

# PSQL v12

---

## *What's New in PSQL v12 SP1* *Release 12.10*

**An Overview of New Features and Changed Behavior**



## *disclaimer*

ACTIAN CORPORATION LICENSES THE SOFTWARE AND DOCUMENTATION PRODUCT TO YOU OR YOUR COMPANY SOLELY ON AN “AS IS” BASIS AND SOLELY IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THE ACCOMPANYING LICENSE AGREEMENT. ACTIAN CORPORATION MAKES NO OTHER WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THE SOFTWARE OR THE CONTENT OF THE DOCUMENTATION; ACTIAN CORPORATION HEREBY EXPRESSLY STATES AND YOU OR YOUR COMPANY ACKNOWLEDGES THAT ACTIAN CORPORATION DOES NOT MAKE ANY WARRANTIES, INCLUDING, FOR EXAMPLE, WITH RESPECT TO MERCHANTABILITY, TITLE, OR FITNESS FOR ANY PARTICULAR PURPOSE OR ARISING FROM COURSE OF DEALING OR USAGE OF TRADE, AMONG OTHERS.

## *trademarks*

Btrieve, Client/Server in a Box, and Pervasive are registered trademarks of Actian Corporation. Built on Pervasive Software, DataExchange, MicroKernel Database Engine, MicroKernel Database Architecture, Pervasive.SQL, Pervasive PSQL, Solution Network, Ultralight, and ZDBA are trademarks of Actian Corporation.

Apple, Macintosh, Mac, and OS X are registered trademarks of Apple Inc.

Microsoft, MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows Millennium, Windows 2000, Windows 2003, Windows 2008, Windows 7, Windows 8, Windows 10, Windows Server 2003, Windows Server 2008, Windows Server 2012, Windows XP, Win32, Win32s, and Visual Basic are registered trademarks of Microsoft Corporation.

NetWare and Novell are registered trademarks of Novell, Inc. NetWare Loadable Module, NLM, Novell DOS, Transaction Tracking System, and TTS are trademarks of Novell, Inc.

Oracle, Java, all trademarks and logos that contain Oracle, or Java, are trademarks or registered trademarks of Oracle Corporation.

All other company and product names are the trademarks or registered trademarks of their respective companies.

© Copyright 2016 Actian Corporation. All rights reserved. Reproduction, photocopying, or transmittal of this publication, or portions of this publication, is prohibited without the express prior written consent of the publisher.

This product includes software developed by Powerdog Industries. © Copyright 1994 Powerdog Industries.

All rights reserved. This product includes software developed by KeyWorks Software. © Copyright 2002

KeyWorks Software. All rights reserved. This product includes software developed by DUNDAS

SOFTWARE. © Copyright 1997-2000 DUNDAS SOFTWARE LTD., all rights reserved. This product

includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product uses the free unixODBC Driver Manager as written by Peter Harvey

([pharvey@codebydesign.com](mailto:pharvey@codebydesign.com)), modified and extended by Nick Gorham ([nick@easysoft.com](mailto:nick@easysoft.com)), with local

modifications from Actian Corporation. Actian Corporation will donate their code changes to the current

maintainer of the unixODBC Driver Manager project, in accordance with the LGPL license agreement of

this project. The unixODBC Driver Manager home page is located at [www.unixodbc.org](http://www.unixodbc.org). For further

information on this project, contact its current maintainer: Nick Gorham ([nick@easysoft.com](mailto:nick@easysoft.com)).

A copy of the GNU Lesser General Public License (LGPL) is included on the distribution media for this

product. You may also view the LGPL at [www.fsf.org/licenses/lgpl.html](http://www.fsf.org/licenses/lgpl.html).

**What's New in PSQL v12 SP1  
June 2016**



# Contents

<b>About This Document</b> . . . . .	<b>vii</b>
<b>1 What's New in PSQL v12 SP1</b> . . . . .	<b>1</b>
<i>An Overview of New and Changed Features</i>	
Database Defragmentation . . . . .	2
Cancel All . . . . .	2
Automatic Defragmentation . . . . .	2
Disk Space Management . . . . .	2
Defragmenting Multiple Files from the Command Line . . . . .	2
Defragmentation with Open File Handles . . . . .	2
Unicode Support for Globalization . . . . .	3
NCHAR Support in PCC . . . . .	3
Collation Capabilities . . . . .	3
SDK Access Methods . . . . .	4
ADO.NET . . . . .	4
JDBC and JCL . . . . .	4
PDAC . . . . .	4
<b>A What Was New in PSQL v12</b> . . . . .	<b>5</b>
<i>An Overview of New and Changed Features</i>	
Database Defragmentation . . . . .	6
Using Defragmenter . . . . .	6
Limitations . . . . .	7
Unicode Support for Globalization . . . . .	8
The Unicode Standard . . . . .	8
Unicode Encodings . . . . .	8
Declaring Encodings . . . . .	8
PSQL Support for Unicode Features . . . . .	9
New Wide Character Data Types . . . . .	10
Restrictions on Unicode Support . . . . .	10
Unicode Support in Utilities . . . . .	11
Bulk Data Utility (BDU) . . . . .	11
PSQL Control Center (PCC) . . . . .	11
Query Plan Viewer . . . . .	12
Installation . . . . .	13
Rebranding . . . . .	13
PSQL Workgroup Run as a Service by Default . . . . .	13
Windows Operating System Detection in Installation Executables . . . . .	13
JRE Version . . . . .	14
Custom Installation on Windows . . . . .	14
Silent Installation on Windows . . . . .	14
Installation on Linux and OS X . . . . .	15
64-bit Installation Packages . . . . .	15
Improved Performance on Linux and OS X . . . . .	15
Support for 32-bit Applications on 64-bit Linux . . . . .	16
Samba Support . . . . .	16
Licensing . . . . .	17
Live Migration Capability in All Editions of PSQL . . . . .	17

License Enforcement . . . . .	17
Internet Connectivity Requirements for Vx. . . . .	17
Multiplexed Applications . . . . .	17
Concurrent Sessions. . . . .	17
SDK Access Methods. . . . .	18
ADO.NET . . . . .	18
JDBC . . . . .	18
ODBC . . . . .	18
ODBC Driver for Applications with Wide Character Data . . . . .	18
New ODBC Guide and Release Notes. . . . .	19
PDAC. . . . .	19
Wait Lock Timeouts . . . . .	20
Discontinued Features . . . . .	21
Unsupported Features . . . . .	22
Windows Features Not Supported by PSQL . . . . .	22
Windows To Go . . . . .	22
OneDrive . . . . .	22
PSQL Features Not Supported by Windows . . . . .	22



# *About This Document*

---

This document contains information about the features and enhancements in this release of PSQL. It does not provide comprehensive usage instructions for the software. Its purpose is to summarize what is new and different from the previous release.

The PSQL v12 family of products includes three editions of the database engine:

- PSQL Server
- PSQL Vx Server
- PSQL Workgroup

Throughout the PSQL documentation, all content covers all three editions unless differences are explicitly called out. For example, you will occasionally encounter topics that apply only to PSQL Workgroup. Unless stated otherwise, anything that applies to PSQL Server also applies to PSQL Vx Server.

Actian Corporation would appreciate your comments and suggestions about this document. Your feedback can determine what we write about the use of our products and how we deliver information to you. Please post your feedback in the [Community Forum](#) on the [PSQL website](#).





# *What's New in PSQL v12 SP1*

---

## *An Overview of New and Changed Features*

This General Release of PSQL v12 SP1, version 12.10, includes new features and changes covered in the following topics:

- [Database Defragmentation](#)
  - [Cancel All](#)
  - [Automatic Defragmentation](#)
  - [Disk Space Management](#)
  - [Defragmenting Multiple Files from the Command Line](#)
  - [Defragmentation with Open File Handles](#)
- [Unicode Support for Globalization](#)
  - [NCHAR Support in PCC](#)
  - [Collation Capabilities](#)
- [SDK Access Methods](#)
  - [ADO.NET](#)
  - [JDBC and JCL](#)
  - [PDAC](#)

For information about what was included in the release of PSQL v12, version 12.00, see [What Was New in PSQL v12](#).

This document may be updated after the product release. You can download the latest version from the [PSQL website](#).

## Database Defragmentation

The database defragmentation capability introduced in PSQL v12 has the following new features and enhancements in SP1:

- [Cancel All](#)
- [Automatic Defragmentation](#)
- [Disk Space Management](#)
- [Defragmenting Multiple Files from the Command Line](#)
- [Defragmentation with Open File Handles](#)

### **Cancel All**

From both the Defragmenter window and the `dbdefrag` command line, you can now use `Cancel All` to stop all analysis and defragmentation activity.

### **Automatic Defragmentation**

Automatic defragmentation is a new feature that provides unattended defragmentation of recently opened files. Automatic defragmentation waits one hour after the last defragmentation and then checks for any file that matches the following criteria:

- Not already defragmented in the past 24 hours
- 10 MB or larger
- Fragmented 15% or more
- Containing 15% or more unused space
- Containing 5% or more records out of order

For details, see [Automatic Defragmentation](#). For information about data defragmentation in general, see [Monitoring Data File Fragmentation](#).

### **Disk Space Management**

During defragmentation, the engine monitors available disk space. If your application's need for disk space rises substantially while files are being defragmented and competition for space poses a risk to your application execution, the engine automatically cancels defragmentation requests to release resources.

### **Defragmenting Multiple Files from the Command Line**

The `dbdefrag` command line utility now accepts multiple file names.

### **Defragmentation with Open File Handles**

In PSQL v12, if a file were opened while it was undergoing defragmentation, additional defragmentation requests would be denied until that open file handle closed. In 12.10, additional defragmenting is no longer blocked.

---

## Unicode Support for Globalization

PSQL v12 SP1 includes additional support for globalization as explained in the following topics. The *Advanced Operations Guide* covers these features under [Database Globalization](#).

- [NCHAR Support in PCC](#)
- [Collation Capabilities](#)

### ***NCHAR Support in PCC***

PSQL Control Center (PCC) now supports editing of view, procedure, and trigger body text and default column values that contain NCHAR text outside of the database code page.

### ***Collation Capabilities***

At the Btrieve level, PSQL v12 SP1 supports two new Unicode collations based on International Components for Unicode (ICU) for use with Btrieve STRING and WSTRING types. The STRING types are assumed to be encoded in UTF-8. The collation called root is the default Unicode collation defined by ICU. The collation called u54-msft\_enus\_0 emulates the PSQL ISR collation MSFT\_ENUS01252\_0 for CP1252 text when applied to UTF-8 or UTF-16 data.

For more information see [Collation Support Using an ICU Unicode Collation](#).

## SDK Access Methods

PSQL v12 SP1 includes additions or enhancements to the following SDK access methods:

- [ADO.NET](#)
- [JDBC and JCL](#)
- [PDAC](#)

### **ADO.NET**

The ADO.NET 4.2 SDK has been updated to support Microsoft Visual Studio 2015.

### **JDBC and JCL**

PSQL v12 SP1 now provides 64-bit JDBC and JCL access methods on Microsoft Windows. The new JDBC 64-bit support DLL is w64pvjdbc2.dll. The new JCL 64-bit support DLL is w64jnibtrv.dll.

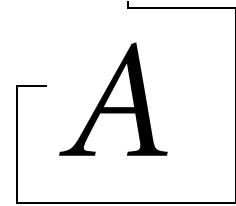
### **PDAC**

PSQL now provides 32-bit design time packages and 32-bit and 64-bit PDAC runtime packages for RAD Studio XE4, XE5, XE6, XE7, XE8, and XE10.

New compile time PDAC binaries for RAD Studio XE4, XE5, XE6, XE7, XE8, and XE10 are available in the PDAC SDK web download package, and new runtime binaries are installed as part of the PDAC module for PSQL Server, Vx Server, Workgroup, and Client.

See [Using Direct Access Components](#) in *PSQL Direct Access Components Guide*.

# *What Was New in PSQL v12*



---

## *An Overview of New and Changed Features*

The General Release of PSQL v12, version 12.00, included new features and changes covered in the following topics:

- [Database Defragmentation](#)
- [Unicode Support for Globalization](#)
- [Installation](#)
- [Custom Installation on Windows](#)
- [Silent Installation on Windows](#)
- [Installation on Linux and OS X](#)
- [Licensing](#)
- [SDK Access Methods](#)
- [Wait Lock Timeouts](#)
- [Discontinued Features](#)
- [Unsupported Features](#)

For information about what is included in the release of PSQL v12 SP1, version 12.10, see [What's New in PSQL v12 SP1](#).

This document may be updated after the product release. You can download the latest version from the [PSQL website](#).

## Database Defragmentation

Over time in a busy database, as records are repeatedly created, updated, or deleted, data can become fragmented, lengthening times for file access and transaction response. This fragmentation differs from file system fragmentation on a hard disk because it occurs within the data file itself. As a developer or DBA, you may know when a file is likely to fragment from heavy use, but in some systems, you may be guessing.

Defragmenter is a new feature that helps you solve this problem by finding and correcting data fragmentation. It rearranges records and rebuilds indexes in data files and removes unused space so that data can be efficiently accessed again. Defragmentation does not alter data in any way, and records can be created, read, updated, or deleted while their files are being defragmented. You can use Defragmenter features during database engine execution with no need for down time or disruption of business operations in most cases.

This section provides the following topics:

- [Using Defragmenter](#)
- [Limitations](#)

### Using Defragmenter

Opened from the Start menu or as a tool in PSQL Control Center, Defragmenter shows data files in use, including their number of reads and writes so that you can quickly find highly used ones. To add files or tables to the watch list, select them and then drag-and-drop, click a button, or right-click and choose the command. If you know of Btrieve files in other locations, you can browse to them and add them to be watched as well.

In the watch list, you can select one or more items to analyze. Analysis gives the following insights:

Statistic	Measures	Significance
File size	File size in megabytes	The longer the file has been in use and the larger the file, the greater its fragmentation could be.  Smaller file sizes typically take less time to defragment unless they have a large number of indexes.
% Fragmented	Percentage of data divided into small blocks that are not contiguous	A lower percentage indicates fewer, larger data blocks stored closer together, allowing more rapid reads and writes.
% Unused	Percentage of unused space	The lower the percentage, the more compact the file, allowing more rapid reads and writes.  Unused space is often created by update and delete operations.
% Not in Order	Percentage of records not stored sequentially	A lower percentage generally gives higher performance for actions such as table scans on large files.  Inserts over time often result in records stored in nonsequential order.

High values for these statistics can explain loss of performance. By defragmenting a file, you can reduce all four of these numbers. Transactions generally run more quickly against the newly compacted, reordered reindexed file, restoring efficiency, capacity, and performance.

You can also run this feature as the dbdefrag command line utility.

### **Limitations**

- Database and table schema definitions cannot be altered during defragmentation.
- Backup Agent operations and continuous operations for data backup cannot be performed on a file that is undergoing defragmentation.
- If you are using a client cache engine, and it has already opened a file on the server, then server defragmentation is no longer allowed and requests to defragment return an error. To allow defragmenting, you must restart the server to clear the client connection.
- Defragmentation is not currently supported for server engines in environments that use Microsoft Volume Shadow Copy Service (VSS) for backup operations.

## Unicode Support for Globalization

Globalization, as used here, means adapting computer software to different languages. It is now commonplace that users around the globe access the same data and that applications present the data in the user's own language. PSQL support for globalization allows your application to store text in multiple languages in the same database. This capability enables your application to store, process, and retrieve data in whatever language is required.

PSQL v12 includes additional support for globalization as explained in the following topics. *Advanced Operations Guide* also provides a new topic called Database Globalization.

- [The Unicode Standard](#)
- [Declaring Encodings](#)
- [PSQL Support for Unicode Features](#)
- [Unicode Support in Utilities](#)

### **The Unicode Standard**

The Unicode standard defines a character set that contains every character used in spoken languages in the world (see [www.unicode.org](http://www.unicode.org)). Unicode also expands the concept of a character set by defining additional annotation information to specify letter spacing, right-to-left behavior, word and line breaks, and so forth. This allows applications to properly display and manipulate Unicode text. Applications, and the database, also need this additional information for such actions as case conversion and sorting.

PSQL recognizes the Unicode character set and provides support for character data storage and retrieval in the languages required by the application.

### **Unicode Encodings**

In the Unicode character set, each character is assigned a unique value called a code point. That code point value is then encoded for storage. The code points are organized into planes. Each plane can contain 65536 code points. The first plane, plane 0, is named the Basic Multilingual Plane (BMP) and contains the majority of the code points currently defined. The Unicode standard has several methods of encoding the code points. Two that are commonly used are UTF-8 and UCS-2. UTF-8 encodes character code point values to a byte string using one to four bytes per character. UCS-2 encodes character code point values using 16-bit values, often referred to as wide characters.

PSQL recognizes the Basic Multilingual Plane code points and is compatible with applications that use the Unicode encodings UTF-8 for byte strings and UCS-2 for wide character strings. The binary unit for UTF-8 is 8-bit. The binary unit for UCS-2 is 16-bit, that is, wide character.

### **Declaring Encodings**

The database code page is a PSQL database property that declares the encoding of character data stored in the database. Its purpose is to help ensure that character data can be interpreted correctly. However, the database code page property is just a declaration. PSQL does not validate the encoding of the data and metadata that an application inserts into a database. The application is responsible for ensuring that character data is stored and retrieved in a particular encoding. Note that the database code page only applies to text encoded with legacy code pages or UTF-8. Wide character text is always encoded using UCS-2. A proper setting is required if the engine will be converting between wide character text and



byte-string text, such as with CAST or ALTER TABLE. The default value of the database code page is the system code page of the operating system where the engine is running.

The PSQL SQL access methods infer a client code page for byte strings exchanged between the application and the access method. (Wide character strings are always encoded with UCS-2.) On Windows, the access method assumes that the application is respecting the Active Code Page (ACP) for byte strings. On Linux and OS X, the access method assumes that the application is respecting the encoding of the locale, which is usually given by the LANG environment variable.

PSQL provides methods to ensure compatible encoding between the database engine and clients. For example, an application can specify that it wants the PSQL SQL client to translate data automatically between the database code page and the client application. This is referred to as automatic translation.

Note, however, that automatic translation can translate characters only if they are present in the character sets of both code pages: the code page on the server machine and the code page on the client machine.

For backward compatibility, automatic translation in access methods is turned off by default. The application must configure the access method to enable automatic translation. When possible, the recommended method is to set the database code page and configure the access method to read and use that value.

## PSQL Support for Unicode Features

A previous PSQL release already provided UTF-8 and UCS-2 string storage for Btrieve applications and UTF-8 string storage for SQL applications. This section summarizes PSQL v12 Unicode features, new data types, and restrictions.

Table 1 Unicode Features in PSQL v12

Feature	Discussion
New Relational data types NCHAR, NVARCHAR, and NLONGVARCHAR for wide character data.  These data types are available only for the PSQL ADO.NET Data Provider, ODBC, and JDBC access methods.	See <a href="#">New Wide Character Data Types</a> .  Text is encoded as UCS-2.
New ODBC driver for wide character applications	See <a href="#">ODBC Driver for Applications with Wide Character Data</a> .
SQL access to the Btrieve data types WSTRING and WZSTRING in DDF Builder	See <a href="#">Table Definition Editor</a> in <i>DDF Builder User's Guide</i> .
NCHAR literals in SQL queries (a string literal preceded by an N, as in N'Måløy')	A wide character SQL query can use string-literal content that does not map to the byte-string encoding used for the client, server, or database.
Support for Unicode in PSQL utilities	See <a href="#">Bulk Data Utility (BDU)</a> , <a href="#">PSQL Control Center (PCC)</a> , and <a href="#">Query Plan Viewer</a> .
New APIs	APIs have been added to the ODBC and JDBC access methods to offer some support for Unicode.

## New Wide Character Data Types

PSQL v12 provides the new relational data types NCHAR, NVARCHAR, and NLONGVARCHAR for use with wide character data.

Table 2 New Relational Data Types

Transactional Type	Relational Type	PSQL Metadata Type Code	Size (bytes)	Create and Add Parameters <sup>1</sup>	Data Type Notes
WSTRING	NCHAR	25	2–8000	size (1–4000) not null	Padded with spaces
WZSTRING	NVARCHAR	26	2–8000	size (1–4000) not null	Not padded
none	NLONGVARCHAR	21	N/A	not null	Not padded. Cannot be indexed. Flag in X\$Field set for SQL to use wide character. See also COLUMNMAP flags in DTI guide and column flags in DTO guide.

<sup>1</sup> The required parameters are precision and size. The optional parameters are case insensitive, not null, and scale.

## Restrictions on Unicode Support

The following table covers restrictions when using Unicode features.

Table 3 PSQL v12 Restrictions on Unicode Support

Topic	Discussion
Size of index key	The maximum size of an index key is 128 UCS-2 characters. That is, key size remains 256 bytes as with the previous version of PSQL.
Client/Server connections	A v11 client may connect to a v12 server and request data stored in wide character columns, but if the server complies, the client will fail to handle the data successfully. The solution is to upgrade the client to v12.
ODBC Unicode driver	Will not connect to a v11 server.
ODBC ANSI driver	The Microsoft Driver Manager converts wide character data to the ANSI code page of the application before passing it to the PSQL driver. Consequently, Unicode characters outside of the ANSI code page cannot be used with this driver.
SQL grammar	The CASE collation clause on wide character columns is not supported.
Object name metadata	Metadata for database object names is restricted to the database code page.
SQL access methods and wide character data	The only supported SQL access methods for wide character data are ODBC, JDBC, and ADO.NET.
Collations and sorting	Only code point order is supported.
PCC SQL Editor	In v12, you cannot edit views, stored procedures, user-defined functions, or triggers that contain characters outside of the database code page. You must first drop them and then reenter them from an edited source file.

## Unicode Support in Utilities

As part of expanded globalization support, PSQL utilities provide new and revised functionality in the Bulk Data Utility (BDU), PSQL Control Center (PCC), and Query Plan Viewer.

### Bulk Data Utility (BDU)

The Bulk Data Utility (BDU) is a command line utility that allows you to load data from a delimited text file into a PSQL table. A new command line parameter, `-c encoding`, has been added to specify the data encoding to use when loading the data file.

The encoding options are UTF-8, UTF-16LE, and UTF-16BE. If a data file contains a byte order mark (BOM), BDU uses the encoding specified by the BOM. That is, if a data file uses a BOM to indicate an encoding of UTF-8, UTF-16LE, or UTF-16BE, then BDU uses that encoding regardless of the value you enter for the encoding parameter on the command line. Without a BOM or the `-c` parameter, BDU defaults to using the system code page.

In addition, the parameter `-q text-qualifier` has been deprecated. References to it have been removed from the command line usage help for BDU.

See [bdu](#) in *PSQL User's Guide*.

### PSQL Control Center (PCC)

To support wide character data and data types, PCC includes changes to the default value of the connection encoding property, editor and view capabilities, and new dialogs for opening and saving SQL document files.

#### Default Connection Encoding Value

For new databases, PCC defaults to a connection encoding setting of Automatic. The value Automatic means that PCC uses the database code page setting for the connection encoding. When existing databases are upgraded to PSQL v12, PCC retains their previous connection encoding setting. If you are globalizing an existing database and want to get the full benefit of wide character support, declare the database code page for your database and then in PCC set connection encoding to Automatic.

#### Support for Wide Character Types and Data in Editors and Views

PCC now supports wide character data and data types in its editors and views. The data types are NCHAR, NVARCHAR, and NLONGVARCHAR. For details, see these *PSQL User's Guide* topics:

- SQL Editor - Highlights NCHAR types as keywords.
- Table Editor - Includes NCHAR types for column type.
- Grid Window View - Allows editing of data in NCHAR columns.
- Text Window View - The default font is now Consolas. If that is not available, PCC tries to use Lucida Console, Andale Mono, Courier New, or Courier. These fonts can display a wide variety of Unicode characters. Fixed-width fonts in Text Window View enable better text alignment in columns.
- Outline Window View - Can display wide character data from NCHAR string literals.

#### Dialogs for Opening and Saving Files

The PCC dialogs for opening and saving SQL documents, saving exported schemas, and importing and exporting table data have all been enhanced to accommodate a variety of file encodings. Previously, these files were presumed to be in the default system code page. It is now possible to select a number of

Unicode encodings when saving files. When opening a file, the new dialogs detect whether the file uses a BOM (Byte Order Mark) identifying the Unicode encoding. The opening dialogs also allow you to set the expected encoding of the file. For your convenience, a new PCC setting controls the default encoding used in these dialogs.

For more information on these new features, see *PSQL User's Guide* under the topics [Dialogs for File Open and File Save](#), [Wide Character Data Support for Import Data, Export Data, and Export Schema](#), and [File Encoding Preferences](#).

## Query Plan Viewer

Query Plan Viewer is now compatible with wide character data. Query plan files created with versions of PSQL before v12 are still compatible.

For your reference, a query plan file contains the code page identifier of the encoding used for *each* SQL query in the file. (A query plan file can contain plans resulting from multiple queries.) Query Plan Viewer correctly displays wide character data of NCHAR string literals regardless of which database encoding a query used.

See [Query Plan Viewer](#) in *SQL Engine Reference* and [Database Code Page and Client Encoding](#) in *Advanced Operations Guide*.

---

## Installation

The installation of PSQL has the following changes.

- [Rebranding](#)
- [PSQL Workgroup Run as a Service by Default](#)
- [Windows Operating System Detection in Installation Executables](#)
- [JRE Version](#)
- [Custom Installation on Windows](#)
- [Silent Installation on Windows](#)
- [Installation on Linux and OS X](#)

New features in this release are identical on Windows, Linux, and OS X operating systems.

### **Rebranding**

Many display names have been changed from *Pervasive* to *Actian*. The display names for services on Windows have been changed, but the underlying service name remains the same. Scripts that manage services using the display name will require modification, while those that use the service name will continue to work as they are. For example, the display name for the Workgroup engine service has changed to *Actian PSQL Workgroup Engine*, but the system name for the service remains *psqlWGE*.

Access method driver names, such as for ODBC and JDBC, have not changed.

### **PSQL Workgroup Run as a Service by Default**

In previous releases, a new PSQL Workgroup Engine was installed to run as an application. In this release, by default a new engine is installed to run as a service.

For an upgrade, PSQL Workgroup is installed to run in the same manner as the previously installed version. For example, if the previous installation was set to run as an application, the upgrade installation runs that way as well.

During installation, select **Run as a Service** if either of the following is true:

- You want to avoid User Access Control conflicts from the operating system. Running the database engine as a service requires the Log On as Service privilege. If you run the engine as a service under a user account other than the default Local System account, be sure to modify the Log On Properties for the Service using the Windows Control Panel.
- You want to run Workgroup Engine on a machine using the Terminal Services environment.

Alternatively, select **Run as an Application** if data files reside only on a remote machine, are accessed via drive mappings, and no database engine is running on the remote machine.

See [Installing PSQL Workgroup for Windows](#) in *Getting Started with PSQL* for the installation steps. The customizable installation settings for PSQL Workgroup are located in the PTKSetup.ini file. See [Customizing PSQL Installations](#) in *Installation Toolkit Handbook*.

### **Windows Operating System Detection in Installation Executables**

The Windows installation executables for PSQL have been repackaged to detect the bit architecture of the target operating system. This allows you to select a single package that automatically provides the

components for your operating system. In particular, the installers provide all 32- and 64-bit client components appropriate to your system. It is no longer necessary to apply a separate 64-bit client package to 32-bit engines.

The following table describes PSQL v12 installers on Windows operating systems.

PSQL v12 Installation Package (.exe)	Description
Install_PSQL_Server_Engine	Installs 32-bit engine on 32-bit operating system. Installs 64-bit engine on 64-bit operating system. Installs all client components.
Install_PSQL_Vx_Server	Installs 32-bit engine on 32-bit operating system. Installs 64-bit engine on 64-bit operating system. Installs all client components.
Install_PSQL_Workgroup_Engine	Installs 32-bit engine on both 32- and 64-bit operating systems. Installs all client components.
Install_PSQL_Client	Installs 32-bit components on 32-bit operating system. Installs both 32- and 64-bit components on 64-bit system.

### **JRE Version**

The version of the Java Runtime Environment that is installed to support PSQL utilities has been updated from version 6 to version 7. The JRE is installed if the documentation or utilities features are selected in the installer.

### **Custom Installation on Windows**

If you create a custom installation package for PSQL, the required and optional CAB files have changed to support installation packages that are more aware of the bit architecture of the operating system:

- Files \_C32\_64b.cab and \_DRM64.cab have been added to the list of required CAB files.
- Files \_CE32.cab, \_Core.cab, \_PSQL32.cab, and \_SrvrEng.cab are no longer available as required CAB files.
- File PDAC64.cab has been added to the list of optional CAB files. It contains the 64-bit PDAC components.
- File DocsVx.cab is no longer available as an optional CAB file. Docs.cab includes contents formerly in DocsVx.cab.
- File JRE6.cab has been replaced by the JRE.cab file in the list of optional CAB files.

See [Changing Installation Package Size and Features Using CAB Files](#) in *Installation Toolkit Handbook*.

### **Silent Installation on Windows**

In PSQL v12, we now recommend that silent installation use one of the installer executables, named Install\_<product>.exe. If instead you need to use the PSQL .msi file for installation, you must meet additional requirements in the *Installation Toolkit Handbook*.

## Installation on Linux and OS X

PSQL v12 provides the same set of new features for Linux and OS X as the Windows release, but with a few key points unique to Linux and OS X, as described in the following topics:

- [64-bit Installation Packages](#)
- [Improved Performance on Linux and OS X](#)
- [Support for 32-bit Applications on 64-bit Linux](#)
- [Samba Support](#)

### 64-bit Installation Packages

PSQL v12 now consists of pure 64-bit packages for 64-bit Linux and OS X, enabling PSQL Server, Vx Server, and Client to run after installation with no need for additional software. On Linux and OS X, all PSQL command line utilities and PSQL Control Center are now 64-bit.

PSQL v12 offers 64-bit full and client installations as TAR packages for Linux and OS X, as RPM packages for Linux, and as DMG packages for OS X. A full installation includes the engine and client files, utilities, and the complete user documentation. Client installations provide only files needed to support a client, including utilities and documentation. If you have 32-bit applications, support is still provided as an additional client access installation. The following table lists the installation packages.

PSQL v12 Package	Installation Type	Contains
Server 64-bit	Full	64-bit engine, 64-bit client files, 64-bit utilities, and documentation.
Server Vx 64-bit	Full	64-bit engine, 64-bit client files, 64-bit utilities, and documentation.
Client 64-bit	Client	64-bit client files, 64-bit utilities, and documentation.
Client Access 32-bit (Linux only)	Client	<p>32-bit client files for compatibility with your 32-bit applications on 64-bit Linux. Installed after one of the other three packages.</p> <p><b>Note:</b> Glibc and stdc++ packages are prerequisites for the Client Access installation. Before installing Client Access, you can add these packages from the repository of your Linux distribution using a yum command:</p> <pre>yum install glibc.i686 libstdc++.i686</pre> <p>In LD_LIBRARY_PATH, add the /lib directory of the Client Access installation location so that your 32-bit application can find it. For example, /usr/local/psql/lib64:/usr/local/psql/lib.</p>

For installation steps, see [Installing PSQL Server, Vx Server, and Client for Linux and OS X](#) in *Getting Started with PSQL*.

### Improved Performance on Linux and OS X

PSQL v12 on Linux and OS X now provides PSQL users with database access methods for 64-bit JDBC and JCL, supporting SQL and Btrieve, respectively. The new support gives applications full integration with 64-bit Java capabilities such as improved performance and memory management.

## Support for 32-bit Applications on 64-bit Linux

If you plan to run 32-bit applications on 64-bit Linux, in PSQL v12 you need to do only three things:

- 1 On the system where your 32-bit applications run, check for prerequisite packages `glibc.i686` and `stdc++.i686`, and if not found, use `yum` to install them from your Linux distribution repository.
- 2 Install the Client Access 32-bit package.
- 3 In `LD_LIBRARY_PATH`, add the `/lib` directory of the Client Access installation location so that your 32-bit application can find it. For example, `/usr/local/psql/lib64:/usr/local/psql/lib`.

With the release of this 32-bit support on 64-bit systems, PSQL has discontinued its 32-bit Server and Vx Server editions.



---

**Note** The 32-bit Client Access installation is not supported on OS X.

---

## Samba Support

A Samba share is no longer automatically provided in PSQL v12 server installations on Linux or OS X. For information about using Samba shares for Btrieve file access across a network, see [Connecting from a Linux or OS X Client to a Windows Server or Vx Server](#) and [Supported Path Formats for Samba](#) in *Getting Started with PSQL*.



---

## Licensing

PSQL v12 includes the following changes to licensing.

### ***Live Migration Capability in All Editions of PSQL***

In PSQL v12, the live migration capability has been expanded to the Workgroup and Server editions, delivering greater mobility in cloud and virtualized environments.

VM live migration is now available in all PSQL editions.

### ***License Enforcement***

Product authorization now uses simply the machine name for all PSQL editions. This allows for greater hardware and configuration changes without invalidating the PSQL license and also aligns with evolving industry standards to reduce failed validations.

### ***Internet Connectivity Requirements for Vx***

The PSQL v12 Vx Server edition no longer requires Internet connectivity to maintain a valid product key after the product is authorized. This change allows you to authorize your Vx Server once and then operate within a secure environment, such as behind a firewall.

### ***Multiplexed Applications***

PSQL v12 licensing requires that multiplexed applications use Vx Server. Vx Server uses capacity-based licensing rather than measuring concurrent users. Multiplexing includes using hardware or software to pool connections, reroute information, and reduce the number of connections to the database.

### ***Concurrent Sessions***

PSQL Vx Server v12 licensing now counts only open files and no longer measures sessions. This change simplifies determining capacity requirements for PSQL Vx Server.

## SDK Access Methods

PSQL v12 includes additions or enhancements to the following SDK access methods:

- [ADO.NET](#)
- [JDBC](#)
- [ODBC](#)
- [PDAC](#)

### **ADO.NET**

Release 4.2 of the PSQL ADO.NET Data Providers is included with PSQL v12. Release 4.2 supports code first and model first, enumerated data types, and code first migrations, as well as PSQL data types NCHAR, NVARCHAR, and NVARLONGCHAR.

The PSQL ADO.NET Entity Framework Data Provider 4.2 requires Microsoft .NET Framework Version 4.0 or later with Entity Framework 1.0 functionality.

See [Code First and Model First Support](#), [Using Enumerations with the ADO.NET Entity Framework](#), and [Using Code First Migrations with the ADO.NET Entity Framework](#) in *Data Provider for .NET Guide*.

### **JDBC**

The PSQL JDBC driver now supports portions of the JDBC4 API. In particular, the PreparedStatement methods setNString and setNCharacterStream are now implemented. To get the best Unicode support, set the pvtranslate=auto option in the connection string.

The PSQL JDBC driver now implements the PreparedStatement method getParameterMetaData.

The name of the .jar file for the JDBC driver is unchanged.

On Linux, PSQL v12 now provides 64-bit JDBC and JCL access methods. This new support gives applications better memory availability for improved performance.

### **ODBC**

PSQL v12 includes changes to the ODBC access method for client applications to access wide character data. It also includes a newly created *PSQL ODBC Guide*.

### **ODBC Driver for Applications with Wide Character Data**

PSQL v12 provides an ODBC driver for 32-bit and 64-bit applications that use wide character data. The driver is only for Windows operating systems and is an addition to the previous set of drivers.

The PSQL ODBC Unicode Interface driver has the following behavior:

- Connects to a local or remote named database.
- With the 32-bit ODBC Administrator, creates 32-bit DSNs for use by 32-bit applications that use wide character data. The 32-bit driver is installed with all PSQL editions.
- With the 64-bit ODBC Administrator, creates 64-bit DSNs for use by 64-bit applications that use wide character data. The 64-bit driver is installed with any PSQL edition installed on a 64-bit version of Windows.

The driver also always behaves as if the `pvtranslate=auto` option were selected. Thus, it should be used only with databases where the database code page property is correct, either because it is explicitly set or because the default engine encoding is correct. The driver does not have an OEM/ANSI translation option. Instead, set the database code page to your OEM code page.

On Linux, the system encoding is usually UTF-8, which allows SQL text to contain any Unicode character code point. The Pervasive ODBC Unicode Interface driver is not available on Linux because an application can use the Pervasive ODBC Client Interface driver with UTF-8. A Linux application can handle wide character data either as UCS-2 strings (SQL\_C\_WCHAR) or request conversion to the system encoding (usually UTF-8) as SQL\_C\_CHAR. SQL text using UTF-8 is compatible with the existing Pervasive ODBC Client Interface driver, so an additional ODBC driver on Linux is not required.




---

**Note** The ODBC Client Interface driver continues to work with wide character applications and can access wide character database columns. The Microsoft Driver Manager continues to convert wide character queries and data to the ANSI code page of the application before passing them to the driver.

---

## New ODBC Guide and Release Notes

The PSQL documentation now includes a new book, *PSQL ODBC Guide*. The majority of the content was previously in *SQL Engine Reference*. The new book better serves users of the ODBC access method. The ODBC content is more easily referenced because it is no longer mixed with SQL content. Along with its own manual, the ODBC SDK also now has separate release notes.

## PDAC

PSQL now provides 32-bit design time packages and 32-bit and 64-bit PDAC runtime packages for RAD Studio XE4, XE5, XE6, and XE7.

New compile time PDAC binaries for RAD Studio XE4, XE5, XE6, and XE7 are available in the PDAC SDK web download package, and new runtime binaries are installed as part of the PDAC module for PSQL Server, Vx Server, Workgroup, and Client.

See [Using Direct Access Components](#) in *PSQL Direct Access Components Guide*.

## Wait Lock Timeouts

In past releases, page or record locking behavior took widely varying amounts of time to return errors to a calling relational application. This release makes the following changes to improve usability and reliability:

- The default timeout value has been increased from 15,000 to 30,000 milliseconds. Whatever setting you have in your current installation, upgrading to v12 automatically resets it to the new default time.
- Documentation of wait lock timeout settings has been expanded with more specific details. See [Wait Lock Timeout](#) in *Advanced Operations Guide*.

## **Discontinued Features**

PSQL v12 on Windows XP and Windows 2003 is not supported.

The optional database accelerator called Xtreme I/O (XIO) has been discontinued and is no longer included with PSQL.

If you have installed a previous version of PSQL 32-bit Server for Windows with XIO and upgrade to PSQL v12, the PSQL v12 installation removes XIO.

The ADO.NET provider versions 3.2 and 3.5 have been dropped. Their absence should have no effect on applications that are configured to use the default installed version. Applications that are bound to specific provider versions may need to be updated.

In addition, the ADO.NET provider version 4.0 has been deprecated and may be discontinued and removed from a future release of v12.

The pvmdconv utility has been dropped.

## Unsupported Features

This topic discusses Windows 8 and 10 features not supported by PSQL, and also PSQL features not supported by Windows.

### **Windows Features Not Supported by PSQL**

PSQL does not support the following features of Windows 8 and 10. See the Microsoft website for more information about these features.

- WinRT (Windows 8 native runtime)
- ARM Processor Support
- Live Tiles
- Connected Standby
- Low Impact Startup
- Deployment of PSQL through Windows Store at [microsoft.com](http://microsoft.com)

### **Windows To Go**

Windows To Go (WTG) allows Windows 8 and 10 Enterprise to boot and run from USB mass storage devices such as flash drives and external hard drives.

In the sense that a WTG device can be used among multiple machines, PSQL is not supported. You can install PSQL on a WTG device. However, the product key for PSQL remains in the “active” state only for the machine from which you authorized PSQL. If you start up any other machine with the WTG device, the product key is in “failed validation” state.

WTG is useful if you want to run the database engine on the system where you authorized PSQL. Using the WTG device with other systems is limited to a duration time less than the failed-validation period.

### **OneDrive**

Because Microsoft OneDrive provides two-way synchronization or “mirroring” of files, PSQL does not support its use in a multiuser environment. Multiple users accessing data synchronized by OneDrive can result in conflicting updates to files stored in the cloud service, leading to corrupted or lost data.

OneDrive has the same limitations as other cloud storage services intended for a single user. If you use any of these services as a storage location for PSQL data files, then it must be solely for single-user data.

### **PSQL Features Not Supported by Windows**

- **PSQL as a Windows Store app.** PSQL is not a Windows Store app. You cannot install or run the database engine or its client as a Windows Store app. However, a Windows Store app can use PSQL as a data repository. Search for topics on Open Data Protocol (OData) at the PSQL website.
- **PSQL XIO.** XIO is supported only on 32-bit server-class Windows operating systems. Therefore, XIO is not supported on Windows 8 and 10 because they are not server-class, nor on Windows Server 2012 because it is a 64-bit system.