EDB Postgres™ Advanced Server Installation Guide

EDB Postgres™ Advanced Server 10

February 15, 2021

EDB Postgres™ Advanced Server Installation Guide
by EnterpriseDB® Corporation

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

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 10.
- Using a package manager to install and update Advanced Server and its supporting components or utilities.
- Installation options available through the interactive setup wizard on Linux and Windows.
- Managing an Advanced Server installation.
- Configuring an Advanced Server installation.
- Using pg_upgrade to upgrade from an earlier version of Advanced Server to Advanced Server 10.
- 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 10.

2.1 Supported Platforms

To review a list of supported platforms, visit the EnterpriseDB website at:

https://www.enterprisedb.com/product-compatibility

Note:

- The data directory of a production database should not be stored on an NFS file system.

- Advanced Server is no longer supported on RHEL/CentOS/OEL 6.x platforms. It is strongly recommended that EDB products running on these platforms be migrated to a supported platform.

2.2 RPM Installation Pre-Requisites

You can use an RPM package to install Advanced Server and its supporting components on a Linux host. Before installing the Advanced Server, you must:

Request Credentials for the EnterpriseDB Repository

Before installing Advanced Server, you must have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit:

https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf

After receiving your repository credentials you can:

- Create the repository configuration file.

- Modify the file, providing your user name and password.

- Install Advanced Server and its supporting components.

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.

The Advanced Server RPM installs Advanced Server and the core components of the database server. For a complete list of the RPM installers available for Advanced Server and its supporting components, see Section 3.3.1.

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/as10`.

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:

Component	Location
Executables	`/usr/edb/as10/bin`
Libraries	`/usr/edb/as10/lib`
Cluster configuration files	`/etc/edb/as10`
Documentation	`/usr/edb/as10/share/doc`
Contrib	`/usr/edb/as10/share/contrib`
Data	`/var/lib/edb/as10/data`
Logs	`/var/log/as10`
Lock files	`/var/lock/as10`
Log rotation file	`/etc/logrotate.d/as10`
Sudo configuration file	`/etc/sudoers.d/as10`
Binary to access VIP without sudo	`/usr/edb/as10/bin/secure`
Backup area	`/var/lib/edb/as10/backups`
Templates	`/usr/edb/as10/share`
Procedural Languages	`/usr/edb/as10/lib or /usr/edb/as10/lib64`
Development Headers	`/usr/edb/as10/include`
Shared data	`/usr/edb/as10/share`

Component	Location
Regression tests	`/usr/edb/as10/lib/pgxs/src/test/regress`
SGML Documentation	`/usr/edb/as10/share/doc`

3.1 Installing Advanced Server on a CentOS Host

You can use an RPM package to install Advanced Server on a CentOS host.

- To install the repository configuration file, assume superuser privileges and invoke the following platform-specific command:

  On CentOS 7:

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

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

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

- Before installing Advanced Server, you must install the `epel-release` package:

  On CentOS 7:

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

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

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the `enabled` parameter is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you can use yum install command to install Advanced Server. For example, to install the server and its core components, use the command:

- On CentOS 7:

  ```
  yum –y install edb-as10-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 3.4, *Configuring a Package Installation*, for details.

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.

3.2 Installing Advanced Server on a RHEL Host

You can use an RPM package to install Advanced Server on a RHEL host.

- To install the repository configuration file, assume superuser privileges and invoke the following platform-specific command:

  On RHEL 7:

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

- Replace the USERNAME:PASSWORD in the following command with the username and password of a registered EnterpriseDB user:

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

- Before installing Advanced Server, you must install the epel-release package:

  On RHEL 7:

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

- Enable the repository:

  On RHEL7, enable the `optional, extras`, and `HA` repositories to satisfy package dependencies:

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

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

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the `enabled` parameter is `1,` and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you can use `yum install` command to install Advanced Server. For example, to install the server and its core components, use the command:

- On RHEL 7:

  ```
  yum –y install edb-as10-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 3.4, *Configuring a Package Installation*, for details.

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.

3.3 Installing Advanced Server on a CentOS/RHEL 7 ppc64le Host

You can use an RPM package to install Advanced Server on a CentOS or RHEL 7 ppc64le host.

- To install Advance Toolchain repository:

  On CentOS or RHEL 7 ppc64le:

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

  The repository configuration file is named `advance-toolchain.repo`. The file resides in `/etc/yum.repos.d`.

- After creating the `advance-toolchain.repo` file, use your choice of editor to set the value of the `enabled` parameter to `1,` and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

  ```
  [advance-toolchain]
  name=Advance Toolchain IBM FTP
  baseurl=https://public.dhe.ibm.com/software/server/POWER/Linux
  /toolchain/at/redhat/RHEL7
  failovermethod=priority
  enabled=1
  gpgcheck=1
  gpgkey=ftp://public.dhe.ibm.com/software/server/POWER/Linux/to
  olchain/at/redhat/RHELX/gpg-pubkey-6976a827-5164221b
  ```

- To install the repository configuration file, assume superuser privileges and invoke the following platform-specific command:

  On CentOS or RHEL 7 ppc64le:

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

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

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

- Before installing Advanced Server, you must install the `epel-release` package:

On CentOS or RHEL 7 ppc64le:

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

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

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

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

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the `enabled` parameter is `1,` and replace the `username` and `password` placeholders in the `baseurl` specification with a registered EnterpriseDB username and password.

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

After saving your changes to the configuration file, you can use `yum install` command to install Advanced Server. For example, to install the server and its core components, use the command:

- On CentOS or RHEL 7 ppc64le:

  ```
  yum –y install edb-as10-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 3.4, *Configuring a Package Installation*, for details.

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.

3.3.1 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:

- On RHEL or CentOS 7:

```
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.

The following table lists the packages that are stored in the Advanced Server repository and the corresponding software installed by those packages:

Package Name	Package Installs
`edb-as10-server`	This package installs core components of the Advanced Server database server.
`edb-as10-server-client`	The `edb-as10-server-client` package contains client programs and utilities that you can use to access and manage Advanced Server.
`edb-as10-server-contrib`	The `edb-as10-contrib` package installs contributed tools and utilities that are distributed with Advanced Server. Files for these modules are installed in: Documentation: `/usr/edb/as10/share/doc` Loadable modules: `/usr/edb/as10/lib` Binaries: `/usr/edb/as10/bin`
`edb-as10-server-core`	The `edb-as10-server-core` package includes the programs needed to create the core functionality behind the Advanced Server database.
`edb-as10-server-devel`	The `edb-as10-server-devel` package contains the header files and libraries needed to compile C or C++ applications that directly interact with an Advanced Server server and the ecpg or ecpgPlus C preprocessor.
`edb-as10-server-docs`	The `edb-as10-server-docs` package installs the readme file.
`edb-as10-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-as10-server-libs`	The `edb-as10-server-libs` package provides the essential shared libraries for any Advanced Server client program or interface.
`edb-as10-server-pldebugger`	This package implements an API for debugging PL/pgSQL functions on Advanced Server.
`edb-as10-server-plperl`	The `edb-as10-server-plperl` package installs the PL/Perl procedural language for Advanced Server. Please note that the `edb-as10-server-plperl` package is dependent on the platform-supplied version of Perl.
`edb-as10-server-plpython`	The `edb-as10-server-plpython` package installs the PL/Python procedural language for Advanced Server. Please note that the `edb-as10-server-plpython` package is dependent on the platform-

Package Name	Package Installs
	supplied version of Python.
edb-as10-server-pltcl	The edb-as10-pltcl package installs the PL/Tcl procedural language for Advanced Server. Please note that the edb-as10-pltcl package is dependent on the platform-supplied version of TCL.
edb-as10-server-sqlprofiler	This package installs Advanced Server's SQL Profiler feature. SQL Profiler helps identify and optimize SQL code.
edb-as10-server-sqlprotect	This package installs Advanced Server's SQL Protect feature. SQL Protect provides protection against SQL injection attacks.
edb-as10-server-sslutils	This package installs functionality that provides SSL support.
edb-as10-server-cloneschema	This package installs the EDB Clone Schema extension. For more information about EDB Clone Schema, see the EDB Postgres Advanced Server Guide.
edb-as10-server-parallel-clone	This package installs functionality that supports the EDB Clone Schema extension.
edb-as10-edbplus	The edb-edbplus package contains the files required to install the EDB*Plus command line client. EDB*Plus commands are compatible with Oracle's SQL*Plus.
edb-as10-pgagent	This package installs pgAgent; pgAgent is a job scheduler for Advanced Server. Before installing this package, you must install EPEL.
edb-icache	This package installs InfiniteCache.
edb-icache-devel	This is a supporting package for InfiniteCache.
edb-as10-pgsnmpd	SNMP (Simple Network Management Protocol) is a protocol that allows you to supervise an apparatus connected to the network.
edb-as10-pljava	This package installs PL/Java, providing access to Java stored procedures, triggers and functions via the JDBC interface.
edb-as10-pgpool35-extensions	This package creates pgPool extensions required by the server.
edb-as10-slony-replication	This package contains the meta installer for Slony replication and documentation. Slony facilitates primary-standby replication, and is suited for large databases with a limited number of standbys.
edb-as10-slony-replication-core	This package contains the files required to install Slony replication. Slony facilitates primary-standby replication, and is suited for large databases with a limited number of standby systems.
edb-as10-slony-replication-docs	This package contains the Slony project documentation (in pdf form).
edb-as10-slony-replication-tools	This package contains the Slony altperl tools and utilities that are useful when deploying Slony replication environments. Before installing this package, you must install EPEL.

The following table lists the packages for Advanced Server 10 supporting components:

Package Name	Package Installs
edb-pgpool35	This package contains the pgPool-II installer. pgPool provides connection pooling for Advanced Server installations.
edb-pgpool35-devel	This package contains the pgPool-II headers and libraries.
edb-jdbc	The edb-jdbc package includes the .jar files needed for Java programs to access an Advanced Server database.
edb-migrationtoolkit	The edb-migrationtoolkit package installs Migration Toolkit, facilitating migration to an Advanced Server database from Oracle, PostgreSQL, MySQL, Sybase and SQL Server.
edb-oci	The edb-oci package installs the EnterpriseDB Open Client library,

Package Name	Package Installs
	allowing applications that use the Oracle Call Interface API to connect to an Advanced Server database.
`edb-oci-devel`	This package installs the OCI include files; install this package if you are developing C/C++ applications that require these files.
`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.

3.3.2 Updating an RPM Installation

If you have an existing Advanced Server RPM installation, you can use yum to upgrade your repository configuration file and update to a more recent product version. To update the `edb.repo` file, assume superuser privileges and enter:

- On RHEL or CentOS 7:

```
yum upgrade edb-repo
```

yum will update the `edb.repo` file to enable access to the current EDB repository, configured to connect with the credentials specified in your `edb.repo` file. Then, you can use yum to upgrade all packages whose names include the expression `edb`:

- On RHEL or CentOS 7:

```
yum upgrade edb*
```

Please note that the `yum upgrade` command will only perform an update between minor releases; to update between major releases, you must use pg_upgrade.

3.4 Configuring a Package Installation

The packages that install the database server component create a 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 `systemd` service manager on RHEL or CentOS 7.x use the service configuration file to configure the environment.

If you are using the interactive graphical installer to install Advanced Server, the installer will invoke `initdb` to create a cluster for you; for details about specifying cluster preferences when using the interactive installer, see ***Setting Cluster Preferences with the Graphical Installer*** in Chapter 4.

3.4.1 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 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/as10/bin/edb-as-10-setup initdb
```

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

```
systemctl { start | stop | restart } edb-as-10
```

For more information about using the service command, please see Section 5.2.

3.4.1.1 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/10/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.

3.4.2 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-10.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/as10/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-10.service
   /etc/systemd/system/
```

After copying the unit file to the new location, modify the service file (`/etc/systemd/system/edb-as-10.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-10
```

3.5 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 7.x

The `edb-as10-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`:

- 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-10.service
/etc/systemd/system/secondary-edb-as-10.service
```

- Edit the file, changing `PGDATA` to point to the new `data` directory that you will create the cluster against.

- Create the target `PGDATA` with user `enterprisedb`.

- Run `initdb,` specifying the setup script:

```
/usr/edb/as10/bin/edb-as-10-setup initdb secondary-
edb-as-10
```

- Edit the `postgresql.conf` file for the new instance, specifying the port, the IP address, TCP/IP settings, etc.

- Make sure that new cluster runs after a reboot:

```
systemctl enable secondary-edb-as-10
```

- Start the second cluster with the following command:

```
systemctl start secondary-edb-as-10
```

3.6 Installing Advanced Server on an Isolated Network

You can create a local yum 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:

- Use yum to install the `yum-utils` and `createrepo` packages:

  On RHEL or CentOS 7:

  ```
  yum install yum-utils
  yum install createrepo
  ```

- Create a directory in which to store the repository:

  ```
  mkdir /srv/repos
  ```

- Copy the RPM installation packages to your local network repository. You can download the individual RPM files from:

  yum.enterprisedb.com

- Sync the RPM packages and create the repository.

  ```
  reposync -r edbas10 -p /srv/repos
  createrepo /srv/repos
  ```

- 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. For example, you might install `lighttpd`:

  ```
  yum install lighttpd
  ```

- If you are using `lighttpd`, you must provide a configuration file that identifies the location of the repository on your local network. For example, the configuration file might contain:

  ```
  $HTTP["host"] == "yum.domain.com"{
    server.document-root = "/srv/repos"
    server.errorlog="/var/log/lighttpd/yum_error.log"
  ```

```
      accesslog.filename =
    "/var/log/lighttpd/yum_access.log"}
```

For detailed information about installing, configuring and using lighttpd, visit the official project site at:

http://redmine.lighttpd.net/projects/1/wiki/Docs

- On each isolated database server, configure yum to pull updates from the mirrored repository on your local network. For example, you might create a file called `/etc/yum.repos.d/edb-repo` with connection information that specifies:

```
[edbas10]
name=EnterpriseDB Advanced Server 10
baseurl=http://yum.domain.com/edbas10
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 -y install edb-as10
```

For more information about creating a local repository, visit:

http://yum.baseurl.org/

4 Installing Advanced Server with the Interactive Installer

The Advanced Server installer is available from the EnterpriseDB website at:

http://www.enterprisedb.com/downloads/postgres-postgresql-downloads

After navigating to the `Software Downloads` page, use the drop-down listboxes to select the Advanced Server version you wish to install and your platform, and then click the `Download Now` button. When the download completes, extract files using your system-specific file extractor.

You can use the extracted installer in different installation modes to perform an Advanced Server installation:

- For information about using the extracted files to perform a graphical installation on Windows, See Section 4.3.1.
- For information about performing a graphical installation on Linux, see Section 4.3.2.
- For information about using the installer to perform a command line installation, see Section 4.4.
- For information about performing an unattended installation, see Section 4.4.2.
- For information about performing an installation with limited privileges, see Section 4.4.3.
- For information about the command line options you can use when invoking the installer, see Section 4.4.4.

During the installation process, the Advanced Server installer program copies a number of temporary files to the location specified by the `TEMP` or `TMP` environment variable (on Windows), or to the `/tmp` directory (on Linux). You can optionally specify an alternate location for the installer to place the temporary files by modifying or creating the `TEMP` environment variable.

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

On Windows, use the command:

```
SET TMP=temp_file_location
```

On Linux, use the command:

```
    export 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, on Linux use an `export` statement to set the value:

```
    export INITDBOPTS="-k -E=UTF-8"
```

or on Windows, use a `SET` statement:

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

On Linux, enclose the options in double-quotes (`"`); on Windows, double-quotes are not required. 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/10/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 Graphical Installation Prerequisites

User Privileges

Before invoking the installer on a Linux system, you must have superuser privileges to perform an Advanced Server installation. 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.

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.

Before invoking the StackBuilder Plus utility on a Linux system, you must install the `redhat-lsb` package. To install the package, open a terminal window, assume superuser privileges, and enter:

```
# yum install redhat-lsb
```

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

SELinux Permissions

Before invoking the 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
```

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 1.7 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.

If you are using Windows, Java installers and instructions are available online at:

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

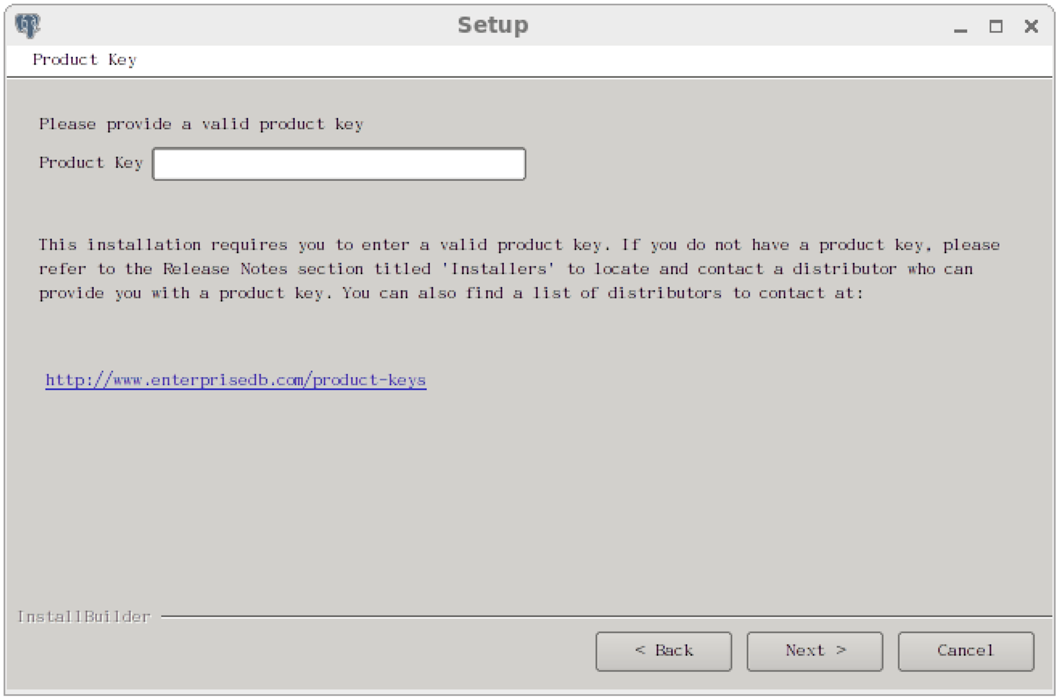
4.2 Locales Requiring Product Keys

The Advanced Server 10 installer will request a product key before completing an installation into a host system using one of the locales listed in the table below. Product keys are available from your local Advanced Server distributor.

Note: The product key applies only to the Advanced Server installation program. The Advanced Server database program has no built-in limitations or expiration features that require a product key or any other activation technique.

Locale	Locale Identifier
Traditional Chinese with Hong Kong SCS	`zh_HK`
Traditional Chinese for Taiwan	`zh_TW`
Simplified Chinese	`zh_CN`
Japanese	`ja_JP`
Korean	`ko_KR`
Argentina – Spanish	`es_ar`
Beliz – English	`en_bz`
Brazil - Portuguese	`pt_br`
Bolivia - Spanish	`es_bo`
Chile - Spanish	`es_cl`
Colombia - Spanish	`es_co`
Costa Rica - Spanish	`es_cr`
Dominican Republic - Spanish	`es_do`
Ecuador - Spanish	`es_ec`
Guatemala - Spanish	`es_gt`
Guyana - English	`en_gy`
Honduras - Spanish	`es_hn`
Mexico - Spanish	`es_mx`
Nicaragua - Spanish	`es_ni`
Panama - Spanish	`es_pa`
Peru - Spanish	`es_pe`
Puerto Rico - Spanish	`es_pr`
Paraguay - Spanish	`es_py`
El Salvador - Spanish	`es_sv`
Uruguay - Spanish	`es_uy`
Venezuela - Spanish	`es_ve`

During an installation in one of the listed locales, the `Product Key` window (shown in Figure 4.1) will open, prompting you to provide a valid product key. Enter a product key, and press `Next` to continue with the installation.



Please provide a valid product key

Product Key

This installation requires you to enter a valid product key. If you do not have a product key, please refer to the Release Notes section titled 'Installers' to locate and contact a distributor who can provide you with a product key. You can also find a list of distributors to contact at:

http://www.enterprisedb.com/product-keys

Figure 4.1 -The Advanced Server Product Key Window

4.3 Performing a Graphical Installation

A graphical installation wizard provides a quick and easy way to install Advanced Server 10 on a Linux or Windows system. As the `Setup` wizard's easy-to-follow dialogs lead you through the installation process, specify information about your system, your system usage, and the modules that will best complement your installation of Advanced Server. When you have completed the dialogs, the installer performs an installation based on the selections made during the setup process.

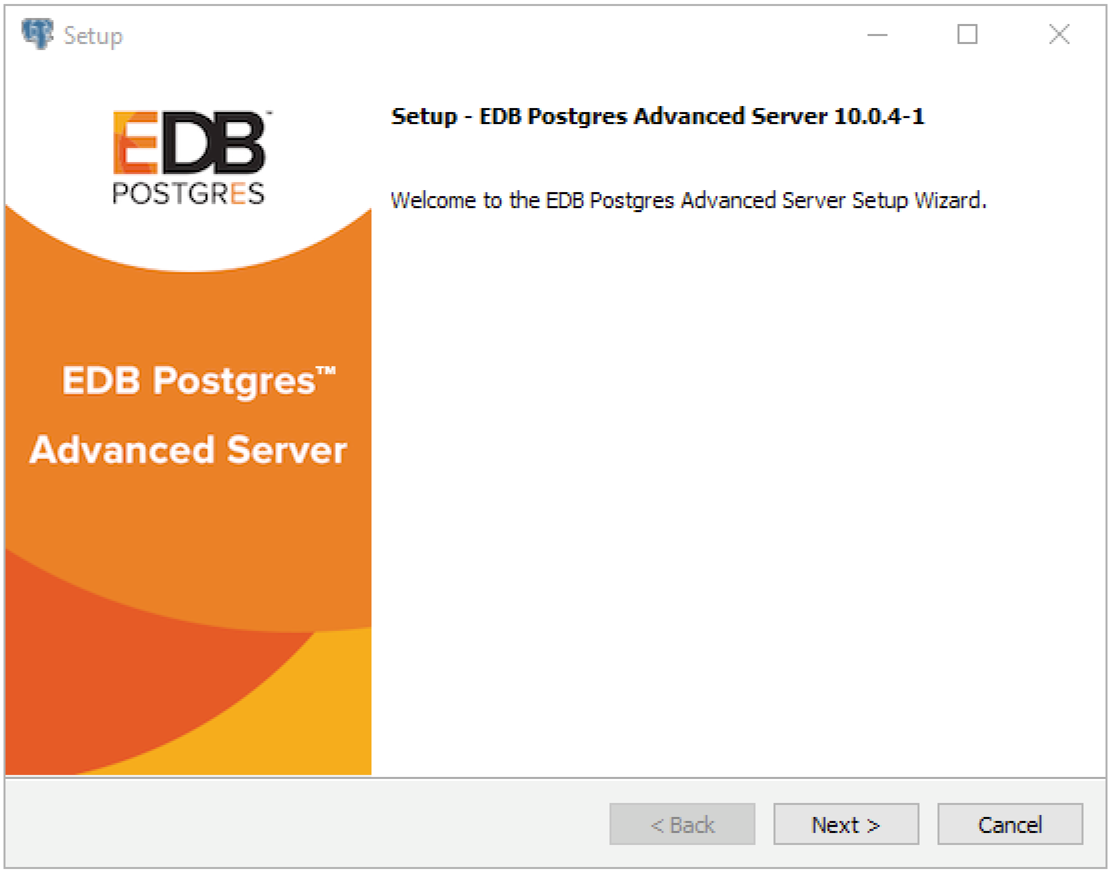
When the Advanced Server installation finishes, you will be offered the option to invoke EDB Postgres StackBuilder Plus. StackBuilder Plus provides an easy-to-use graphical interface that can update installed products, or download and add any omitted modules (and the resulting dependencies) after your Advanced Server setup and installation completes. See Section 4.5 for more information about StackBuilder Plus.

4.3.1 Using the Graphical Installer with Windows

To perform an installation using the graphical wizard on a Windows system, you must have administrator privileges. To start the `Setup` wizard, assume administrator privileges, and double-click the `edb-as10-server-10.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.*

The wizard opens a `Language Selection` popup; select an installation language from the drop-down listbox and click `OK` to continue to the `Setup` window (shown in Figure 4.2):



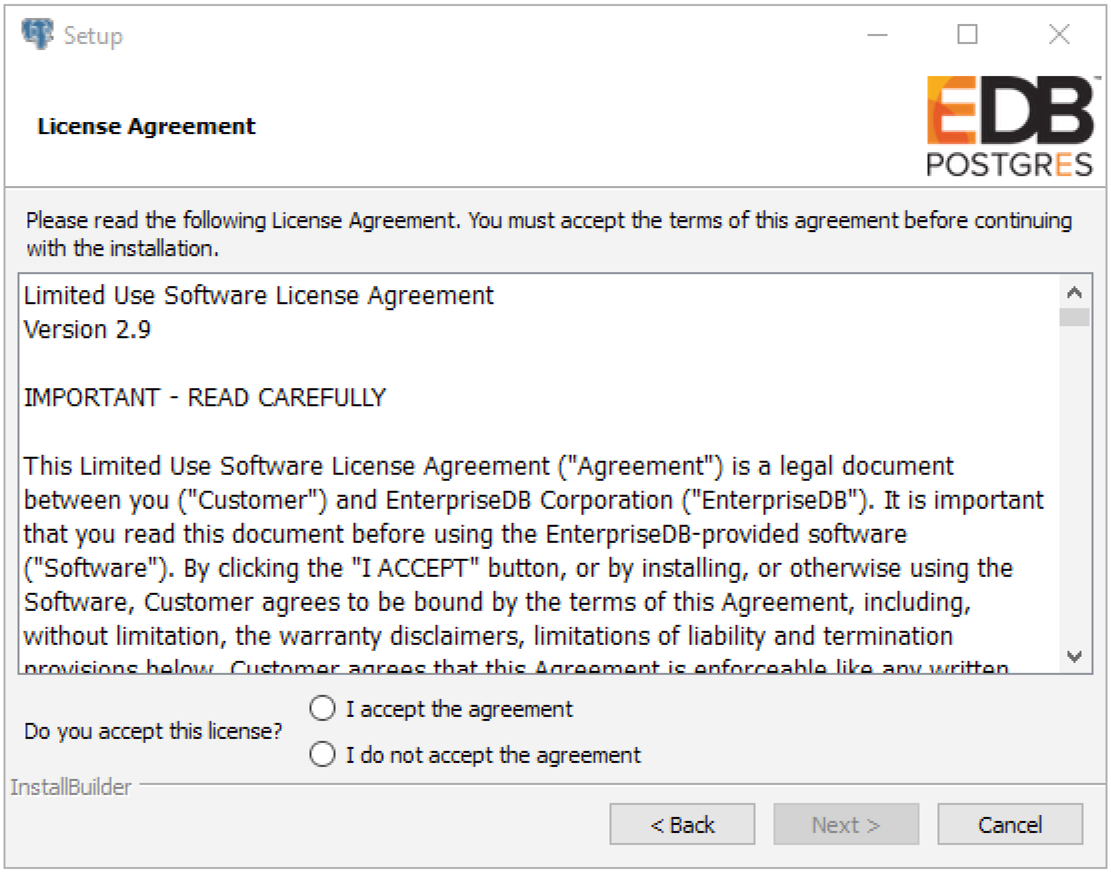
Setup - EDB Postgres Advanced Server 10.0.4-1

Welcome to the EDB Postgres Advanced Server Setup Wizard.

Figure 4.2 -The Advanced Server installer Welcome window

Click `Next` to continue.

The EnterpriseDB `License Agreement` (see Figure 4.3) opens.



Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Limited Use Software License Agreement
Version 2.9

IMPORTANT - READ CAREFULLY

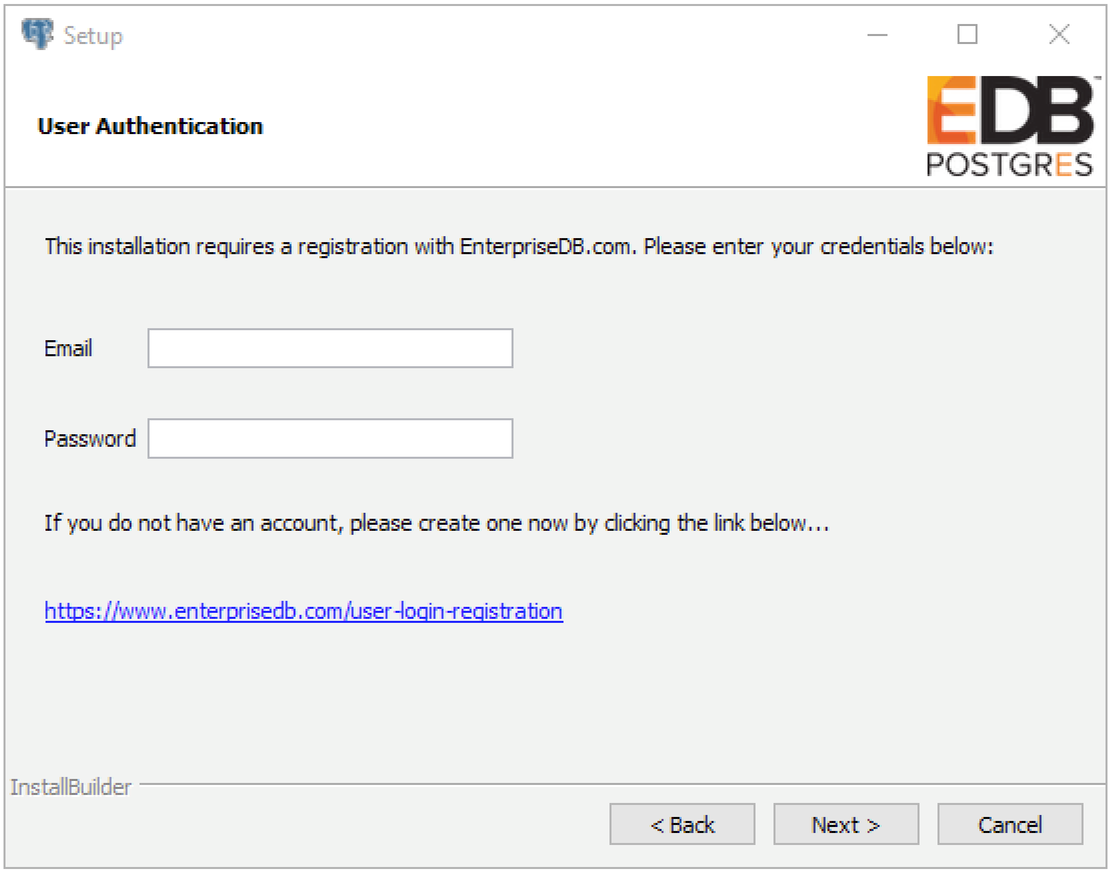
This Limited Use Software License Agreement ("Agreement") is a legal document between you ("Customer") and EnterpriseDB Corporation ("EnterpriseDB"). It is important that you read this document before using the EnterpriseDB-provided software ("Software"). By clicking the "I ACCEPT" button, or by installing, or otherwise using the Software, Customer agrees to be bound by the terms of this Agreement, including, without limitation, the warranty disclaimers, limitations of liability and termination provisions below. Customer agrees that this Agreement is enforceable like any written

Do you accept this license? ○ I accept the agreement ○ I do not accept the agreement

Figure 4.3 -The EnterpriseDB License Agreement

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

The `User Authentication` window opens, as shown in Figure 4.4.



This installation requires a registration with EnterpriseDB.com. Please enter your credentials below:

Email

Password

If you do not have an account, please create one now by clicking the link below...

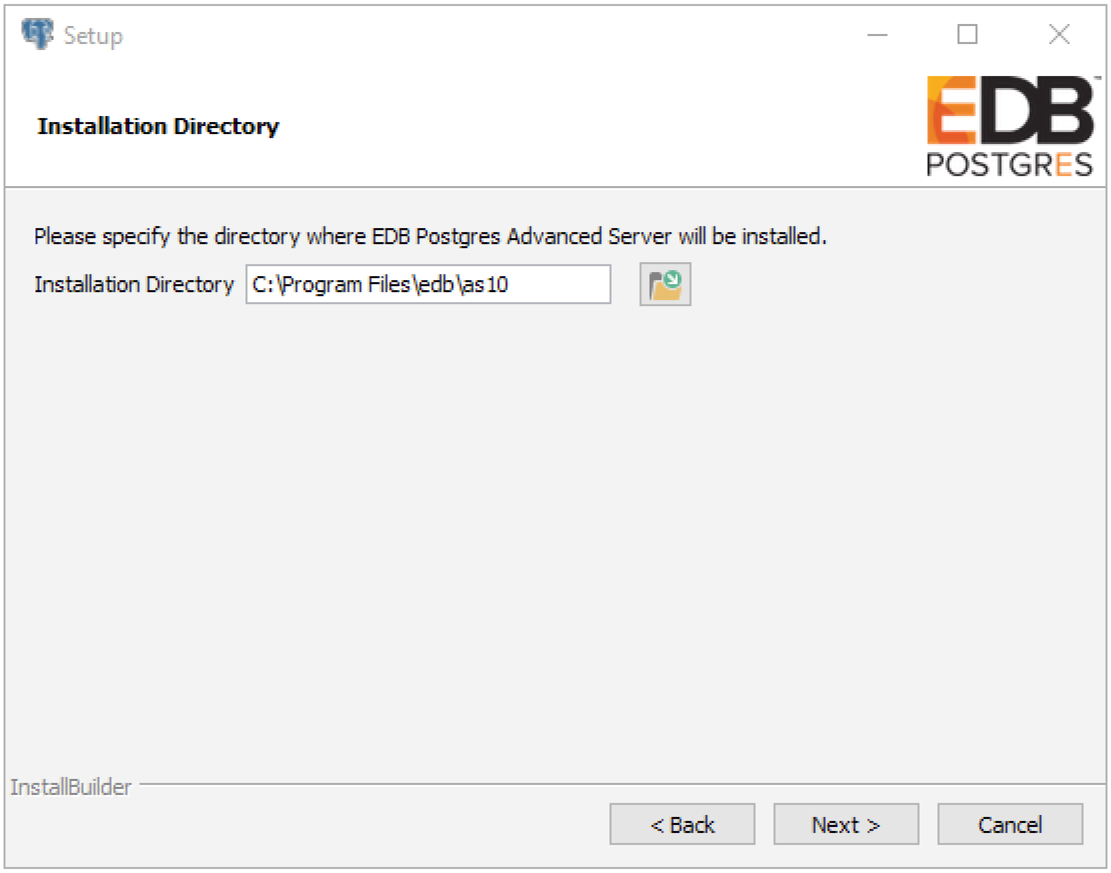
https://www.enterprisedb.com/user-login-registration

Figure 4.4 -The User Authentication window.

Before continuing, you must provide the email address and password associated with your EnterpriseDB user account. Registration is free; if you do not have an EnterpriseDB user account, click the link provided to open a web browser, and supply your user information.

Enter the email address of a registered account in the `Email` field, and the corresponding password in the `Password` field, and click `Next` to continue.

The `Installation Directory` window opens, as shown in Figure 4.5.



Installation Directory

Please specify the directory where EDB Postgres Advanced Server will be installed.

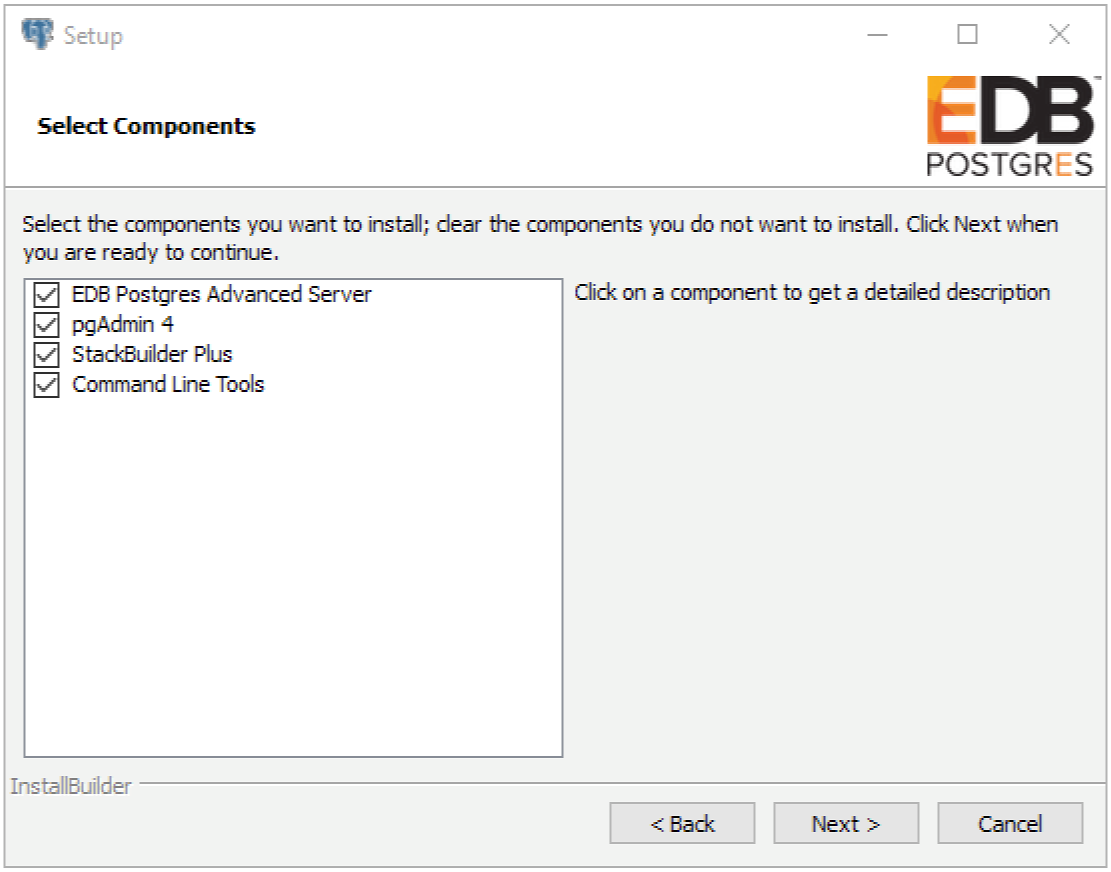
Installation Directory: C:\Program Files\edb\as10

Figure 4.5 -The Installation Directory window.

By default, the Advanced Server installation directory is:

```
C:\Program Files\edb\as10
```

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.



Setup

Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☑ EDB Postgres Advanced Server
☑ pgAdmin 4
☑ StackBuilder Plus
☑ Command Line Tools

Click on a component to get a detailed description

Figure 4.6 -The Select Components window

The `Select Components` window (shown in Figure 4.6) 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 10:

EDB Postgres Advanced Server

> Select the `EDB Postgres Advanced Server` option to install Advanced Server 10.

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 4.5 for more information about StackBuilder Plus.

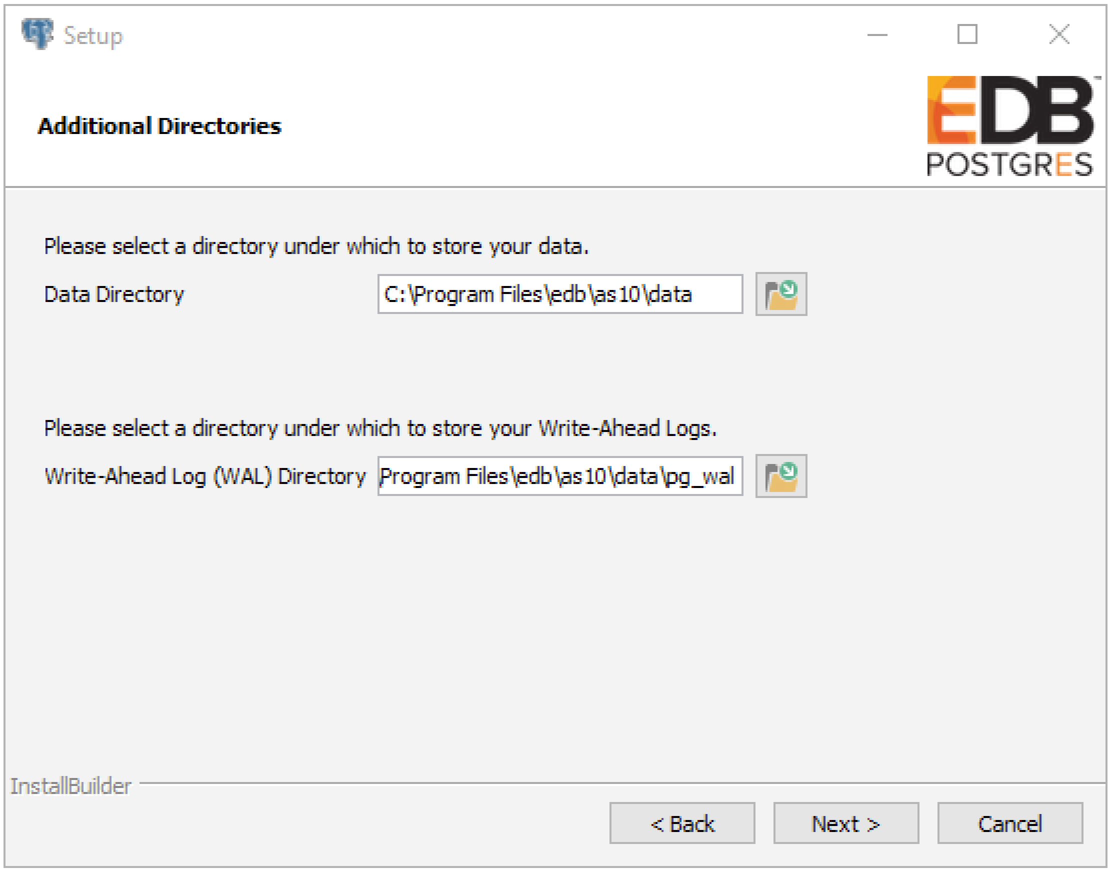
Command Line Tools

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 4.7).



Additional Directories

Please select a directory under which to store your data.

Data Directory: C:\Program Files\edb\as10\data

Please select a directory under which to store your Write-Ahead Logs.

Write-Ahead Log (WAL) Directory: Program Files\edb\as10\data\pg_wal

Figure 4.7 -The Additional Directories window.

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

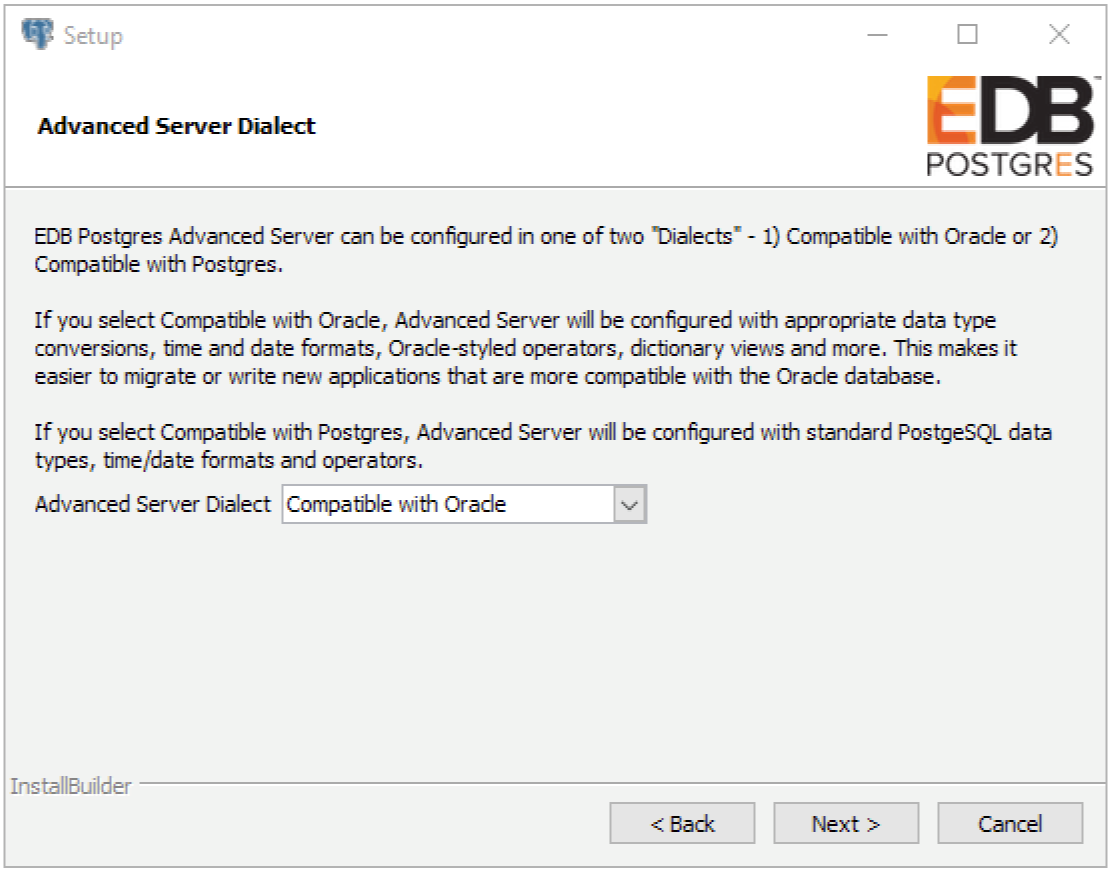
```
C:\Program Files\edb\as10\data
```

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

```
C:\Program Files\edb\as10\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 (shown in Figure 4.8).



Advanced Server Dialect

EDB Postgres Advanced Server can be configured in one of two "Dialects" - 1) Compatible with Oracle or 2) Compatible with Postgres.

If you select Compatible with Oracle, Advanced Server will be configured with appropriate data type conversions, time and date formats, Oracle-styled operators, dictionary views and more. This makes it easier to migrate or write new applications that are more compatible with the Oracle database.

If you select Compatible with Postgres, Advanced Server will be configured with standard PostgeSQL data types, time/date formats and operators.

Advanced Server Dialect: Compatible with Oracle

Figure 4.8 -The Advanced Server Dialect window.

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

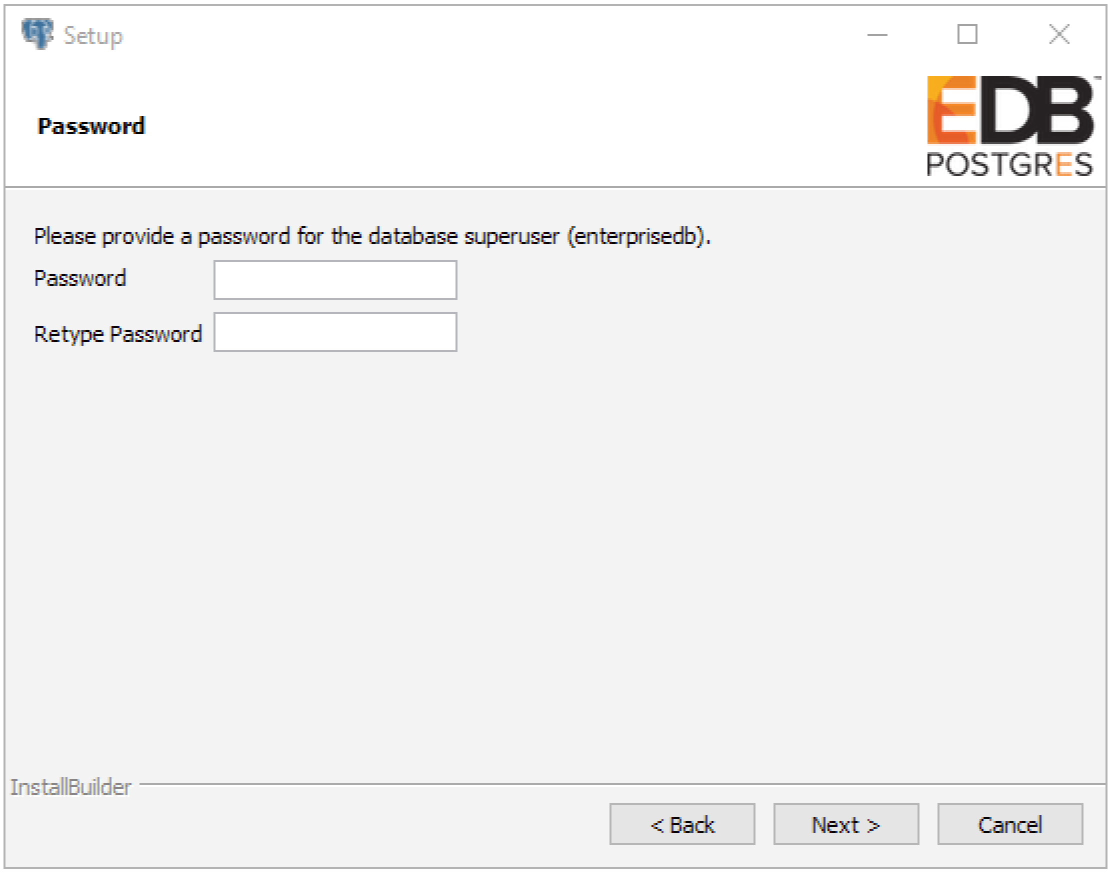
Compatible with PostgreSQL

If you select `Compatible with PostgreSQL`, Advanced Server will exhibit compatibility with PostgreSQL version 10. 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 4.9).



Password

Please provide a password for the database superuser (enterprisedb).

Password

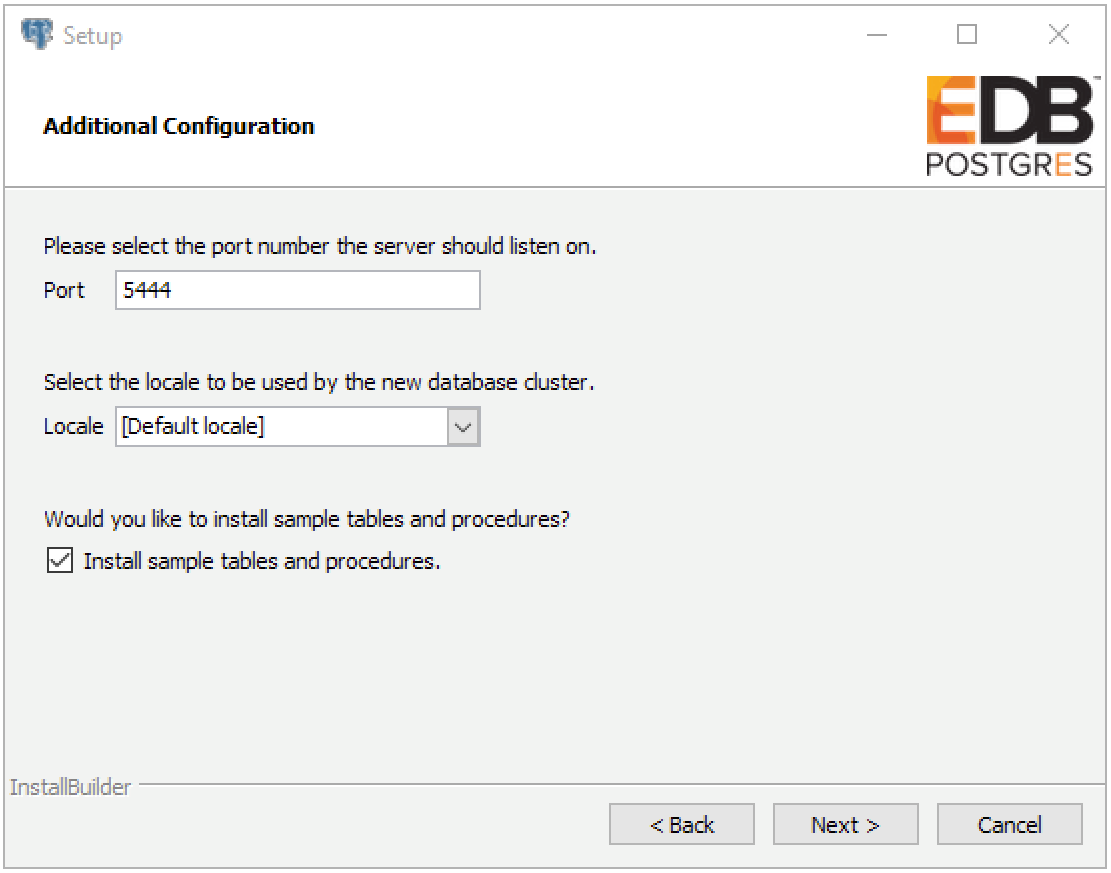
Retype Password

Figure 4.9 -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 4.10).



Please select the port number the server should listen on.

Port 5444

Select the locale to be used by the new database cluster.

Locale [Default locale]

Would you like to install sample tables and procedures?

☑ Install sample tables and procedures.

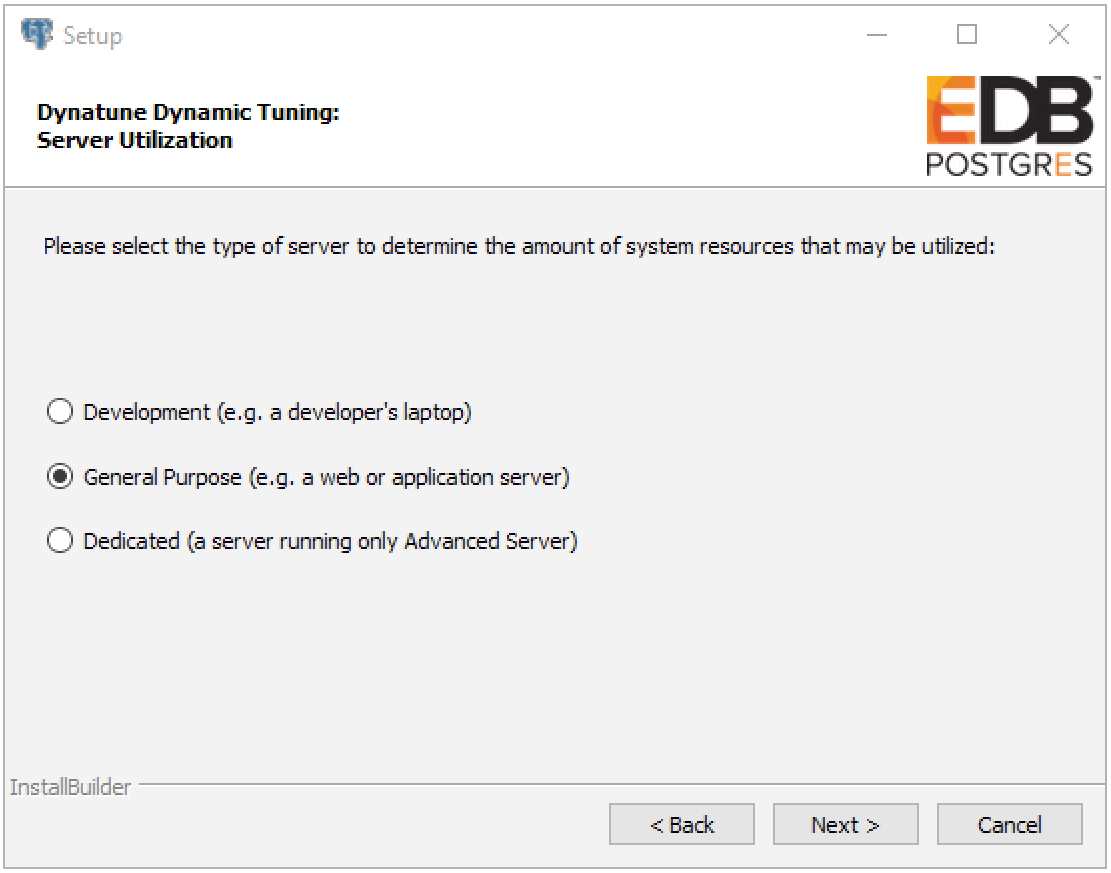
Figure 4.10 -The Additional Configuration window.

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 4.11).

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.



Dynatune Dynamic Tuning:
Server Utilization

Please select the type of server to determine the amount of system resources that may be utilized:

○ Development (e.g. a developer's laptop)

● General Purpose (e.g. a web or application server)

○ Dedicated (a server running only Advanced Server)

Figure 4.11 -The Dynatune Dynamic Tuning: Server Utilization window.

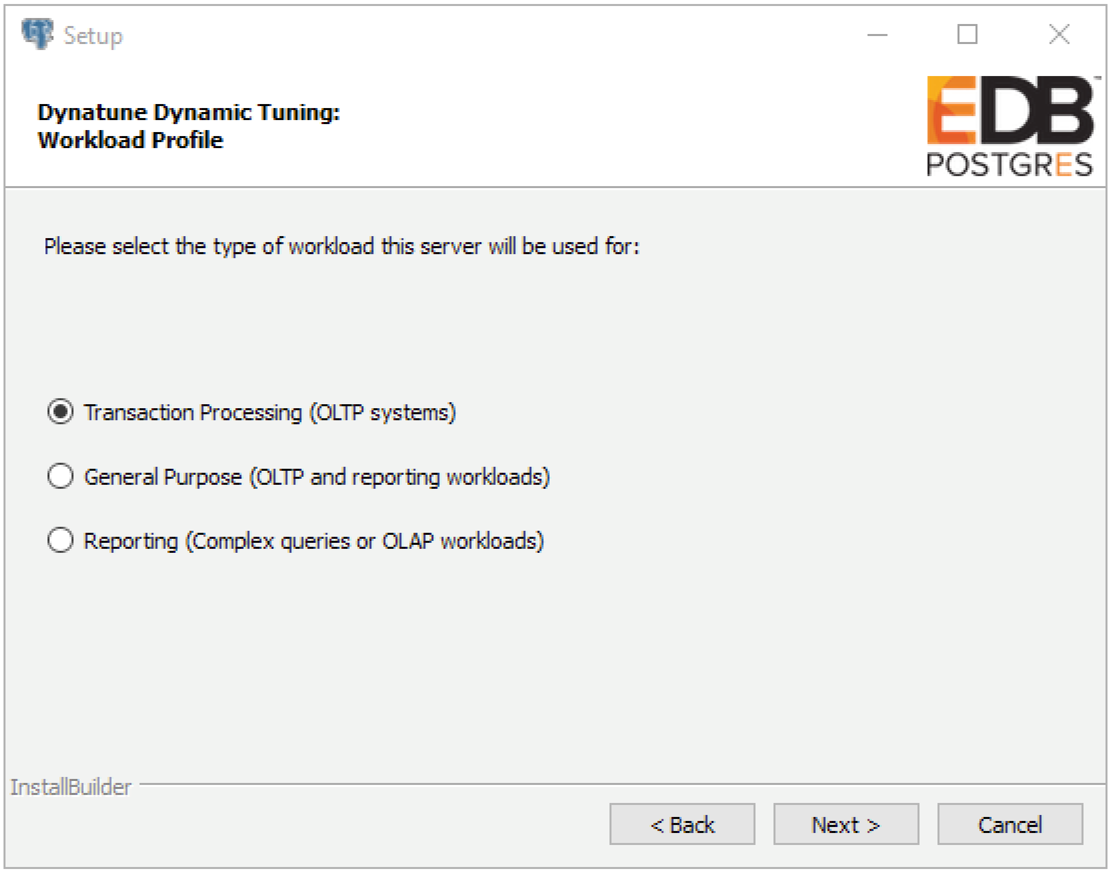
The edb_dynatune configuration parameter determines how Advanced Server allocates system resources. Use the radio buttons on the Server Utilization window to set the initial value of the edb_dynatune configuration parameter:

- 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 4.12).



**Dynatune Dynamic Tuning:
Workload Profile**

Please select the type of workload this server will be used for:

● Transaction Processing (OLTP systems)

○ General Purpose (OLTP and reporting workloads)

○ Reporting (Complex queries or OLAP workloads)

Figure 4.12 -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.

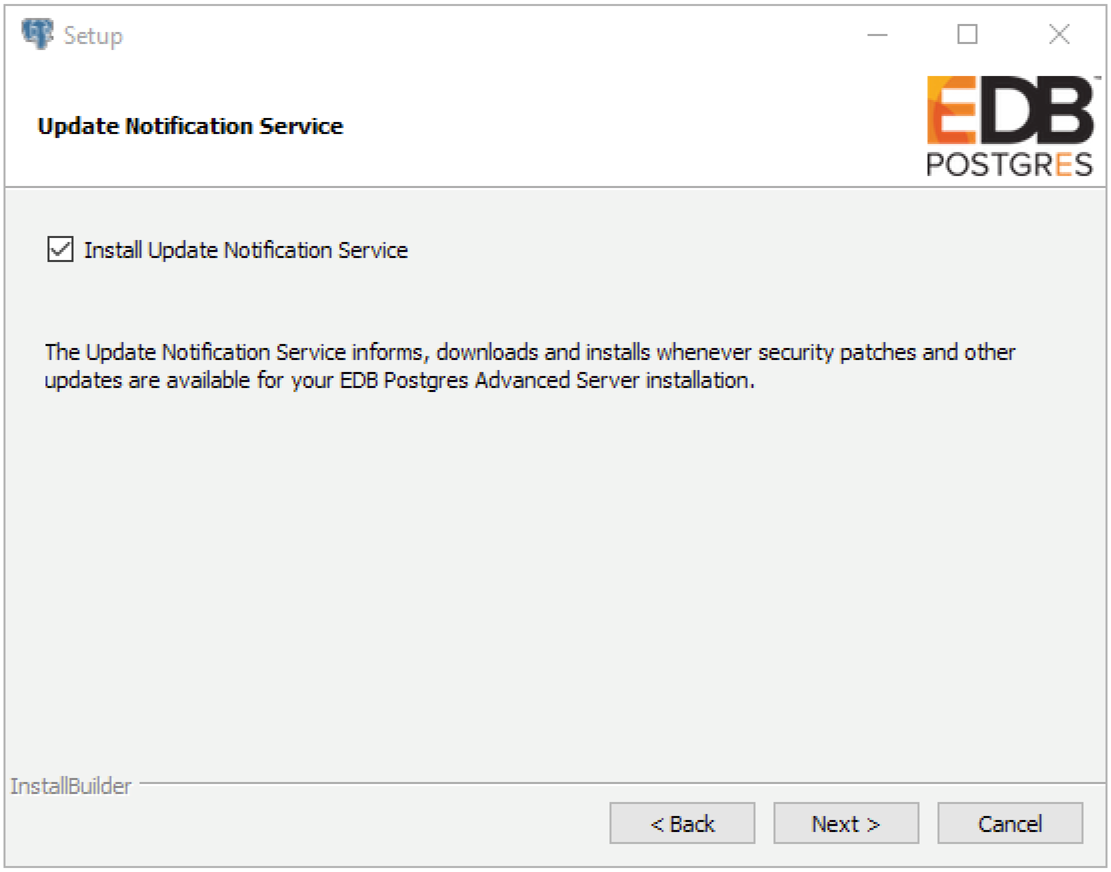
- Select Transaction Processing (OLTP systems) to specify an edb_dynatune_profile value of oltp. Recommended when Advanced Server is supporting heavy online transaction processing.
- Select General Purpose (OLTP and reporting workloads) to specify an edb_dynatune_profile value of mixed. Recommended for 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 4.13) opens.



Update Notification Service

☑ Install Update Notification Service

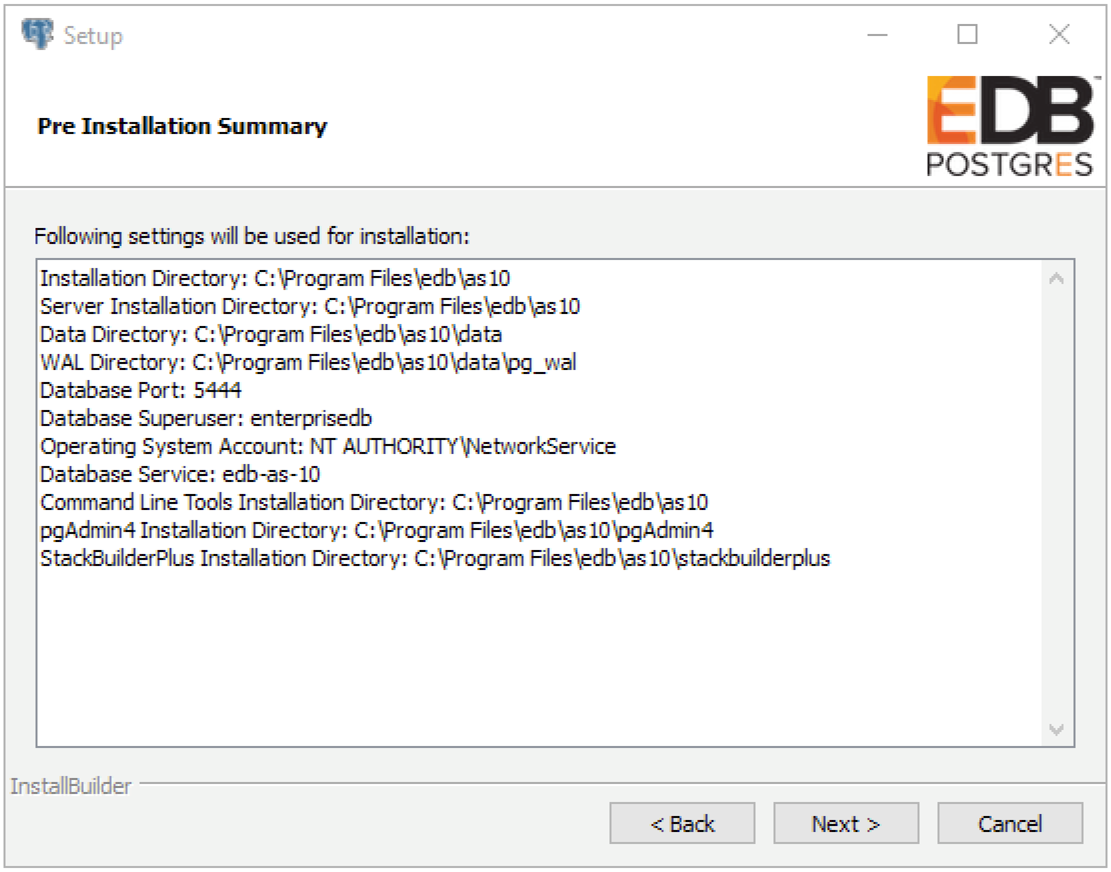
The Update Notification Service informs, downloads and installs whenever security patches and other updates are available for your EDB Postgres Advanced Server installation.

Figure 4.13 -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 4.14.



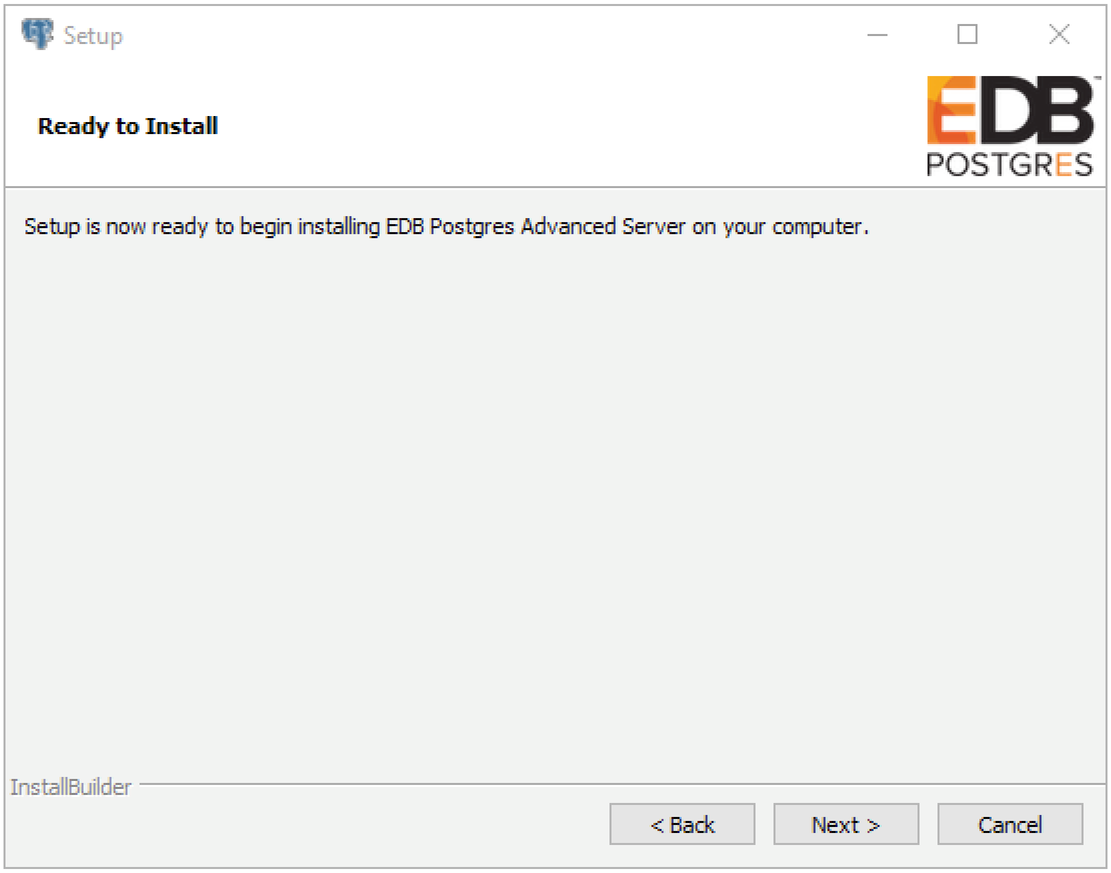
Following settings will be used for installation:

```
Installation Directory: C:\Program Files\edb\as10
Server Installation Directory: C:\Program Files\edb\as10
Data Directory: C:\Program Files\edb\as10\data
WAL Directory: C:\Program Files\edb\as10\data\pg_wal
Database Port: 5444
Database Superuser: enterprisedb
Operating System Account: NT AUTHORITY\NetworkService
Database Service: edb-as-10
Command Line Tools Installation Directory: C:\Program Files\edb\as10
pgAdmin4 Installation Directory: C:\Program Files\edb\as10\pgAdmin4
StackBuilderPlus Installation Directory: C:\Program Files\edb\as10\stackbuilderplus
```

Figure 4.14 -The Pre Installation Summary.

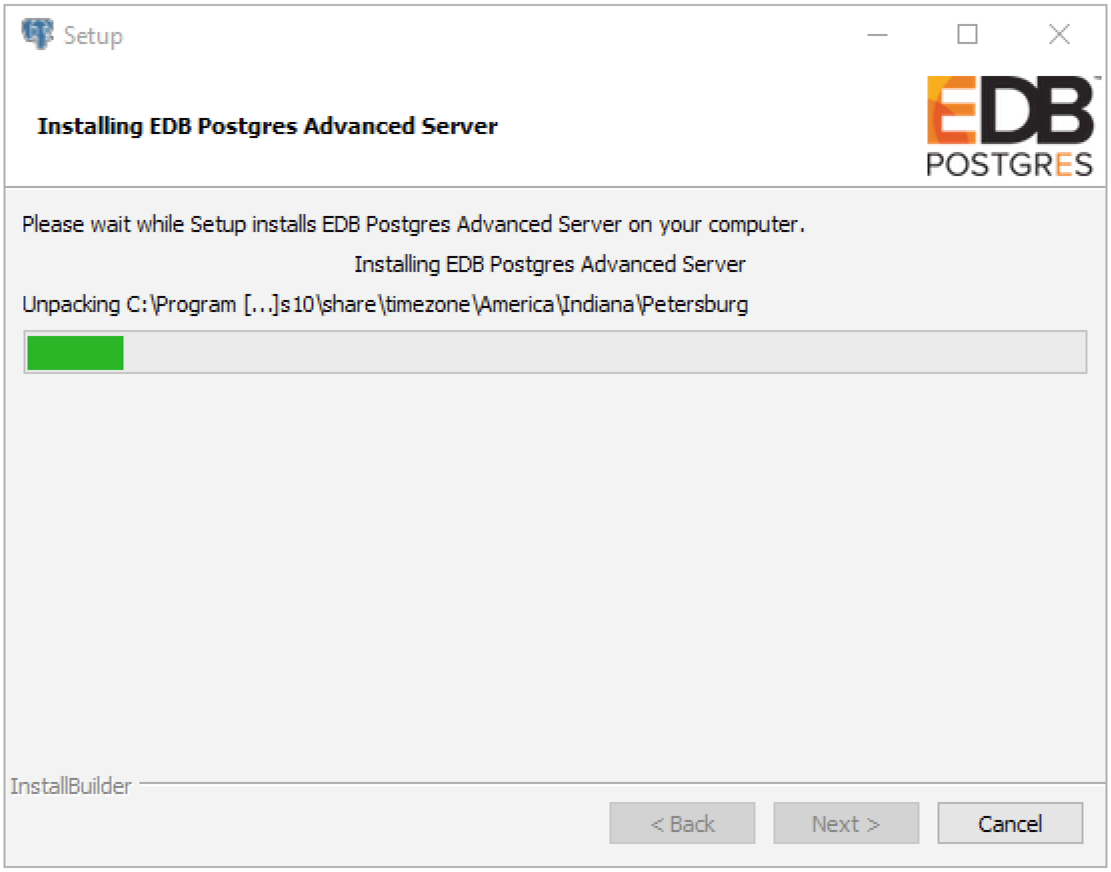
The `Pre Installation Summary` provides an overview of the options specified during the `Setup` process. Review the options before clicking `Next`; use the `Back` button to navigate back through the dialogs and update any options.

The `Ready to Install` window (see Figure 4.15) confirms that the installer has the information it needs about your configuration preferences to install Advanced Server. Click `Next` to continue.



Setup is now ready to begin installing EDB Postgres Advanced Server on your computer.

Figure 4.15 -The Ready to Install window.



Installing EDB Postgres Advanced Server

Please wait while Setup installs EDB Postgres Advanced Server on your computer.

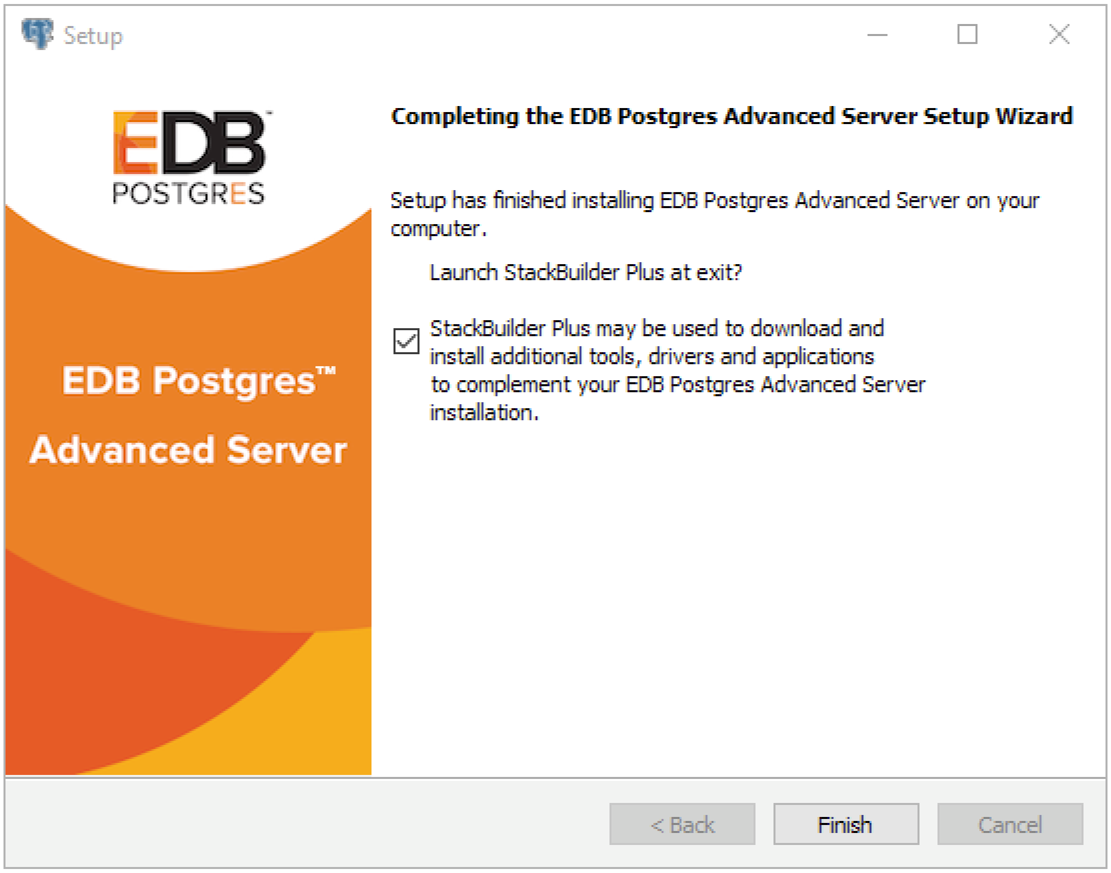
Installing EDB Postgres Advanced Server

Unpacking C:\Program [...]s10\share\timezone\America\Indiana\Petersburg

Figure 4.16 -Popup dialogs confirm the installation of supporting modules.

As each supporting module is unpacked and installed, the module's installation is confirmed with a progress bar (see Figure 4.16).

Before the `Setup` wizard completes the Advanced Server installation, it offers to `Launch StackBuilder Plus at exit?` (see Figure 4.17).



Completing the EDB Postgres Advanced Server Setup Wizard

Setup has finished installing EDB Postgres Advanced Server on your computer.

Launch StackBuilder Plus at exit?

☑ StackBuilder Plus may be used to download and install additional tools, drivers and applications to complement your EDB Postgres Advanced Server installation.

Figure 4.17 -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.

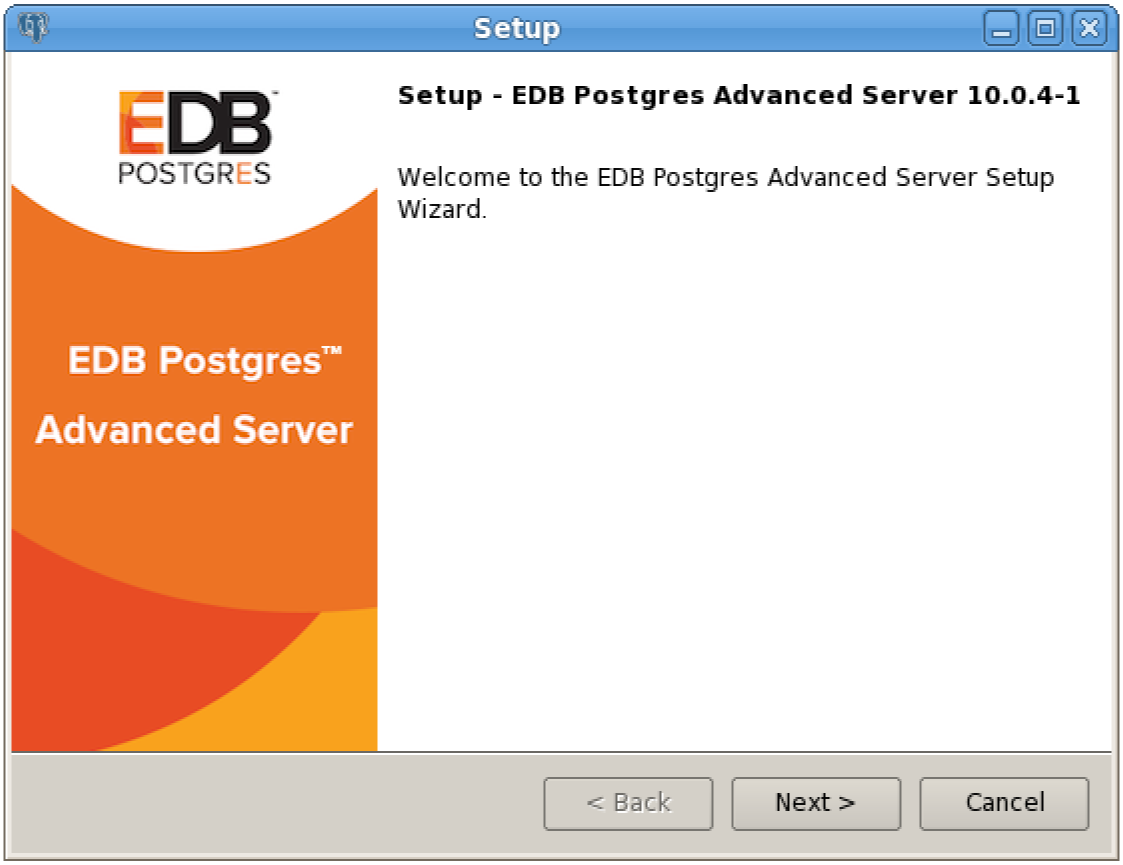
EDB Postgres StackBuilder Plus is included with the installation of Advanced Server and its core supporting components. StackBuilder Plus is a graphical tool that can update installed products, or download and add supporting modules (and the resulting dependencies) after your Advanced Server setup and installation completes. See Section 4.5 for more information about StackBuilder Plus.

4.3.2 Using the Graphical Installer on a Linux System

To use the graphical installation wizard on a Linux system, you must have superuser privileges. To invoke the `Setup` wizard, open a `Terminal` window, navigate to the directory that contains the Advanced Server installer, and enter the command:

```
./edb-as10-server-10.x.x.x-linux-x64.run
```

The wizard opens a `Language Selection` popup; select an installation language from the drop-down listbox and click `OK` to continue to the `Welcome` window (shown in Figure 4.18).



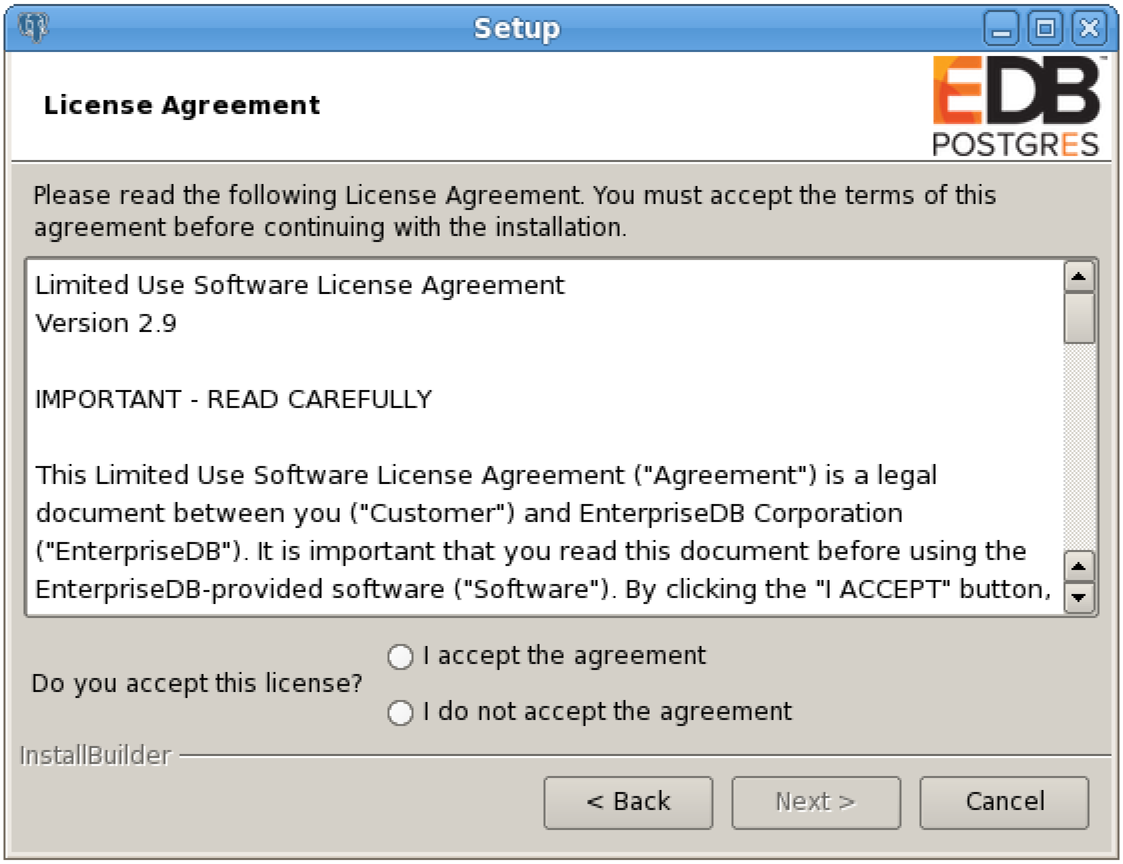
Setup - EDB Postgres Advanced Server 10.0.4-1

Welcome to the EDB Postgres Advanced Server Setup Wizard.

Figure 4.18 -The Advanced Server installer Setup welcome.

Click `Next` to continue.

The `License Agreement` window (shown in Figure 4.19) opens.



Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Limited Use Software License Agreement
Version 2.9

IMPORTANT - READ CAREFULLY

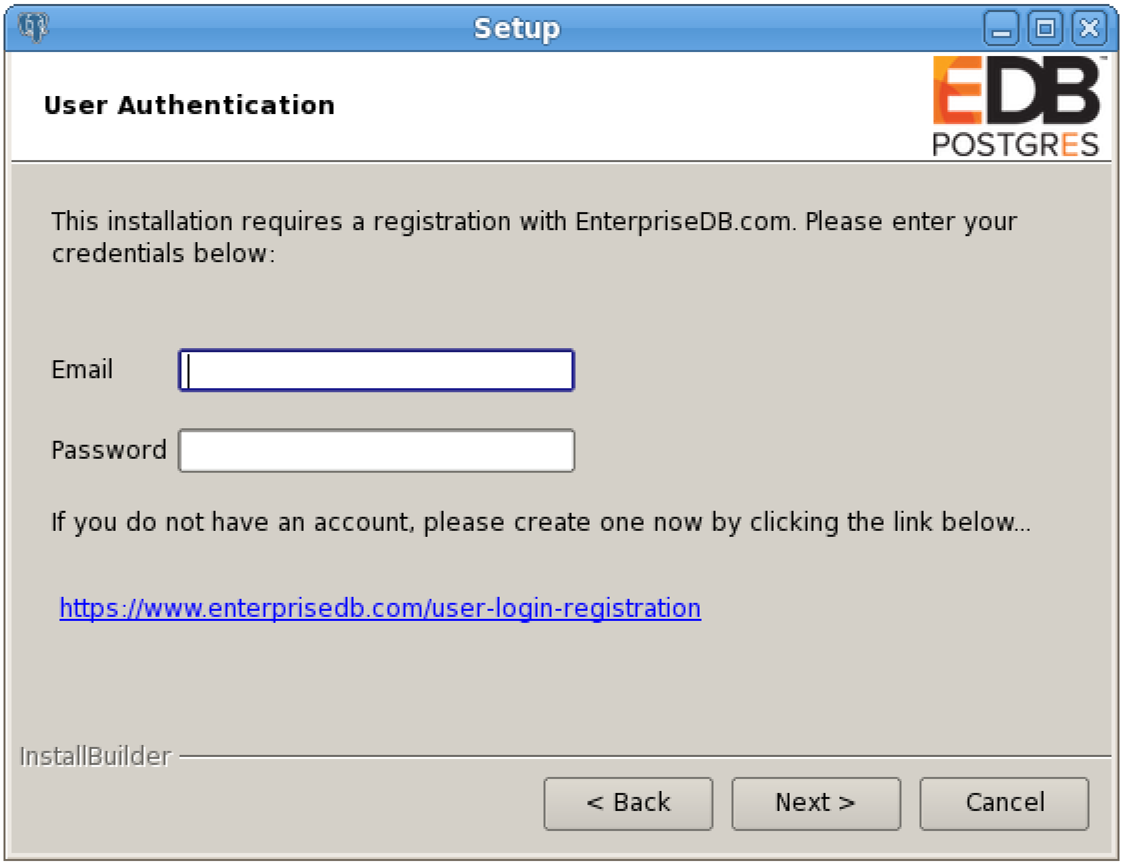
This Limited Use Software License Agreement ("Agreement") is a legal document between you ("Customer") and EnterpriseDB Corporation ("EnterpriseDB"). It is important that you read this document before using the EnterpriseDB-provided software ("Software"). By clicking the "I ACCEPT" button,

Do you accept this license? ○ I accept the agreement ○ I do not accept the agreement

Figure 4.19 -The EnterpriseDB License Agreement.

Review the EnterpriseDB `License Agreement` carefully before selecting the radio button next to `I accept the agreement`. Click `Next` to continue to the `User Authentication` window.

The `User Authentication` window opens, as shown in Figure 4.20.



This installation requires a registration with EnterpriseDB.com. Please enter your credentials below:

Email

Password

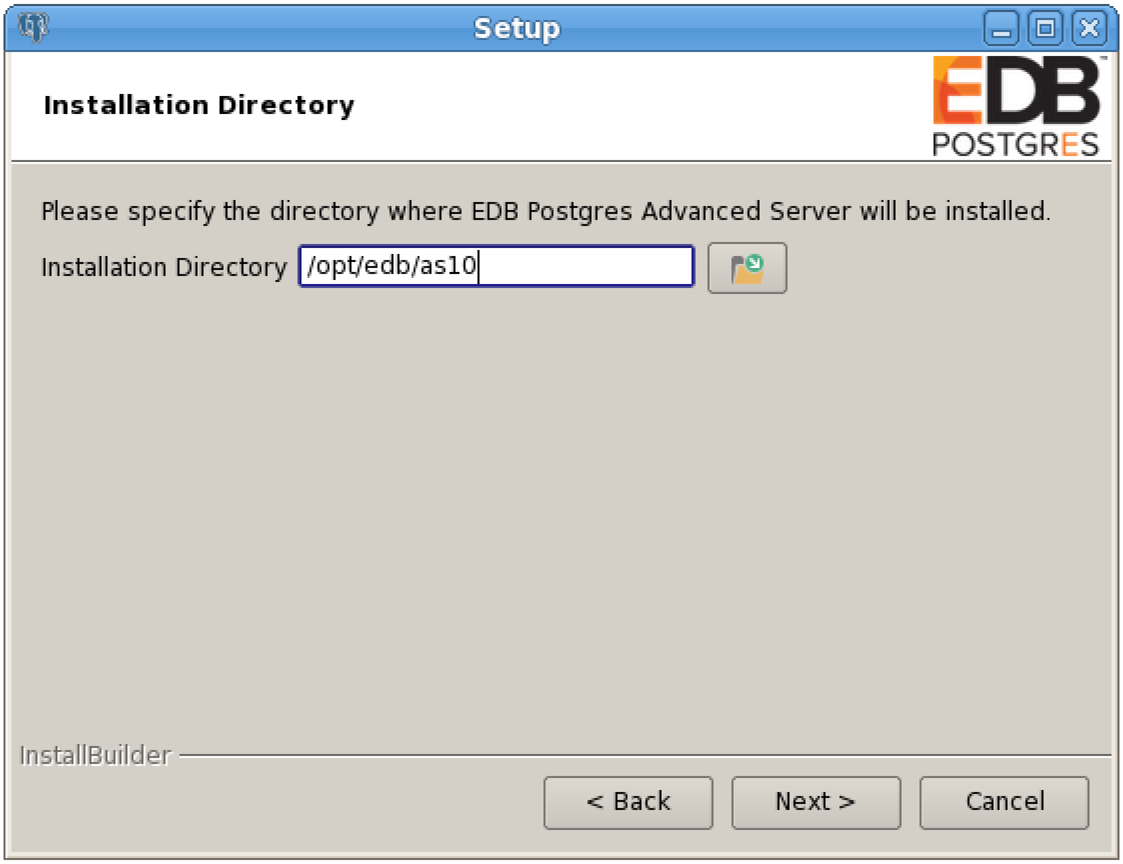
If you do not have an account, please create one now by clicking the link below...

https://www.enterprisedb.com/user-login-registration

Figure 4.20 -The User Authentication window.

Before continuing, you must provide the email address and password associated with your EnterpriseDB user account. Registration is free; if you do not have an EnterpriseDB user account, click the link provided to open a web browser, and enter your user information.

Enter the email address of a registered account in the `Email` field, and the corresponding password in the `Password` field, and click `Next` to continue to the `Installation Directory` window (see Figure 4.21).



Please specify the directory where EDB Postgres Advanced Server will be installed.

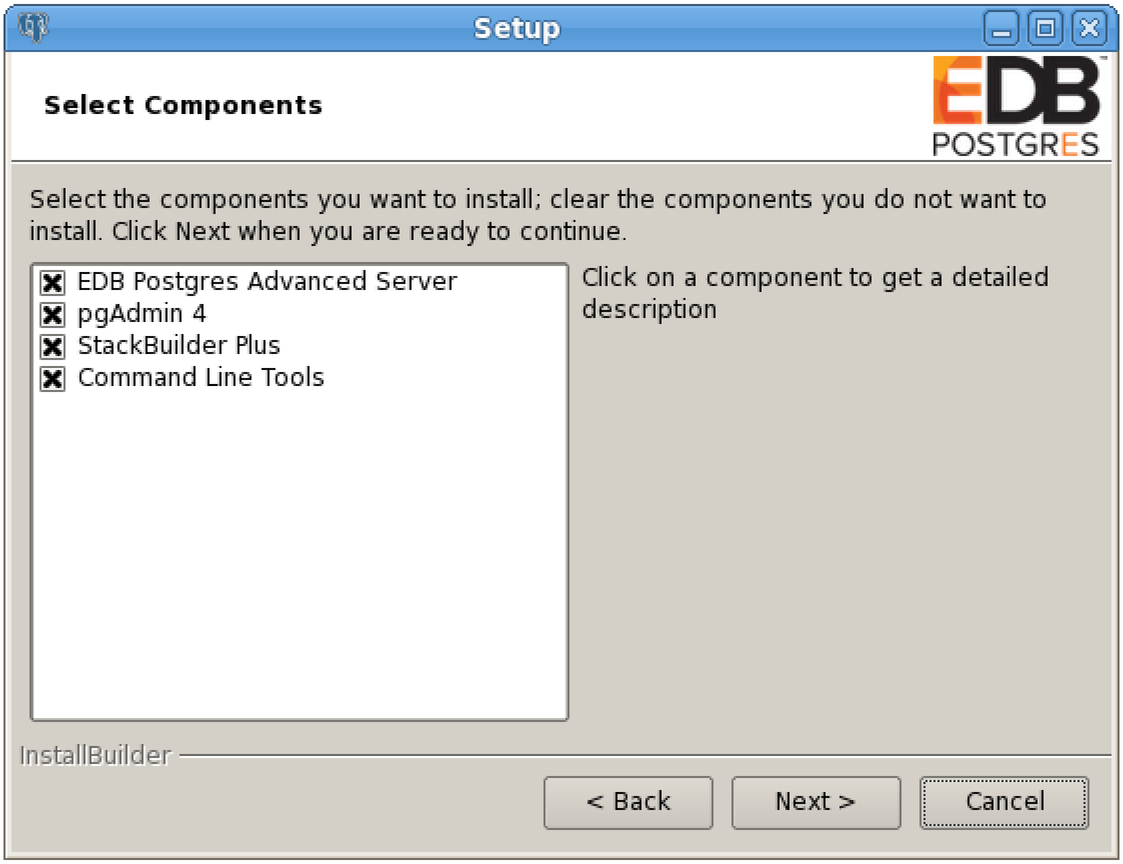
Installation Directory /opt/edb/as10

Figure 4.21 -The Installation Directory window.

By default, the Advanced Server installation directory is:

```
/opt/edb/as10
```

You can accept the default installation location, and click `Next` to continue, or click the `File Browser` icon to open a dialog and choose an alternate installation directory.



Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☒ EDB Postgres Advanced Server
☒ pgAdmin 4
☒ StackBuilder Plus
☒ Command Line Tools

Click on a component to get a detailed description

Figure 4.22 -The Select Components window.

The `Select Components` window (shown in Figure 4.22) contains a list of the tools and utilities that you can install with the Advanced Server `Setup` wizard. To omit a component from your installation, deselect the check to the left of the component name.

The `Setup` wizard can install the following components while installing Advanced Server 10:

EDB Postgres Advanced Server

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

pgAdmin 4

Select the `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 4.5 for more information about StackBuilder Plus.

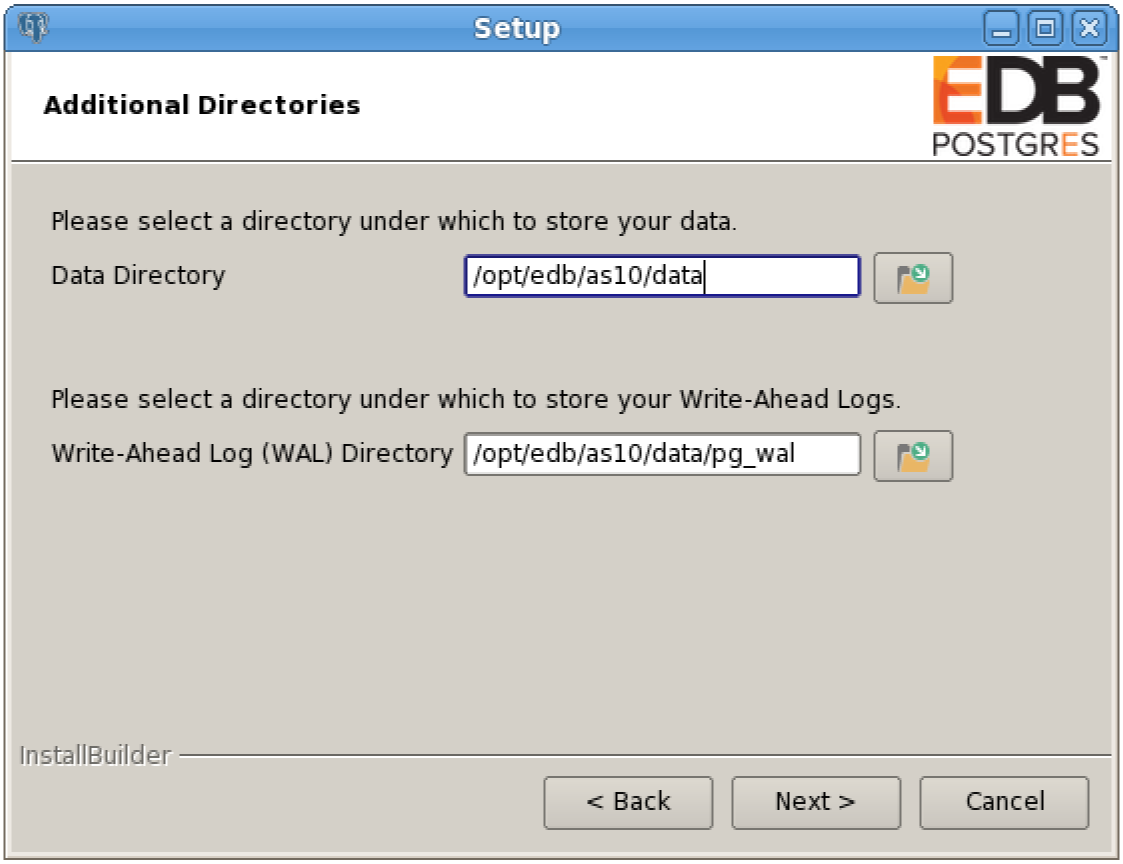
Command Line Tools

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.

This option is 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 4.23).



Please select a directory under which to store your data.

Data Directory /opt/edb/as10/data

Please select a directory under which to store your Write-Ahead Logs.

Write-Ahead Log (WAL) Directory /opt/edb/as10/data/pg_wal

Figure 4.23 -The Additional Directories window.

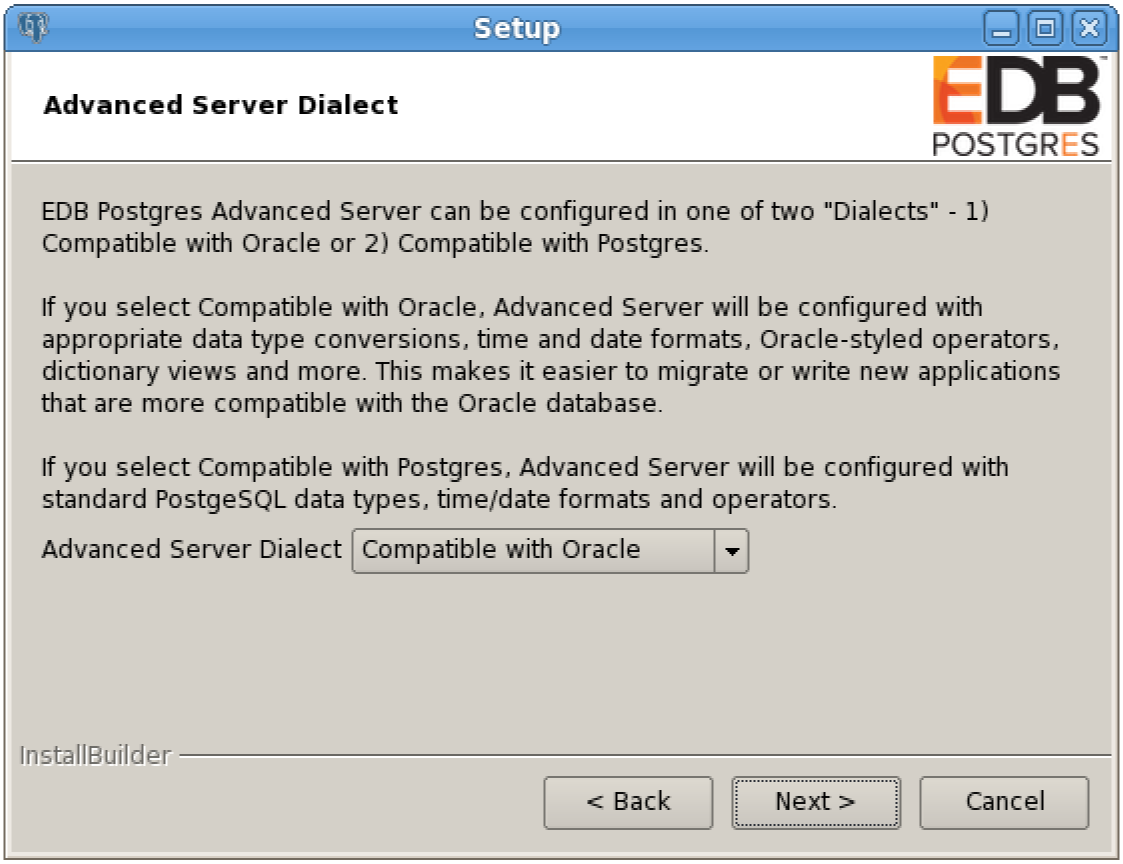
Use the fields in the `Additional Directories` window to specify locations for the Advanced Server `Data Directory` and `Write-Ahead Log (WAL) Directory`.

The default `Data Directory` is `/opt/edb/as10/data`. You can use the file selector icon to specify an alternate location.

The default location of the Advanced Server `Write-Ahead Log (WAL) Directory` is `opt/edb/as10/data/pg_wal`. Accept the default location, or specify an alternate location with the file selector icon.

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.

Click `Next` to continue to the `Advanced Server Dialect` window (shown in Figure 4.24).



Advanced Server Dialect

EDB Postgres Advanced Server can be configured in one of two "Dialects" - 1) Compatible with Oracle or 2) Compatible with Postgres.

If you select Compatible with Oracle, Advanced Server will be configured with appropriate data type conversions, time and date formats, Oracle-styled operators, dictionary views and more. This makes it easier to migrate or write new applications that are more compatible with the Oracle database.

If you select Compatible with Postgres, Advanced Server will be configured with standard PostgeSQL data types, time/date formats and operators.

Advanced Server Dialect: Compatible with Oracle

Figure 4.24 -The Configuration Mode window.

Use the drop-down listbox on the `Advanced Server Dialect` window to choose a server dialect. The server dialect specifies the compatibility features supported by Advanced Server.

By default, Advance Server installs with database compatibility with Oracle; you can choose between `Compatible with Oracle` and `Compatible with PostgreSQL` installation modes.

Compatible with Oracle Mode

If you select `Compatible with Oracle`, the installation will include the following features:

- Dictionary views compatible with Oracle databases.
- Oracle data type conversions.
- 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:

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

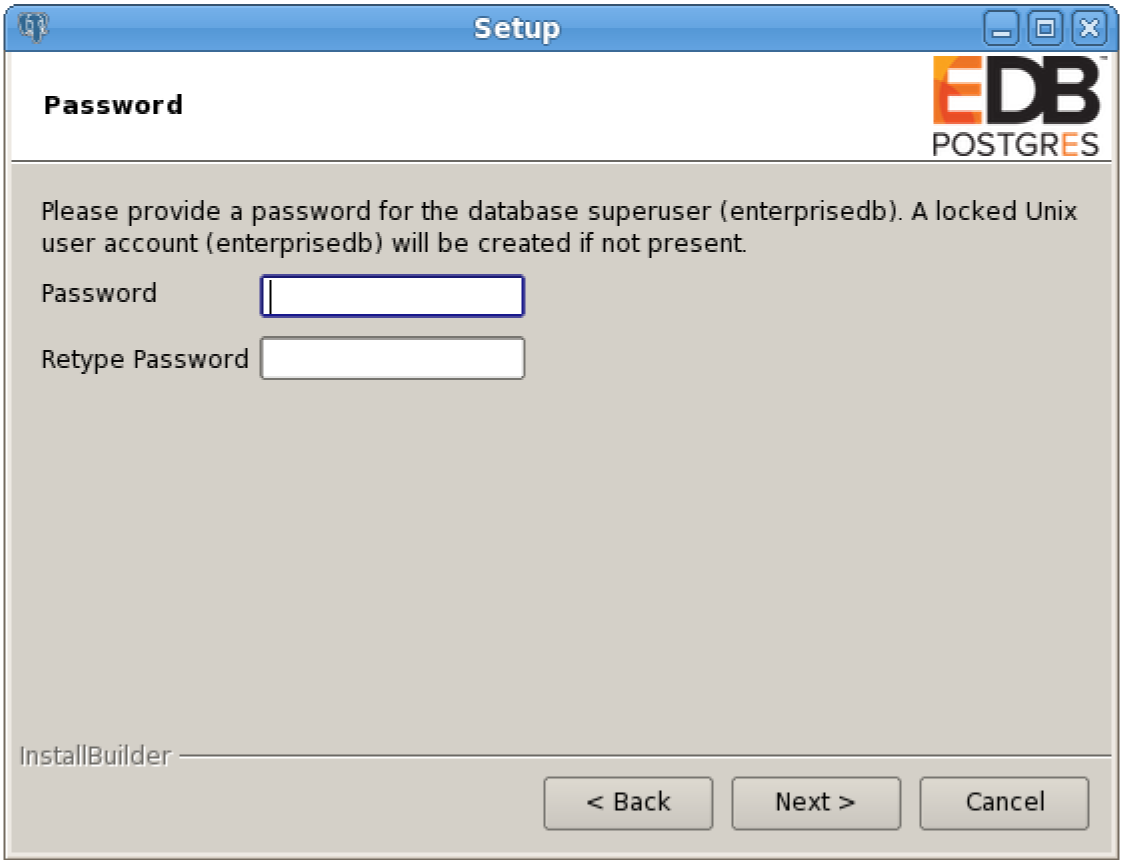
When installed in `Compatible with PostgreSQL` mode, Advanced Server exhibits complete compatibility with Postgres version 10.

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

http://www.postgresql.org

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

After specifying a configuration mode, click `Next` to continue to the `Password` window (shown in Figure 4.25).



Password

Please provide a password for the database superuser (enterprisedb). A locked Unix user account (enterprisedb) will be created if not present.

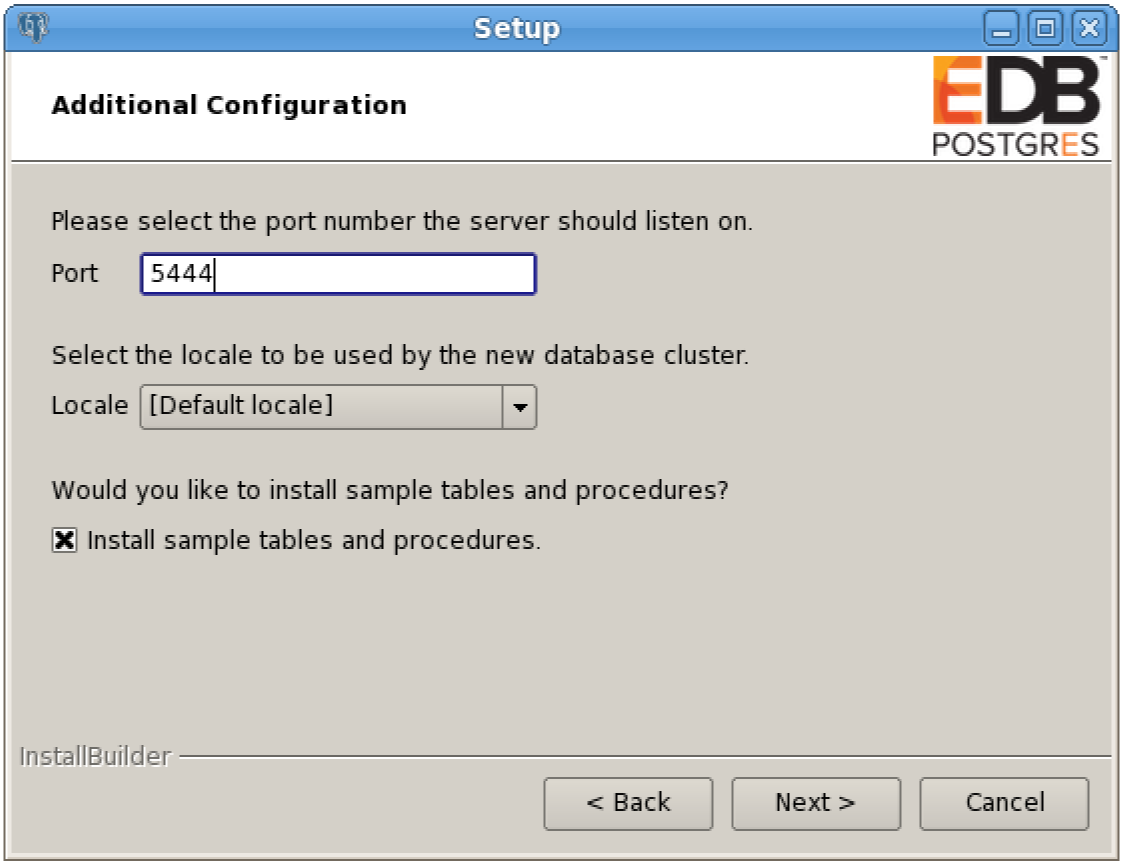
Password

Retype Password

Figure 4.25 -The Password window.

Advanced Server uses the password specified in 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.



Setup

Additional Configuration

Please select the port number the server should listen on.

Port 5444

Select the locale to be used by the new database cluster.

Locale [Default locale]

Would you like to install sample tables and procedures?

☒ Install sample tables and procedures.

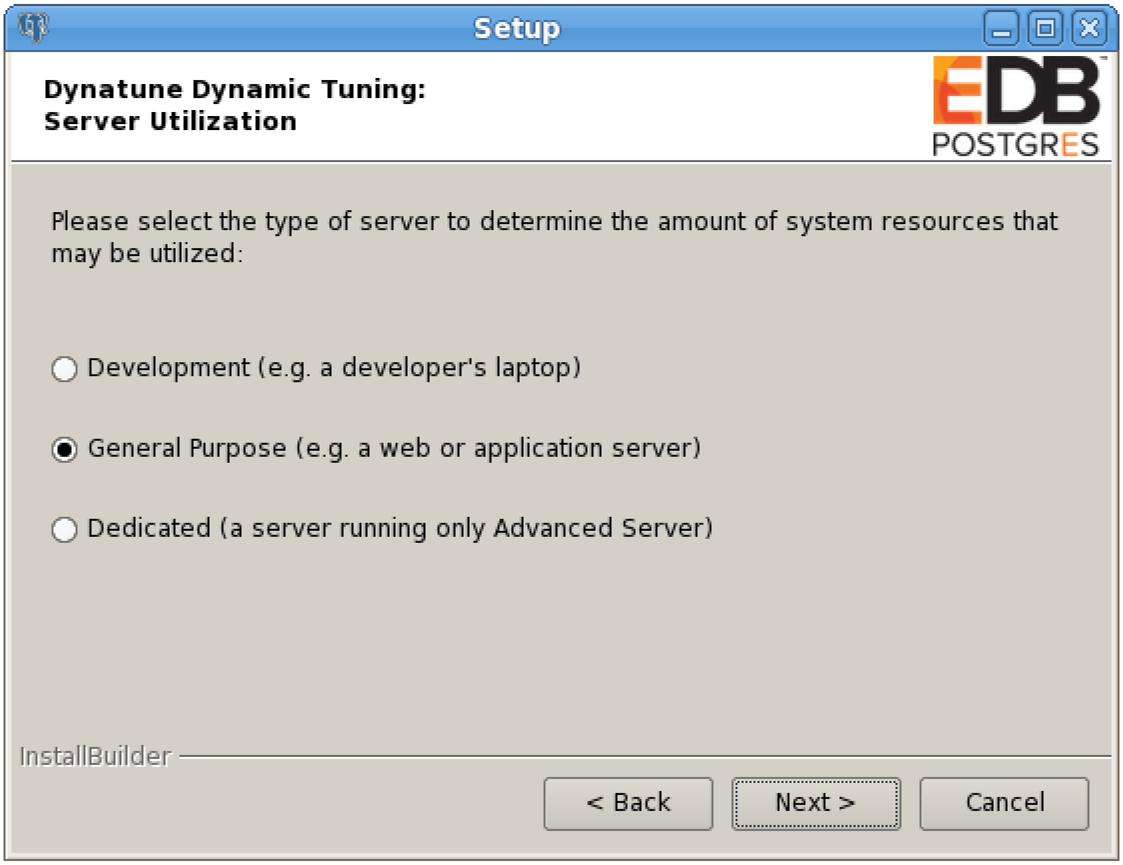
Figure 4.26 -The Additional Configuration window.

Use the fields on the Additional Configuration window (shown in Figure 4.26) 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 (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 selections on the `Additional Configuration` window, click `Next` to open the `Dynatune Dynamic Tuning: Server Utilization` window (shown in Figure 4.27).

The `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.



Please select the type of server to determine the amount of system resources that may be utilized:

○ Development (e.g. a developer's laptop)

● General Purpose (e.g. a web or application server)

○ Dedicated (a server running only Advanced Server)

Figure 4.27 -The Server Utilization window.

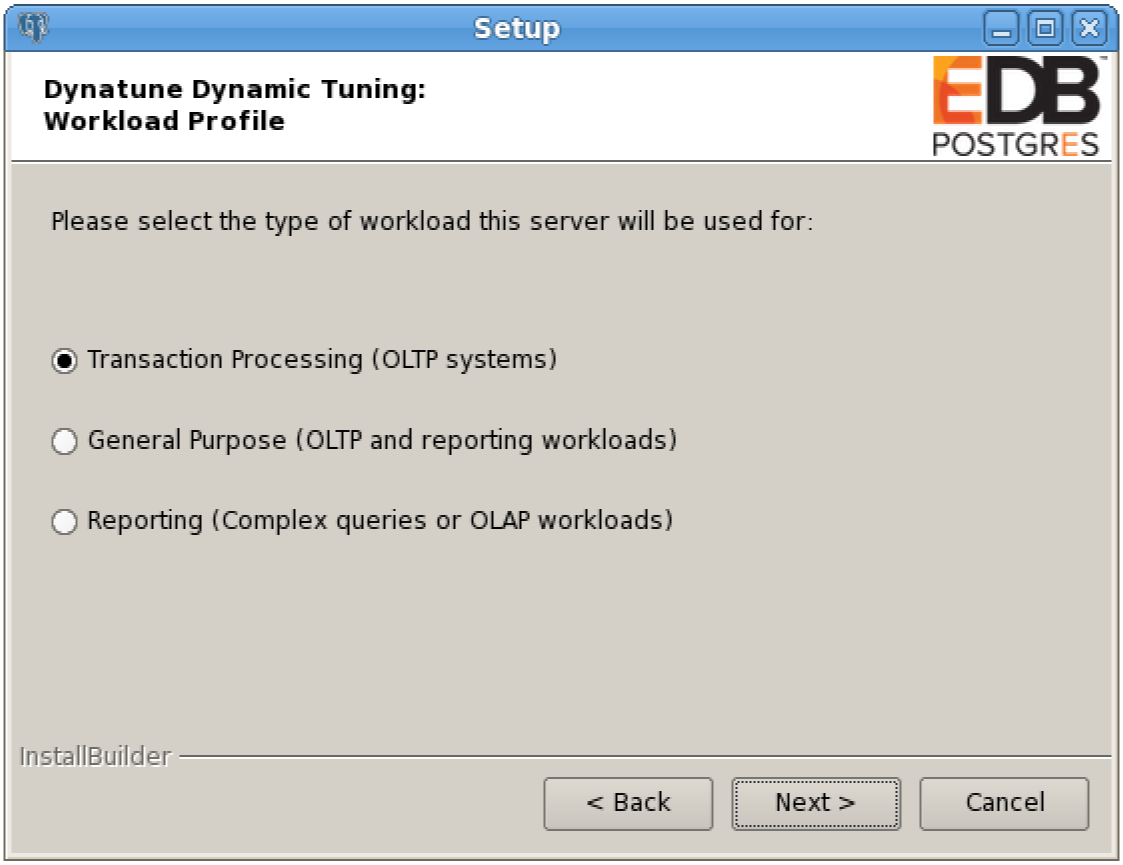
The `edb_dynatune configuration` parameter determines how Advanced Server allocates system resources. The radio buttons on the `Server Utilization` window set the initial value of the `edb_dynatune` configuration parameter.

- 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 the 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 4.28).



Dynatune Dynamic Tuning:
Workload Profile

Please select the type of workload this server will be used for:

● Transaction Processing (OLTP systems)

○ General Purpose (OLTP and reporting workloads)

○ Reporting (Complex queries or OLAP workloads)

Figure 4.28 -The 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.

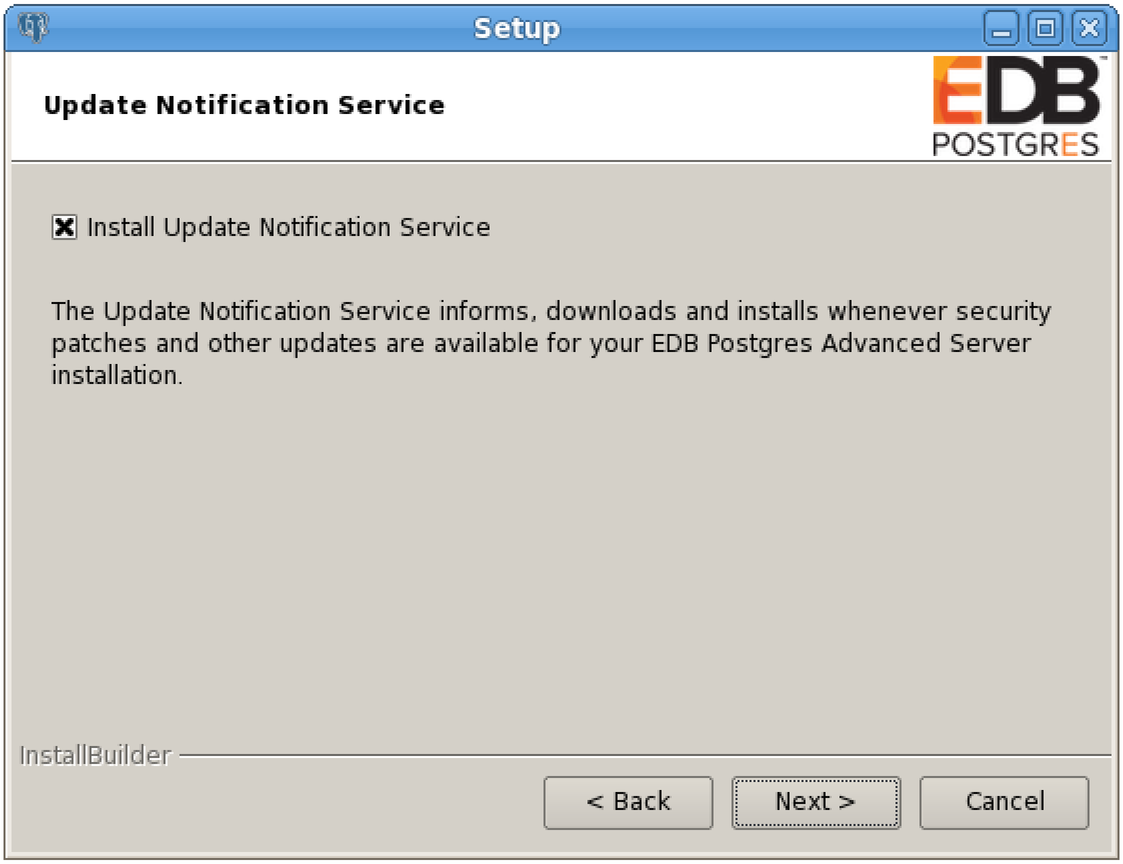
- Select `Transaction Processing (OLTP systems)` to specify an `edb_dynatune_profile` value of `oltp`. Recommended when Advanced Server is supporting heavy online transaction processing workloads.
- Select `General Purpose (OLTP and reporting workloads)` to specify an `edb_dynatune_profile` value of `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 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

After selecting the radio button that best describes the use of the system, click `Next` to continue. The `Advanced Configuration` window (shown in Figure 4.29) opens.



Update Notification Service

☒ Install Update Notification Service

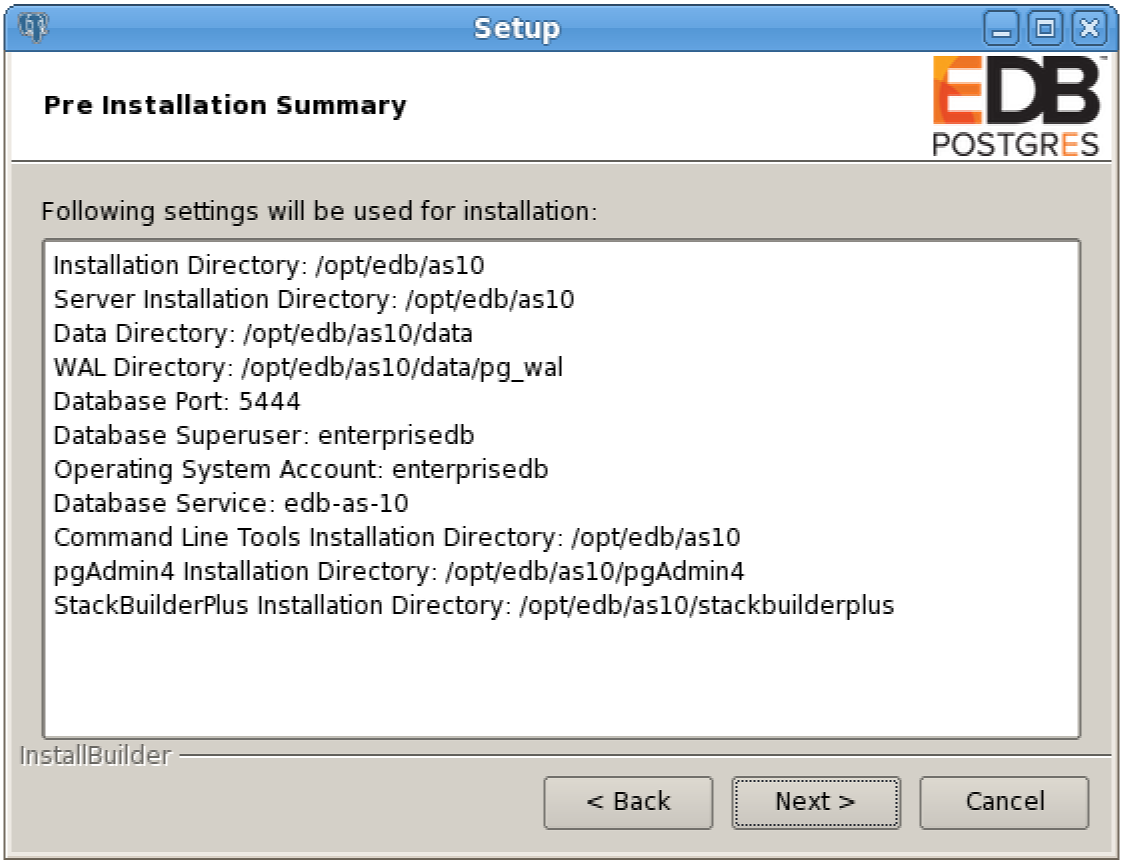
The Update Notification Service informs, downloads and installs whenever security patches and other updates are available for your EDB Postgres Advanced Server installation.

Figure 4.29 –The Update Notification Service selection window.

Check the box to the left of `Install Update Notification Service` to install the service and enable notifications. 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 install and enable the service when the system boots; clear the checkbox to skip the installation, or accept the defaults, and click `Next` to continue.

The `Pre Installation Summary` window opens (shown in Figure 4.30).



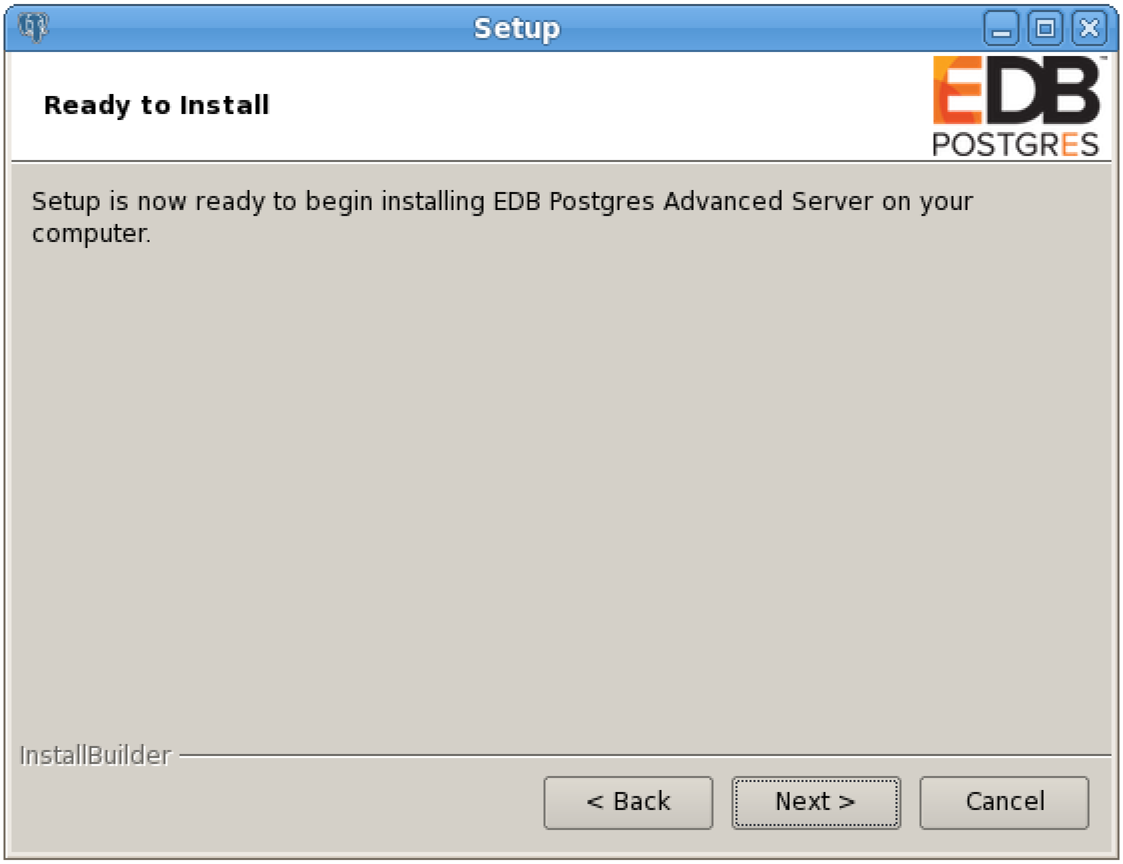
Following settings will be used for installation:

Installation Directory: /opt/edb/as10
Server Installation Directory: /opt/edb/as10
Data Directory: /opt/edb/as10/data
WAL Directory: /opt/edb/as10/data/pg_wal
Database Port: 5444
Database Superuser: enterprisedb
Operating System Account: enterprisedb
Database Service: edb-as-10
Command Line Tools Installation Directory: /opt/edb/as10
pgAdmin4 Installation Directory: /opt/edb/as10/pgAdmin4
StackBuilderPlus Installation Directory: /opt/edb/as10/stackbuilderplus

Figure 4.30 -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 to update any options.

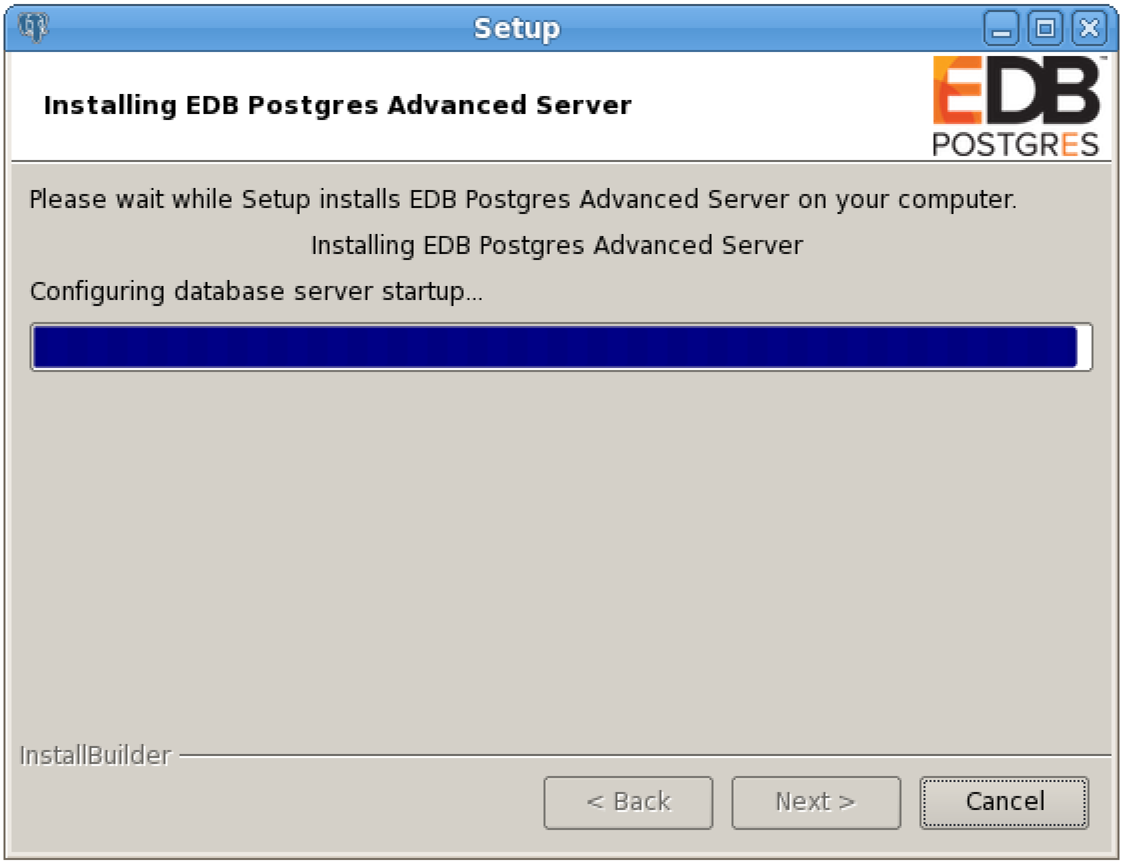
The `Ready to Install` window (shown in Figure 4.31) confirms that the installer has the information it needs about your configuration preferences to install Advanced Server.



Setup is now ready to begin installing EDB Postgres Advanced Server on your computer.

Figure 4.31 -The Ready to Install window.

Click `Next` to continue. The `Setup` wizard confirms the installation progress of Advanced Server via a progress bar (shown in Figure 4.32).



Installing EDB Postgres Advanced Server

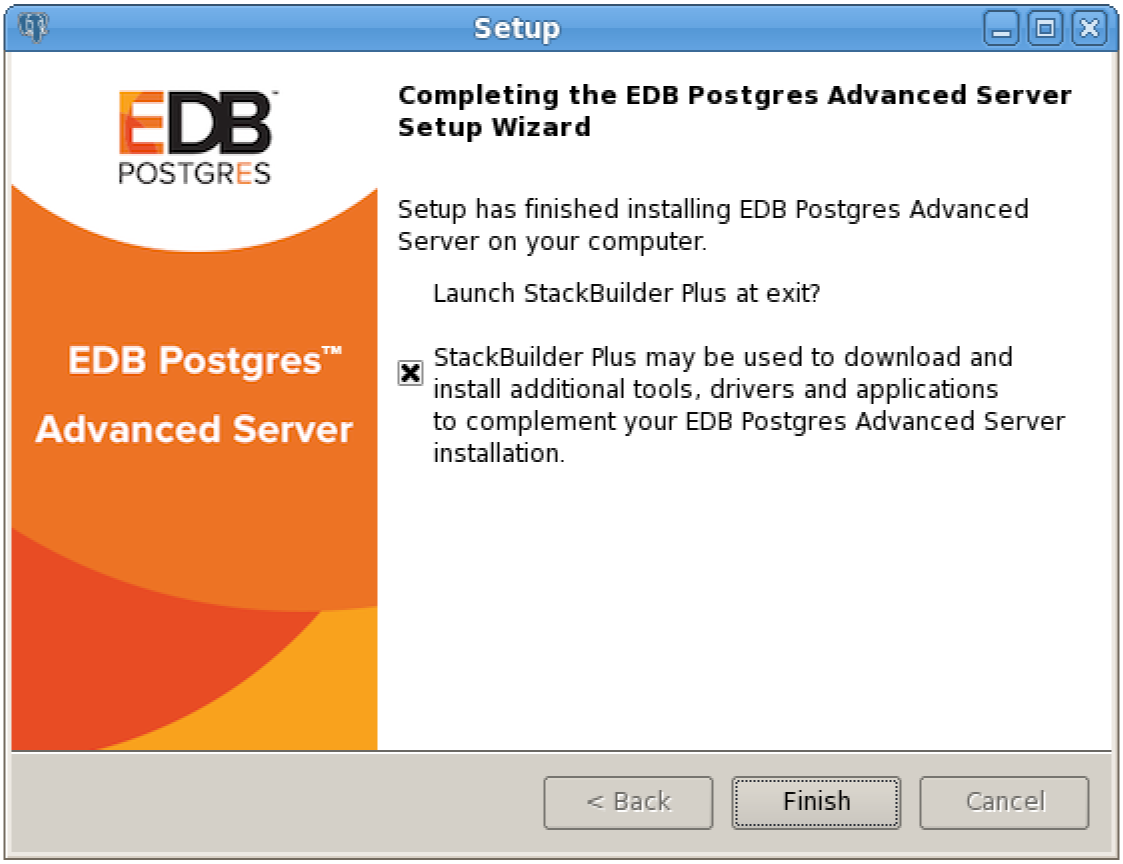
Please wait while Setup installs EDB Postgres Advanced Server on your computer.

Installing EDB Postgres Advanced Server

Configuring database server startup...

Figure 4.32 –The wizard confirms the installation progress.

If you have elected to add StackBuilder Plus to your installation, the `Setup` wizard will offer to `Launch Stack Builder Plus at exit?` (see Figure 4.33).



Completing the EDB Postgres Advanced Server Setup Wizard

Setup has finished installing EDB Postgres Advanced Server on your computer.

Launch StackBuilder Plus at exit?

☒ StackBuilder Plus may be used to download and install additional tools, drivers and applications to complement your EDB Postgres Advanced Server installation.

Figure 4.33 -The Setup wizard offers to Launch StackBuilder Plus at exit.

Clear the `StackBuilder Plus` checkbox and click `Finish` to complete the Advanced Server installation, or accept the default and proceed to StackBuilder Plus.

StackBuilder Plus provides a graphical interface that updates, downloads and installs applications and drivers that work with Advanced Server. For more information about StackBuilder Plus, see Section 4.5, *Using StackBuilder Plus*.

4.4 Invoking the Installer from the Command Line

The command line options of the Advanced Server installer offer functionality 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.
- Include the `--mode text` option when invoking the installer to perform an installation from the command line.
- 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 platforms. For a complete reference guide to the command line options, see Section 4.4.4, *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.

4.4.1 Performing a Text Mode Installation

To specify that the installer should run in text mode, include the `--mode text` command line option when invoking the installer. Text-mode installations are useful if you need to install on a remote server using ssh tunneling (and have access to a minimal amount of bandwidth), or if you do not have access to a graphical interface.

In text mode, the installer uses a series of command line questions to establish the configuration parameters. Text-mode installations are valid only on Linux systems.

You must assume superuser privileges before performing a text-mode installation. To perform a text-mode installation on a Linux system, navigate to the directory that contains the installation binary file and enter:

```
# ./edb-as10-server-10.x.x.x-linux-x64.run --mode text
```

At any point during the installation process, you can press `Ctrl-C` to abort the installation.

The text mode installer welcomes you to the `Setup Wizard`, and introduces the `License Agreement`. Use the `Enter` key to page through the license agreement:

```
Welcome to the EDB Postgres Advanced Server Setup Wizard.

----------------------------------------------------------
Please read the following License Agreement. You must
accept the terms of this agreement before continuing with
the installation.
Press [Enter] to continue:
```

Press `Enter` to continue reviewing the license agreement. After displaying the license, the installer prompts you to accept the license:

```
Do you accept this license? [y/n]:
```

After reading the license agreement, enter `y` to accept the agreement and proceed with the installation. Enter `n` if you do not accept the license agreement; this will abort the installation.

```
Press Enter to proceed.
```

Next, the installer will prompt you for the `User Authentication` information associated with your EnterpriseDB user account. There is no charge to register for an EnterpriseDB user account; if you do not have a user account, visit http://www.enterprisedb.com/user-login-registration to register.

```
This installation requires a registration with
EnterpriseDB.com. Please enter your credentials below. If
you do not have an account, Please create one now on
https://www.enterprisedb.com/user-login-registration
```

When prompted, enter the email address of a registered account, and then the corresponding password:

```
Email []:

Password :
```

By default, Advanced Server is installed in `/opt/edb/as10`:

```
Please specify the directory where EDB Postgres Advanced
Server will be installed.

Installation Directory [/opt/edb/as10]:
```

Enter an alternate location, or press `Enter` to accept the default and continue to the component selection portion of the installation process.

Next, the installer prompts you individually for each component that is to be installed with Advanced Server:

```
Select the components you want to install.
```

Enter `Y` or press `Enter` to accept the default value of `yes` after each component that you wish to include with the installation. Enter `n` to omit a component from the installation.

```
EDB Postgres Advanced Server [Y/n] :
pgAdmin 4 [Y/n] :
StackBuilder Plus [Y/n] :
Command Line Tools [Y/n] :
```

The Advanced Server components are:

EDB Postgres Advanced Server

Press `Y` or `enter` after the `EDB Postgres Advanced Server` option to install Advanced Server 10.

pgAdmin 4

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 4.5 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.

This option is required if you are installing Advanced Server or pgAdmin 4.

After selecting components for installation, confirm that the list is correct by entering `Y`; enter `n` to iterate through the list of components a second time.

```
Is the selection above correct? [Y/n]:
```

Next, the installer prompts you to specify the location of the `Additional Directories` required by Advanced Server. The default data directory is `/opt/edb/as10/data`. You can specify an alternate location, or press `Enter` to accept the default and continue.

```
Please select a directory under which to store your data.
Data Directory [/opt/edb/as10/data]:
```

The default location of the Advanced Server `Write-Ahead Log (WAL) Directory` is `/opt/edb/as10/data/pg_wal`. Press `Enter` to accept the default location and continue, or specify an alternate location.

```
Please select a directory under which to store your Write-
Ahead Logs.
Write-Ahead Log (WAL) Directory [/opt/edb/as10/data/pg_wal]:
```

Advanced Server uses write-ahead logs to help ensure 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 COMMIT a transaction, Advance Server writes contents of the write-ahead log to disk.

Next, the installer prompts you to select an Advanced Server Dialect:

```
EDB Postgres Advanced Server can be configured in one of
two "Dialects" - 1) Compatible with Oracle or 2) Compatible
with Postgres.

If you select Compatible with Oracle, Advanced Server will
be configured with appropriate data type conversions, time
and date formats, Oracle-styled operators, dictionary views
and more. This makes it easier to migrate or write new
applications that are more compatible with the Oracle
database.

If you select Compatible with Postgres, Advanced Server
will be configured with standard PostgeSQL data types,
time/date formats and operators.

Advanced Server Dialect

[1] Compatible with Oracle
[2] Compatible with PostgreSQL
Please choose an option [1] :
```

The configuration mode specifies the server dialect with which Advanced Server will be compatible; you can choose between Compatible with Oracle and Compatible with PostgreSQL installation modes.

Compatible with Oracle Mode

Installing Advanced Server in Compatible with Oracle mode provides 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:

Package	Functionality Compatible with Oracle Databases
dbms_alert	Provides the ability to register for, send and receive alerts.
dbms_aq	Provides queueing functionality for Advanced Server.

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

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

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

Compatible with PostgreSQL Mode

When installed in `Compatible with PostgreSQL` mode, Advanced Server exhibits complete compatibility with Postgres version 10. For more information about PostgreSQL functionality, visit the official PostgreSQL website at:

http://www.postgresql.org

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

Press `Enter` to accept the default configuration mode (`Compatible with Oracle`) and continue; enter `2` and press `Enter` to install in `Compatible with PostgreSQL` mode.

Next, the installer prompts you for a database superuser password:

```
Please provide a password for the database superuser
(enterprisedb). A locked Unix user account (enterprisedb)
will be created if not present.

Password :

Retype Password :
```

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

After entering a password in the `Password` field, confirm the password and press `Enter` to continue.

The installer prompts you for `Additional Configuration` information:

```
Additional Configuration
```

When prompted, enter the `Port` that the Advanced Server service will monitor for connections. By default, Advanced Server chooses the first available port after port number `5444`:

```
Please select the port number the server should listen on.

Port [5444]:
```

Specify a `Locale` by entering a locale number from the list shown. Accept the `Default locale` value to instruct the installer to use the system locale as the server locale.

```
Select the locale to be used by the new database cluster.

Locale

[1] [Default locale]
[2] C
[3] POSIX
```

```
Please choose an option [1] :
```

When prompted, enter `Y` (or press `Enter` to accept the default value) to install the sample tables and procedures for the database dialect specified by the compatibility mode (Oracle or PostgreSQL):

```
Would you like to install sample tables and procedures?
Install sample tables and procedures. [Y/n]:
```

Dynatune functionality allows Advanced Server to make optimal usage of the system resources available on the host machine. To facilitate performance tuning through Dynatune, the installer prompts you first for `Server Utilization` information:

```
Dynatune Dynamic Tuning: Server Utilization

Please select the type of server to determine the amount of
system resources that may be utilized:

[1] Development (e.g. a developer's laptop)
[2] General Purpose (e.g. a web or application server)
[3] Dedicated (a server running only EDB Postgres)
Please choose an option [2] :
```

The `edb_dynatune` configuration parameter determines how Advanced Server allocates system resources. Your selection will establish the initial value of `edb_dynatune`.

- Specify `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.
- Specify `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.
- Specify `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.

Enter a value of `1`, `2`, or `3`, or accept the default value of `2` (to indicate that the server will be used for `General Purpose` processing) and press `Enter` to continue.

Next, the Advanced Server installer prompts for information about the type of workload the system will be processing:

```
Dynatune Dynamic Tuning:
Workload Profile
```

```
Please select the type of workload this server will be used
for:

[1] Transaction Processing (OLTP systems)
[2] General Purpose (OLTP and reporting workloads)
[3] Reporting (Complex queries or OLAP workloads)
Please choose an option [1] :
```

The installer uses the `Workload Profile` to establish 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.

- Enter `1` to indicate `Transaction Processing (OLTP systems)` and set the value of `edb_dynatune_profile` to `oltp`. Recommended when Advanced Server is supporting heavy online transaction processing workloads.
- Enter `2` to indicate `General Purpose (OLTP and reporting workloads)` and set the value of `edb_dynatune_profile` to `mixed`. Recommended for servers that provide a mix of transaction processing and data reporting.
- Enter `3` to indicate `Reporting (Complex queries or OLAP workloads)` and set the value of `edb_dynatune_profile` to `reporting`. Recommended for database servers used for heavy data reporting.

After choosing a `Workload Profile`, press `Enter` to continue.

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

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

Update Notification Service

If enabled, the Update Notification Service notifies you of any available updates and security patches for your installation of Advanced Server.

```
Update Notification Service

Update Notification Service [Y/n]:
```

By default, the installer specifies that Advanced Server should enable the notification service when the system boots; specify `n` to disable the notification service, or accept the default, and press `Enter` to continue to the `Pre Installation Summary`:

```
Pre Installation Summary

Following settings will be used for installation:

Installation Directory: /opt/edb/as10
Data Directory: /opt/edb/as10/data
WAL Directory: /opt/edb/as10/data/pg_wal
Database Port: 5444
Database Superuser: enterprisedb
Operating System Account: enterprisedb
Database Service: edb-as-10
Command Line Tools Installation Directory: /opt/edb/as10
pgAdmin4 Installation Directory: /opt/edb/as10/pgAdmin4
StackBuilderPlus Installation Directory:
/opt/edb/as10/stackbuildeplus
Press [Enter] to continue:
```

The `Pre Installation Summary` lists the options specified during the installation setup process; review the listing and press `Enter` to continue; press `Enter` again to start the installation process.

```
Setup is now ready to begin installing EDB Postgres
Advanced Server on your computer.

Do you want to continue? [Y/n]:
```

The installer extracts the Advanced Server files and proceeds with the installation:

```
Please wait while Setup installs EDB Postgres Advanced
Server on your computer.

 Installing EDB Postgres Advanced Server
 0% ______________ 50% ______________ 100%
 ########################################
```

The installer informs you when the installation is complete.

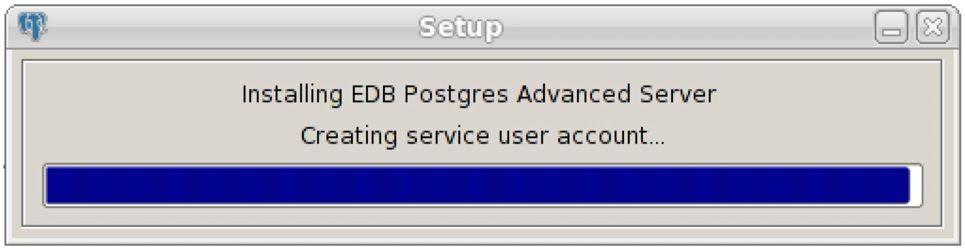
```
Setup has finished installing EDB Postgres Advanced Server
on your computer.
```

4.4.2 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 (shown in Figure 4.34).



Installing EDB Postgres Advanced Server

Creating service user account...

Figure 4.34 –The installation progress bar.

You must have superuser privileges to install Advanced Server using the `--mode unattended` option on a Linux system. On a Windows system, administrative privileges are required. If you are using the `--mode unattended` option to install Advanced Server with another installer, the calling installer must be invoked with superuser or administrative privileges.

On Linux

To install in unattended mode on a Linux machine, navigate to the directory that contains the Advanced Server installer and enter:

```
./edb-as10-server-10.x.x-x-linux-x64.run --mode unattended
--superpassword database_superuser_password --webusername
edb_user_name@email.com --webpassword edb_user_password
```

The `--superpassword` option specifies a password for the database superuser. If you omit the option, the database superuser password defaults to `enterprisedb`. The default password can be easily guessed by a potential intruder; be sure to provide a stronger password with the `--superpassword` option.

You must include the `--webusername` and `--webpassword` options to specify the identity of a registered EnterpriseDB user. There is no charge to register for an EnterpriseDB user account; if you do not have an account, you can create one at:

http://www.enterprisedb.com/user-login-registration

You can control configuration parameters for Advanced Server by specifying options at the command line, or by including the parameters (in option=value pairs) in a configuration file. A sample configuration file might include:

```
mode=unattended
prefix=/opt/edb/as10
datadir=/opt/edb/as10/data
serverport=5444
webusername=edb_user_name@email.com
webpassword=edb_user_password
```

Then, when you invoke the installer, include the `--optionfile` parameter, and the complete path to the configuration parameter file:

```
# ./edb-as10-server-10.x.x-x-linux-x64.run --optionfile
/$HOME/config_param
```

For more information about the command line options supported during an unattended installation, see Section 4.4.4, *Reference - Command Line Options*.

On Windows

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

```
edb-as10-server-10.x.x-x-windows-x64.exe --mode unattended
--superpassword database_superuser_password --
servicepassword system_password --webusername
edb_user_name@email.com --webpassword edb_user_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`).

Use the `--webusername` and `--webpassword` options to specify the identity of a registered EnterpriseDB user; if you do not have an account, you can create one at:

http://www.enterprisedb.com/user-login-registration

4.4.3 Performing an Installation with Limited Privileges

To perform an abbreviated installation of Advanced Server without access to root or administrative privileges, invoke the installer from the command line and include the `--extract-only` option. Invoking the installer with 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 and stop 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/10/static/app-initdb.html

https://www.postgresql.org/docs/10/static/app-pg-ctl.html

If you include the `--extract-only` option when you invoke the installer, 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. To invoke the installation script, you must have superuser privileges on Linux or administrative privileges on Windows.

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.

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:

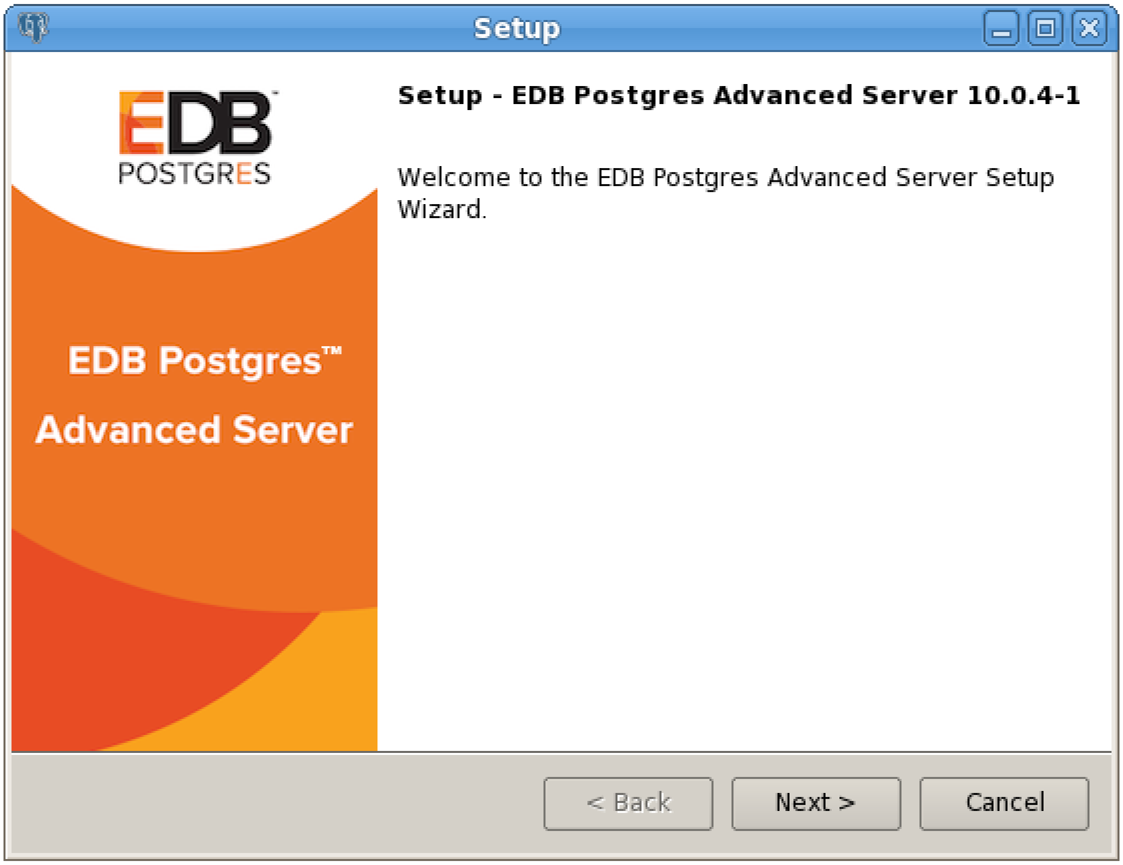
On Linux:

```
      ./edb-as10-server-10.x.x-x-linux-x64.run --extract-only yes
```

On Windows:

```
      edb-as10-server-10.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 4.35).

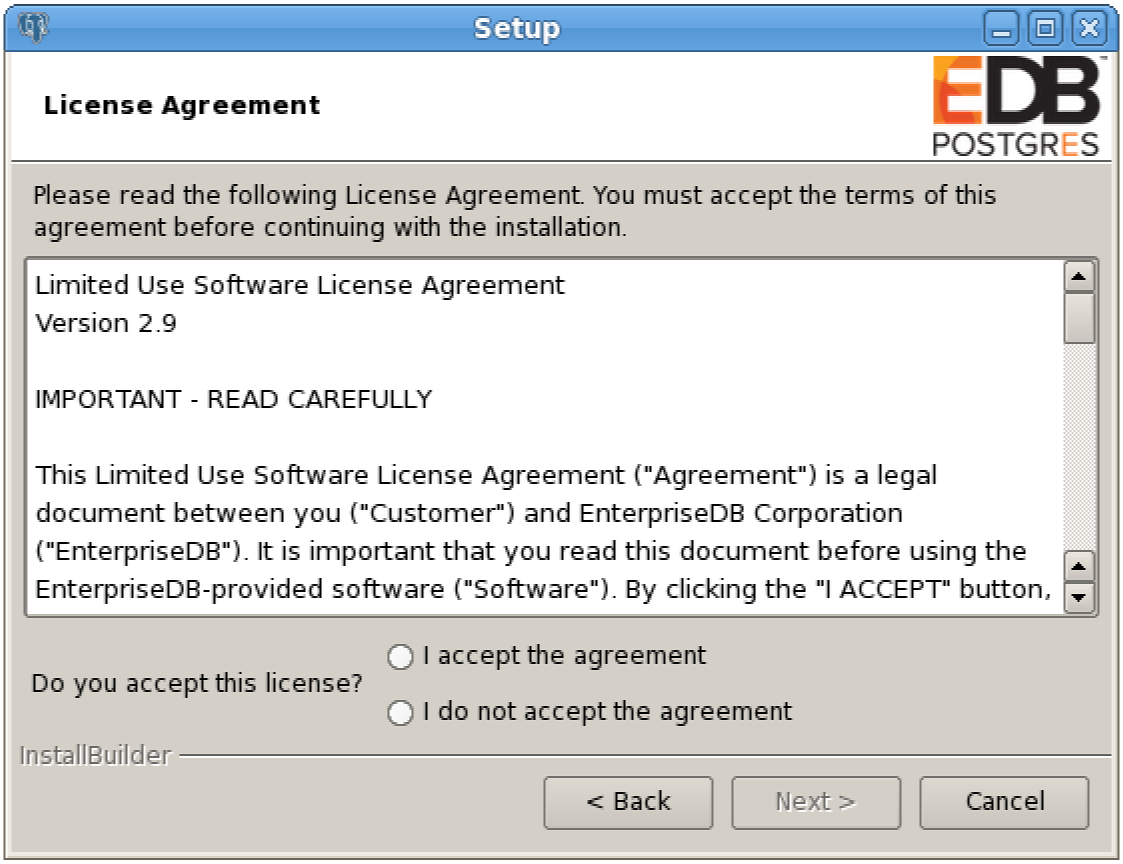


Setup - EDB Postgres Advanced Server 10.0.4-1

Welcome to the EDB Postgres Advanced Server Setup Wizard.

Figure 4.35 -The Welcome window.

Click `Next` to continue to the Advanced Server `License Agreement` (shown in Figure 4.36).



Setup

License Agreement

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Limited Use Software License Agreement
Version 2.9

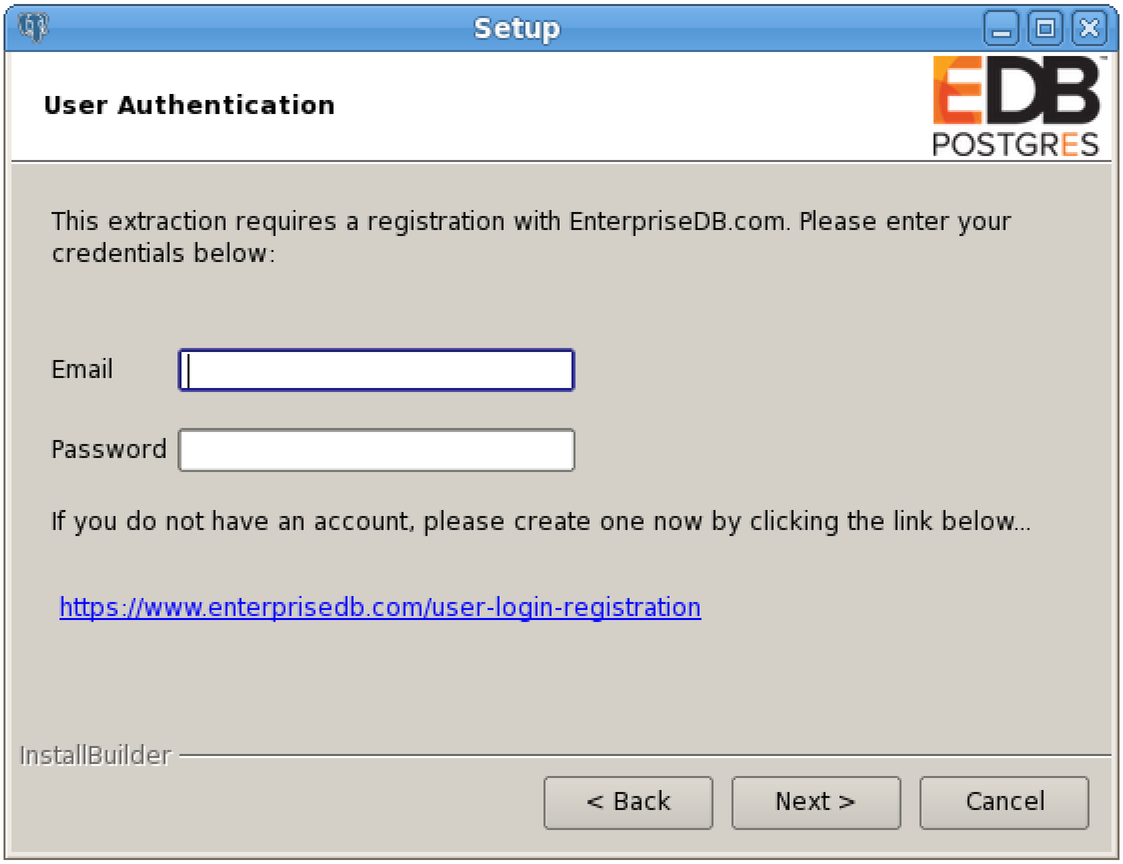
IMPORTANT - READ CAREFULLY

This Limited Use Software License Agreement ("Agreement") is a legal document between you ("Customer") and EnterpriseDB Corporation ("EnterpriseDB"). It is important that you read this document before using the EnterpriseDB-provided software ("Software"). By clicking the "I ACCEPT" button,

Do you accept this license?
○ I accept the agreement
○ I do not accept the agreement

Figure 4.36 -The Advanced Server license agreement.

After reading the license agreement, select the appropriate radio button and click `Next` to continue to the `User Authentication` window (shown in Figure 4.37).



This extraction requires a registration with EnterpriseDB.com. Please enter your credentials below:

Email

Password

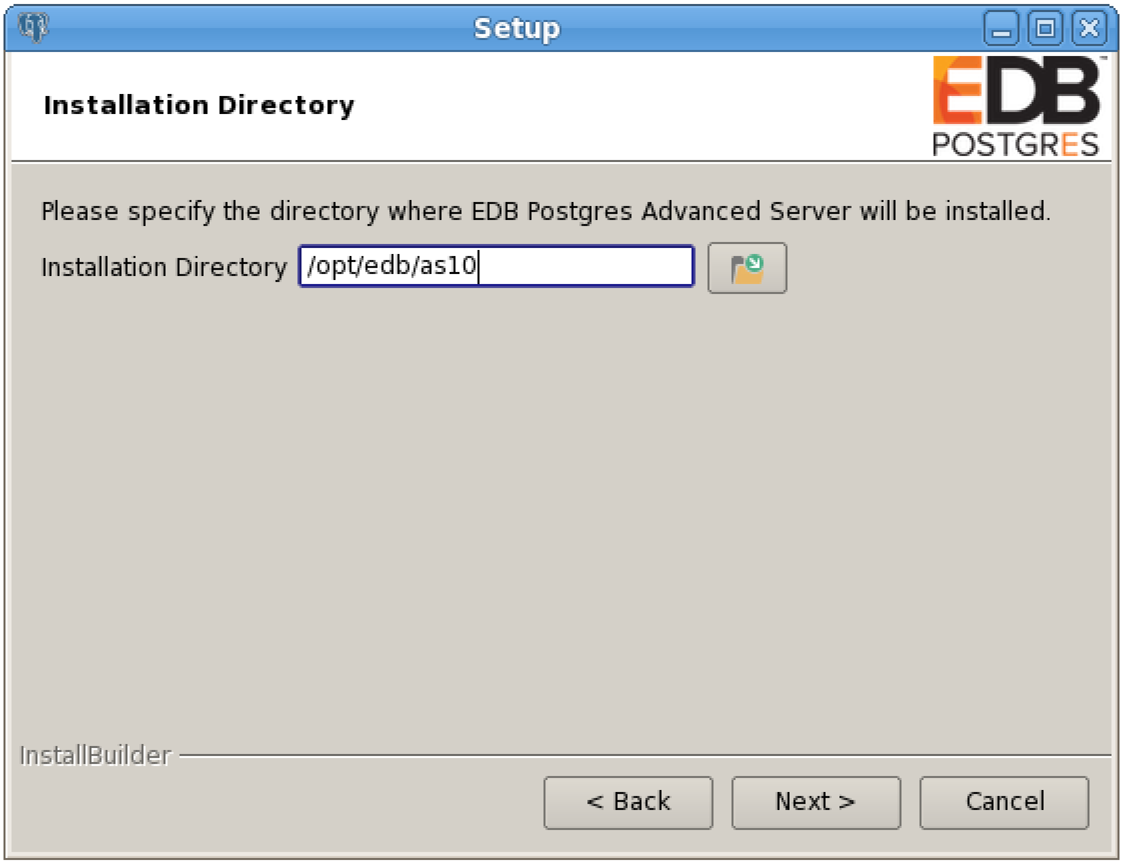
If you do not have an account, please create one now by clicking the link below...

https://www.enterprisedb.com/user-login-registration

Figure 4.37 -The Advanced Server User Authentication window.

Before continuing, you must provide the email address and password associated with your EnterpriseDB user account. Registration is free; if you do not have a user account, click the link provided to open a web browser, and register your user information.

Enter the email address of a registered account in the `Email` field, and the corresponding password in the `Password` field, and click `Next` to continue.



Please specify the directory where EDB Postgres Advanced Server will be installed.

Installation Directory /opt/edb/as10

Figure 4.38 -Specify an installation directory.

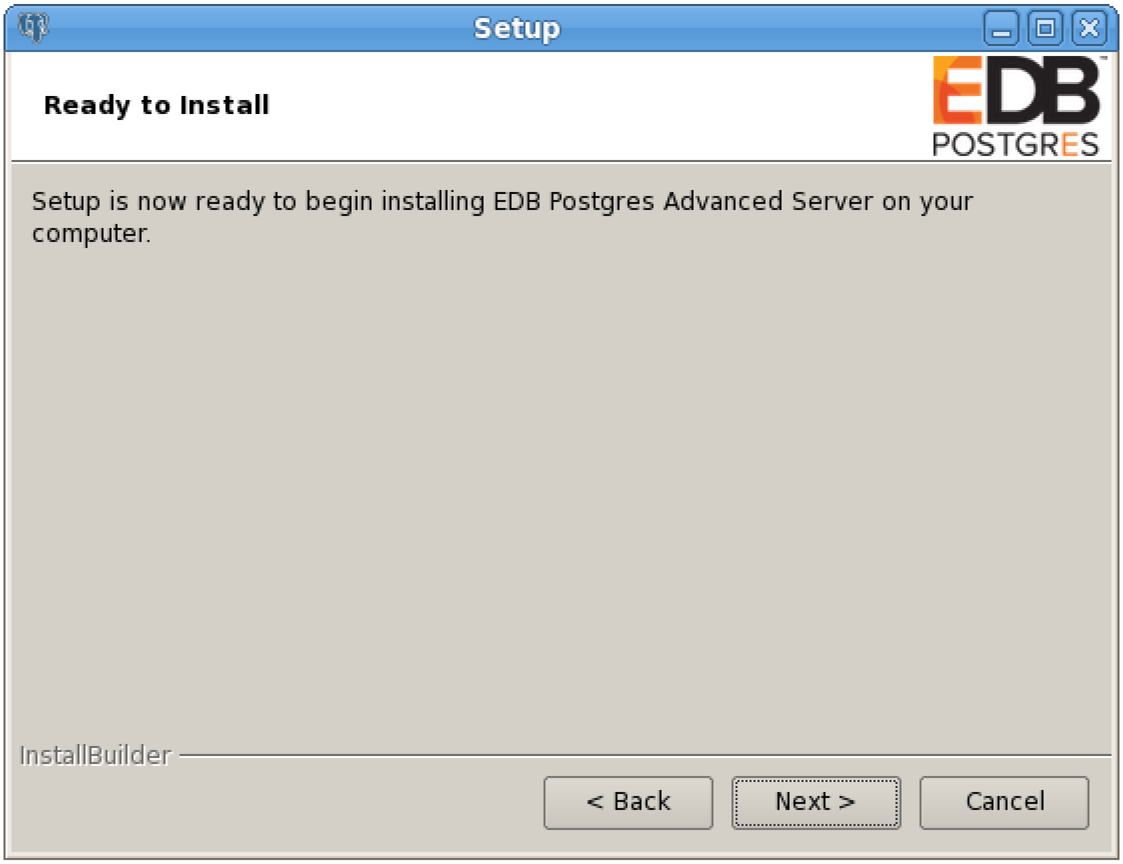
On Linux, the default Advanced Server installation directory is:

```
/opt/edb/as10
```

On Windows, the default Advanced Server installation directory is:

```
C:\Program Files\edb\as10
```

You can accept the default installation location, and click `Next` to continue to the `Ready to Install` window (shown in Figure 4.39), or optionally click the `File Browser` icon to choose an alternate installation directory.

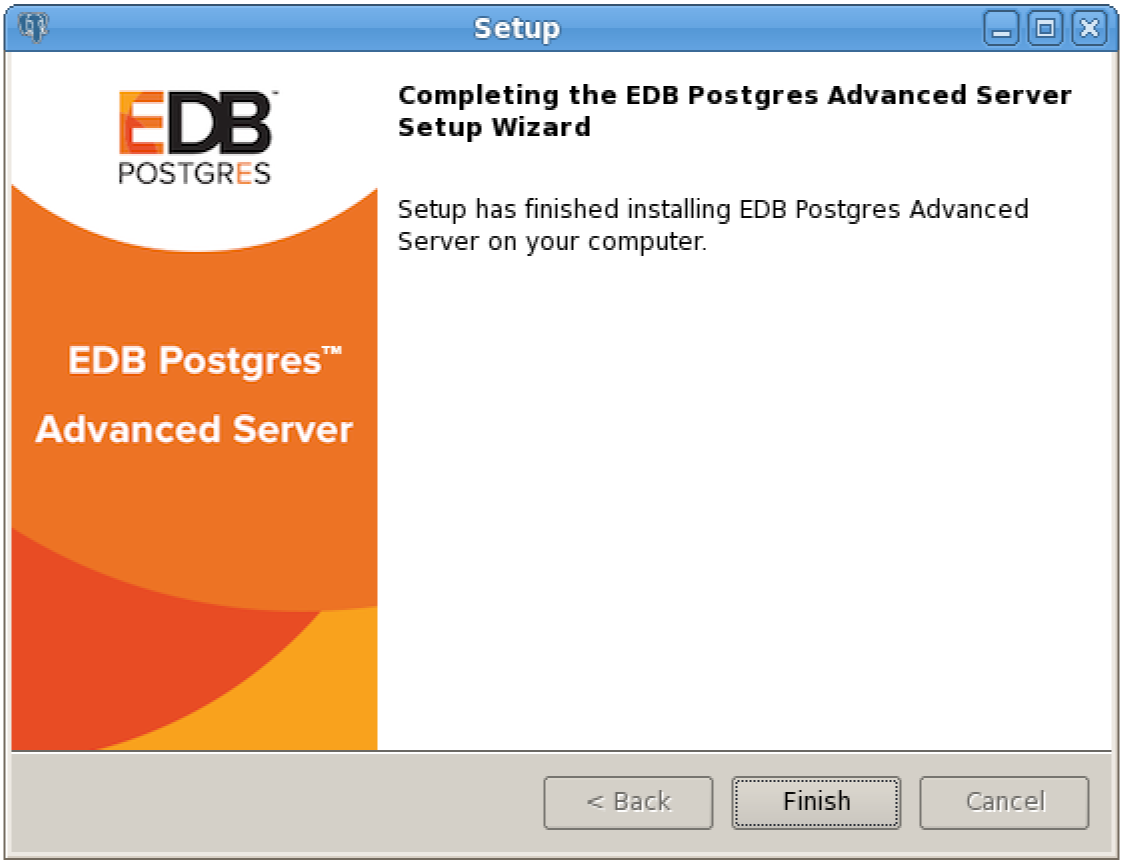


Ready to Install

Setup is now ready to begin installing EDB Postgres Advanced Server on your computer.

Figure 4.39 -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 4.40).



Completing the EDB Postgres Advanced Server Setup Wizard

Setup has finished installing EDB Postgres Advanced Server on your computer.

Figure 4.40 -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 the following directories:

On Linux:

```
/opt/edb
```

On Windows:

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

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

On Linux:

```
./runAsRoot.sh
```

On Windows:

```
     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 on a Linux system. The actual installation dialog will vary by platform and reflect the options specified during the installation.

```
======================
INSTALLATION DIRECTORY
======================
Please enter the installation directory [ /opt/edb ] :
```

The installation directory is the directory where Advanced Server is installed.

```
==============
DATA DIRECTORY
==============
NOTE: If data directory exists and postgresql.conf file exists in
that directory, we will not initialize the cluster.

Please enter the data directory path: [ /opt/edb/as10/data ] :
```

The data directory is the directory where Advanced Server data is stored.

```
=============
WAL DIRECTORY
=============
Please enter the Write-Ahead Log (WAL) directory path:
     [ /opt/edb/as10/data/pg_wal ] :
```

The WAL directory is where the write-ahead log will be written.

```
=============
DATABASE MODE
=============
Please enter database mode: [ oracle ] :
```

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

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 `postgresql` to install Advanced Server with complete compatibility with Postgres version 10.

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`.

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

Please enter port: [ 5444 ] :
```

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

```
======
LOCALE
======
Please enter the locale: [ DEFAULT ] :
```

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

```
=============
SAMPLE TABLES
=============
Install sample tables and procedures? (Y/n): [ Y ] :
```

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.

```
===========================
DATABASE SUPERUSER PASSWORD
===========================
Please provide password for the super-user(enterprisedb): [ ] :
Please re-type password for the super-user(enterprisedb): [ ] :
```

Specify and confirm a password for the database superuser. By default, the database superuser is named `enterprisedb`. (On Windows, there is no password validation if you are logged in as an administrator, but you may be prompted to supply a service account password.)

```
==================
SERVER UTILIZATION
==================
Please enter the server utilization: [ 66 ] :
```

Specify a value between `1` and `100`.

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.

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.

```
================
WORKLOAD PROFILE
================
Please enter the workload profile: [ oltp ] :
```

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.

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.

Before continuing with the installation, the installer displays the selected options and initializes the database cluster in preparation for the installation of individual components. When the installer has prepared the system for the installation, the installation begins. Before installing a component, the installer prompts you to select modules for installation. With each component, onscreen warnings may alert you to unresolved dependencies.

Please note that the available components are different for each platform, and the prompts that follow may vary.

The installer saves the specified configuration values in the following file:

```
'/opt/edb/.rar_options_xxxxx'
```

After continued processing, the Advanced Server installation is complete.

4.4.4 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 10.

`pgadmin4` (Linux and Windows only.)

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. This option cannot be specified when installing on Linux. The `--enable_acledit 1` option particularly should be specified when a *`discretionary access control list`* (DACL) needs to be set for allowing access to objects on a Windows host 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

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 {qt | gtk | xwindow | text | unattended}`

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

`qt` - Specify `qt` to tell the installer to use the Qt graphical toolkit

`gtk` - Specify `gtk` to tell the installer to use the GTK graphical toolkit.

`xwindow` - Specify `xwindow` to tell the installer to use the X Window graphical toolkit.

`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`*`/as9.`*`x`*

Use the `--prefix` parameter to specify an installation directory for Advanced Server. The installer will append a version-specific sub-directory (i.e. `as10`) to the specified directory. By default, on a Linux system, Advanced Server is installed in:

```
/opt/edb/as10
```

The default installation directory on a Windows system is:

```
C:\Program Files\edb\as10
```

`--productkey` *`product_key`*

Use the `--productkey` parameter to specify a value for the product key.

The `--productkey` parameter is only required when the specified system locale is Japanese, Chinese or Korean.

`--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:** For installing on Windows hosts, see the `--enable_acledit` option in this section for additional information relevant to a Windows environment. **Note:** Specification of the `--enable_acledit` option is permitted only when installing on Windows. The `--enable_acledit` option cannot be specified when installing on Linux.

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-10`.

`--servicepassword` *`user_password`* (Windows only)

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 10 --- Built on 2017-06-15
00:04:00 IB: 15.10.1-201511121057
```

`--webusername` *`{registered_username}`*

You must specify the name of a registered user and password when performing an installation of EDB Postgres Advanced Server 10. Use the `--webusername` parameter to specify the name of the registered EnterpriseDB user that is performing the installation.

`registered_username` must be an email address.

If you do not have a *`registered user name`*, visit the EnterpriseDB website at:

http://www.enterprisedb.com/user-login-registration

`--webpassword` *`{associated_password}`*

Use the `--webpassword` parameter to specify the password associated with the registered EnterpriseDB user that is performing the installation.

`--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

`--xlogdir` *`directory_name`* (Linux only.)

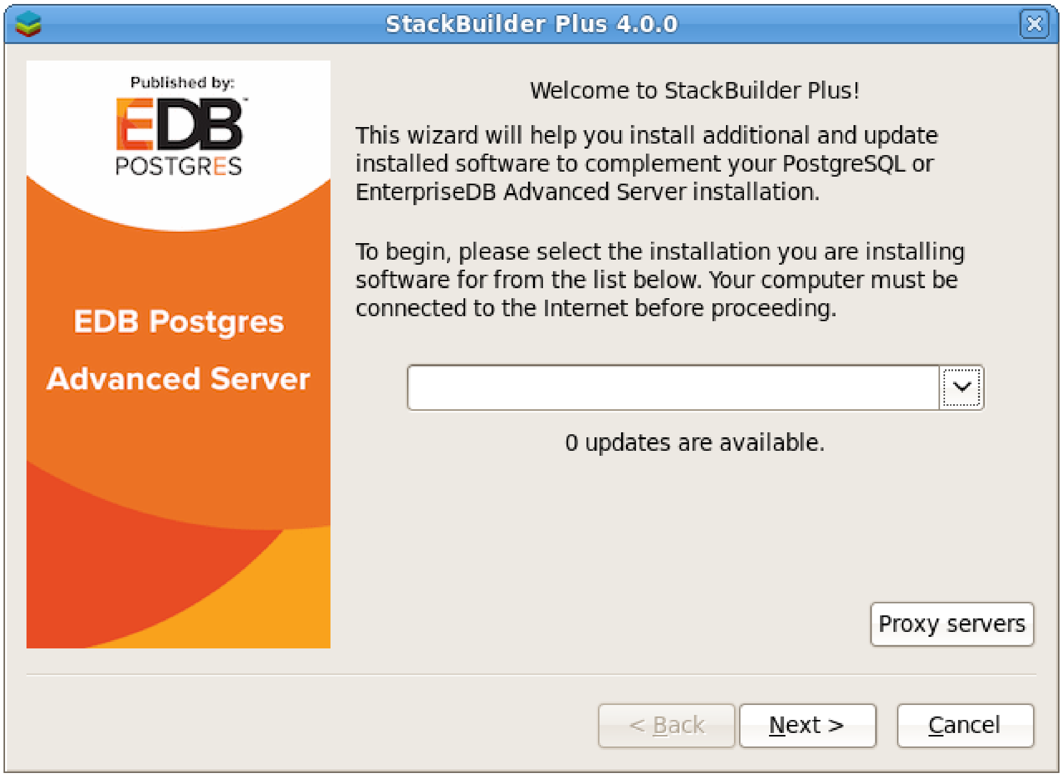
Use the `--xlogdir` parameter to specify a location for the write-ahead log. The default value is `/opt/edb/as10/data/pg_wal`.

4.5 Using StackBuilder Plus

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.

Please note: If your installation resides on a Linux system, you must install the `redhat-lsb` package before invoking StackBuilder Plus. For more information, see Section 4.1.

You can invoke StackBuilder Plus at any time after the installation has completed by selecting the `StackBuilder Plus` menu option from the `EDB Postgres` → `Advanced Server 10` menu (Linux) or from the `Apps` menu (Windows). Enter your system password (if prompted), and the StackBuilder Plus welcome window opens (shown in Figure 4.41).



Welcome to StackBuilder Plus!

This wizard will help you install additional and update installed software to complement your PostgreSQL or EnterpriseDB Advanced Server installation.

To begin, please select the installation you are installing software for from the list below. Your computer must be connected to the Internet before proceeding.

0 updates are available.

Figure 4.41 -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 4.42).

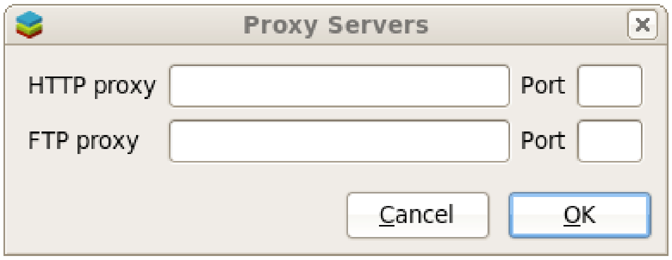
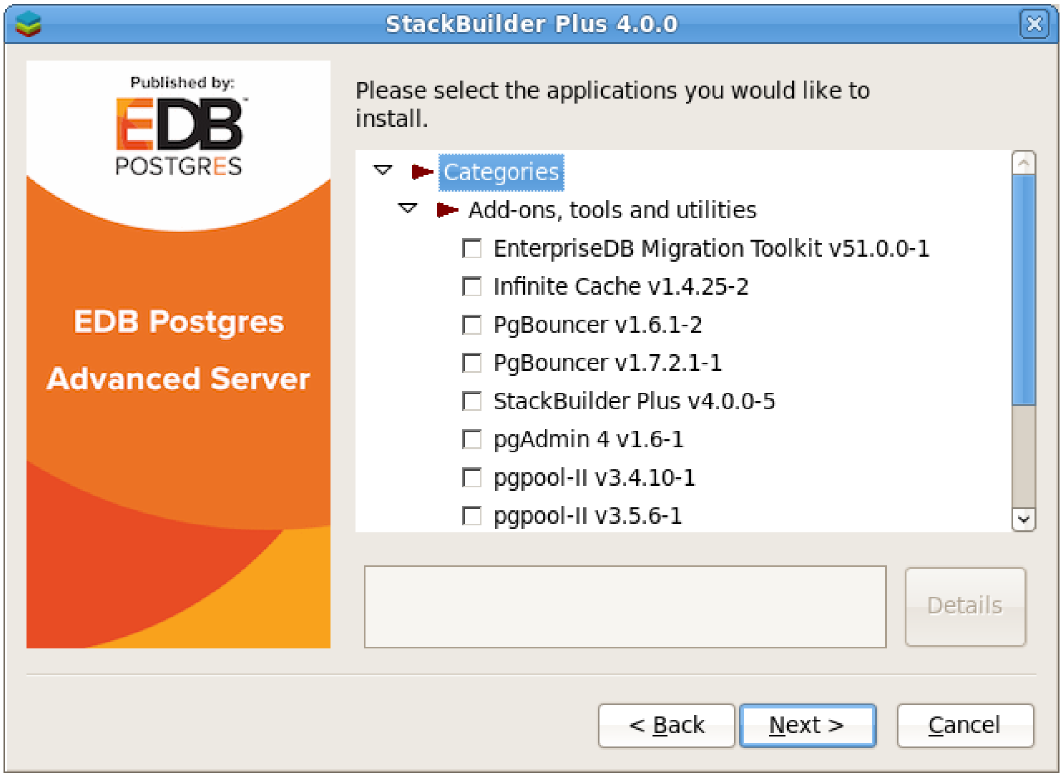


Figure 4.42 –The Proxy servers dialog.

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.



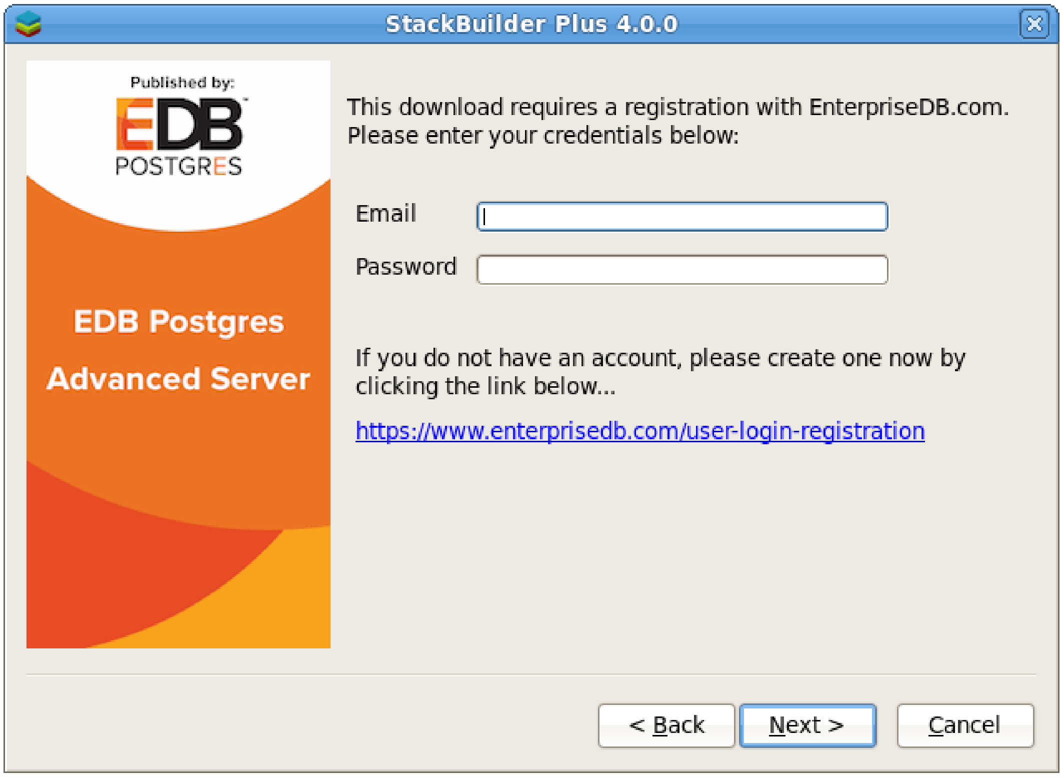
Please select the applications you would like to install.

- Categories
  - Add-ons, tools and utilities
    - □ EnterpriseDB Migration Toolkit v51.0.0-1
    - □ Infinite Cache v1.4.25-2
    - □ PgBouncer v1.6.1-2
    - □ PgBouncer v1.7.2.1-1
    - □ StackBuilder Plus v4.0.0-5
    - □ pgAdmin 4 v1.6-1
    - □ pgpool-II v3.4.10-1
    - □ pgpool-II v3.5.6-1

Figure 4.43 –The StackBuilder Plus module selection window.

The tree control on the StackBuilder Plus module selection window (shown in Figure 4.43) displays a node for each module category.

To add a new component to the selected Advanced Server installation or to upgrade a component, check the box to the left of the module name and click `Next`. A window opens, requesting your EnterpriseDB registration information (as shown in Figure 4.44).



This download requires a registration with EnterpriseDB.com.
Please enter your credentials below:

Email

Password

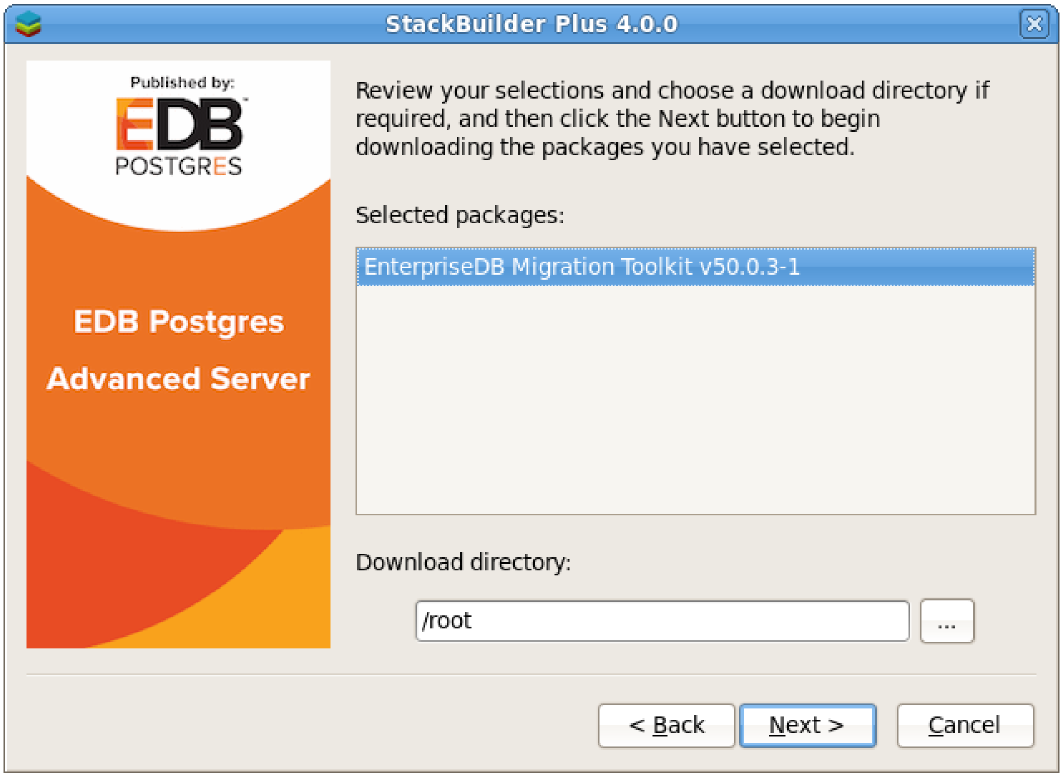
If you do not have an account, please create one now by clicking the link below...

https://www.enterprisedb.com/user-login-registration

Figure 4.44 –The User Authenication window.

Before downloading and installing modules and updates with StackBuilder Plus, you must enter the user information associated with your EnterpriseDB account. If you do not have an EnterpriseDB user account, click the link provided to open a web browser, and enter your user information.

Enter the email address of a registered account in the `Email` field, and the corresponding password in the `Password` field, and click `Next` to continue. The next dialog confirms the packages selected (Figure 4.45).



Review your selections and choose a download directory if required, and then click the Next button to begin downloading the packages you have selected.

Selected packages:

EnterpriseDB Migration Toolkit v50.0.3-1

Download directory:

/root

Figure 4.45 -A summary window displays a list of selected packages.

By default, the selected package installers are downloaded to:

On Windows:

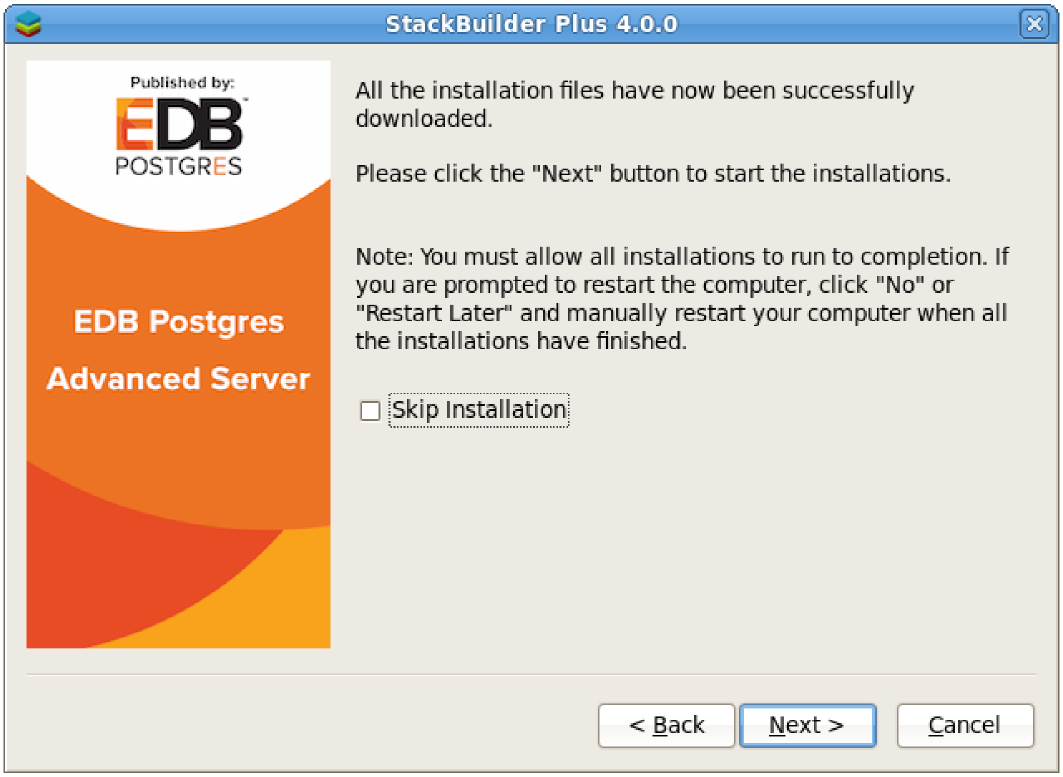
```
C:\Users\Administrator
```

On Linux:

```
/root
```

You can change the directory; use the 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 4.46).



All the installation files have now been successfully downloaded.

Please click the "Next" button to start the installations.

Note: You must allow all installations to run to completion. If you are prompted to restart the computer, click "No" or "Restart Later" and manually restart your computer when all the installations have finished.

□ Skip Installation

Figure 4.46 -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.

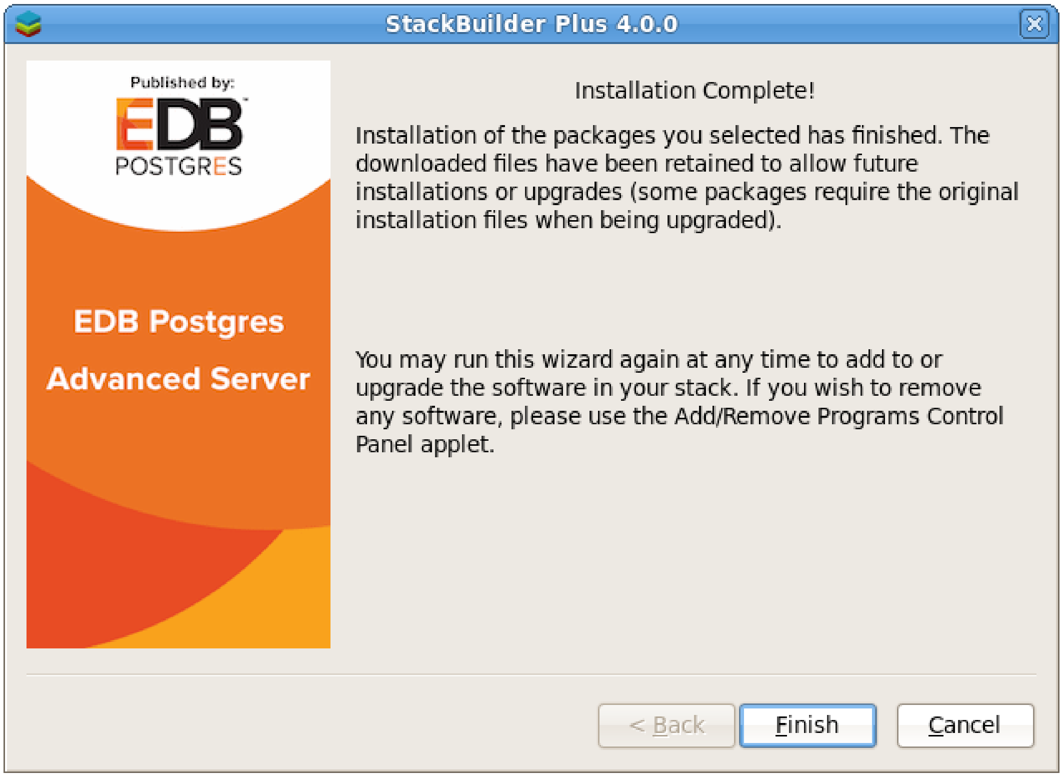
Each downloaded installer has different requirements. As the installers execute, they may prompt you to confirm acceptance of license agreements, to enter passwords, and enter configuration information.

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:

On Windows: `%TEMP%`

On Linux: `/root`



Installation Complete!

Installation of the packages you selected has finished. The downloaded files have been retained to allow future installations or upgrades (some packages require the original installation files when being upgraded).

You may run this wizard again at any time to add to or upgrade the software in your stack. If you wish to remove any software, please use the Add/Remove Programs Control Panel applet.

Figure 4.47 -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 4.47). If you were prompted by an installer to restart your computer, reboot now.

The following table lists some of the modules supported by StackBuilder Plus. Please note that the list is subject to change and varies by platform.

Category and Module Name	**Description**
Add-ons, tools and utilities	
EnterpriseDB Migration Toolkit	Migration Toolkit is a command line tool that facilitates migration from Oracle databases into Advanced Server
Infinite Cache	Infinite Cache (for Linux only) allows you to utilize memory on other computers connected to your network to increase the amount of memory in the shared buffer cache.
PgBouncer	Connection pooler for Postgres Server, packaged by EnterpriseDB.
StackBuilder Plus	An advanced application stack builder that provides an easy interface for downloading and installing Advanced Server updates and modules.
pgAdmin 4	A full-featured graphical client that can manage multiple databases.
pgAgent	pgAgent is a job scheduling agent for Postgres, capable of running multi-step batch/shell and SQL tasks on complex schedules
pgPool-II	pgPool-II provides load balancing, connection pooling, high availability, and connection limits for Advanced Server databases.
Database Drivers	
EnterpriseDB Connectors	A collection of drivers. Includes .NET, ODBC, JDBC and libpq drivers for

Category and Module Name	Description
	Advanced Server
Database Server	
Advanced Server	The EDB Postgres Advanced Server database server.
EnterpriseDB Tools	
Postgres Enterprise Manager Agent	The PEM Agent is responsible for executing tasks and reporting statistics from the host and monitored Postgres instances to the PEM Server.
Postgres Enterprise Manager Client	The PEM Client is a full-featured graphical interface that allows you to schedule tasks and report statistics for the host and monitored instances.
Postgres Enterprise Manager Server	The PEM Server is used as the data repository for monitoring data and as a server to which the agents and client connect.
Replication Server	Replication Server is an asynchronous, primary-to-standby replication system enabling replication of tables from an Oracle or SQL Server database to an Advanced Server database.
Replication Solutions	
Slony Replication	Slony is a primary to multiple standbys replication system that supports cascading and failover. Packaged by EnterpriseDB.
Spatial Extensions	
PostGIS	PostGIS enables Advanced Server to store spatial data for geographic information systems (GIS).
Web Development	
ApachePHP	A distribution of the Apache webserver and PHP, preconfigured for use with Advanced Server. Packaged by EnterpriseDB.
PEM-HTTPD	A pre-configured Apache webserver for use with PostgreSQL. Packaged by EnterpriseDB.

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 Linux 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.

Applications Fail to Launch During an Advanced Server Installation on Linux

If applications fail to launch (such as StackBuilder Plus or your web browser) during the installation process on a Linux system, verify that the xdg-open program is on your system. If `xdg-open` is missing, install the `xdg-utils` package.

If you are using the GNOME desktop, load the root profile before running the Advanced Server installation script. To load the root profile, issue the command, `su - root` instead of `su root` before installing Advanced Server.

Configuration File Editors Close Spontaneously

If you are using a Linux system with the gnome console, a bug in the gnome shell may cause configuration file editors accessed via the Expert Configuration menu (under the Advanced Server Application menu) to close spontaneously. To correct this error, open a terminal window and enter:

```
dconf write /org/gnome/settings-daemon/plugins/cursor/active
false
```

Please note that each time you reboot your system, you must invoke the command, resetting the value.

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.

Difficulty Installing the EPEL Release Package

If you encounter difficulty when installing the EPEL release package, you can use the following command to install the epel-release package on RHEL or CentOS 7 platform:

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

Please note that you may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).

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 –y install epel-release
```

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-10	edb-as-10
Infinite Cache	edb-icache	N/A
pgAgent	edb-pgagent-10	EDB Postgres Advanced Server 10 Scheduling Agent
PgBouncer	edb-pgbouncer-1.7	edb-pgbouncer-1.7
pgPool-II	edb-pgpool-3.5	N/A
Slony	edb-slony-replication-10	edb-slony-replication-10

Advanced Server's database server, and the services of Advanced Server's supporting components can be controlled at the command line or through operating system-specific graphical interfaces.

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 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/10/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.

Note that when you invoke the installer with the `--extract-only` option, the installer does not create a service, it merely unpacks the server. If you have installed Advanced

Server by invoking the installer with the `--extract-only` option, you must use the `pg_ctl` command to control the server.

5.2.3 Using the edbstart and edbstop Utilities

Note: edbstart and edbstop functionality is supported only on Linux hosts that are running Advanced Server installations performed with the Interactive installer. RPM installations do not support edbstart and edbstop.

While the autostart scripts created during an Advanced Server installation control a single database cluster, the edbstart and edbstop utilities can control multiple database clusters on the same host, with a single configuration file.

The edbstart and edbstop utilities use a file named `edbtab` (described below) to determine which instances of Advanced Server should start when the operating system boots, and stop when the host is shut down.

Before using the edbstart or edbstop utilities, you should disable the Advanced Server autostart scripts. The commands that disable the scripts are platform specific;

on Fedora/Redhat:

```
chkconfig --level 2345 edb-as-10 off
```

on Debian/Ubuntu:

```
update-rc.d edb-as-10 disable
```

After stopping the Advanced Server service, use an editor to create a file named `edbtab` in the `/etc` directory; you can copy the sample file located in:

```
/opt/edb/as10/scripts/server/autostart
```

Edit the `edbtab` file, specifying which Advanced Server clusters that the edbstart and edbstop programs will control, and indicating if the cluster should be automatically started and stopped.

Each `edbtab` file entry should take the form:

```
edb_home directory:edb_data_directory:N|Y
```

`edb_home_directory`

`edb_home_directory` specifies the home directory of the Advanced Server installation that the edbstart/edbstop utilities will control.

`edb_data_directory`

> *edb_data_directory* specifies the data directory of the database cluster that the edbstart/edbstop utilities will control. *edb_data_directory* is the same as the value of $PGDATA for a specified cluster.

N|Y

> Y specifies that edbstart and edbstop should control the service; N specifies that the user will control the service manually.

Include a separate entry in the edbtab file for each Advanced Server cluster that you wish to control with the edbstart and edbstop utilities.

After editing the edbtab file, copy the edb_autostart script to /etc/init.d. By default, the edb_autostart script is located in:

```
/opt/edb/as10/scripts/server/autostart
```

Copy the edbstart and edbstop scripts to $EDBHOME. Make the scripts executable with the following command:

```
chmod +x edbstart
chmod +x edbstop
chmod +x edbstart edbstop /etc/init.d/edb_autostart
```

Enable the edb_autostart service with the commands:

```
chkconfig --level 2345 edb_autostart on
chkconfig --add edb_autostart
```

For the service to take effect, you must restart your system.

Manually Controlling the Server with edbstart and edbstop

You can use edbstart and edbstop at the command line to manually control all of the clusters specified in the edbtab file, or to control an individual cluster. Call edbstart without an argument to start all of the clusters listed within the edbtab file; invoke edbstop without an argument to stop all of the clusters listed in the edbtab file. You can control an individual cluster by specifying the cluster's data directory as an argument. The following command starts a cluster:

```
edbstart /opt/edb/as10/data
```

While the following command stops a cluster:

```
edbstop /opt/edb/as10/data
```

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
```

On a Debian-compatible Linux system, use the command:

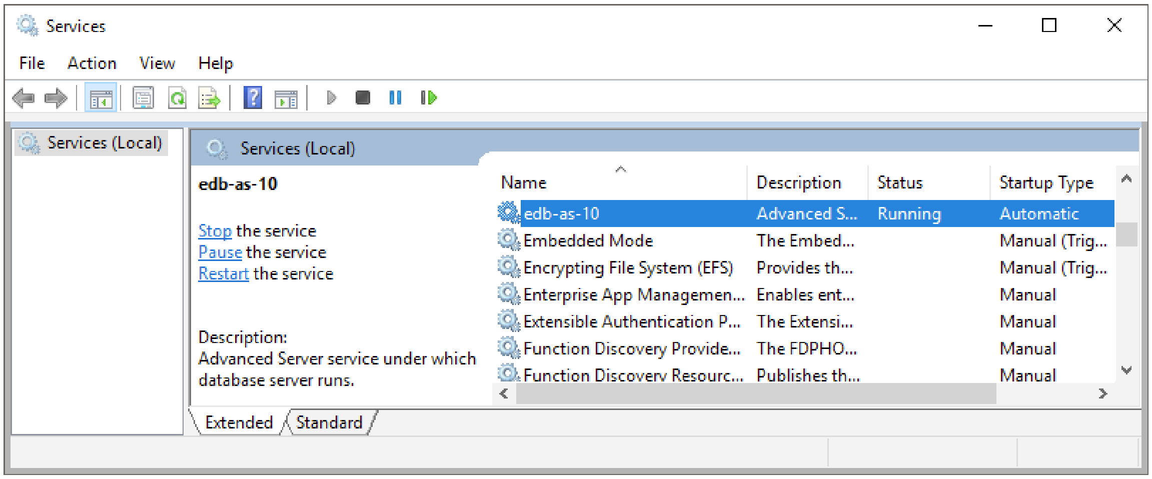
```
/usr/sbin/update-rc.d service_name enable
```

Where *`service_name`* specifies the name of the service.

Please note that if you are using a Windows system, the Slony service will be configured to autostart by default. On Windows, you can use the `Service Properties` dialog to control the service startup type. For more information about controlling a service on Windows, see Section 5.3.

5.3 Controlling a Service on Windows

The Windows operating system includes a graphical service controller that offers graphical control of Advanced Server and the services associated with Advanced Server components. The Windows Services utility can be accessed through the `Administrative Tools` section of the `Control Panel`, or by navigating through the `Apps` menu to `Run`; when the `Run` dialog opens, enter `services.msc` and click `OK`.



edb-as-10

Stop the service
Pause the service
Restart the service

Description:
Advanced Server service under which database server runs.

Name	Description	Status	Startup Type
edb-as-10	Advanced S...	Running	Automatic
Embedded Mode	The Embed...		Manual (Trig...
Encrypting File System (EFS)	Provides th...		Manual (Trig...
Enterprise App Managemen...	Enables ent...		Manual
Extensible Authentication P...	The Extensi...		Manual
Function Discovery Provide...	The FDPHO...		Manual
Function Discovery Resourc...	Publishes th...		Manual

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

When the `Services` window opens, use the scroll bar to move through the listed services to highlight `edb-as-10` (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.

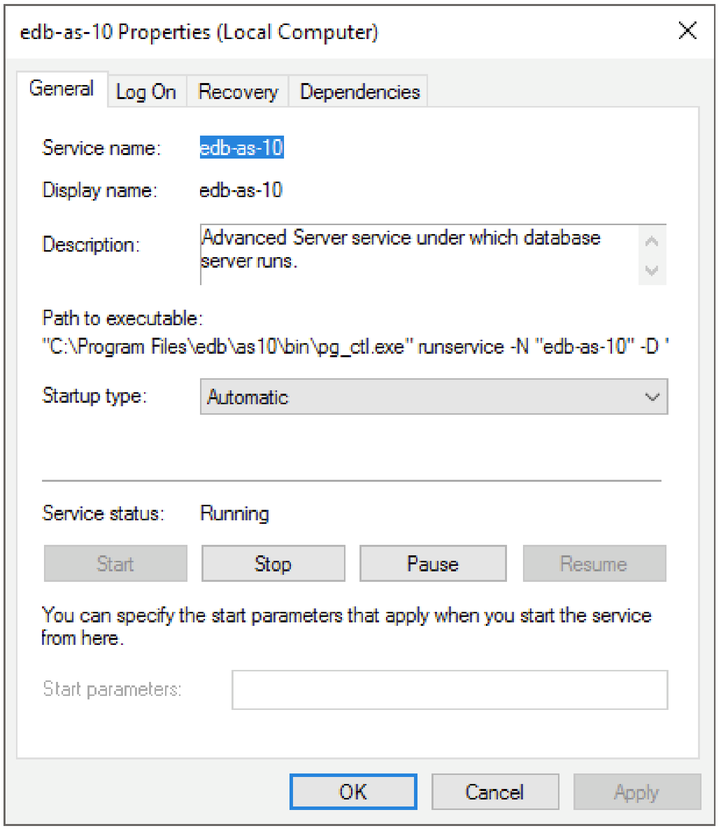
  Please Note: A limitation in Windows may cause Advanced Server to generate an error message after performing a parameter reload. To confirm that the reload command has successfully updated the parameters, query the `pg_settings` table to verify that the change has taken effect.
- Use the `Restart the 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. To alter the startup properties of a server, navigate through the `Control Panel` to the `Services` window, or navigate through the `Apps` menu to `Run`; when the `Run` dialog opens, enter `services.msc` and click `OK`.

Right click on the name of the service you wish to change 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.



edb-as-10 Properties (Local Computer)

Service name: edb-as-10

Display name: edb-as-10

Description: Advanced Server service under which database server runs.

Path to executable:
"C:\Program Files\edb\as10\bin\pg_ctl.exe" runservice -N "edb-as-10" -D '

Startup type: Automatic

Service status: Running

You can specify the start parameters that apply when you start the service from here.

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`.

6 Configuring Advanced Server

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

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 navigate through a menu to open the file:

- On a Windows system, a link to each configuration file is available on the `Apps` menu.
- To update configuration files in Linux, navigate through the `EDB Postgres` menu selection on the `Applications` menu to the `Advanced Server 10` menu; use the `Expert Configuration` menu to select the configuration file that you would like to edit (see Figure 6.1).

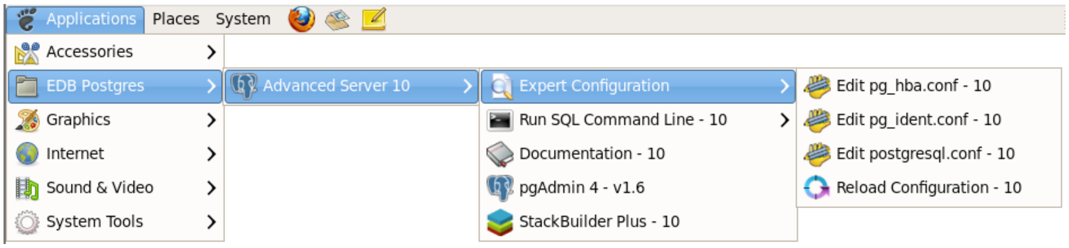
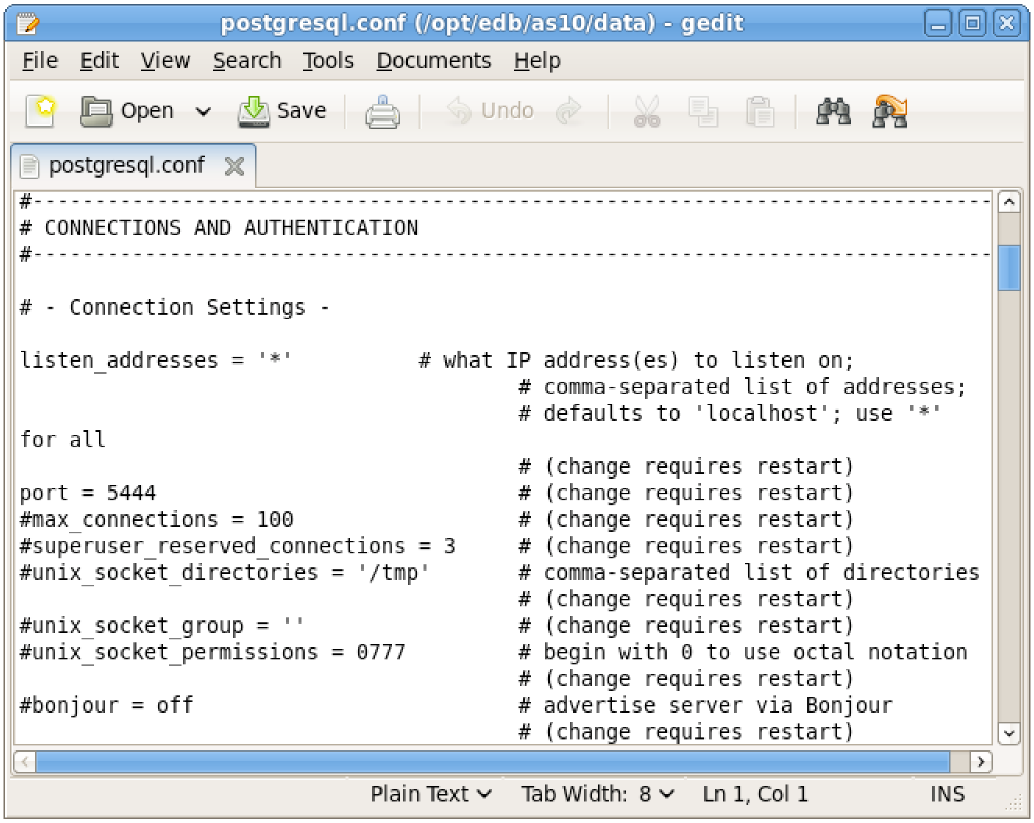


Figure 6.1 -Accessing the configuration files through the Applications menu.

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 (/opt/edb/as10/data) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
postgresql.conf
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

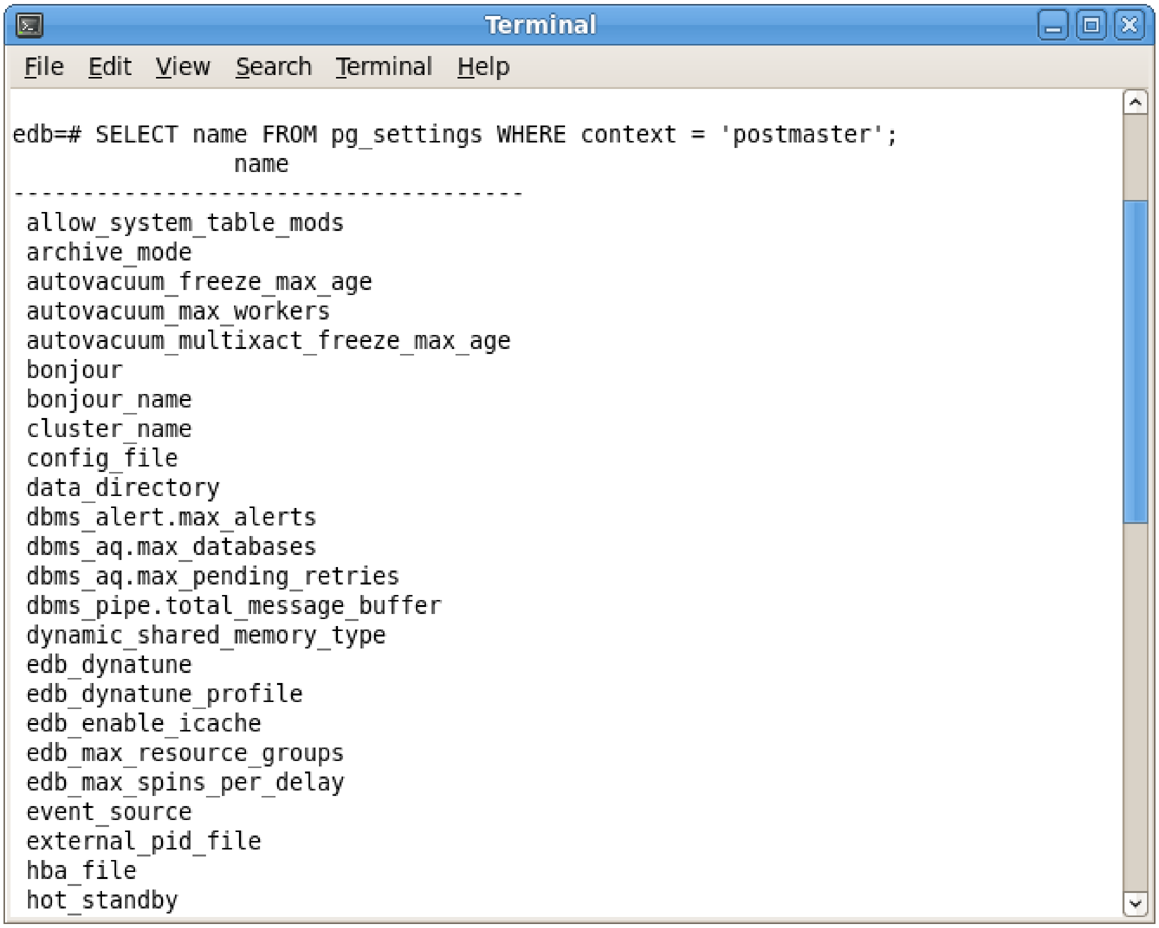
```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*'
for all
                                        # (change requires restart)
port = 5444                             # (change requires restart)
#max_connections = 100                  # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/tmp'       # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
```

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 EDB-PSQL command line (see Figure 6.3):

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



```
edb=# SELECT name FROM pg_settings WHERE context = 'postmaster';
                name
------------------------------------
 allow_system_table_mods
 archive_mode
 autovacuum_freeze_max_age
 autovacuum_max_workers
 autovacuum_multixact_freeze_max_age
 bonjour
 bonjour_name
 cluster_name
 config_file
 data_directory
 dbms_alert.max_alerts
 dbms_aq.max_databases
 dbms_aq.max_pending_retries
 dbms_pipe.total_message_buffer
 dynamic_shared_memory_type
 edb_dynatune
 edb_dynatune_profile
 edb_enable_icache
 edb_max_resource_groups
 edb_max_spins_per_delay
 event_source
 external_pid_file
 hba_file
 hot_standby
```

Figure 6.3 - Configuration parameters that require a server restart.

If you are changing a parameter that requires a server restart, see Section 5.1, *Starting and Stopping Advanced Server* for information about restarting Advanced Server.

On a Linux system, you can reload the system configuration parameter values by navigating through the `EDB Postgres` menu to the `Advanced Server 10` menu; then, navigate through the `Expert Configuration` menu, selecting `Reload Configuration`. Reloading the configuration parameters does not require Advanced Server users to log out of their current Advanced Server sessions.

On a Windows system, you will find the `Reload Configuration` menu selection on the `Apps` menu.

6.2 Modifying the pg_hba.conf File

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     enterprisedb                                 md5
#host    replication     enterprisedb        127.0.0.1/32            md5
#host    replication     enterprisedb        ::1/128                 md5
```

Appropriate authetication methods provide protection and security. Please consult the PostgreSQL documentation for details about authentication options:

https://www.postgresql.org/docs/10/static/auth-methods.html

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 services utility (Windows), or use the following command to restart the server and apply the changes:

On Linux 7.x:

```
systemctl restart edb-as-10
```

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

https://www.postgresql.org/docs/10/static/auth-pg-hba-conf.html

6.3 Setting Advanced Server Environment Variables

The graphical installers provide a script that simplifies the task of setting environment variables, allowing a user to more easily invoke client applications at the command line. 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 or pg_env 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:

On Linux:

```
source /opt/edb/as10/pgplus_env.sh
```

On Windows:

```
C:\Program Files\edb\10AS\pgplus_env.bat
```

As the pgplus_env.sh script executes (on Linux), it sets the following environment variables:

```
export PATH=/opt/edb/as10/bin:$PATH
export EDBHOME=/opt/edb/as10
export PGDATA=/opt/edb/as10/data
export PGDATABASE=edb
# export PGUSER=enterprisedb
export PGPORT=5444
export PGLOCALEDIR=/opt/edb/as10/share/locale
```

As the pgplus_env.bat script executes (on Windows), it sets the following environment variables:

```
PATH="C:\Program Files\edb\as10\bin";%PATH%
EDBHOME=C:\Program Files\edb\as10
PGDATA=C:\Program Files\edb\as10\data
PGDATABASE=edb
REM @SET PGUSER=enterprisedb
PGPORT=5444
PGLOCALEDIR=C:\Program Files\edb\as10\share\locale
```

If you have used an installer created by EnterpriseDB to install PostgreSQL, the pg_env script performs the same function. To invoke the pg_env script, open a command line, and enter:

On Linux:

```
source /opt/PostgreSQL/10/pg_env.sh
```

On Windows:

```
C:\Progra~1\PostgreSQL\10\pg_env.bat
```

As the `pg_env.sh` script executes (on Linux), it sets the following environment variables:

```
PATH=/home/opt/PostgreSQL/10/bin:$PATH
PGDATA=/home/opt/PostgreSQL/10/data
PGDATABASE=postgres
PGUSER=postgres
PGPORT=5432
PGLOCALEDIR=/home/opt/PostgreSQL/10/share/locale
MANPATH=$MANPATH:/home/opt/PostgreSQL/10/share/man
```

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

```
PATH="C:\Program Files\PostgreSQL\10\bin";%PATH%
PGDATA=C:\Program Files\PostgreSQL\10\data
PGDATABASE=postgres
PGUSER=postgres
PGPORT=5432
PGLOCALEDIR=C:\Program Files\PostgreSQL\10\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 edb-psql client, the client must be in your search path. The executable resides in the `bin` directory, under your Advanced Server installation.

On Linux:

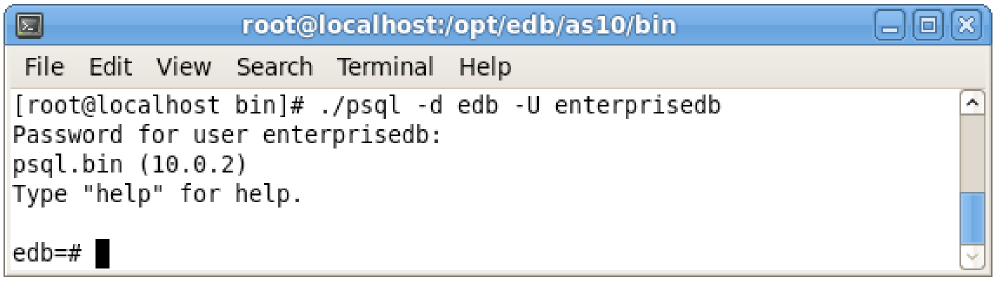
```
/opt/edb/as10/bin/psql
```

On Windows:

```
C:\Program Files\edb\as10\bin\psql
```

Use the following command and command options to start the psql client:

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



root@localhost:/opt/edb/as10/bin
File Edit View Search Terminal Help

```
[root@localhost bin]# ./psql -d edb -U enterprisedb
Password for user enterprisedb:
psql.bin (10.0.2)
Type "help" for help.

edb=# 
```

Figure 6.4 - 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 through the `Applications` or `Start` menu. Navigate through the `EDB Postgres` menu to the `Advanced Server 10` menu; then, navigate through the `Run SQL Command Line` menu, selecting `EDB-PSQL`. When the `Terminal` window opens, provide connection information for your session.

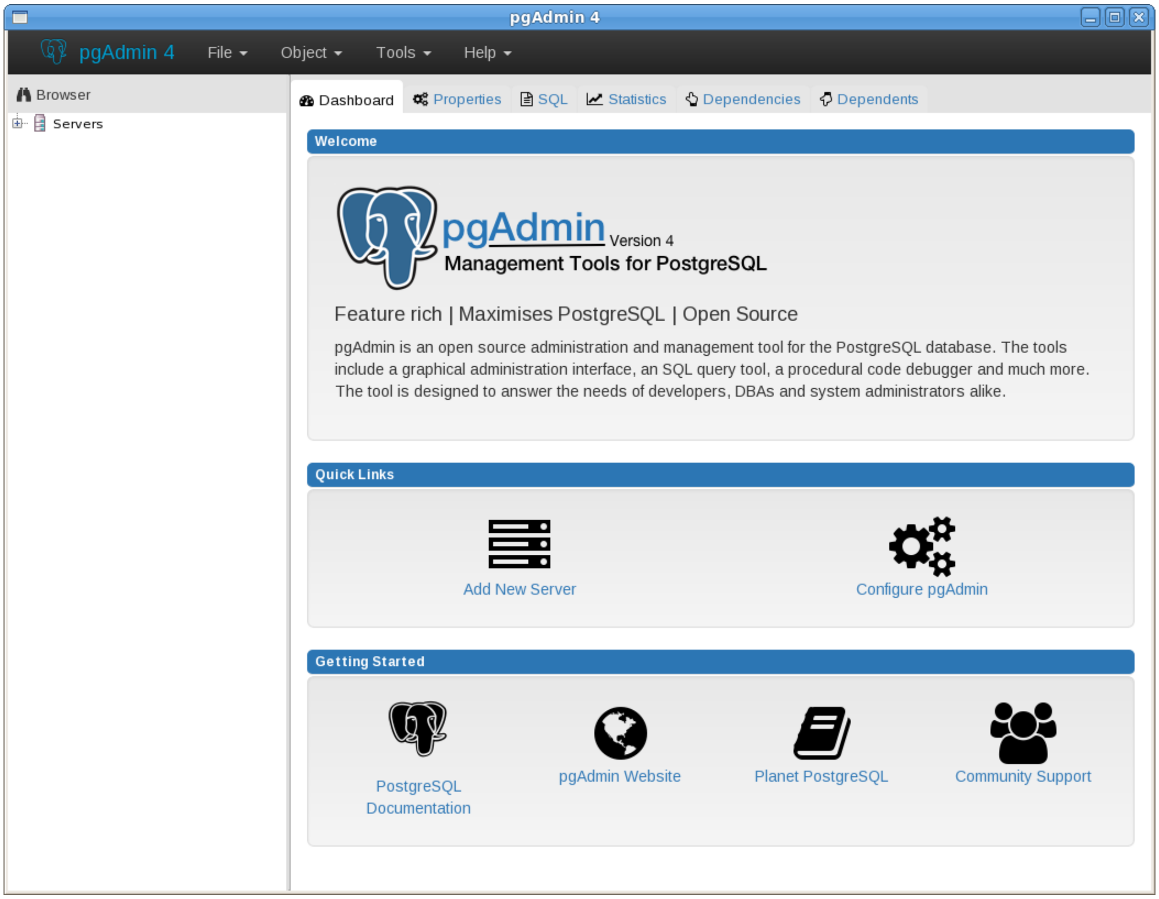
For more information about using the command line client, please refer to the PostgreSQL Core Documentation at:

https://www.postgresql.org/docs/10/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.

To open pgAdmin, use the Linux `Applications` or Windows `Start` menu to access the `EDB Postgres` menu; navigate through the `Advanced Server 10` menu to select `pgAdmin`. The client opens as shown in Figure 6.5.

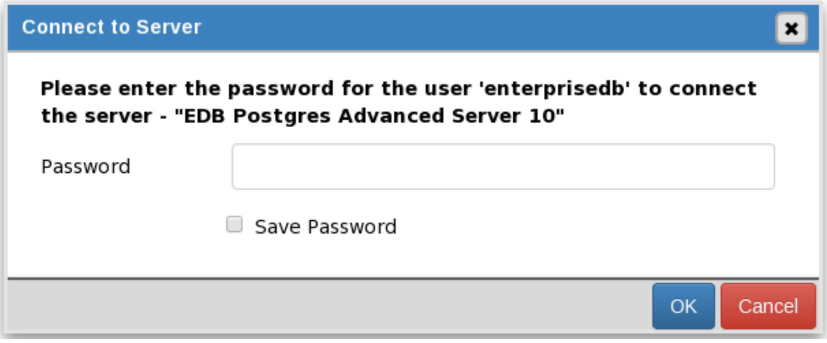


Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an open source administration and management tool for the PostgreSQL database. The tools include a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Figure 6.5 – The pgAdmin 4 client.

To connect to the Advanced Server database server, expand the server 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 6.6).



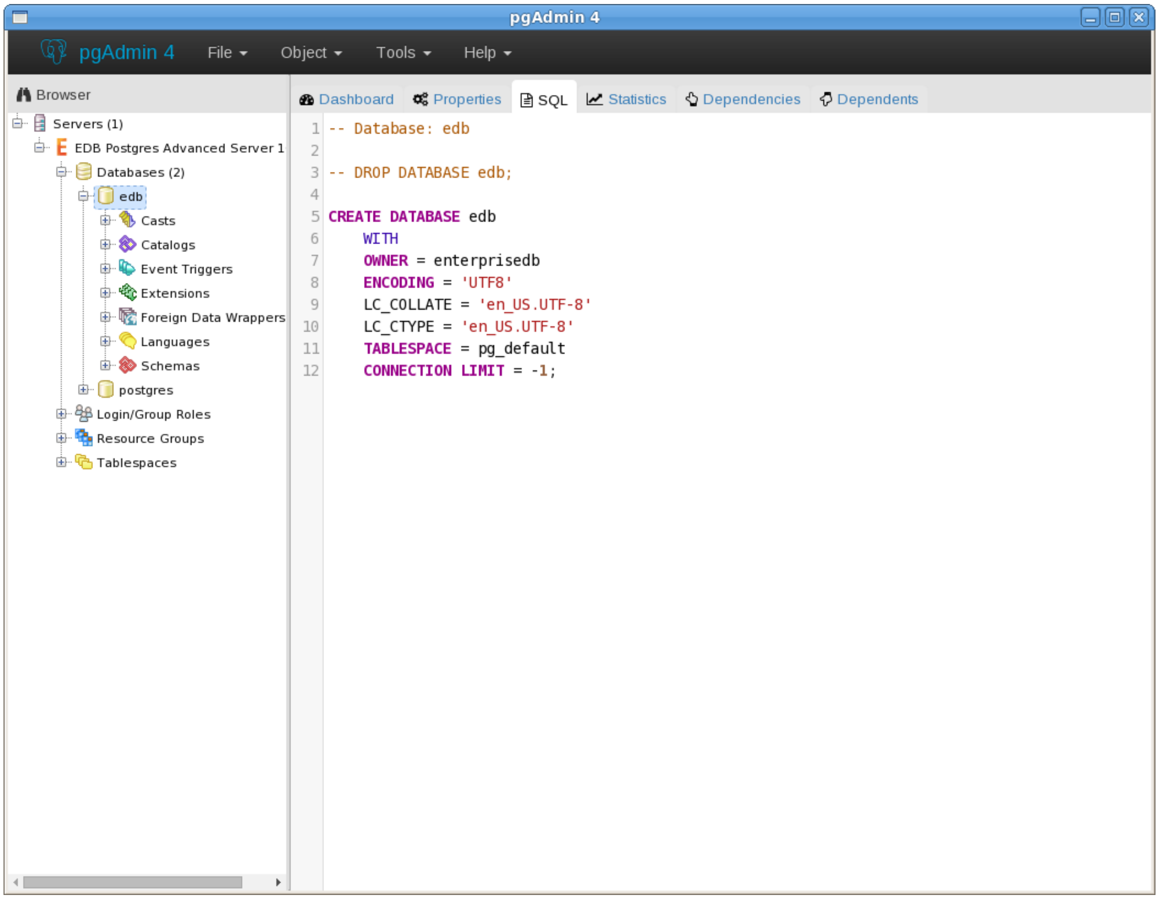
Connect to Server

Please enter the password for the user 'enterprisedb' to connect the server - "EDB Postgres Advanced Server 10"

Password

Figure 6.6 – The pgAdmin 4 client.

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



```
-- Database: edb

-- DROP DATABASE edb;

CREATE DATABASE edb
    WITH
    OWNER = enterprisedb
    ENCODING = 'UTF8'
    LC_COLLATE = 'en_US.UTF-8'
    LC_CTYPE = 'en_US.UTF-8'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

Figure 6.7 – The pgAdmin client.

When the client connects (see Figure 6.7), 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 Limitations

- The pg_upgrade utility cannot upgrade a partitioned table if a foreign key refers to the partitioned table.
- If you are upgrading from the version 9.4 server or a lower version of Advanced Server, and you use partitioned tables that include a `SUBPARTITION BY` clause, you must use `pg_dump` and `pg_restore` to upgrade an existing Advanced Server installation to a later version of Advanced Server. To upgrade, you must:
    - Use `pg_dump` to preserve the content of the subpartitioned table.
    - Drop the table from the Advanced Server 9.4 database or a lower version of Advanced Server database.
    - Use `pg_upgrade` to upgrade the rest of the Advanced Server database to a more recent version.
    - Use `pg_restore` to restore the subpartitioned table to the latest upgraded Advanced Server database.

8 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 10, 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. If you have a significant amount of data, that can take a considerable amount of time and planning. You may also have to use additional storage to temporarily accommodate both the original data and the exported 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 10, and is installed as part of the `Database Server` component; no additional installation or configuration steps are required.

8.1 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.

8.1.1 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.

8.2 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 9.6 to Advanced Server 10 takes the form:

```
pg_upgrade --old-datadir path_to_9.6_data_directory --new-
datadir path_to_10_data_directory --user superuser_name --
old-bindir path_to_9.6_bin_directory --new-bindir
path_to_10_bin_directory --old-port 9.6_port --new-port
10_port
```

Where:

`--old-datadir` *`path_to_9.6_data_directory`*

Use the `--old-datadir` option to specify the complete path to the `data` directory within the Advanced Server 9.6 installation.

`--new-datadir` *`path_to_10_data_directory`*

Use the `--new-datadir` option to specify the complete path to the `data` directory within the Advanced Server 10 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_9.6_bin_directory`*

Use the `--old-bindir` option to specify the complete path to the `bin` directory in the Advanced Server 9.6 installation.

`--new-bindir` *`path_to_10_bin_directory`*

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

`--old-port` *`9.6_port`*

Include the `--old-port` option to specify the port on which Advanced Server 9.6 listens for connections.

`--new-port` *`10_port`*

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

8.2.1 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 10 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.

8.3 Upgrading to Advanced Server 10 – 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 10 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 10 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 10, 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 10. 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, invoke the command:

```
service edb-pgagent-10 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.

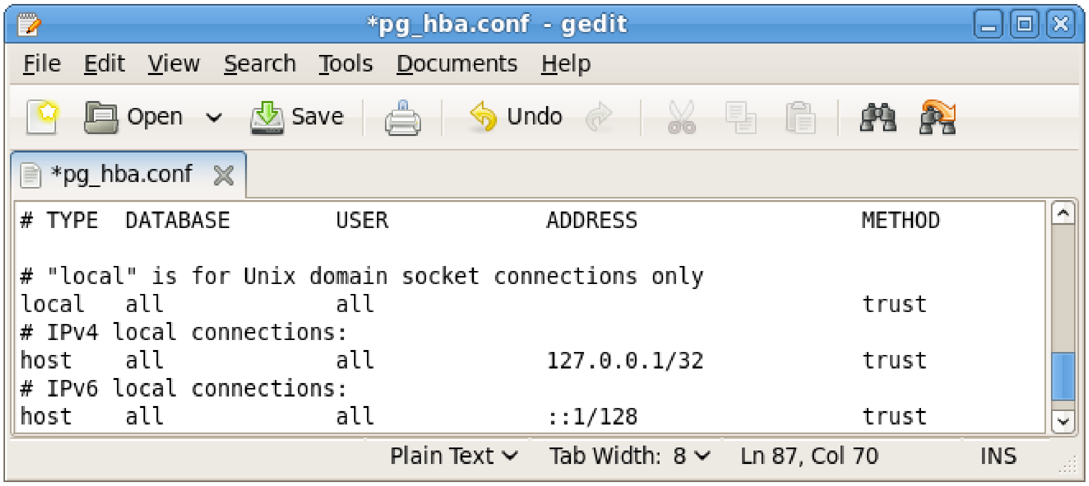
Then, create an empty database based on the contents of the `template1` database (see Figure 8.1):

```
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 10 servers. Edit the `pg_hba.conf` file for both installations of Advanced Server as shown in Figure 7.1.



```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
```

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/10/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 10 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
EnterpriseDB Postgres Advanced Server 9.6	edb-as-9.6	edb-as-9.6
EnterpriseDB Postgres Advanced Server 10	edb-as-10	edb-as-10
Advanced Server 9.6 Scheduling Agent (pgAgent)	edb-pgagent-9.6	EnterpriseDB Postgres Advanced Server 9.6 Scheduling Agent
Infinite Cache 9.6	edb-icache	N/A
Infinite Cache 10	edb-icache	N/A
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	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 9.6	edb-slony-replication-9.6	edb-slony-replication-9.6
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/as10/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\as10\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\as10\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/9.6AS/data -D
/opt/edb/as10/data -U enterprisedb -b
/opt/PostgresPlus/9.6AS/bin -B /opt/edb/as10/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\9.6AS
\data" -D "C:\Program Files\edb\as10\data" -U enterprisedb
-b  "C:\Program Files\PostgresPlus\9.6AS\bin" -B
"C:\Program Files\edb\as10\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 10 service. After restarting the service, open StackBuilder Plus by navigating through the `Start` menu to the `Advanced Server 10` 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/9.6AS/data -D
/opt/edb/as10/data -U enterprisedb -b
/opt/PostgresPlus/9.6AS/bin -B /opt/edb/as10/bin -p 5444 -P
5445
```

On Windows:

```
pg_upgrade.exe -d "C:\Program
Files\PostgresPlus\9.6AS\data"
-D "C:\Program Files\edb\as10\data" -U enterprisedb
-b  "C:\Program Files\PostgresPlus\9.6AS\bin" -B
"C:\Program Files\edb\as10\bin" -p 5444 -P 5445
```

pg_upgrade will display the progress of the upgrade onscreen:

```
$ pg_upgrade -d /opt/edb/as10/data -D /opt/edb/as10/data -U enterprisedb -b
/opt/edb/as10/bin -B /opt/edb/as10/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.

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:

- Delete any tablespace directories created by the new cluster.
- Rename `$PGDATA/global/pg_control`, removing the `.old` suffix.
- Rename the old cluster tablespace directory names, removing the `.old` suffix.
- 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.

8.4 pg_upgrade Troubleshooting

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

8.4.1 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.

8.4.2 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.

8.4.3 Upgrade Error - New cluster is not empty; exiting

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

8.4.4 Upgrade Error - Failed to load library

If the original Advanced Server cluster included libraries that are not included in the Advanced Server 10 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:

- Restart the Advanced Server service.

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

- Stop the Advanced Server service.

- Resume the upgrade process: invoke pg_upgrade to perform consistency checking.

- 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.

8.5 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.

8.6 Migration to Version 10

A dump/restore using pg_dumpall or use of pg_upgrade or logical replication is required for those wishing to migrate data from any previous release. See Section 8, *Upgrading an Installation With pg_upgrade* for general information on migrating to new major releases.

Version 10 contains a number of changes that may affect compatibility with previous releases. Observe the following incompatibilities:

- Hash indexes must be rebuilt after pg_upgrade from any previous major PostgreSQL version.

  Major hash index improvements necessitated this requirement. pg_upgrade will create a script to assist with this.

- Rename write-ahead log directory pg_xlog to pg_wal, and rename transaction status directory pg_clog to pg_xact.

  Users have occasionally thought that these directories contained only inessential log files, and proceeded to remove write-ahead log files or transaction status files manually, causing irrecoverable data loss. These name changes are intended to discourage such errors in future.

- Rename SQL functions, tools, and options that reference “xlog” to “wal”.

  For example, pg_switch_xlog() becomes pg_switch_wal(), pg_receivexlog becomes pg_receivewal, and --xlogdir becomes --waldir. This is for consistency with the change of the pg_xlog directory name; in general, the “xlog” terminology is no longer used in any user-facing places.

- Rename WAL-related functions and views to use lsn instead of location.

  There was previously an inconsistent mixture of the two terminologies.

- Change the implementation of set-returning functions appearing in a query’s SELECT list.

  Set-returning functions are now evaluated before evaluation of scalar expressions in the SELECT list, much as though they had been placed in a LATERAL FROM-clause item. This allows saner semantics for cases where multiple set-returning functions are present. If they return different numbers of rows, the shorter results are extended to match the longest result by adding nulls. Previously the results were cycled until they all terminated at the same time, producing a number of rows equal to the least common multiple of the functions' periods. In addition, set-returning functions are now disallowed within `CASE` and `COALESCE` constructs.

- Use standard row constructor syntax in `UPDATE ... SET (`*`column_list`*`) = `*`row_constructor`*.

  The *`row_constructor`* can now begin with the keyword `ROW`; previously that had to be omitted. If just one column name appears in the *`column_list`*, then the *`row_constructor`* now must use the `ROW` keyword, since otherwise it is not a valid row constructor but just a parenthesized expression. Also, an occurrence of *`table_name`*`.*` within the *`row_constructor`* is now expanded into multiple columns, as occurs in other uses of *`row_constructors`*.

- When `ALTER TABLE ... ADD PRIMARY KEY` marks columns `NOT NULL`, that change now propagates to inheritance child tables as well.

- Prevent statement-level triggers from firing more than once per statement.

  Cases involving writable CTEs updating the same table updated by the containing statement, or by another writable CTE, fired `BEFORE STATEMENT` or `AFTER STATEMENT` triggers more than once. Also, if there were statement-level triggers on a table affected by a foreign key enforcement action (such as `ON DELETE CASCADE`), they could fire more than once per outer SQL statement. This is contrary to the SQL standard, so change it.

- Move sequences' metadata fields into a new `pg_sequence` system catalog.

  A sequence relation now stores only the fields that can be modified by `nextval()`, that is `last_value, log_cnt,` and `is_called`. Other sequence properties, such as the starting value and increment, are kept in a corresponding row of the `pg_sequence` catalog. `ALTER SEQUENCE` updates are now fully transactional, implying that the sequence is locked until commit. The `nextval()` and `setval()` functions remain nontransactional.

  The main incompatibility introduced by this change is that selecting from a sequence relation now returns only the three fields named above. To obtain the sequence's other properties, applications must look into `pg_sequence`. The new system view `pg_sequences` can also be used for this purpose; it provides column names that are more compatible with existing code.

  Also, sequences created for `SERIAL` columns now generate positive 32-bit wide values, whereas previous versions generated 64-bit wide values. This has no visible effect if the values are only stored in a column.

The output of psql's `\d` command for a sequence has been redesigned, too.

- Make `pg_basebackup` stream the WAL needed to restore the backup by default.

  This changes `pg_basebackup's -X/--wal-method` default to `stream`. An option value `none` has been added to reproduce the old behavior. The `pg_basebackup` option `-x` has been removed (instead, use `-X fetch`).

- Change how logical replication uses `pg_hba.conf`.

  In previous releases, a logical replication connection required the `replication` keyword in the database column. As of this release, logical replication matches a normal entry with a database name or keywords such as `all`. Physical replication continues to use the `replication` keyword. Since built-in logical replication is new in this release, this change only affects users of third-party logical replication plugins.

- Make all `pg_ctl` actions wait for completion by default.

  Previously some `pg_ctl` actions didn't wait for completion, and required the use of `-w` to do so.

- Change the default value of the `log_directory` server parameter from `pg_log` to `log`.

- Add configuration option `ssl_dh_params_file` to specify file name for custom OpenSSL DH parameters.

  This replaces the hardcoded, undocumented file name `dh1024.pem`. Note that `dh1024.pem` is no longer examined by default; you must set this option if you want to use custom DH parameters.

- Increase the size of the default DH parameters used for OpenSSL ephemeral DH ciphers to 2048 bits.

  The size of the compiled-in DH parameters has been increased from 1024 to 2048 bits, making DH key exchange more resistant to brute-force attacks. However, some old SSL implementations, notably some revisions of Java Runtime Environment version 6, will not accept DH parameters longer than 1024 bits, and hence will not be able to connect over SSL. If it's necessary to support such old clients, you can use custom 1024-bit DH parameters instead of the compiled-in defaults.

- Remove the ability to store unencrypted passwords on the server.

The password_encryption server parameter no longer supports off or plain. The UNENCRYPTED option is no longer supported in CREATE/ALTER USER ... PASSWORD. Similarly, the --unencrypted option has been removed from createuser. Unencrypted passwords migrated from older versions will be stored encrypted in this release. The default setting for password_encrytion is still md5.

- Add min_parallel_table_scan_size and min_parallel_index_scan_size server parameters to control parallel queries.

  These replace min_parallel_relation_size, which was found to be too generic.

- Don't downcase unquoted text within shared_preload_libraries and related server parameters.

  These settings are really lists of file names, but they were previously treated as lists of SQL identifiers, which have different parsing rules.

- Remove sql_inheritance server parameter.

  Changing this setting from the default value caused queries referencing parent tables to not include child tables. The SQL standard requires them to be included, however, and this has been the default since PostgreSQL 7.1.

- Allow multi-dimensional arrays to be passed into PL/Python functions, and returned as nested Python lists.

  This feature requires a backwards-incompatible change to the handling of arrays of composite types in PL/Python. Previously, you could return an array of composite values by writing, e.g., [[col1, col2], [col1, col2]]; but now that is interpreted as a two-dimensional array. Composite types in arrays must now be written as Python tuples, not lists, to resolve the ambiguity; that is, write [(col1, col2), (col1, col2)] instead.

- Remove PL/Tcl's “module” auto-loading facility.

  This functionality has been replaced by new server parameters pltcl.start_proc and pltclu.start_proc, which are easier to use and more similar to features available in other PLs.

- Remove pg_dump/pg_dumpall support for dumping from pre-8.0 servers.

  Users needing to dump from pre-8.0 servers will need to use dump programs from PostgreSQL 9.6 or earlier. The resulting output should still load successfully into newer servers.

- Remove support for floating-point timestamps and intervals.

  This removes configure's `--disable-integer-datetimes` option. Floating-point timestamps have few advantages and have not been the default since PostgreSQL 8.3.

- Remove server support for client/server protocol version 1.0.

  This protocol hasn't had client support since PostgreSQL 6.3.

- Remove `contrib/tsearch2` module.

  This module provided compatibility with the version of full text search that shipped in pre-8.3 PostgreSQL releases.

- Remove createlang and droplang command-line applications.

  These had been deprecated since PostgreSQL 9.1. Instead, use `CREATE EXTENSION` and `DROP EXTENSION` directly.

- Remove support for version-0 function calling conventions.

  Extensions providing C-coded functions must now conform to version 1 calling conventions. Version 0 has been deprecated since 2001.

- The `SELECT DISTINCT...ORDER BY` clause of the `SELECT DISTINCT` query behavior differs after upgrade.

  If `SELECT DISTINCT` is specified or if a `SELECT` statement includes the `SELECT DISTINCT ...ORDER BY` clause then all the expressions in `ORDER BY` must be present in the select list of the `SELECT DISTINCT` query (applicable when upgrading from version 9.6 to any higher version of Advanced Server).

9 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.

9.1 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:

- On RHEL or CentOS 7:

```
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.

9.2 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 bundled with the installer (pgAdmin 4, StackBuilder Plus, or the command line tools). If you uninstall an Advanced Server component, the remainder of the Advanced Server installation will remain intact.

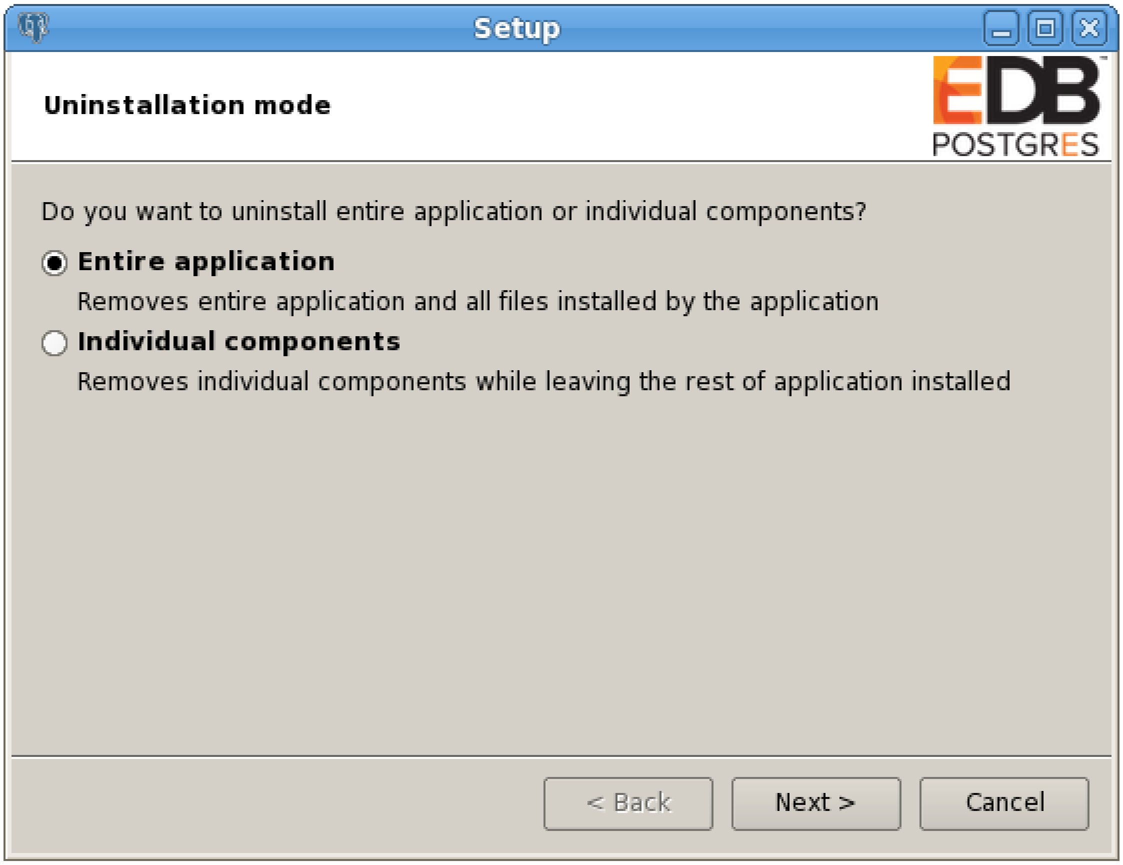
On Linux, the uninstaller is created in `/opt/edb/as10`. To open the uninstaller, assume superuser privileges, navigate into the directory that contains the uninstaller, and enter:

```
./uninstall-edb-as10-server
```

On Windows, the uninstaller is created in `C:\Program Files\edb\as10`. To open the uninstaller, assume superuser privileges, navigate into the directory that contains the uninstaller, and enter:

```
uninstall-edb-as10-server.exe
```

The uninstaller opens as shown in Figure 8.1.



Uninstallation mode

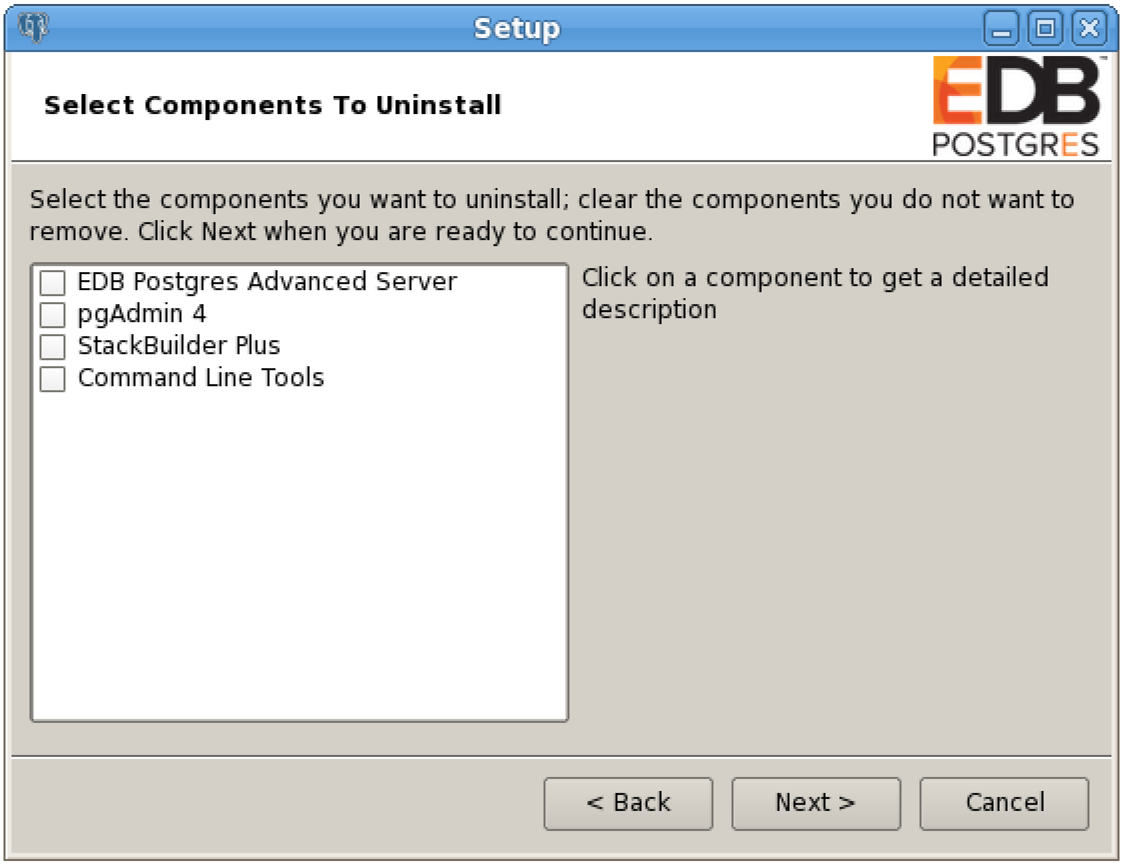
Do you want to uninstall entire application or individual components?

● **Entire application**
Removes entire application and all files installed by the application

○ **Individual components**
Removes individual components while leaving the rest of application installed

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; click `Next`.



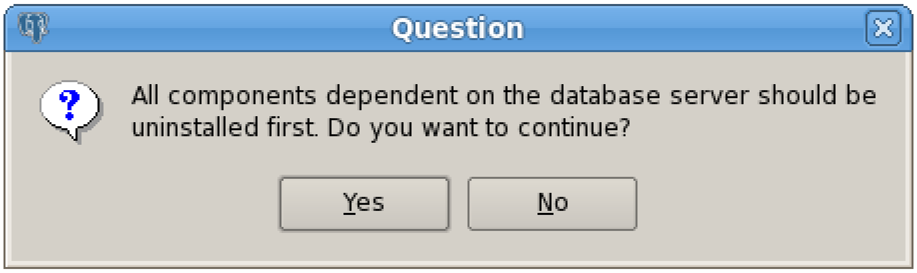
Select the components you want to uninstall; clear the components you do not want to remove. Click Next when you are ready to continue.

- □ EDB Postgres Advanced Server
- □ pgAdmin 4
- □ StackBuilder Plus
- □ Command Line Tools

Click on a component to get a detailed description

Figure 8.2 – Select components for uninstallation.

Check the box to the left of a component name to select a component for removal and click `Next` to continue (see Figure 8.2).

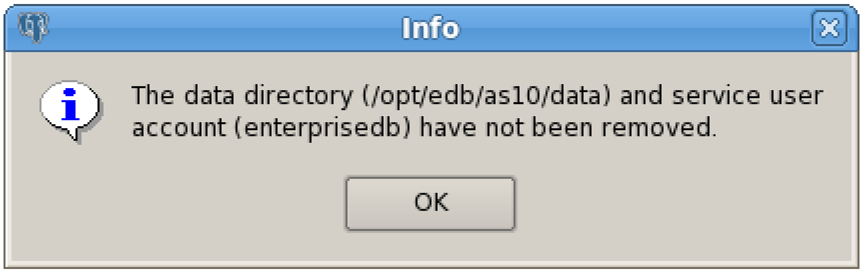


All components dependent on the database server should be uninstalled first. Do you want to continue?

Figure 8.3 – Acknowledge that dependent components are removed first.

If you choose 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 8.3).

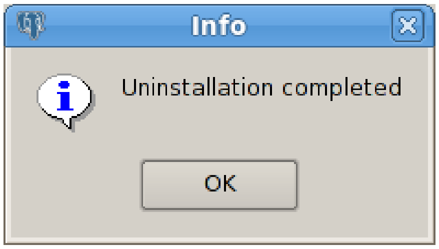
When the uninstaller completes, a popup confirms that the data directory and service account have not been removed (see Figure 8.4).



The data directory (/opt/edb/as10/data) and service user account (enterprisedb) have not been removed.

Figure 8.4 - A dialog confirms that the data directory and service user have not been removed.

When the uninstallation is complete, an Info dialog opens to confirm that Advanced Server (and/or its components) has been removed (see Figure 8.5).



Uninstallation completed

Figure 8.5 - The uninstallation is complete.