



EDB Postgres Language Pack Guide

Release 1.0

May 22, 2020

1	Supported Database Server Versions	2
2	Installing Language Pack	4
2.1	Invoking the Graphical Installer	4
2.2	Installing Language Pack with StackBuilder Plus	9
2.3	Configuring Language Pack on an Advanced Server Host	10
2.3.1	Configuring Language Pack on Windows	10
2.3.2	Configuring Language Pack on a PostgreSQL Host	10
2.3.3	Configuring Language Pack on OSX	10
3	Using the Procedural Languages	12
3.1	PL/Perl	12
3.2	PL/Python	14
3.3	PL/Tcl	15
4	Conclusion	16
	Index	17

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 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.

This document uses the term *Postgres* to mean either EDB Postgres Advanced Server or EDB PostgreSQL. For more information about using EDB Postgres products, please visit the EnterpriseDB website at:

<https://www.enterprisedb.com/edb-docs>

StackBuilder Plus is distributed with Advanced Server; Stack Builder (distributed with PostgreSQL) provides comparable functionality. This document uses the term *StackBuilder Plus* to mean either StackBuilder Plus or Stack Builder.

CHAPTER 1

Supported Database Server Versions

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

Linux:

Advanced Server or PostgreSQL Version	Language Pack Installer Version	Installs Language Versions
9.5	9.5	Perl 5.20, Python 3.3, Tcl 8.5
9.6 through 10	1.0	Perl 5.26, Python 3.7, Tcl 8.6

For detailed information about using an RPM package to add Language Pack, please see the EDB Postgres Advanced Server Installation Guide for Linux, available at:

<https://www.enterprisedb.com/edb-docs/p/edb-postgres-advanced-server>

Mac OS:

Advanced Server or PostgreSQL Version	Language Pack Installer Version	Installs Language Version
9.5 through 12	1.0	Perl 5.26, Python 3.7, Tcl 8.6

Windows 32:

Advanced Server or PostgreSQL Version	Language Pack Installer Version	Installs Language Version
9.5	9.5	Perl 5.20, Python 3.3, Tcl 8.5
9.6 through 12	1.0	Perl 5.26, Python 3.7, Tcl 8.6

Windows 64:

Advanced Server or PostgreSQL Version	Language Pack Installer Version	Installs Language Version
PostgreSQL 9.5	9.5	Perl 5.20, Python 3.3, Tcl 8.5
PostgreSQL 9.6 through 12	1.0	Perl 5.26, Python 3.7, Tcl 8.8
Advanced Server 12	1.0	Perl 5.26, Python 3.7, Tcl 8.6

Installing Language Pack

The graphical installer is available from the EnterpriseDB website or via StackBuilder Plus.

2.1 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.

The installer `Welcome` window opens.

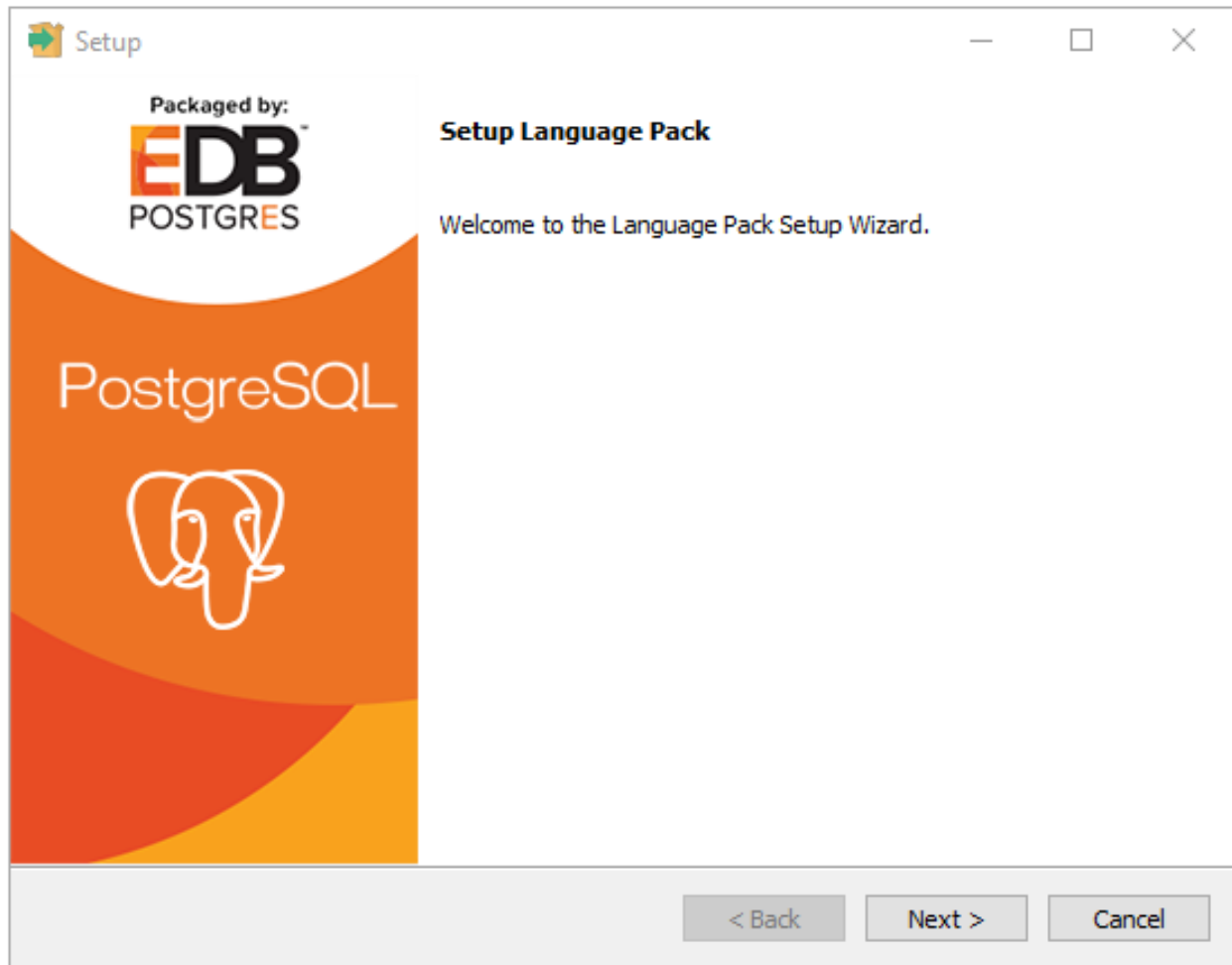


Fig. 2.1: *The Language Pack Welcome Window*

Click **Next** to continue.

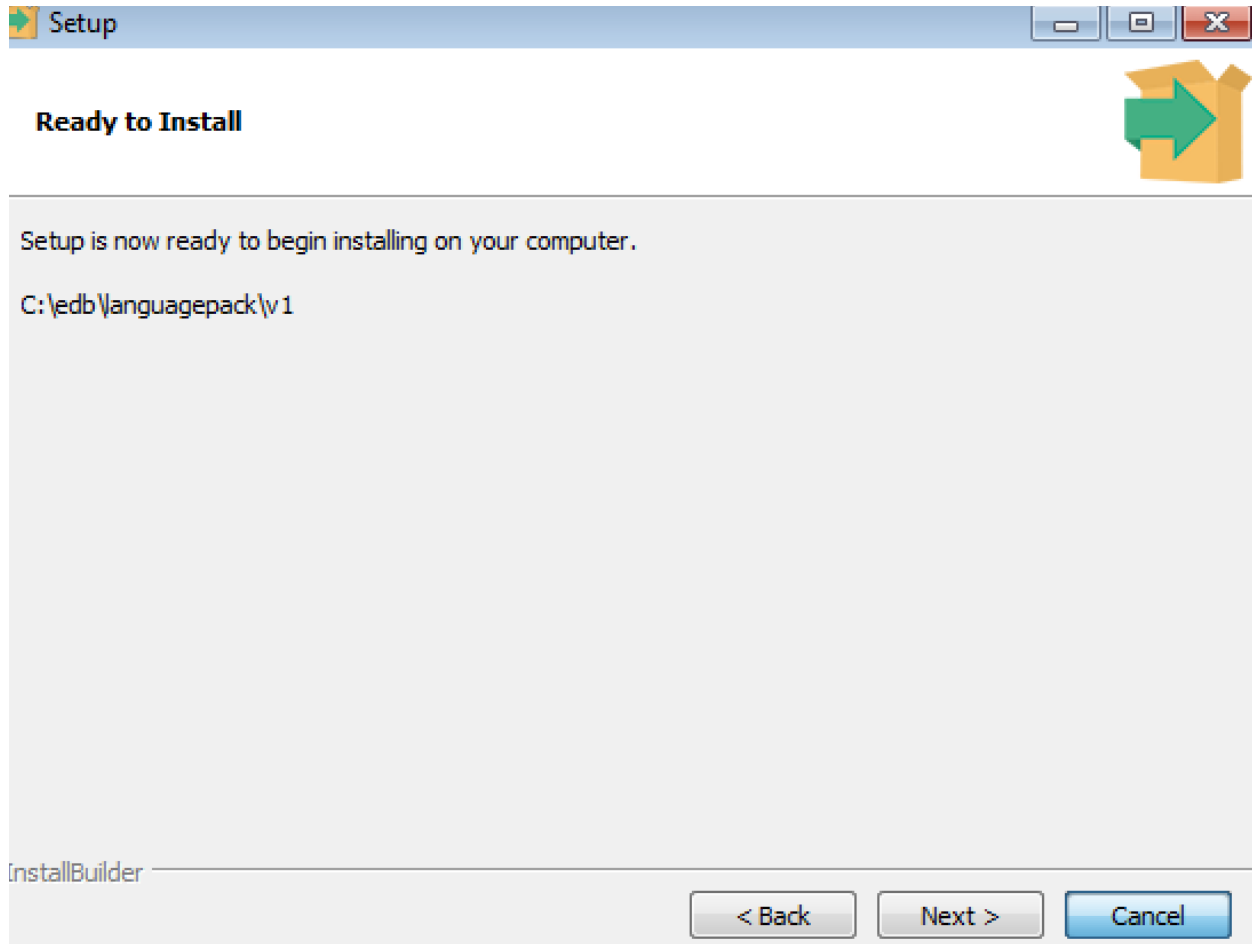


Fig. 2.2: *The Ready to Install dialog*

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

On Windows 64: C:\edb\languagepack\v1

On OSX: /Library/edb/languagepack/v1

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

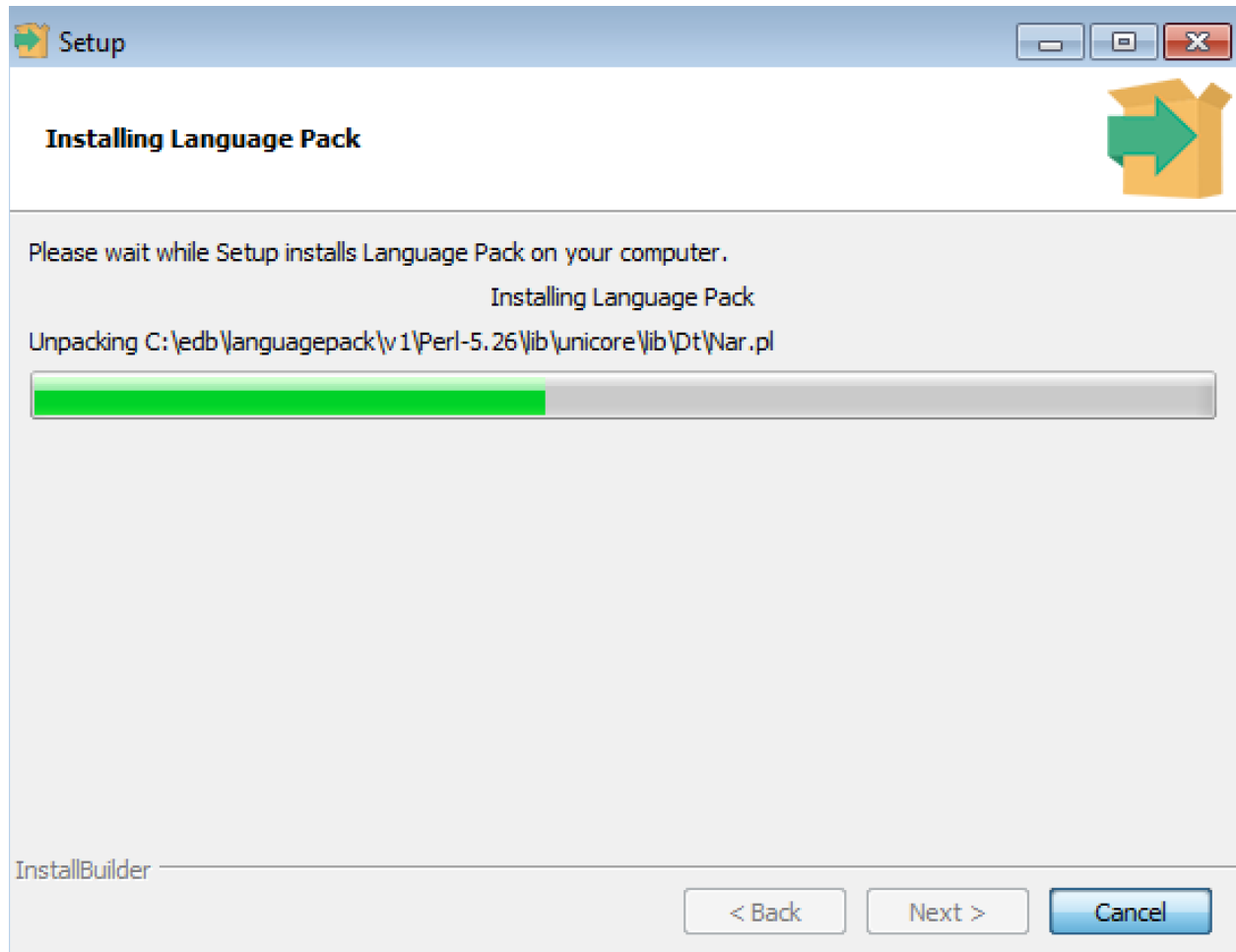


Fig. 2.3: *The Installing dialog*

A progress bar marks installation progress; click `Next` to continue.

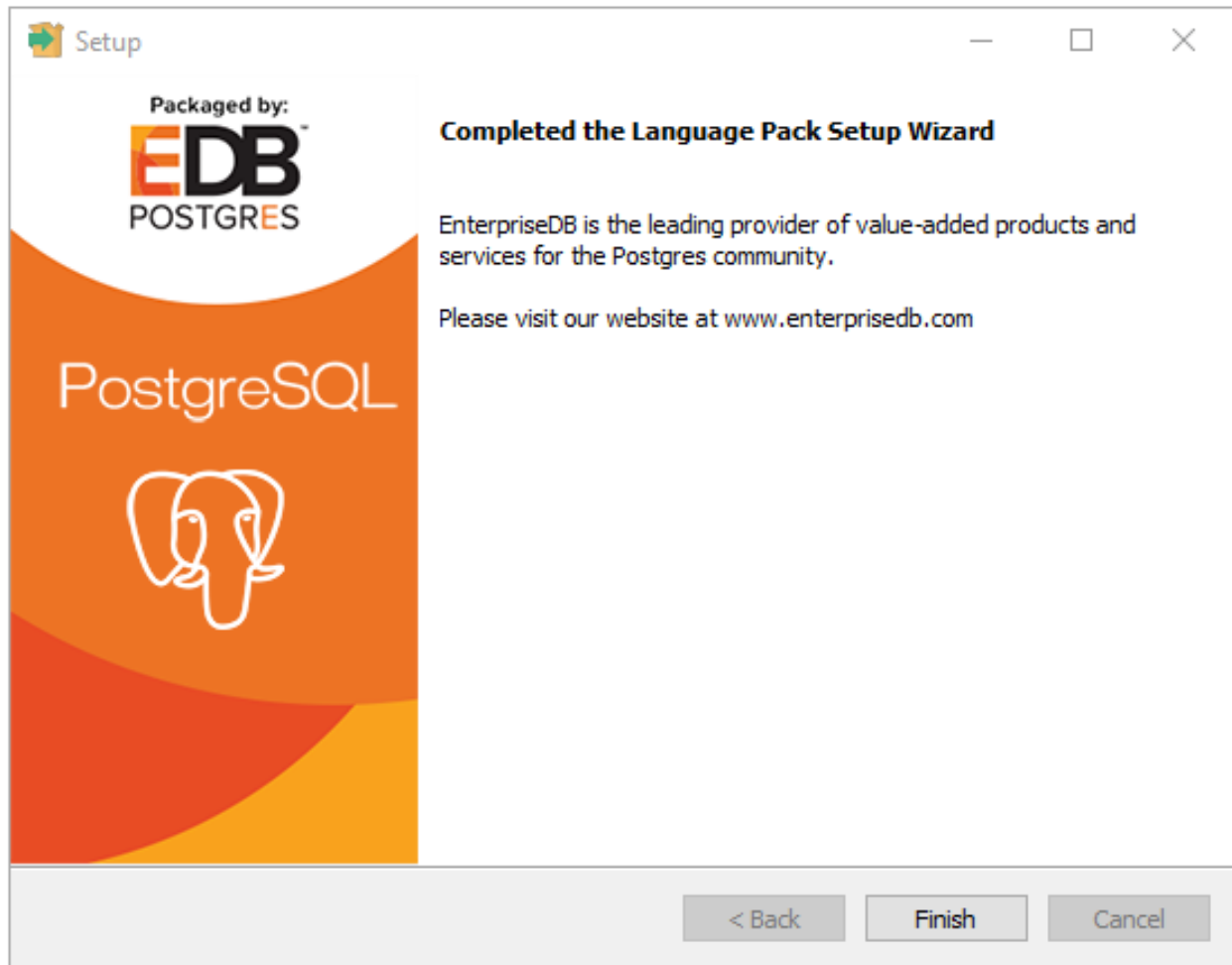


Fig. 2.4: *The Language Pack Setup Complete dialog*

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

2.2 Installing Language Pack with StackBuilder Plus

You can use StackBuilder Plus to download and invoke the Language Pack graphical installer. To open StackBuilder Plus, select the StackBuilder Plus menu item from the version-specific EDB Postgres sub-menu.

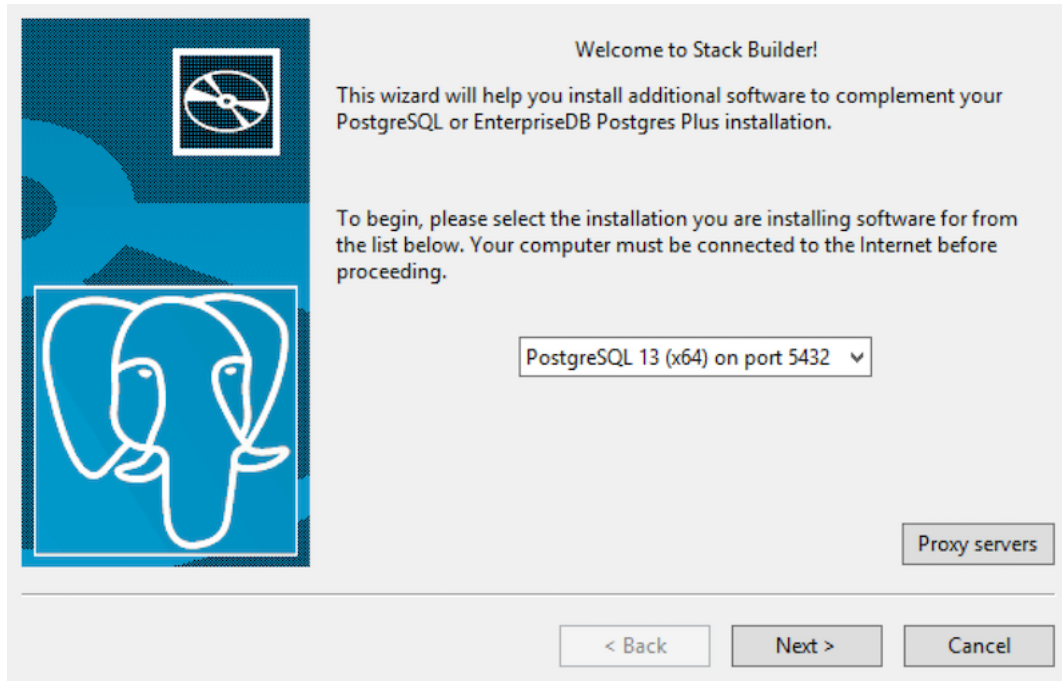


Fig. 2.5: *The StackBuilder Plus Welcome Window*

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

Expand the `Add-ons, tools and utilities` node of the `Categories` tree control, and check the box to the left of `EDB Language Pack`; click Next to continue.

StackBuilder Plus will confirm your package selection before downloading the installer. When the download completes, StackBuilder Plus will offer to invoke the installer for you, or will delay the installation until a more convenient time.

For details about using the graphical installer, see *Invoking the Graphical Installer*.

2.3 Configuring Language Pack on an Advanced Server Host

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

2.3.1 Configuring Language Pack on Windows

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

```
C:\edb\languagepack\v1
```

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

```
set PYTHONHOME=C:\edb\languagepack\v1\Python-3.7
```

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

```
set PATH=C:\edb\languagepack\v1\Python-3.7;
C:\edb\languagepack\v1\Perl-5.26\bin;
C:\edb\languagepack\v1\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 database server.

2.3.2 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 tell the Python interpreter where to find Python:

```
set PYTHONHOME=C:\edb\languagepack\v1\Python-3.7
```

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

```
set PATH=C:\edb\languagepack\v1\Python-3.7;
C:\edb\languagepack\v1\Perl-5.26\bin;
C:\edb\languagepack\v1\Tcl-8.6\bin;%PATH%
```

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

2.3.3 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/v1/Perl-5.26
PYTHONHOME=/Library/edb/languagepack/v1/Python-3.7
TCLHOME=/Library/edb/languagepack/v1/Tcl-8.6
```

Then, 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, use the variables to tell 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 Advanced Server installation. For more information about using an RPM package, please see the EDB Advanced Server Installation Guide, available at:

<https://www.enterprisedb.com/edb-docs>

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 Postgres documentation available at:

<https://www.postgresql.org/docs/13/static/plperl.html>

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 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.

Please 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 available at:

<https://www.postgresql.org/docs/13/static/plpython.html>

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 $$
  if {[argisnull 1]} {
    if {[argisnull 2]} { return_null }
    return $2
  }
  if {[argisnull 2]} { return $1 }
  if {$1 > $2} {return $1}
  return $2
$$ 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 Postgres documentation available at:

<https://www.postgresql.org/docs/13/static/pltcl.html>

CHAPTER 4

Conclusion

EDB Postgres Language Pack Guide

Copyright © 2013 - 2020 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

C

Conclusion, [16](#)

Configuring Language Pack on a PostgreSQL Host,
[10](#)

Configuring Language Pack on an Advanced Server
Host, [9](#)

Configuring Language Pack on Windows, [10](#)

I

Installing Language Pack, [4](#)

Installing Language Pack with StackBuilder Plus, [8](#)

Invoking the Graphical Installer, [4](#)

P

PL/Perl, [12](#)

PL/Python, [13](#)

PL/Tcl, [14](#)

S

Supported Database Server Versions, [2](#)

U

Using the Procedural Languages, [12](#)