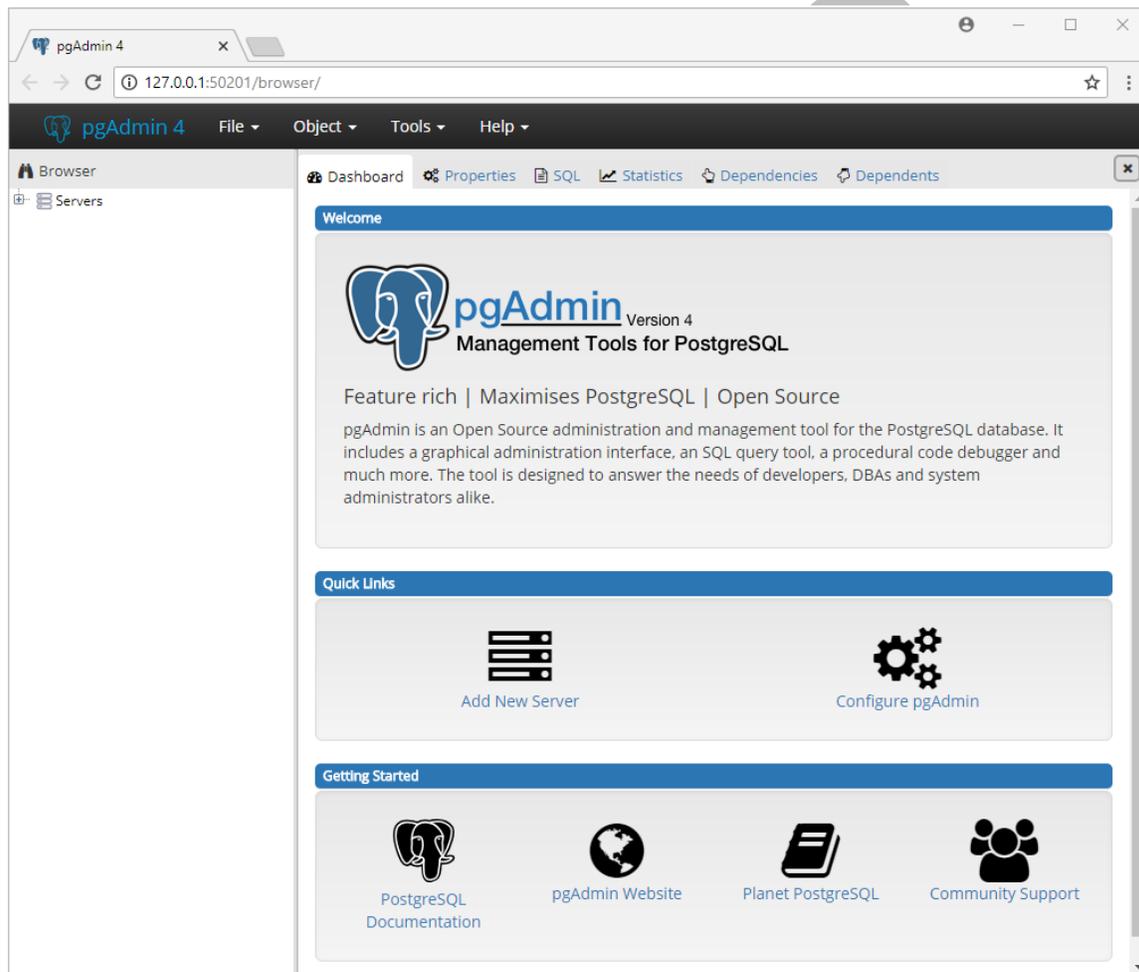


Figure 6.4 – The pgAdmin 4 client Dashboard.

To connect to the Advanced Server database server, expand the `Servers` node of the `Browser` tree control, and right click on the `EDB Postgres Advanced Server` node. When the context menu opens, select `Connect Server`. The `Connect to Server` dialog opens (see Figure 6.5).

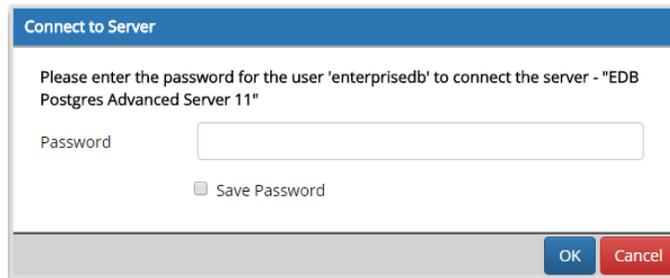


Figure 6.5 – The Connect to Server dialog.

Provide the password associated with the database superuser in the `Password` field, and click `OK` to connect.

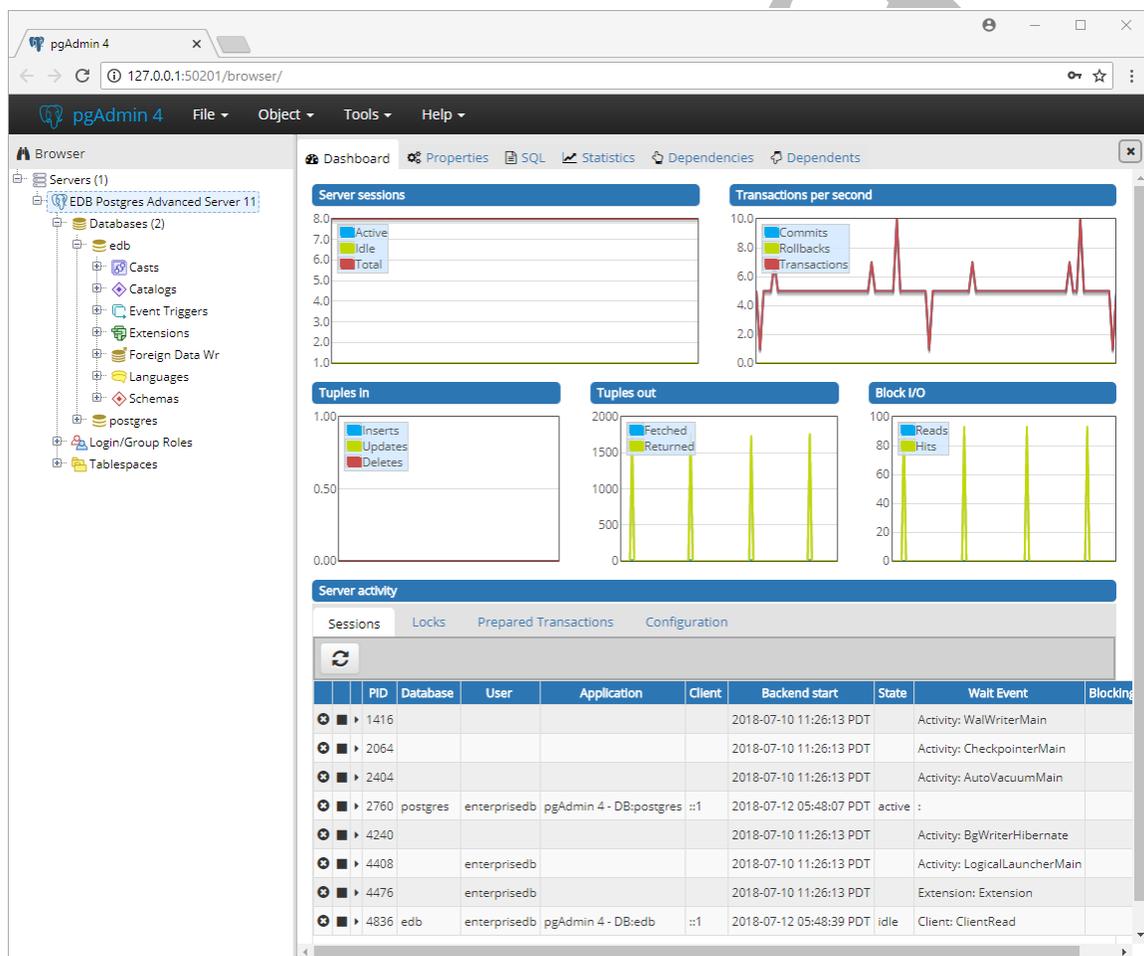


Figure 6.6 – Connecting to an Advanced Server database.

When the client connects (see Figure 6.6), you can use the `Browser` tree control to retrieve information about existing database objects, or to create new objects. For more information about using the `pgAdmin` client, use the `Help` drop-down menu to access the online help files.

7 Upgrading an Installation With `pg_upgrade`

While minor upgrades between versions are fairly simple, and require only the installation of new executables, past major version upgrades have been both expensive and time consuming. `pg_upgrade` facilitates migration between any version of Advanced Server (version 9.0 or later), and any subsequent release of Advanced Server that is supported on the same platform.

Without `pg_upgrade`, to migrate from an earlier version of Advanced Server to Advanced Server 11, you must export all of your data using `pg_dump`, install the new release, run `initdb` to create a new cluster, and then import your old data.

`pg_upgrade` can reduce both the amount of time required and the disk space required for many major-version upgrades.

The `pg_upgrade` utility performs an in-place transfer of existing data between Advanced Server and any subsequent version.

Several factors determine if an in-place upgrade is practical:

- The on-disk representation of user-defined tables must not change between the original version and the upgraded version.
- The on-disk representation of data types must not change between the original version and the upgraded version.
- To upgrade between major versions of Advanced Server with `pg_upgrade`, both versions must share a common binary representation for each data type. Therefore, you cannot use `pg_upgrade` to migrate from a 32-bit to a 64-bit Linux platform.

Before performing a version upgrade, `pg_upgrade` will verify that the two clusters (the old cluster and the new cluster) are compatible.

If the upgrade involves a change in the on-disk representation of database objects or data, or involves a change in the binary representation of data types, `pg_upgrade` will be unable to perform the upgrade; to upgrade, you will have to `pg_dump` the old data and then import that data into the new cluster.

The `pg_upgrade` executable is distributed with Advanced Server 11, and is installed as part of the Database Server component; no additional installation or configuration steps are required.

7.1 Performing an Upgrade - Overview

To upgrade an earlier version of Advanced Server to the current version, you must:

- Install the current version of Advanced Server. The new installation must contain the same supporting server components as the old installation.
- Empty the target database or create a new target cluster with `initdb`.
- Place the `pg_hba.conf` file for both databases in `trust` authentication mode (to avoid authentication conflicts).
- Shut down the old and new Advanced Server services.
- Invoke the `pg_upgrade` utility.

When `pg_upgrade` starts, it performs a compatibility check to ensure that all required executables are present and contain the expected version numbers. The verification process also checks the old and new `$PGDATA` directories to ensure that the expected files and subdirectories are in place. If the verification process succeeds, `pg_upgrade` starts the old `postmaster` and runs `pg_dumpall --schema-only` to capture the metadata contained in the old cluster. The script produced by `pg_dumpall` is used in a later step to recreate all user-defined objects in the new cluster.

Note that the script produced by `pg_dumpall` recreates only user-defined objects and not system-defined objects. The new cluster will *already* contain the system-defined objects created by the latest version of Advanced Server.

After extracting the metadata from the old cluster, `pg_upgrade` performs the bookkeeping tasks required to sync the new cluster with the existing data.

`pg_upgrade` runs the `pg_dumpall` script against the new cluster to create (empty) database objects of the same shape and type as those found in the old cluster. Then, `pg_upgrade` links or copies each table and index from the old cluster to the new cluster.

Please note: if you are upgrading from a version of Advanced Server prior to 9.5 to Advanced Server 11 and have installed the `edb_dblink_oci` or `edb_dblink_libpq` extension, you must drop the extension before performing an upgrade. To drop the extension, connect to the server with the `psql` or PEM client, and invoke the commands:

```
DROP EXTENSION edb_dblink_oci;
DROP EXTENSION edb_dblink_libpq;
```

When you've completed upgrading, you can use the `CREATE EXTENSION` command to add the current versions of the extensions to your installation.

7.1.1 Linking versus Copying

When invoking `pg_upgrade`, you can use a command-line option to specify whether `pg_upgrade` should *copy* or *link* each table and index in the old cluster to the new cluster.

Linking is much faster because `pg_upgrade` simply creates a second name (a hard link) for each file in the cluster; linking also requires no extra workspace because `pg_upgrade` does not make a copy of the original data. When linking the old cluster and the new cluster, the old and new clusters share the data; note that after starting the new cluster, your data can no longer be used with the previous version of Advanced Server.

If you choose to copy data from the old cluster to the new cluster, `pg_upgrade` will still reduce the amount of time required to perform an upgrade compared to the traditional `dump/restore` procedure. `pg_upgrade` uses a file-at-a-time mechanism to copy data files from the old cluster to the new cluster (versus the row-by-row mechanism used by `dump/restore`). When you use `pg_upgrade`, you avoid building indexes in the new cluster; each index is simply copied from the old cluster to the new cluster. Finally, using a `dump/restore` procedure to upgrade requires a great deal of workspace to hold the intermediate text-based dump of all of your data, while `pg_upgrade` requires very little extra workspace.

Data that is stored in user-defined tablespaces is not copied to the new cluster; it stays in the same location in the file system, but is copied into a subdirectory whose name reflects the version number of the new cluster. To manually relocate files that are stored in a tablespace after upgrading, move the files to the new location and update the symbolic links (located in the `pg_tblspc` directory under your cluster's `data` directory) to point to the files.

7.2 Invoking `pg_upgrade`

When invoking `pg_upgrade`, you must specify the location of the old and new cluster's PGDATA and executable (`/bin`) directories, as well as the name of the Advanced Server superuser, and the ports on which the installations are listening. A typical call to invoke `pg_upgrade` to migrate from Advanced Server 10 to Advanced Server 11 takes the form:

```
pg_upgrade --old-datadir path_to_10_data_directory --new-
datadir path_to_11_data_directory --user superuser_name --
old-bindir path_to_10_bin_directory --new-bindir
path_to_11_bin_directory --old-port 10_port --new-port
11_port
```

Where:

```
--old-datadir path_to_10_data_directory
```

Use the `--old-datadir` option to specify the complete path to the data directory within the Advanced Server 10 installation.

```
--new-datadir path_to_11_data_directory
```

Use the `--new-datadir` option to specify the complete path to the data directory within the Advanced Server 11 installation.

```
--username superuser_name
```

Include the `--username` option to specify the name of the Advanced Server superuser. The superuser name should be the same in both versions of Advanced Server. By default, when Advanced Server is installed in Oracle mode, the superuser is named `enterprisedb`. If installed in PostgreSQL mode, the superuser is named `postgres`.

If the Advanced Server superuser name is not the same in both clusters, the clusters will not pass the `pg_upgrade` consistency check.

```
--old-bindir path_to_10_bin_directory
```

Use the `--old-bindir` option to specify the complete path to the `bin` directory in the Advanced Server 10 installation.

```
--new-bindir path_to_11_bin_directory
```

Use the `--new-bindir` option to specify the complete path to the `bin` directory in the Advanced Server 11 installation.

```
--old-port 10_port
```

Include the `--old-port` option to specify the port on which Advanced Server 10 listens for connections.

```
--new-port 11_port
```

Include the `--new-port` option to specify the port on which Advanced Server 11 listens for connections.

Beta

7.2.1 Command Line Options - Reference

`pg_upgrade` accepts the following command line options; each option is available in a long form or a short form:

```
-b path_to_old_bin_directory
--old-bindir path_to_old_bin_directory
```

Use the `-b` or `--old-bindir` keyword to specify the location of the old cluster's executable directory.

```
-B path_to_new_bin_directory
--new-bindir path_to_new_bin_directory
```

Use the `-B` or `--new-bindir` keyword to specify the location of the new cluster's executable directory.

```
-c
--check
```

Include the `-c` or `--check` keyword to specify that `pg_upgrade` should perform a consistency check on the old and new cluster without performing a version upgrade.

```
-d path_to_old_data_directory
--old-datadir path_to_old_data_directory
```

Use the `-d` or `--old-datadir` keyword to specify the location of the old cluster's data directory.

```
-D path_to_new_data_directory
--new-datadir path_to_new_data_directory
```

Use the `-D` or `--new-datadir` keyword to specify the location of the new cluster's data directory.

Please note: Data that is stored in user-defined tablespaces is not copied to the new cluster; it stays in the same location in the file system, but is copied into a subdirectory whose name reflects the version number of the new cluster. To manually relocate files that are stored in a tablespace after upgrading, you must move the files to the new location and update the symbolic links (located in the `pg_tblspc` directory under your cluster's data directory) to point to the files.

```
-j
--jobs
```

Include the `-j` or `--jobs` keyword to specify the number of simultaneous processes or threads to use during the upgrade.

```
-k
--link
```

Include the `-k` or `--link` keyword to create a hard link from the new cluster to the old cluster. See Section [8.1.1, *Linking versus Copying*](#) for more information about using a symbolic link.

```
-o options
--old-options options
```

Use the `-o` or `--old-options` keyword to specify options that will be passed to the old `postgres` command. Enclose options in single or double quotes to ensure that they are passed as a group.

```
-O options
--new-options options
```

Use the `-O` or `--new-options` keyword to specify options to be passed to the new `postgres` command. Enclose options in single or double quotes to ensure that they are passed as a group.

```
-p old_port_number
--old-port old_port_number
```

Include the `-p` or `--old-port` keyword to specify the port number of the Advanced Server installation that you are upgrading.

```
-P new_port_number
--new-port new_port_number
```

Include the `-P` or `--new-port` keyword to specify the port number of the new Advanced Server installation.

Please note: If the original Advanced Server installation is using port number 5444 when you invoke the Advanced Server 11 installer, the installer will recommend using listener port 5445 for the new installation of Advanced Server.

```
-r
--retain
```

During the upgrade process, `pg_upgrade` creates four append-only log files; when the upgrade is completed, `pg_upgrade` deletes these files. Include the `-r` or `--retain` option to specify that the server should retain the `pg_upgrade` log files.

```
-U user_name  
--username user_name
```

Include the `-U` or `--username` keyword to specify the name of the Advanced Server database superuser. The same superuser must exist in both clusters.

```
-v  
--verbose
```

Include the `-v` or `--verbose` keyword to enable verbose output during the upgrade process.

```
-V  
--version
```

Use the `-V` or `--version` keyword to display version information for `pg_upgrade`.

```
-?  
-h  
--help
```

Use `-?`, `-h` or `--help` options to display `pg_upgrade` help information.

7.3 Upgrading to Advanced Server 11 – Step-by-Step

You can use `pg_upgrade` to upgrade from an existing installation of Advanced Server into the cluster built by the Advanced Server 11 installer or into an alternate cluster created using the `initdb` command. In this section, we will provide the details of upgrading into the cluster provided by the installer.

The basic steps to perform an upgrade into an empty cluster created with the `initdb` command are the same as the steps to upgrade into the cluster created by the Advanced Server 11 installer, but you can omit Step 2 (*Empty the `edb` database*), and substitute the location of the alternate cluster when specifying a target cluster for the upgrade.

If a problem occurs during the upgrade process, you can revert to the previous version. See Section [8.5](#), *Reverting to the Old Cluster* for detailed information about this process.

You must be an operating system superuser or Windows Administrator to perform an Advanced Server upgrade.

Step 1 - Install the New Server

Install Advanced Server 11, specifying the same non-server components that were installed during the previous Advanced Server installation. Please note that the new cluster and the old cluster must reside in different directories.

Step 2 - Empty the target database

The target cluster must not contain any data; you can create an empty cluster using the `initdb` command, or you can empty a database that was created during the installation of Advanced Server 11. If you have installed Advanced Server in PostgreSQL mode, the installer creates a single database named `postgres`; if you have installed Advanced Server in Oracle mode, it creates a database named `postgres` and a database named `edb`.

The easiest way to empty the target database is to drop the database and then create a new database. Before invoking the `DROP DATABASE` command, you must disconnect any users and halt any services that are currently using the database.

On Windows, navigate through the Control Panel to the Services manager; highlight each service in the Services list, and select Stop.

On Linux, open a terminal window, assume superuser privileges, and manually stop each service; for example, if you are on Linux 6.x, invoke the command:

```
service edb-pgagent-11 stop
```

to stop the pgAgent service.

After stopping any services that are currently connected to Advanced Server, you can use the EDB-PSQL command line client to drop and create a database. When the client opens, connect to the `template1` database as the database superuser; if prompted, provide authentication information. Then, use the following command to drop your database:

```
DROP DATABASE database_name;
```

Where `database_name` is the name of the database.

Then, create an empty database based on the contents of the `template1` database (see Figure 8.1):

```
CREATE DATABASE database_name;
```

Step 3 - Set both servers in trust mode

During the upgrade process, `pg_upgrade` will connect to the old and new servers several times; to make the connection process easier, you can edit the `pg_hba.conf` file, setting the authentication mode to `trust`. To modify the `pg_hba.conf` file, navigate through the Start menu to the EDB Postgres menu; to the Advanced Server menu, and open the Expert Configuration menu; select the Edit `pg_hba.conf` menu option to open the `pg_hba.conf` file.

You should allow trust authentication for the previous Advanced Server installation, and Advanced Server 11 servers. Edit the `pg_hba.conf` file for both installations of Advanced Server as shown in Figure 7.1.

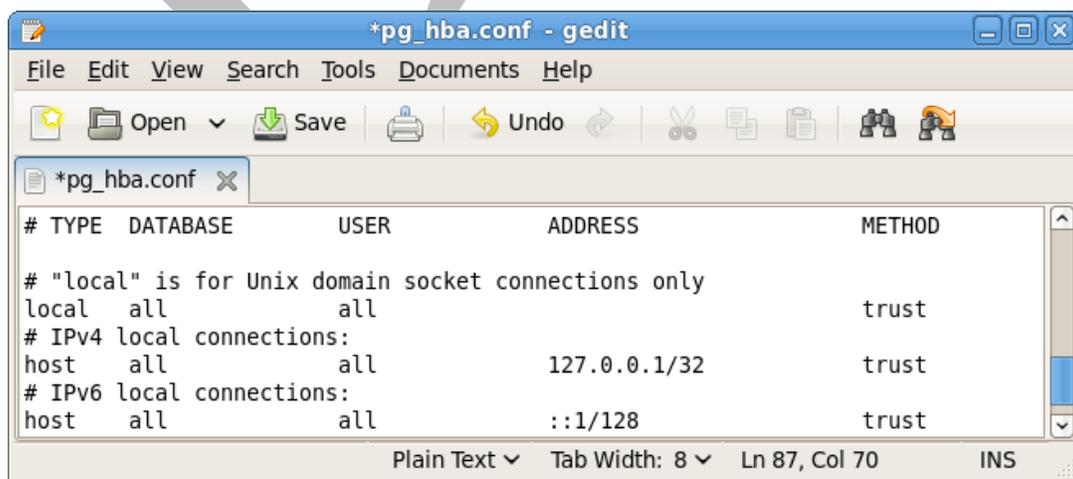


Figure 7.1 - Configuring Advanced Server to use trust authentication.

After editing each file, save the file and exit the editor.

If the system is required to maintain md5 authentication mode during the upgrade process, you can specify user passwords for the database superuser in a password file (`pgpass.conf` on Windows, `.pgpass` on Linux). For more information about configuring a password file, see the PostgreSQL Core Documentation, available through:

<https://www.postgresql.org/docs/11/static/libpq-pgpass.html>

Step 4 - Stop All Component Services and Servers

Before you invoke `pg_upgrade`, you must stop any services that belong to the original Advanced Server installation, Advanced Server 11 or the supporting components. This ensures that a service will not attempt to access either cluster during the upgrade process.

The services that are most likely to be running in your installation are:

Service:	On Linux:	On Windows
Postgres Plus Advanced Server 9.0	<code>ppas-9.0</code>	<code>ppas-9.0</code>
Postgres Plus Advanced Server 9.1	<code>ppas-9.1</code>	<code>ppas-9.1</code>
Postgres Plus Advanced Server 9.2	<code>ppas-9.2</code>	<code>ppas-9.2</code>
Postgres Plus Advanced Server 9.3	<code>ppas-9.3</code>	<code>ppas-9.3</code>
Postgres Plus Advanced Server 9.4	<code>ppas-9.4</code>	<code>ppas-9.4</code>
Postgres Plus Advanced Server 9.5	<code>ppas-9.5</code>	<code>ppas-9.5</code>
EnterpriseDB Postgres Advanced Server 9.6	<code>edb-as-9.6</code>	<code>edb-as-9.6</code>
EnterpriseDB Postgres Advanced Server 10	<code>edb-as-10</code>	<code>edb-as-10</code>
EnterpriseDB Postgres Advanced Server 11	<code>edb-as-11</code>	<code>edb-as-11</code>
Advanced Server 9.0 Scheduling Agent	<code>ppasAgent-90</code>	Postgres Plus Advanced Server 90 Scheduling Agent
Advanced Server 9.1 Scheduling Agent	<code>ppasAgent-91</code>	Postgres Plus Advanced Server 91 Scheduling Agent
Advanced Server 9.2 Scheduling Agent	<code>ppas-agent-9.2</code>	Postgres Plus Advanced Server 9.2 Scheduling Agent
Advanced Server 9.3 Scheduling Agent	<code>ppas-agent-9.3</code>	Postgres Plus Advanced Server 9.3 Scheduling Agent
Advanced Server 9.4 Scheduling Agent	<code>ppas-agent-9.4</code>	Postgres Plus Advanced Server 9.4 Scheduling Agent
Advanced Server 9.5 Scheduling Agent	<code>ppas-agent-9.5</code>	Postgres Plus Advanced Server 9.5 Scheduling Agent
Advanced Server 9.6 Scheduling Agent (pgAgent)	<code>edb-pgagent-9.6</code>	EnterpriseDB Postgres Advanced Server 9.6 Scheduling Agent

Service:	On Linux:	On Windows
Infinite Cache 9.2	ppas-infinitecache-9.2	N/A
Infinite Cache 9.3	ppas-infinitecache-9.3	N/A
Infinite Cache 9.4	ppas-infinitecache	N/A
Infinite Cache 9.5	ppas-infinitecache	N/A
Infinite Cache 9.6	edb-icache	N/A
Infinite Cache 10	edb-icache	N/A
PgBouncer 9.0	pgbouncer-90	pgbouncer-90
PgBouncer 9.1	pgbouncer-91	pgbouncer-91
PgBouncer 9.2	pgbouncer-9.2	pgbouncer-9.2
PgBouncer 9.3	pgbouncer-9.3	pgbouncer-9.3
PgBouncer	Pgbouncer	pgbouncer
PgBouncer 1.6	ppas-pgbouncer-1.6 or ppas-pgbouncer16	ppas-pgbouncer-1.6
PgBouncer 1.7	edb-pgbouncer-1.7	edb-pgbouncer-1.7
PgPool 9.2	ppas-pgpool-9.2	N/A
PgPool 9.3	ppas-pgpool-9.3	N/A
PgPool	ppas-pgpool	N/A
PgPool 3.4	ppas-pgpool-3.4 or ppas-pgpool34 or	N/A
pgPool-II	edb-pgpool-3.5	N/A
Slony 9.2	ppas-replication-9.2	ppas-replication-9.2
Slony 9.3	ppas-replication-9.3	ppas-replication-9.3
Slony 9.4	ppas-replication-9.4	ppas-replication-9.4
Slony 9.5	ppas-replication-9.5	ppas-replication-9.5
Slony 9.6	edb-slony-replication-9.6	edb-slony-replication-9.6
xDB Publication Server 9.0	edb-xdbpubserver-90	Publication Service 90
xDB Publication Server 9.1	edb-xdbpubserver-91	Publication Service 91
xDB Subscription Server	edb-xdbsubserver-90	Subscription Service 90
xDB Subscription Server	edb-xdbsubserver-91	Subscription Service 91
EDB Replication Server v6.x	edb-xdbpubserver	Publication Service for xDB Replication Server
EDB Subscription Server v6.x	edb-xdbsubserver	Subscription Service for xDB Replication Server

To stop a service on Windows:

Open the Services applet; highlight each Advanced Server or supporting component service displayed in the list, and select Stop.

To stop a service on Linux:

Open a terminal window and manually stop each service at the command line.

Step 5 for Linux only - Assume the identity of the cluster owner

If you are using Linux, assume the identity of the Advanced Server cluster owner. (The following example assumes Advanced Server was installed in the default, compatibility

with Oracle database mode, thus assigning `enterprisedb` as the cluster owner. If installed in compatibility with PostgreSQL database mode, `postgres` is the cluster owner.)

```
su - enterprisedb
```

Enter the Advanced Server cluster owner password if prompted. Then, set the path to include the location of the `pg_upgrade` executable:

```
export PATH=$PATH:/opt/edb/as11/bin
```

During the upgrade process, `pg_upgrade` writes a file to the current working directory of the `enterprisedb` user; you must invoke `pg_upgrade` from a directory where the `enterprisedb` user has `write` privileges. After performing the above commands, navigate to a directory in which the `enterprisedb` user has sufficient privileges to write a file.

```
cd /tmp
```

Proceed to Step 6.

Step 5 for Windows only - Assume the identity of the cluster owner

If you are using Windows, open a terminal window, assume the identity of the Advanced Server cluster owner and set the path to the `pg_upgrade` executable.

If the `--serviceaccount service_account_user` parameter was specified during the initial installation of Advanced Server, then `service_account_user` is the Advanced Server cluster owner and is the user to be given with the `RUNAS` command.

```
RUNAS /USER:service_account_user "CMD.EXE"  
SET PATH=%PATH%;C:\Program Files\edb\as11\bin
```

During the upgrade process, `pg_upgrade` writes a file to the current working directory of the service account user; you must invoke `pg_upgrade` from a directory where the service account user has `write` privileges. After performing the above commands, navigate to a directory in which the service account user has sufficient privileges to write a file.

```
cd %TEMP%
```

Proceed to Step 6.

If the `--serviceaccount` parameter was omitted during the initial installation of Advanced Server, then the default owner of the Advanced Server service and the database cluster is `NT AUTHORITY\NetworkService`.

When `NT AUTHORITY\NetworkService` is the service account user, the `RUNAS` command may not be usable as it prompts for a password and the `NT AUTHORITY\NetworkService` account is not assigned a password. Thus, there is typically a failure with an error message such as, “Unable to acquire user password”.

Under this circumstance a Windows utility program named `PsExec` must be used to run `CMD.EXE` as the service account `NT AUTHORITY\NetworkService`.

The `PsExec` program must be obtained by downloading `PsTools`, which is available at the following site:

<https://technet.microsoft.com/en-us/sysinternals/bb897553.aspx>

You can then use the following command to run `CMD.EXE` as `NT AUTHORITY\NetworkService`, and then set the path to the `pg_upgrade` executable.

```
psexec.exe -u "NT AUTHORITY\NetworkService" CMD.EXE
SET PATH=%PATH%;C:\Program Files\edb\as11\bin
```

During the upgrade process, `pg_upgrade` writes a file to the current working directory of the service account user; you must invoke `pg_upgrade` from a directory where the service account user has `write` privileges. After performing the above commands, navigate to a directory in which the service account user has sufficient privileges to write a file.

```
cd %TEMP%
```

Proceed with Step 6.

Step 6 - Perform a consistency check

Before attempting an upgrade, perform a consistency check to assure that the old and new clusters are compatible and properly configured. Include the `--check` option to instruct `pg_upgrade` to perform the consistency check.

The following example demonstrates invoking `pg_upgrade` to perform a consistency check on Linux:

```
pg_upgrade -d /opt/PostgresPlus/10AS/data -D
/opt/edb/as11/data -U enterprisedb -b
/opt/PostgresPlus/10AS/bin -B /opt/edb/as11/bin -p 5444 -P
5445 --check
```

If the command is successful, it will return `*Clusters are compatible*`.

If you are using Windows, you must quote any directory names that contain a space:

```
pg_upgrade.exe -d "C:\Program Files\ PostgresPlus\10AS
\data" -D "C:\Program Files\edb\as11\data" -U enterprisedb
-b "C:\Program Files\PostgresPlus\10AS\bin" -B "C:\Program
Files\edb\as11\bin" -p 5444 -P 5445 --check
```

During the consistency checking process, `pg_upgrade` will log any discrepancies that it finds to a file located in the directory from which `pg_upgrade` was invoked. When the consistency check completes, review the file to identify any missing components or upgrade conflicts. You must resolve any conflicts before invoking `pg_upgrade` to perform a version upgrade.

If `pg_upgrade` alerts you to a missing component, you can use StackBuilder Plus to add the component that contains the component. Before using StackBuilder Plus, you must restart the Advanced Server 11 service. After restarting the service, open StackBuilder Plus by navigating through the Start menu to the Advanced Server 11 menu, and selecting StackBuilder Plus. Follow the onscreen advice of the StackBuilder Plus wizard to download and install the missing components.

For more information about using StackBuilder Plus, please see Section 4.5, *Using StackBuilder Plus*.

When `pg_upgrade` has confirmed that the clusters are compatible, you can perform a version upgrade.

Step 7 - Run `pg_upgrade`

After confirming that the clusters are compatible, you can invoke `pg_upgrade` to upgrade the old cluster to the new version of Advanced Server.

On Linux:

```
pg_upgrade -d /opt/PostgresPlus/10AS/data -D
/opt/edb/as11/data -U enterprisedb -b
/opt/PostgresPlus/10AS/bin -B /opt/edb/as11/bin -p 5444 -P
5445
```

On Windows:

```
pg_upgrade.exe -d "C:\Program Files\PostgresPlus\10AS\data"
-D "C:\Program Files\edb\as11\data" -U enterprisedb
-b "C:\Program Files\PostgresPlus\10AS\bin" -B "C:\Program
Files\edb\as11\bin" -p 5444 -P 5445
```

`pg_upgrade` will display the progress of the upgrade onscreen:

```
$ pg_upgrade -d /opt/edb/as11/data -D /opt/edb/as11/data -U enterprisedb -b
/opt/edb/as11/bin -B /opt/edb/as11/bin -p 5444 -P 5445
Performing Consistency Checks
```

```

-----
Checking current, bin, and data directories           ok
Checking cluster versions                           ok
Checking database user is a superuser                ok
Checking for prepared transactions                  ok
Checking for reg* system OID user data types        ok
Checking for contrib/isn with bigint-passing mismatch ok
Creating catalog dump                               ok
Checking for presence of required libraries         ok
Checking database user is a superuser                ok
Checking for prepared transactions                  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 on the new cluster                 ok
Deleting files from new pg_clog                      ok
Copying old pg_clog to new server                   ok
Setting next transaction ID for new cluster          ok
Resetting WAL archives                              ok
Setting frozenxid counters in new cluster            ok
Creating databases in the new cluster                ok
Adding support functions to new cluster              ok
Restoring database schema to new cluster             ok
Removing support functions from new cluster          ok
Copying user relation files                          ok

Setting next OID for new cluster                     ok
Creating script to analyze new cluster               ok
Creating script to delete old cluster                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

```

While `pg_upgrade` runs, it may generate SQL scripts that handle special circumstances that it has encountered during your upgrade. For example, if the old cluster contains large objects, you may need to invoke a script that defines the default permissions for the objects in the new cluster. When performing the pre-upgrade consistency check `pg_upgrade` will alert you to any script that you may be required to run manually.

You must invoke the scripts after `pg_upgrade` completes. To invoke the scripts, connect to the new cluster as a database superuser with the EDB-PSQL command line client, and invoke each script using the `\i` option:

```
\i complete_path_to_script/script.sql
```

It is generally unsafe to access tables referenced in rebuild scripts until the rebuild scripts have completed; accessing the tables could yield incorrect results or poor performance. Tables not referenced in rebuild scripts can be accessed immediately.

Please Note: If `pg_upgrade` fails to complete the upgrade process, the old cluster will be unchanged, except that `$PGDATA/global/pg_control` is renamed to `pg_control.old` and each tablespace is renamed to `tablespace.old`. To revert to the pre-invocation state:

1. Delete any tablespace directories created by the new cluster.
2. Rename `$PGDATA/global/pg_control`, removing the `.old` suffix.
3. Rename the old cluster tablespace directory names, removing the `.old` suffix.
4. Remove any database objects (from the new cluster) that may have been moved before the upgrade failed.

After performing these steps, resolve any upgrade conflicts encountered before attempting the upgrade again.

When the upgrade is complete, `pg_upgrade` may also recommend vacuuming the new cluster, and will provide a script that allows you to delete the old cluster.

Before removing the old cluster, ensure that the cluster has been upgraded as expected, and that you have preserved a backup of the cluster in case you need to revert to a previous version.

Step 8 - Restore the authentication settings in the `pg_hba.conf` file

If you modified the `pg_hba.conf` file to permit `trust` authentication, update the contents of the `pg_hba.conf` file to reflect your preferred authentication settings.

Step 9 - Move and identify user-defined tablespaces (*Optional*)

If you have data stored in a user-defined tablespace, you must manually relocate tablespace files after upgrading; move the files to the new location and update the symbolic links (located in the `pg_tblspc` directory under your cluster's `data` directory) to point to the files.

7.4 Upgrading a pgAgent Installation

If your existing Advanced Server installation uses pgAgent, you can use a script provided with the Advanced Server 11 installer to update pgAgent. The script is named `dbms_job.upgrade.script.sql`, and is located in the `/share/contrib/` directory under your Advanced Server installation.

If you are using `pg_upgrade` to upgrade your installation, you should:

1. Perform the upgrade.
2. Invoke the `dbms_job.upgrade.script.sql` script to update the catalog files. If your existing pgAgent installation was performed with a script, the update will convert the installation to an extension.

7.5 *pg_upgrade* Troubleshooting

The troubleshooting tips in this section address problems you may encounter when using *pg_upgrade*.

7.5.1 Upgrade Error - There seems to be a postmaster servicing the cluster

If *pg_upgrade* reports that a postmaster is servicing the cluster, please stop all Advanced Server services and try the upgrade again.

7.5.2 Upgrade Error - *fe_sendauth*: no password supplied

If *pg_upgrade* reports an authentication error that references a missing password, please modify the *pg_hba.conf* files in the old and new cluster to enable *trust* authentication, or configure the system to use a *pgpass.conf* file.

7.5.3 Upgrade Error - New cluster is not empty; exiting

If *pg_upgrade* reports that the new cluster is not empty, please empty the new cluster. The target cluster may not contain any user-defined databases.

7.5.4 Upgrade Error - Failed to load library

If the original Advanced Server cluster included libraries that are not included in the Advanced Server 11 cluster, *pg_upgrade* will alert you to the missing component during the consistency check by writing an entry to the *loadable_libraries.txt* file in the directory from which you invoked *pg_upgrade*. Generally, for missing libraries that are not part of a major component upgrade, perform the following steps:

1. Restart the Advanced Server service.

Use StackBuilder Plus to download and install the missing module as described in Chapter 4, *Using StackBuilder Plus*. Then:

2. Stop the Advanced Server service.
3. Resume the upgrade process: invoke *pg_upgrade* to perform consistency checking.
4. When you have resolved any remaining problems noted in the consistency checks, invoke *pg_upgrade* to perform the data migration from the old cluster to the new cluster.

7.6 Reverting to the Old Cluster

The method used to revert to a previous cluster varies with the options specified when invoking `pg_upgrade`:

- If you specified the `--check` option when invoking `pg_upgrade`, an upgrade has not been performed, and no modifications have been made to the old cluster; you can re-use the old cluster at any time.
- If you included the `--link` option when invoking `pg_upgrade`, the data files are shared between the old and new cluster after the upgrade completes. If you have started the server that is servicing the new cluster, the new server has written to those shared files and it is unsafe to use the old cluster.
- If you ran `pg_upgrade` without the `--link` specification or have not started the new server, the old cluster is unchanged, except that the `.old` suffix has been appended to the `$PGDATA/global/pg_control` and tablespace directories.
- To reuse the old cluster, delete the tablespace directories created by the new cluster and remove the `.old` suffix from `$PGDATA/global/pg_control` and the old cluster tablespace directory names and restart the server that services the old cluster.

8 Uninstalling Advanced Server

Note that after uninstalling Advanced Server, the cluster data files remain intact and the service user persists. You may manually remove the cluster data and service user from the system.

8.1 Uninstalling an RPM Package

You can use variations of the `rpm` or `yum` command to remove installed packages. Note that removing a package does not damage the Advanced Server `data` directory.

Include the `-e` option when invoking the `rpm` command to remove an installed package; the command syntax is:

```
rpm -e package_name
```

Where `package_name` is the name of the package that you would like to remove.

You can use the `yum remove` command to remove a package installed by `yum`. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

```
yum remove package_name
```

Where `package_name` is the name of the package that you would like to remove.

Note: `yum` and `RPM` will not remove a package that is required by another package. If you attempt to remove a package that satisfies a package dependency, `yum` or `RPM` will provide a warning.

8.2 Using Advanced Server Uninstallers at the Command Line

The Advanced Server interactive installer creates an uninstaller that you can use to remove Advanced Server or components that reside on a Windows host. The uninstaller is created in `C:\Program Files\edb\as11`. To open the uninstaller, assume superuser privileges, navigate into the directory that contains the uninstaller, and enter:

```
uninstall-edb-as11-server.exe
```

The uninstaller opens as shown in Figure 8.1.

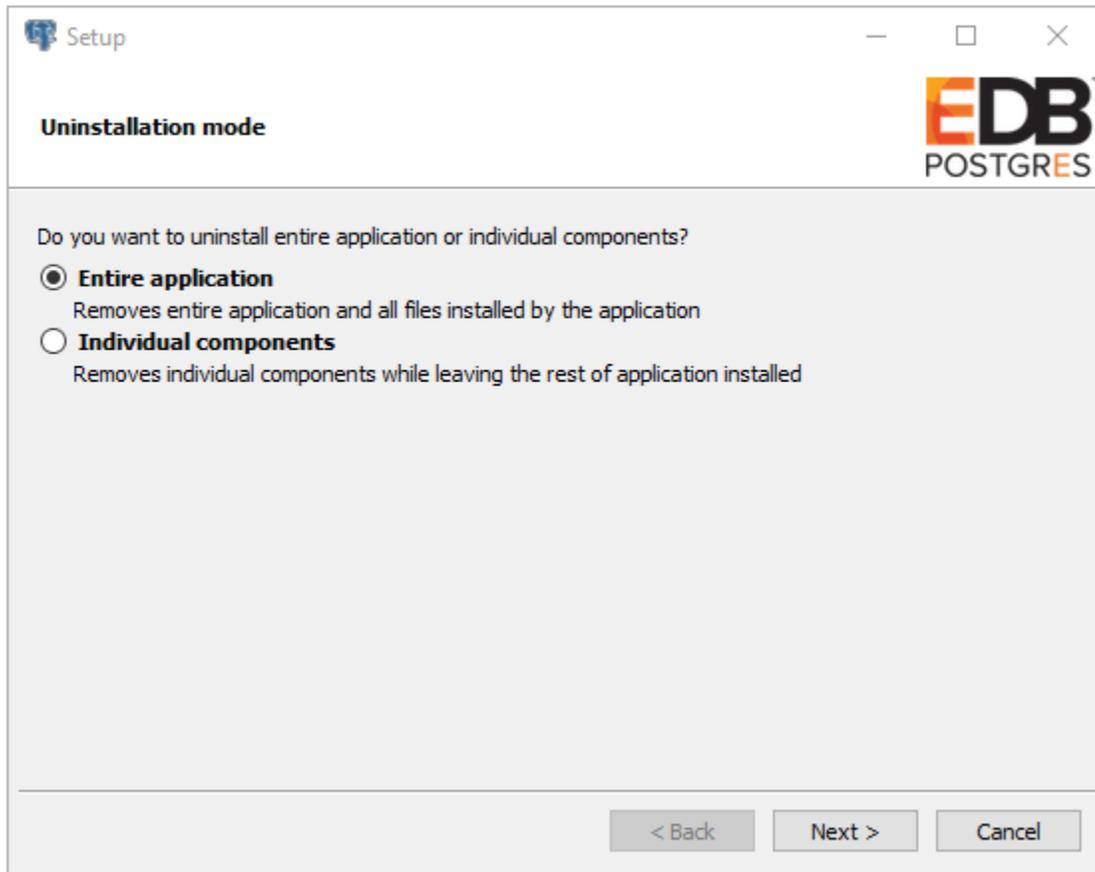


Figure 8.1 – The Advanced Server uninstaller.

You can remove the `Entire application` (the default), or select the radio button next to `Individual components` to select components for removal; if you select `Individual components`, a dialog will open, prompting you to select the components you wish to remove. After making your selection, click `Next`.

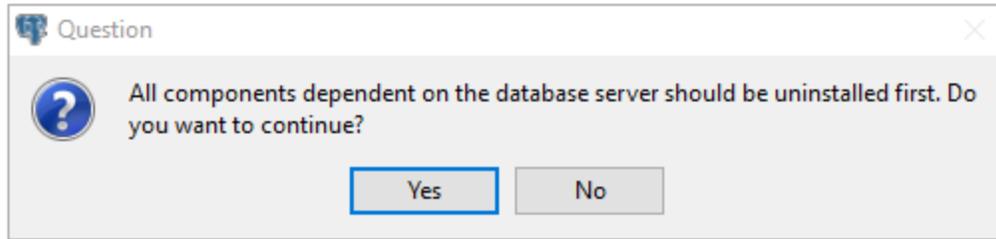


Figure 8.2 – Acknowledge that dependent components are removed first.

If you have elected to remove components that are dependent on Advanced Server, those components will be removed first; click `Yes` to acknowledge that you wish to continue (see Figure 8.2).

Progress bars are displayed as the software is removed. When the uninstallation is complete, an `Info` dialog opens to confirm that Advanced Server (and/or its components) has been removed (see Figure 8.3).

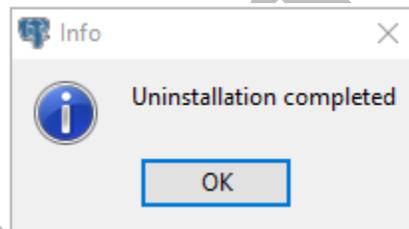


Figure 8.3 - The uninstallation is complete.