



EDB

EDB Postgres Language Pack Guide

Release 14.1.0

EnterpriseDB

Dec 18, 2021

CONTENTS

1	Supported Database Server Versions	3
2	Installing and Configuring Language Pack	5
2.1	Installing Language Pack	5
2.1.1	Invoking the Graphical Installer	5
2.1.2	Installing Language Pack with Stack Builder	9
2.2	Configuring Language Pack	14
2.2.1	Configuring Language Pack on an Advanced Server Host	14
2.2.2	Configuring Language Pack on a PostgreSQL Host	14
3	Using the Procedural Languages	17
3.1	PL/Perl	17
3.2	PL/Python	18
3.3	PL/Tcl	18
4	Uninstalling Language Pack	21

This guide provides information about how to install and configure Language Pack, as well as how to use the procedural languages (PL/Perl, PL/Python, and PL/TCL).

Language pack installers contain supported languages that may be used with EDB Postgres Advanced Server and EnterpriseDB PostgreSQL database installers. The language pack installer allows you to install Perl, TCL/TK, and Python without installing supporting software from third-party vendors.

The Language Pack 2.0 installer includes:

- TCL **with** TK version **8.6**

- Perl version **5.26**

- Python version **3.7**

The Perl package contains the `cpan` package manager, and Python contains `pip` and `easy_install` package managers. There is no package manager for TCL/TK.

In previous Postgres releases, `plpython` was statically linked with ActiveState's python library. The Language Pack Installer dynamically links with our shared object for python. In ActiveState Linux installers for Python, there is no dynamic library. As a result of these changes, `plpython` will no longer work with ActiveState installers.

Convention Used in this Guide

The term *Postgres* refers to either PostgreSQL or EDB Postgres Advanced Server.

SUPPORTED DATABASE SERVER VERSIONS

Language Pack installers are version and platform specific; select the Language Pack installer that corresponds to your EDB Postgres Advanced Server or PostgreSQL server version:

Linux:

EDB Postgres Advanced Server/PostgreSQL Version	Language Pack Version	Procedural Language Version
9.6, 10	1.0	Perl 5.26, Python 3.7, Tcl 8.6

For detailed information about using an RPM package to add Language Pack, see the *EDB Postgres Advanced Server Installation Guide*, available at the [EDB website](#).

Mac OS:

PostgreSQL Version	Language Pack Version	Procedural Language Version
9.6, 10, 11, 12, 13	1.0	Perl 5.26, Python 3.7, Tcl 8.6

Windows 32:

EDB Postgres Advanced Server/PostgreSQL Version	Language Pack Version	Procedural Language Version
9.6, 10	1.0	Perl 5.26, Python 3.7, Tcl 8.6

Windows 64:

EDB Postgres Advanced Server/PostgreSQL Version	Language Pack Version	Procedural Language Version
<i>PostgreSQL 9.6, 10, 11, 12, 13</i>	1.0	Perl 5.26, Python 3.7, Tcl 8.8
<i>EDB Postgres Advanced Server 12</i>	1.0	Perl 5.26, Python 3.7, Tcl 8.6

INSTALLING AND CONFIGURING LANGUAGE PACK

This section walks you through installing and configuring Language Pack.

2.1 Installing Language Pack

The graphical installer is available from the [EDB website](#), as well as via Stack Builder and StackBuilder Plus. StackBuilder Plus is distributed with EDB Postgres Advanced Server and Stack Builder is distributed with PostgreSQL.

2.1.1 Invoking the Graphical Installer

Assume Administrator privileges, and double-click the installer icon; if prompted, provide the password associated with the Administrator account. When prompted, select an installation language, and click OK.

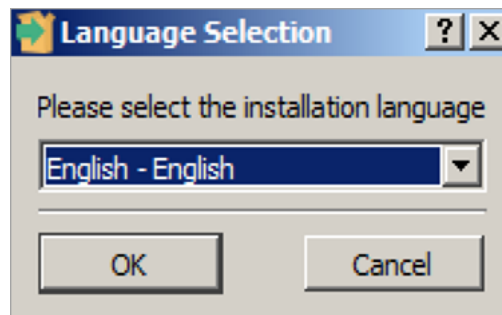


Fig. 1: The Language Selection Window

Fig. 1: The Language Selection Window

The Language Pack setup wizard welcome window opens.

Fig. 2: The Language Pack Welcome Window

Click **Next** to continue.

The Ready to Install window displays the Language Pack installation directory:

On Windows 64: C:/edb/languagepack/v2

On OSX: /Library/edb/languagepack/v2

Fig. 3: The Ready to Install dialog

You cannot modify the installation directory. Click **Next** to continue.

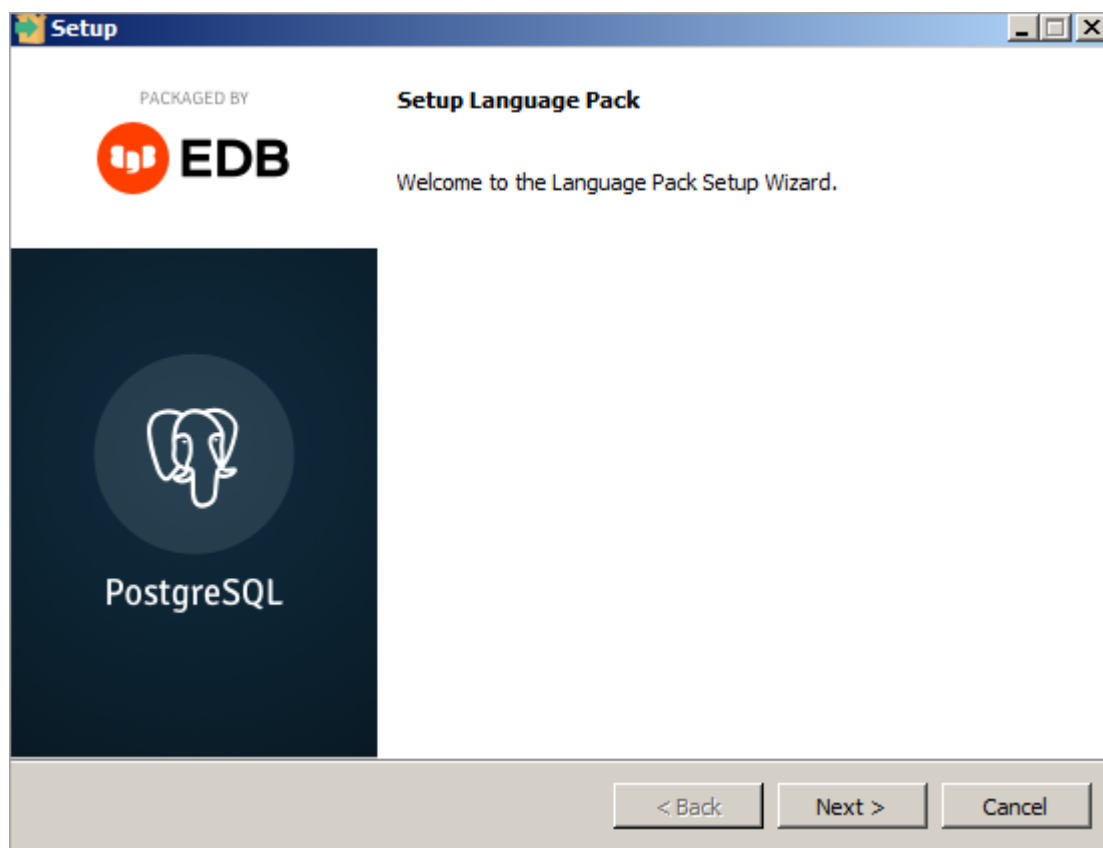


Fig. 2: The Language Pack Welcome Window

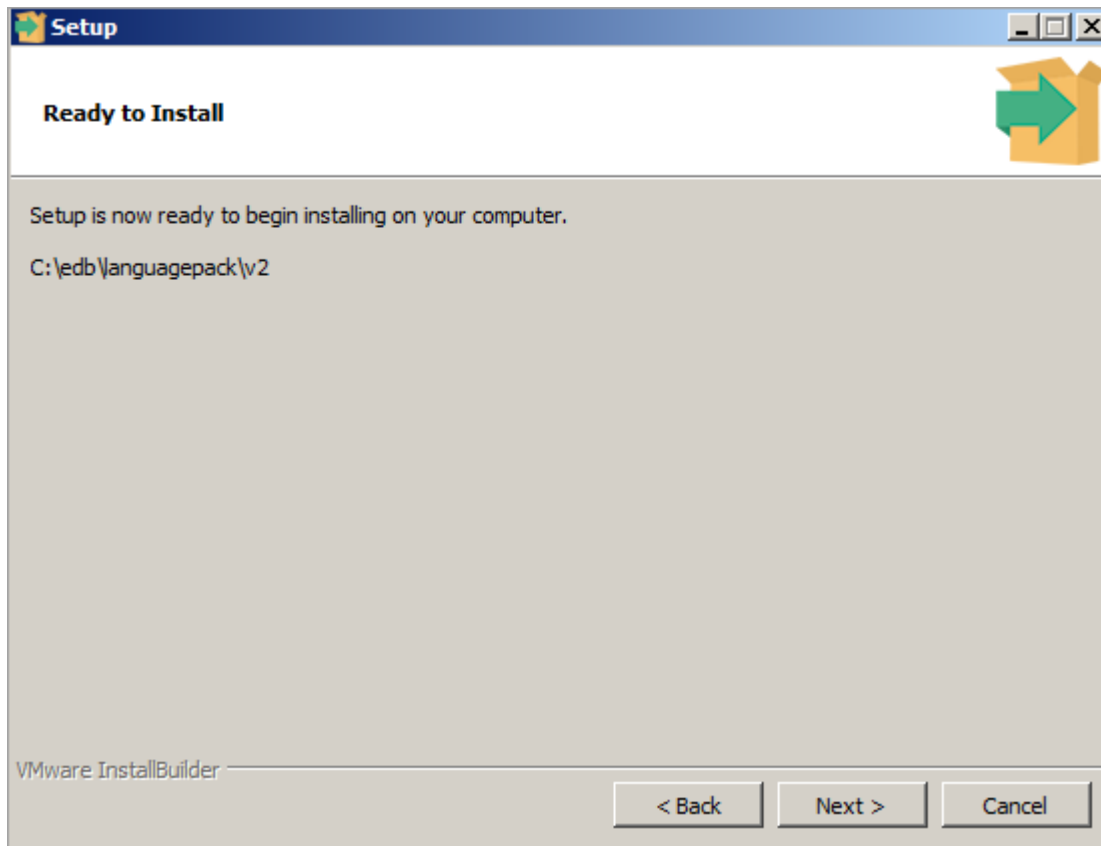


Fig. 3: The Ready to Install dialog

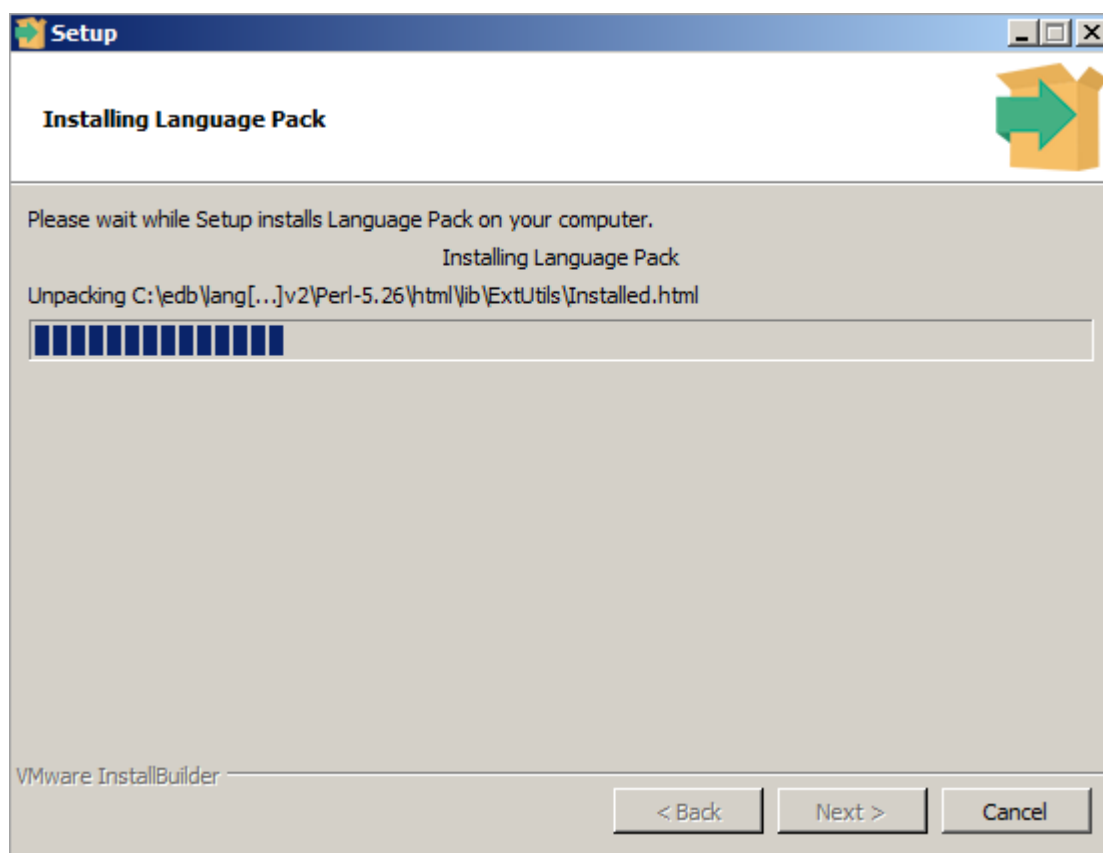


Fig. 4: The Installing dialog

A progress bar marks installation progress.

Fig. 4: The Installing dialog

Click **Next** to continue.

The installer will inform you that the Language Pack installation has completed; click **Finish** to exit the installer.

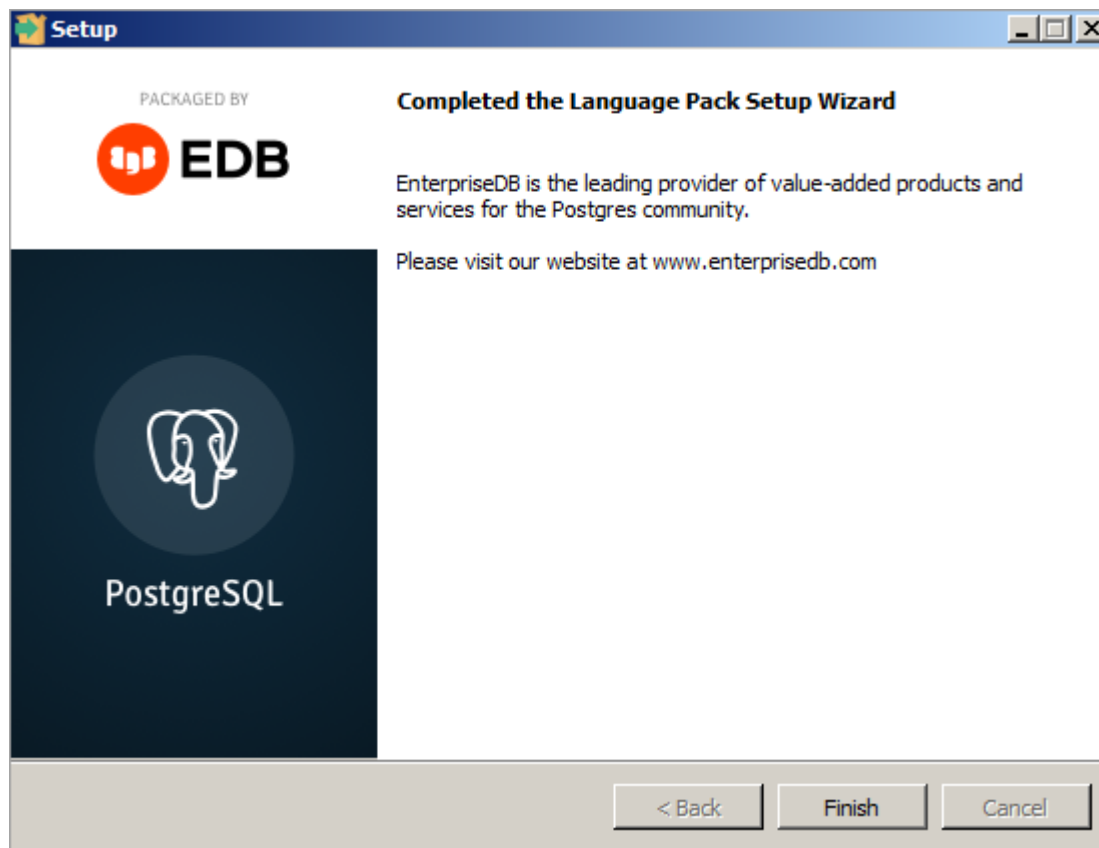


Fig. 5: The Language Pack Setup Complete dialog

Fig. 5: The Language Pack Setup Complete dialog

2.1.2 Installing Language Pack with Stack Builder

You can use Stack Builder or StackBuilder Plus to download and invoke the Language Pack graphical installer. StackBuilder Plus is distributed with EDB Postgres Advanced Server and Stack Builder is distributed with PostgreSQL.

The following section walks you through installing Language Pack with Stack Builder.

The Stack Builder utility provides a graphical interface that simplifies the process of downloading and installing modules that complement your PostgreSQL installation.

Stack Builder requires Internet access; if your installation of PostgreSQL resides behind a firewall (with restricted Internet access), Stack Builder 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.

You can invoke Stack Builder at any time after the installation has completed by selecting the **Application Stack Builder** menu option from the **PostgreSQL 14** menu.

Select your server from the drop-down menu on the Stack Builder Welcome window and click Next to continue.

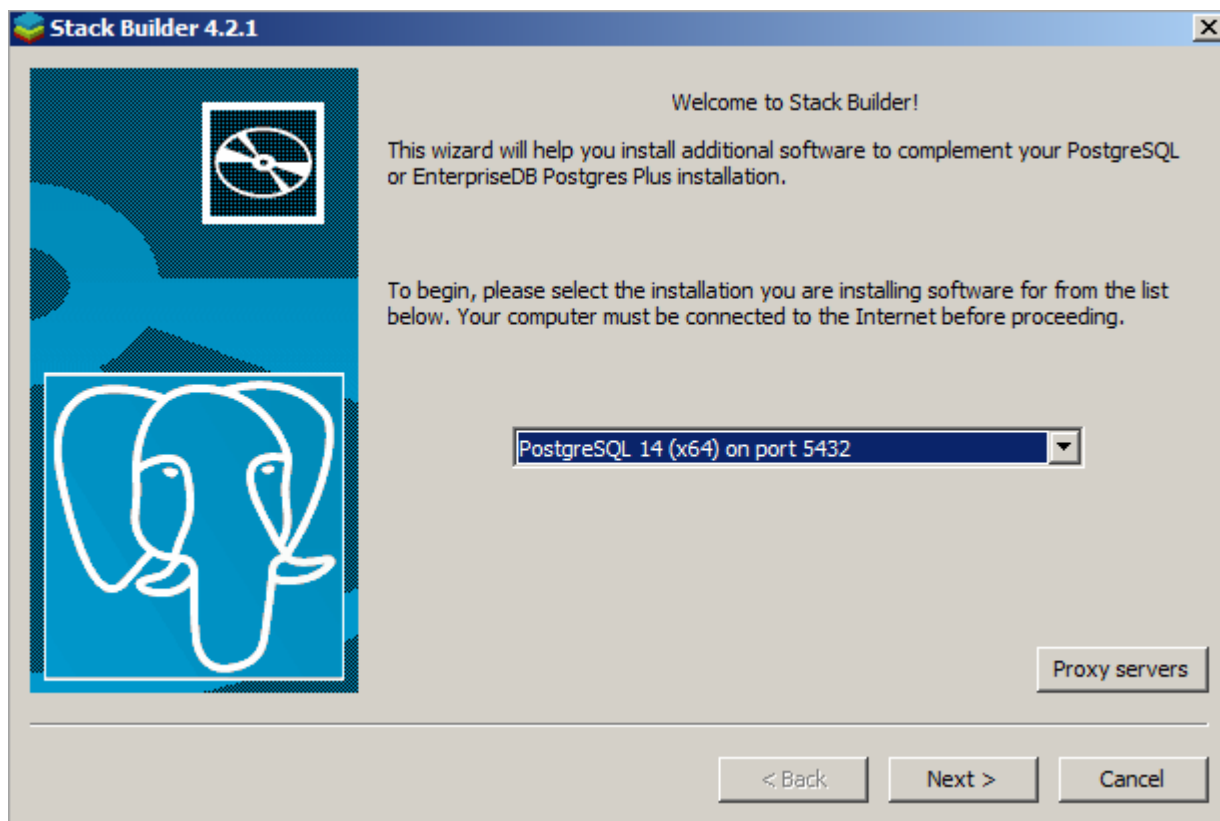


Fig. 6: The Stack Builder Welcome Window

Fig. 6: The Stack Builder Welcome Window

Expand the Add-ons, tools and utilities node of the Categories tree control, and check the box next to EDB Language Pack v2.0-1. Click Next to continue.

Fig. 7: The Language Pack Selection Window

Stack Builder will confirm your package selection before downloading the installer. Click Next to continue.

Fig. 8: The Selected Packages Window

When the download completes, Stack Builder will offer to invoke the installer for you, or will delay the installation until a more convenient time. To invoke the installer, click Next and follow the steps provided in the *Invoking the Graphical Installer* section.

Fig. 9: The StackBuilder Installation Complete

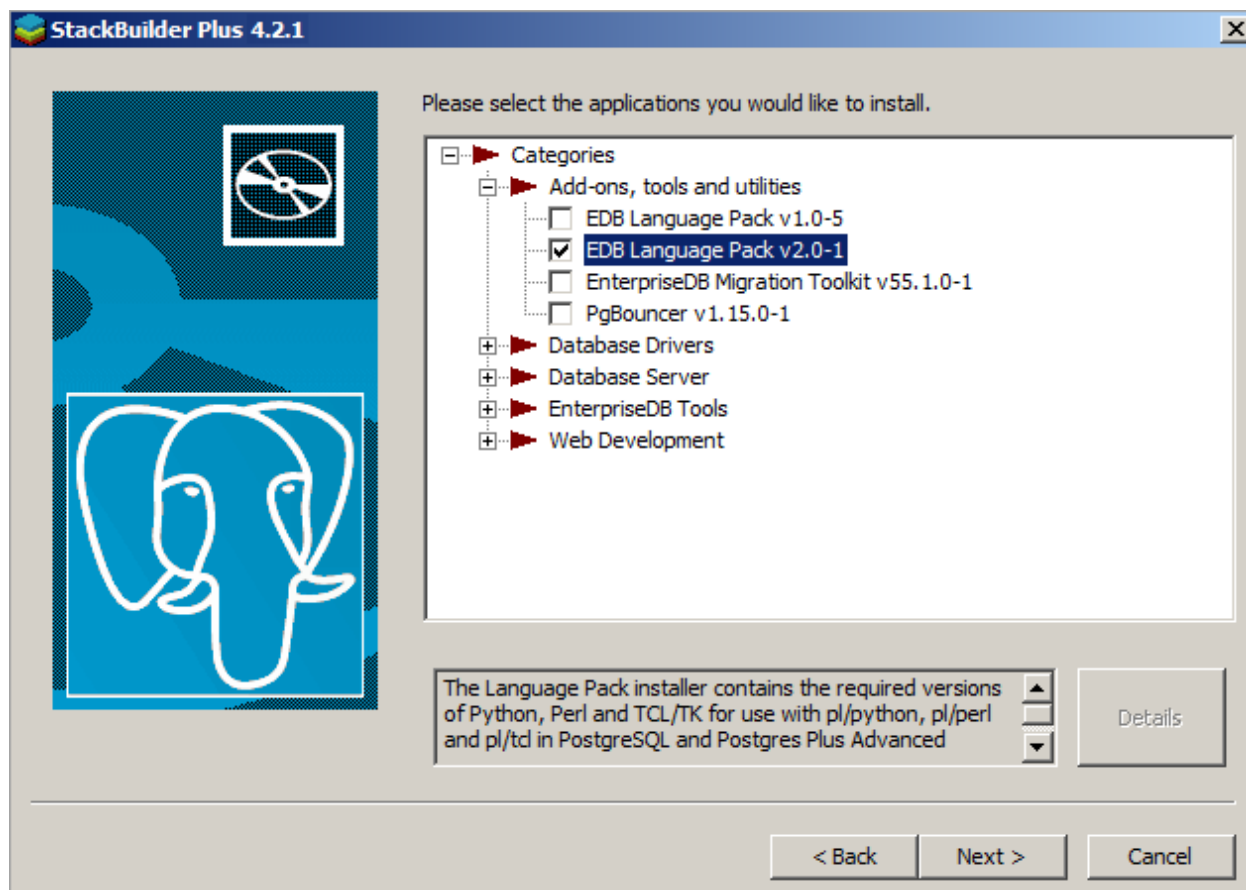


Fig. 7: The Language Pack Selection Window

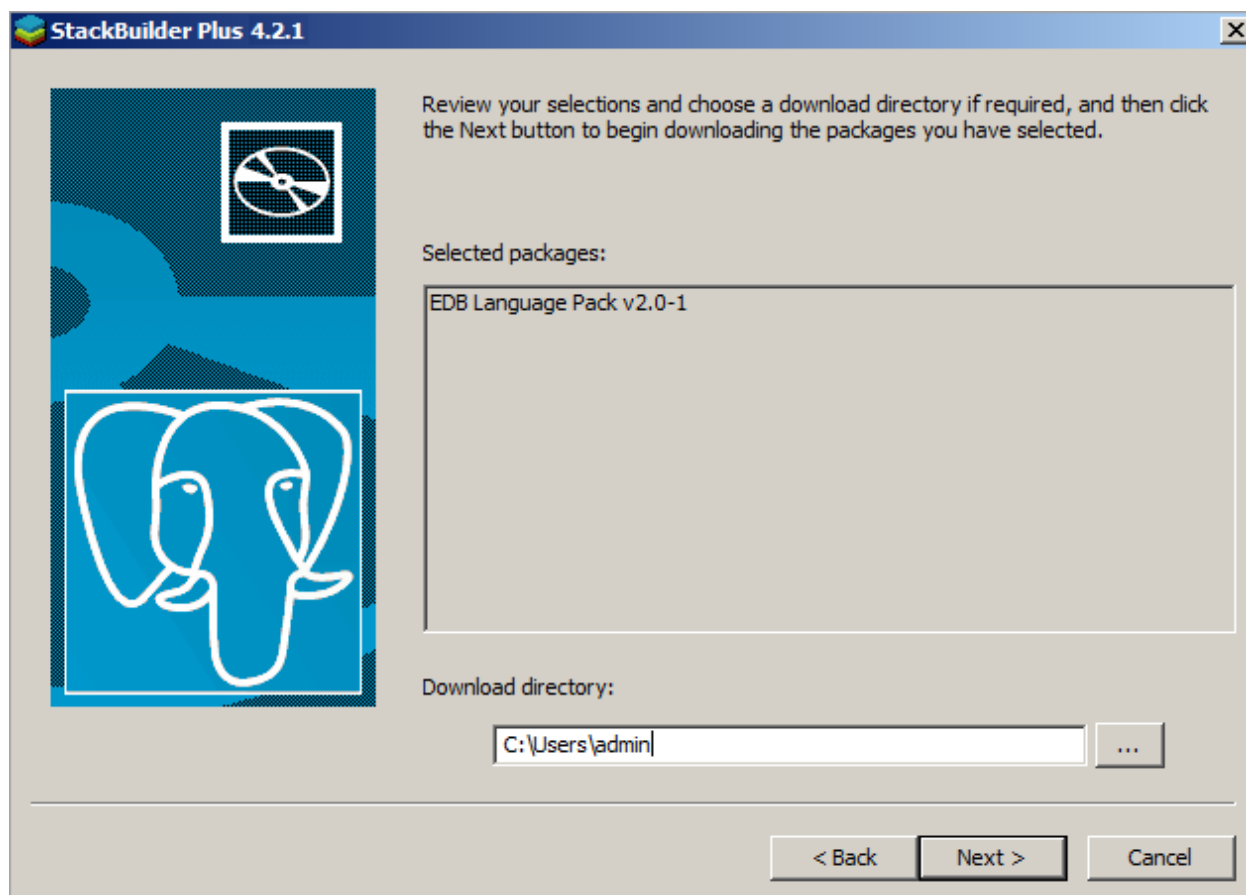


Fig. 8: The Selected Packages Window

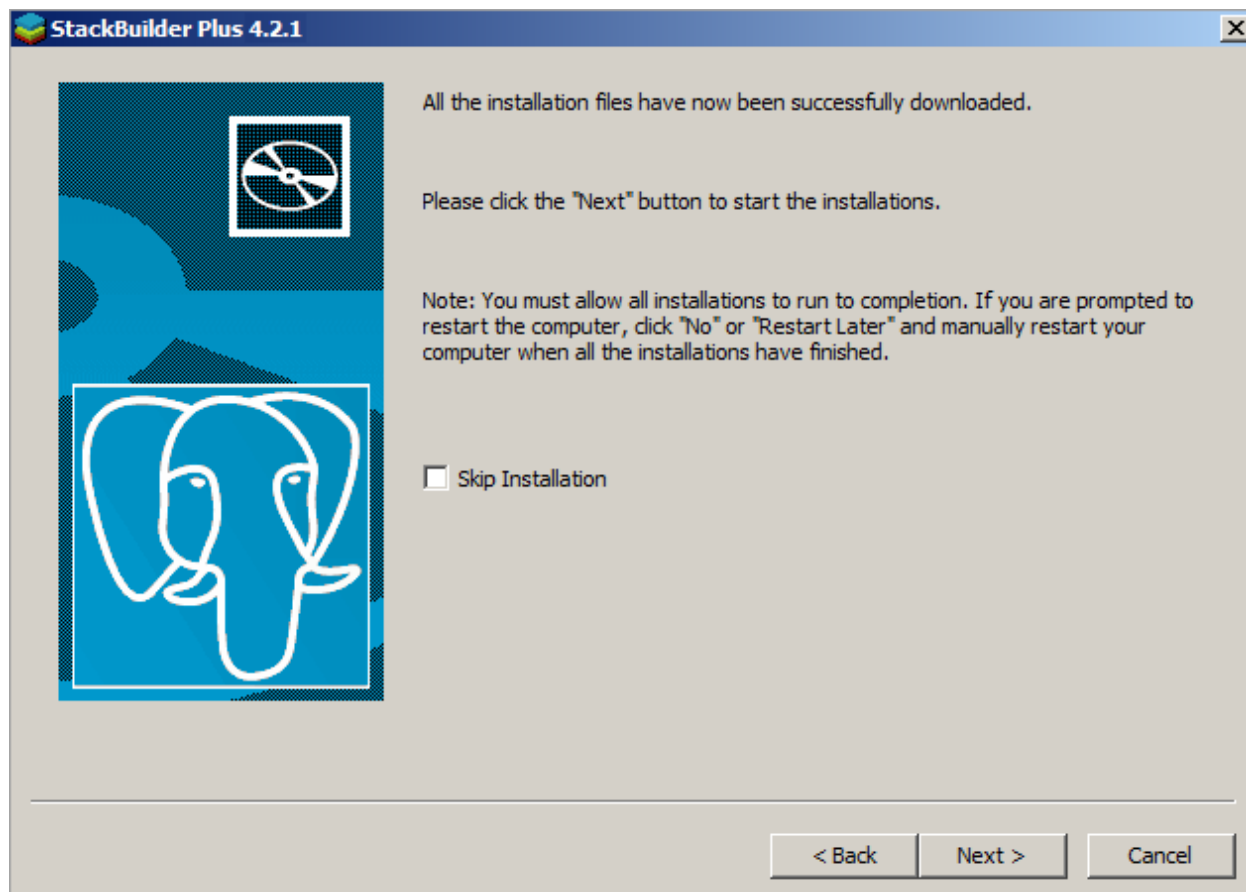


Fig. 9: The Selected Packages Window

2.2 Configuring Language Pack

This section walks you through configuring Language Pack on an Advanced Server and PostgreSQL hosts.

2.2.1 Configuring Language Pack on an Advanced Server Host

After installing Language Pack on an Advanced Server host, you must configure the installation.

Configuring Language Pack on Windows

On Windows, the Language Pack installer places the languages in:

C:\edb\languagepack\v2

After installing Language Pack, you must set the following variables:

```
set PYTHONHOME=C:\edb\languagepack\v2\Python-3.7
```

Use the following commands to add Python, Perl and Tcl to your search path:

```
set PATH=C:\edb\languagepack\v2\Python-3.7;  
C:\edb\languagepack\v2\Perl-5.26\bin;  
C:\edb\languagepack\v2\Tcl-8.6\bin;%PATH%
```

After performing the steps required to configure Language Pack on Windows, use the Windows Services applet to restart the Advanced Server.

2.2.2 Configuring Language Pack on a PostgreSQL Host

After installing Language Pack on a PostgreSQL host, you must configure the installation.

Configuring Language Pack on Windows

After installing Language Pack, you must set the following variables:

```
set PYTHONHOME=C:\edb\languagepack\v2\Python-3.7
```

Then, use the following commands to add Language Pack to your search path:

```
set PATH=C:\edb\languagepack\v2\Python-3.7;  
C:\edb\languagepack\v2\Perl-5.26\bin;  
C:\edb\languagepack\v2\Tcl-8.6\bin;%PATH%
```

After setting the system-specific steps required to configure Language Pack on Windows, restart the database server.

Configuring Language Pack on OSX

To simplify setting the value of PATH or LD_LIBRARY_PATH, you can create environment variables that identify the installation location:

```
PERLHOME=/Library/edb/languagepack/v2/Perl-5.26  
PYTHONHOME=/Library/edb/languagepack/v2/Python-3.7  
TCLHOME=/Library/edb/languagepack/v2/Tcl-8.6
```

Then, execute the following command to instruct the Python interpreter where to find Python:

```
export PYTHONHOME
```

You can use the same environment variables when setting the value of PATH:

```
export PATH=$PYTHONHOME/bin:  
$PERLHOME/bin:  
$TCLHOME/bin:$PATH
```

Lastly, set the following variables to instruct OSX where to find the shared libraries:

```
export DYLD_LIBRARY_PATH=$PYTHONHOME/lib:  
$PERLHOME/lib/CORE:$TCLHOME/lib:  
$DYLD_LIBRARY_PATH
```


USING THE PROCEDURAL LANGUAGES

The Postgres procedural languages (PL/Perl, PL/Python, and PL/Java) are installed by the Language Pack installer. You can also use an RPM package to add procedural language functionality to your EDB Postgres Advanced Server installation. For more information about using an RPM package, please see the *EDB Postgres Advanced Server Installation Guide*, available at the [EDB website](#).

3.1 PL/Perl

The PL/Perl procedural language allows you to use Perl functions in Postgres applications.

You must install PL/Perl in each database (or in a template database) before creating a PL/Perl function. Use the `CREATE LANGUAGE` command at the EDB-PSQL command line to install PL/Perl. Open the EDB-PSQL client, establish a connection to the database in which you wish to install PL/Perl, and enter the command:

```
CREATE EXTENSION plperl;
```

You can now use a Postgres client application to access the features of the PL/Perl language. The following PL/Perl example creates a function named `perl_max` that returns the larger of two integer values:

```
CREATE OR REPLACE FUNCTION perl_max (integer, integer) RETURNS integer AS
$$
if ($_[0] > $_[1])
{ return $_[0]; }
return $_[1];
$$ LANGUAGE plperl;
```

Pass two values when calling the function:

```
SELECT perl_max(1, 2);
```

The server returns:

```
perl_max
-----
*      2
(1 row)
```

For more information about using the Perl procedural language, consult the official [PostgreSQL documentation](#).

3.2 PL/Python

The PL/Python procedural language allows you to create and execute functions written in Python within Postgres applications. The version of PL/Python used by EDB Postgres Advanced Server and PostgreSQL is untrusted (plpython3u); it offers no restrictions on users to prevent potential security risks.

Install PL/Python in each database (or in a template database) before creating a PL/Python function. You can use the `CREATE LANGUAGE` command at the EDB-PSQL command line to install PL/Python. Use EDB-PSQL to connect to the database in which you wish to install PL/Python, and enter the command:

```
CREATE EXTENSION plpython3u;
```

After installing PL/Python in your database, you can use the features of the PL/Python language.

!!! Note * The indentation shown in the following example must be included as you enter the sample function in EDB-PSQL.

The following PL/Python example creates a function named `pymax` that returns the larger of two integer values:

```
CREATE OR REPLACE FUNCTION pymax (a integer, b integer) RETURNS integer AS
$$
if a > b:
return a
return b
$$ LANGUAGE plpython3u;
```

When calling the `pymax` function, pass two values as shown below:

```
SELECT pymax(12, 3);
```

The server returns:

```
pymax
-----
*      12
(1 row)
```

For more information about using the Python procedural language, consult the official [PostgreSQL documentation](#).

3.3 PL/Tcl

The PL/Tcl procedural language allows you to use Tcl/Tk functions in applications.

You must install PL/Tcl in each database (or in a template database) before creating a PL/Tcl function. Use the `CREATE LANGUAGE` command at the EDB-PSQL command line to install PL/Tcl. Use the `psql` client to connect to the database in which you wish to install PL/Tcl, and enter the command:

```
CREATE EXTENSION pltcl;
```

After creating the `pltcl` language, you can use the features of the PL/Tcl language from within your Postgres server.

The following PL/Tcl example creates a function named `tcl_max` that returns the larger of two integer values:

```
CREATE OR REPLACE FUNCTION tcl_max(integer, integer) RETURNS integer AS
$$
if {[argisnull 1]} {
```

(continues on next page)

(continued from previous page)

```
if {[argisnull 2]} { return_null }
return $2
}
if {[argisnull 2]} { return $1 }
if {$1 > $2} {return $1}
return $2
$$ LANGUAGE pltcl;
```

Pass two values when calling the function:

```
SELECT tcl_max(1, 2);
```

The server returns:

```
tcl_max
-----
*      2
(1 row)
```

For more information about using the Tcl procedural language, consult the official [PostgreSQL documentation](#).

UNINSTALLING LANGUAGE PACK

The following section outlines the process of uninstalling Language Pack.

The Language Pack graphical installer creates an uninstaller that you can use to remove Language Pack. The uninstaller is created in the installation directory.

Perform the following steps to uninstall Language Pack:

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and click Yes to begin uninstalling Language Pack.

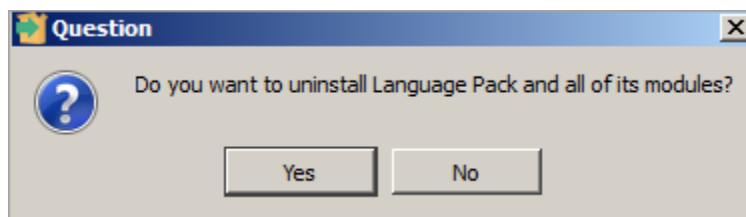


Fig. 1: The Language Pack Uninstaller

Fig. 1: The Language Pack Uninstaller

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

Fig. 2: Uninstalling Language Pack

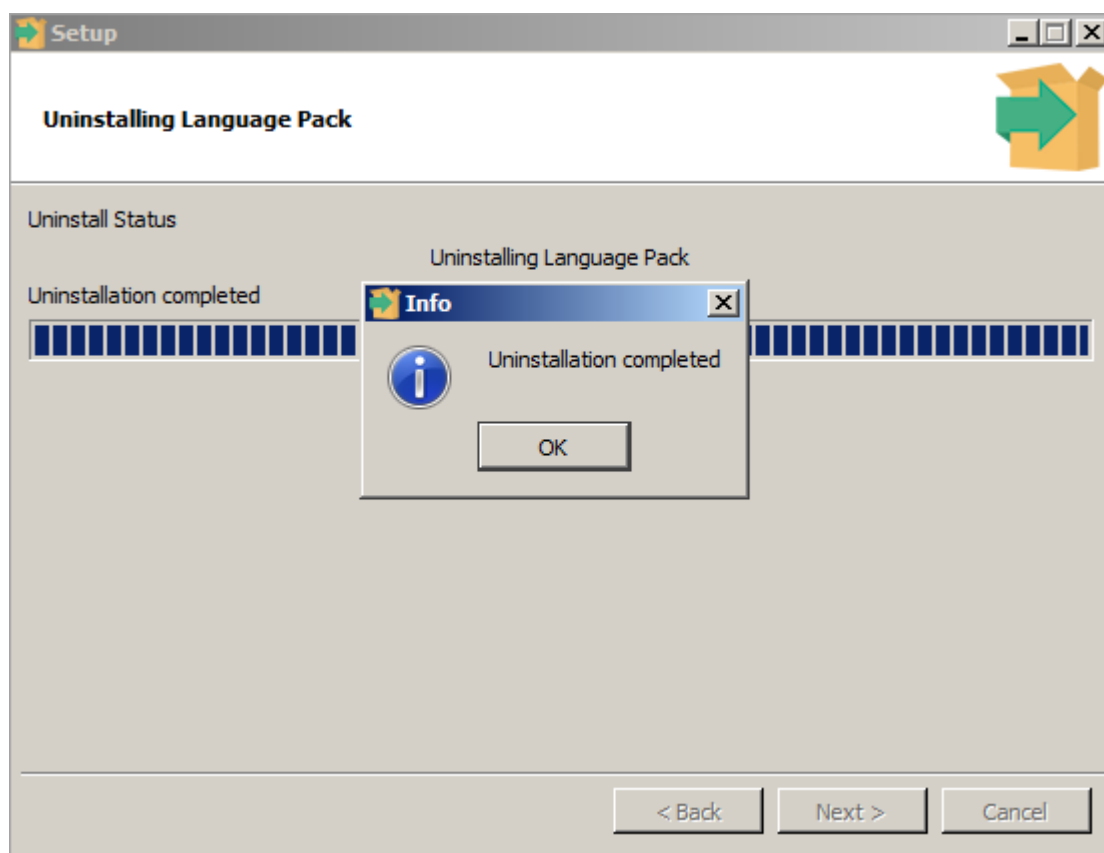


Fig. 2: Uninstalling Language Pack