

Pervasive PSQL v11

What Is New in Pervasive PSQL

An Overview of New Features and Changed Behavior

Pervasive Software Inc.
12365 Riata Trace Parkway
Building B
Austin, TX 78727 USA

Telephone: 512 231 6000 or 800 287 4383
Fax: 512 231 6010
Email: database@pervasive.com
Web: <http://www.pervasivedb.com>



disclaimer

PERVASIVE SOFTWARE INC. 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. PERVASIVE SOFTWARE INC. MAKES NO OTHER WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THE SOFTWARE OR THE CONTENT OF THE DOCUMENTATION; PERVASIVE SOFTWARE INC. HEREBY EXPRESSLY STATES AND YOU OR YOUR COMPANY ACKNOWLEDGES THAT PERVASIVE SOFTWARE INC. 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, Pervasive, Pervasive Software, and the Pervasive Software logo are registered trademarks of Pervasive Software Inc.

Built on Pervasive Software, DataExchange, MicroKernel Database Engine, MicroKernel Database Architecture, Pervasive.SQL, Pervasive PSQL, Solution Network, Ultralight, and ZDBA are trademarks of Pervasive Software Inc.

Microsoft, MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows Millennium, Windows 2000, Windows 2003, Windows 2008, Windows 7, Windows 8, 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.

Sun, Sun Microsystems, Java, all trademarks and logos that contain Sun, Solaris, or Java, are trademarks or registered trademarks of Sun Microsystems.

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

© Copyright 2013 Pervasive Software Inc. 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), modified and extended by Nick Gorham (nick@easysoft.com), with local modifications from Pervasive Software. Pervasive Software 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. For further information on this project, contact its current maintainer: Nick Gorham (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.

What Is New In Pervasive PSQL

January 2013

138-004433-004

Contents

About This Manual	vii
Who Should Read This Manual	viii
Manual Organization	ix
Conventions	x
1 What Is New in Pervasive PSQL v11 SP3	1
<i>An Overview of New and Changed Features</i>	
Support for Windows 8 and Windows Server 2012	2
Editions Supported	2
Windows Server 2012 Application Certification	2
Windows 8 Features	2
Unsupported and Inapplicable Features	7
Installation	9
Additional Interface Support for IPv6	10
Using Pervasive PSQL With IPv6	10
Frequently Asked Questions	12
Pervasive PSQL Utilities and IPv6	12
Licensing	13
Utilities	14
Pervasive Control Center (PCC)	14
License Administrator	15
Status Codes	16
Global Changes	16
New Status Codes	16
Revised Status Codes	16
Relational Interface	17
ORDER BY in Table Subquery	17
CREATE INDEX Using Specified Index Number	17
SDK Access Methods	18
ADO.NET	18
PDAC	18
JPA	18
Features for Original Equipment Manufacturers	19
Features for Update 4	20
A What Was New in Pervasive PSQL v11 SP2	21
<i>An Overview of Features in Pervasive PSQL v11 SP2</i>	
Data Backup Solutions	22
Pervasive Backup Agent	22
Pervasive PSQL VSS Writer	22
Licensing	24
Installation	25
Utilities	26
License Administrator (GUI and CLI)	26
Monitor	26
Query Plan Viewer	26
Status Codes	28
SDK Access Methods	29

DTI	29
DTO	29
PDAC.	29
Features for Original Equipment Manufacturers	30
B What Was New in Pervasive PSQL v11 SP1	31
<i>An Overview of Features in Pervasive PSQL v11 SP1</i>	
Performance Monitoring Counters for Pervasive PSQL	32
Message Logging	33
Licensing Messages	33
Pervasive Notification Viewer	33
Operating System Event Log	34
Pervasive PSQL Event Log	34
Related Topics	34
Installation	35
Firewalls	35
Pervasive Notification Viewer	36
Utilities	37
License Administrator.	37
Pervasive Notification Viewer	37
Pervasive System Analyzer (PSA)	37
Relational Interface.	38
System Catalog Functions.	38
Scalar Functions	38
SDK Access Methods.	40
Btrieve API.	40
DTI	40
DTO	40
PDAC.	41
Linux	42
64-bit ODBC Driver.	42
Red Hat Enterprise Linux 6.	47
Macintosh.	48
Features for Original Equipment Manufacturers	49
C What Was New in Pervasive PSQL v11	51
<i>An Overview of Features in Pervasive PSQL v11</i>	
Multi-core Support.	52
Why Multi-core Support	52
The Multi-core Dilemma	53
Benefiting from the Present While Planning For the Future.	55
Support for IPv6	56
Using Pervasive PSQL With IPv6	56
Frequently Asked Questions About IPv6 Support	59
Pervasive PSQL Utilities and IPv6	60
IPv6 Aspects for Application Programmers	60
64-bit ODBC Driver	64
ODBC and Data Source Names (DSNs).	64
Utilities Affected by ODBC Changes.	69
Support for .NET Framework 3.5 SP1 and 4.0.	70
Pervasive PSQL ADO.NET Data Provider 3.2.	70
Pervasive PSQL ADO.NET Data Provider 3.5.	70

PDAC Development Environments. 72

Enhancements to Other SDK Access Methods. 73

 DTO 73

Product Authorization 74

 Telephone Authorization 74

 Product Authorization for OEMs 74

Configuration Settings 75

 Communications Threads 75

 Listen IP Address 75

Utility Changes 76

 Pervasive PSQL Control Center 76

 ODBC Administrator 76

Deprecated and Discontinued Features. 77

 Deprecated Features. 77

 Discontinued Features 77

About This Manual

This manual contains information about the features and enhancements that are new in this release of Pervasive PSQL v11 SP3.

Who Should Read This Manual

This manual does not provide comprehensive usage instructions for the software. Its purpose is to explain what is new and different in this particular release of the product.

The Pervasive PSQL v11 family of products include three editions of the database engine: Pervasive PSQL Server, Pervasive PSQL Vx Server, and Pervasive PSQL Workgroup.

Throughout the Pervasive PSQL documentation, including this book, all content pertains to all three editions unless differences are explicitly called out. For example, you will occasionally encounter content that applies only to Pervasive PSQL Workgroup. Content that applies to Pervasive PSQL Server also applies to Pervasive PSQL Vx Server unless differences are explicitly called out.

For convenience of reference, the content for Pervasive PSQL Vx Server is consolidated in *Pervasive PSQL Vx Product Guide*. If you are interested only in Pervasive PSQL Vx Server, refer to that guide for ease of reference.

Pervasive Software Inc. would appreciate your comments and suggestions about this manual. As a user of our documentation, you are in a unique position to provide ideas that can have a direct impact on future releases of this and other manuals. If you have comments or suggestions for the product documentation, post your request at the Community Forum on the Pervasive PSQL Web site, www.pervasivedb.com.

Manual Organization

This manual begins with an overview of the new features, then provides links to chapters containing additional details where appropriate. *What Is New in Pervasive PSQL* is divided into the following sections:

- [Chapter 1—What Is New in Pervasive PSQL v11 SP3](#)
This chapter provides an overview of the changes in the current release of the product.
- [Appendix A—What Was New in Pervasive PSQL v11 SP2](#)
This chapter provides an overview of the changes in the second service pack of the product.
- [Appendix B—What Was New in Pervasive PSQL v11 SP1](#)
This chapter provides an overview of the changes in the first service pack of the product.
- [Appendix C—What Was New in Pervasive PSQL v11](#)
This chapter provides an overview of the changes in the initial release of the product.

Conventions

Unless otherwise noted, command syntax, code, and examples use the following conventions:

CASE	Commands and reserved words typically appear in uppercase letters. Unless the manual states otherwise, you can enter these items using uppercase, lowercase, or both. For example, you can type MYPROG, myprog, or MYprog.
Bold	Words appearing in bold include the following: menu names, dialog box names, commands, options, buttons, statements, and so forth.
Monospaced font	Monospaced font is reserved for words you enter, such as command syntax.
[]	Square brackets enclose optional information, as in [<i>log_name</i>]. If information is not enclosed in square brackets, it is required.
	A vertical bar indicates a choice of information to enter, as in [<i>file_name</i> @ <i>file_name</i>].
< >	Angle brackets enclose multiple choices for a required item, as in /D=<5 6 7>.
<i>variable</i>	Words appearing in italics are variables that you must replace with appropriate values, as in <i>file_name</i> .
...	An ellipsis following information indicates you can repeat the information more than one time, as in [<i>parameter</i> ...].
::=	The symbol ::= means one item is defined in terms of another. For example, a::=b means the item <i>a</i> is defined in terms of <i>b</i> .
% <i>string</i> %	A variable defined by the Windows operating system. <i>String</i> represents the variable text. Example: %ProgramFiles% is a variable for the location C:\Program Files.
<i>\$string</i>	An environment variable defined by the Linux operating system. <i>String</i> represents the variable text. Example: \$PATH, which contains a colon-separated list of directories that the shell searches for commands that do not contain a slash in their name.

What Is New in Pervasive PSQL v11 SP3

An Overview of New and Changed Features

The General Release includes new features and changes as summarized in the following topics:

- [Support for Windows 8 and Windows Server 2012](#)
- [Installation](#)
- [Additional Interface Support for IPv6](#)
- [Licensing](#)
- [Utilities](#)
- [Status Codes](#)
- [Relational Interface](#)
- [SDK Access Methods](#)
- [Features for Original Equipment Manufacturers](#)

Update 4 includes new features as summarized in the following topic:

- [Features for Update 4](#)

Support for Windows 8 and Windows Server 2012

Windows 8 features a new user interface designed to better suit touchscreen input, while continuing to accommodate traditional mouse and keyboard input. This section explains the extent to which Pervasive PSQL supports the features of Windows 8 and includes the following topics:

- [Editions Supported](#)
- [Windows Server 2012 Application Certification](#)
- [Windows 8 Features](#)
- [Unsupported and Inapplicable Features](#)

Editions Supported

The following table lists the Windows 8 editions supported by Pervasive PSQL.

Table 1 Pervasive PSQL Support for Windows 8 Editions

Windows 8 Edition	Pervasive PSQL Product			
	Server	Vx Server	Workgroup	Client
Windows 8 (32-bit and 64-bit)	Yes	Yes	Yes	Yes
Windows 8 Professional (32-bit and 64-bit)	Yes	Yes	Yes	Yes
Windows 8 Enterprise (32-bit and 64-bit)	Yes	Yes	Yes	Yes
Windows RT	No	No	No	No
Windows Server 2012 64-bit (32-bit not available)	Yes	Yes	Yes	Yes

Windows Server 2012 Application Certification

Pervasive PSQL Vx Server, 64-bit, has obtained the Windows Server Application Certification Program Logo for Windows Server 2012.

Windows 8 Features

The following topics discuss the primary features of Windows 8 as they pertain to Pervasive PSQL.

- [Start Screen and Apps Screen](#)
- [Microsoft Live ID](#)
- [Storage Spaces and Thin Provisioning](#)
- [ReFS File System](#)
- [4K Read Sector Size](#)
- [Live Migration without Shared Storage](#)
- [SMB 3.0 with Failover Cluster Environment](#)
- [NIC Teaming](#)
- [High Availability](#)
- [Pooled Virtual Desktop Infrastructure](#)
- [SkyDrive](#)

- [Refresh and Reset](#)
- [DisktoVHD and Bootable VHD](#)
- [Internet Explorer 10](#)
- [Windows Defender](#)

Start Screen and Apps Screen

In Windows 8, a Start Screen and an Apps Screen replace the traditional Start menu. Both screens use tiles to display Windows Store apps and shortcuts to display applications not from Windows Store.

The Start Screen is considered a user-preference location to be individually customized. Therefore, by default, no shortcuts for Pervasive PSQL components appear on the Start Screen.

The following shortcuts for Pervasive PSQL components appear on the Apps Screen.

Table 2 Pervasive PSQL Component Shortcuts on Windows 8 Apps Screen

Pervasive PSQL Component	Shortcut Name on Apps Screen^{1, 2}
Pervasive PSQL Control Center and Documentation	PSQL Control Center & Documentation
License Administrator	Pervasive License Administrator
PVSW Log	PSQL PVSW Log
Start Workgroup Engine	Start PSQL Workgroup Engine ³
Stop Workgroup Engine	Stop PSQL Workgroup Engine ³
DDF Builder	DDF Builder
Function Executor	PSQL Function Executor
Gateway Locator	PSQL Gateway Locator ³
Maintenance	PSQL Maintenance
Monitor	PSQL Monitor
Notification Viewer	PSQL Notification Viewer
Pervasive System Analyzer	Pervasive System Analyzer
Query Plan Viewer	PSQL Query Plan Viewer
Rebuild	PSQL Rebuild
<p>¹ "PSQL" has been added to the majority of shortcut names to help you identify Pervasive PSQL components. Shortcuts are organized under single folder names. The Pervasive PSQL folder name on the Apps Screen is "Pervasive PSQL 11."</p> <p>² No shortcut is provided for the XIO Statistics component because it does not apply. See Inapplicable Features.</p> <p>³ These screen components are present only if Pervasive PSQL Workgroup is installed as an application.</p>	

Microsoft Live ID

By default with Windows 8, a local user account is associated with a Live ID for signing on to the operating system. Signing on with Live ID automatically populates certain personalized settings associated with the Live ID. In addition, apps purchased from Windows Store roam with the Live ID account.

Microsoft Live ID offers no roaming advantage for Pervasive PSQL because Pervasive PSQL is not a Windows Store app. (However, a Windows Store app can use Pervasive PSQL as a data repository. Refer to the content on Open Data Protocol (OData) available on the Pervasive PSQL Web site.)

However, if you choose, you can use a Live ID account to log on to the machine where you installed Pervasive PSQL. In that case, Pervasive PSQL functions just as it does for a local user account. For example, the transactional and relational services start successfully. Similarly, the Pervasive PSQL Workgroup Engine and Cache Engine auto-load and auto-unload as required when you log on or log off. Pervasive PSQL security honors Windows 8 security based on the Windows file system, networking, and peer-to-peer sharing configured to use Microsoft Live ID for authentication.

Storage Spaces and Thin Provisioning

Storage Spaces is a collection of drives grouped into a single logical drive resource. Thin Provisioning is a storage technology that allows the over-allocation of capacity without actually reserving physical storage. A file system can be much larger than the underlying storage, if that storage is expandable.

Storage Spaces

Pervasive PSQL licensing is not affected by the use of Storage Spaces. You can set up a Storage Spaces drive array, remove a disk from the array, or remove the entire array.

Pervasive PSQL can use a Storage Spaces drive as regular physical storage. The database engine can access the drive configuration and all features function normally. If the Storage Spaces configuration runs out of storage, the database engine returns an “out of disk space” status. Pervasive PSQL VSS Writer functions normally with backup operations using Storage Spaces.

Thin Provisioning

Pervasive PSQL can use Thin Provisioning storage. For an “out of disk space” condition, the operating system takes the disks offline. The database engine returns an “out of disk space” status.

ReFS File System

ReFS (Resilient File System) is built on the foundations of NTFS to maintain compatibility yet is architected for a new generation of storage technologies. At the application level, ReFS stored data is accessible just as is NTFS data. Pervasive PSQL can use physical storage formatted as ReFS.

One incompatibility with ReFS is DOS short names. They are an NTFS feature but are not supported with ReFS.

4K Read Sector Size

Pervasive PSQL can use disk storage with 4K sector size.

However, if the physical page size of the data file being opened is less than the section size of the storage (4K in this case), the database engine uses system cache when it opens the file. This applies even if the configuration setting [System Cache](#) is turned off. In addition, the engine writes a warning message to

PVSW.LOG. Only one warning is written as long as the engine retains pages in cache for the file. A `butil -stat` command issued on a file also writes a warning. This can result in numerous warnings if the `-stat` command is issued repeatedly while no other operations occur to the file.

If you want to avoid the use of system cache and messages being written to PVSW.LOG, rebuild the data file with a page size of 4K or greater, or use physical storage with a conventional sector size.

Live Migration without Shared Storage

Microsoft Hyper-V allows for the migration of a running VM from one host to another without requiring shared storage.

Pervasive PSQL Vx Server can be migrated under Hyper-V and requires no reconnection from the PSQL Clients. The key for PSQL Vx Server remains in the “active” state provided the virtual MAC addresses and host names remain the same after the VMs migrate.

You may prefer not to migrate Pervasive PSQL Server or Workgroup under Hyper-V unless you intent to migrate back to original host fairly quickly. The key for either product changes to “failed validation” state when a VM migrates to another host. The database engine runs only for a limited number of days in “failed validation” state.

For details, see [High Availability Support](#) in *Advanced Operations Guide*.

SMB 3.0 with Failover Cluster Environment

Server Message Block (SMB) 3.0 can provide shared access to files from a failover cluster. Pervasive PSQL Workgroup running outside of the cluster environment can use data that resides on SMB cluster storage and will continue to work uninterrupted when a cluster failover occurs. Set access permissions to the shared storage through Failover Cluster Manager, not through the operating system (such as with My Computer or File Explorer).

Pervasive PSQL Server and Vx Server do not support the use of data that resides on separate SMB cluster storage. They require shared storage that the engine accesses as if it were the local file system.

See also [High Availability Support](#) in *Advanced Operations Guide*.

NIC Teaming

Network interface card (NIC) teaming is a feature that allows two or more specialized NICs to be set up as a “team” for fault tolerance or load balancing.

NIC teaming can be used on a physical or virtual machine with Pervasive PSQL Server, Vx Server, or Workgroup. However, ensure that you configure the NIC teaming *before* you authorize the Pervasive PSQL product. If you subsequently need to change the NIC teaming, first deauthorize the Pervasive PSQL product, make the changes, and then reauthorize the product.

High Availability

Pervasive PSQL can be used in high availability environments. See [High Availability Support](#) in *Advanced Operations Guide*.

Pooled Virtual Desktop Infrastructure

Virtual desktop infrastructure (VDI) is a feature that hosts a user's desktop operating system as a VM on a server. Pooled VDI creates a pool of VMs from which a newly initialized VM instance is created when a user logs in.

Pervasive PSQL Client is compatible with pooled VDI and can access remote data. The Client can be added to a VM template and functions normally when accessed from the pool by multiple users.

Pervasive PSQL Server, Vx Server, and Workgroup are *not* compatible with pooled VDI VMs. A VM template can be created that includes the database engine, but each instance of the VM accessed from the pool violates Pervasive PSQL licensing.

Refresh and Reset

Refresh and Reset allows the existing installation of Windows 8 to be overwritten with a new image of the operating system. Both Refresh and Reset remove Pervasive PSQL from the machine (the product key, all Registry entries, and the installation folders and files).

If you want to use Refresh or Reset, first deauthorize the Pervasive PSQL product key on the machine. This allows you to authorize the key on another machine or on the same machine after the Refresh and Reset completes. See [To Deauthorize a Key](#) and [To Authorize a Key](#), both in *Pervasive PSQL User's Guide*.

Reset

Reset removes all personal data, applications—including Windows Store apps—and customized settings.

Refresh

Refresh retains any Windows Store apps that were installed from the Microsoft Windows Store. The installations (and Registry entries) for applications not from Windows Store, such as Pervasive PSQL, are lost when the new image of the operating system is written. However, also note the following.

- Databases created outside the default Pervasive PSQL installation directory structure are retained on the file system.
- Databases created within the default Pervasive PSQL installation directory structure are archived in folder C:\Windows.old.

For example, a database named “testdb1” created at C:\Program Files\Pervasive Software\testdb1 is retained in C:\Windows.old\Program Files\Pervasive Software\testdb1.

- If Pervasive PSQL was installed with the default paths, DBNAMES.CFG is retained in C:\Windows.old\ProgramData\Pervasive Software\PSQL.
- File paths and data paths that do not include the operating system folders “Program Files,” “Program Files (86),” or “ProgramData” are *not* archived in C:\Windows.old.

For example, you perform a custom installation of Pervasive PSQL and specify C:\PervasiveSoftware1 for the data path and C:\PervasiveSoftware2 for the file path. Those two directories, which include DBNAMES.CFG and the other files, are retained after a Refresh, but no files in those directories are archived to C:\Windows.old.

DisktoVHD and Bootable VHD

DisktoVHD allows the creation of a virtual machine (VM) from a physical machine instance. DisktoVHD is similar to other physical-to-virtual utilities provided with hypervisor products. See *Pervasive PSQL Vx Server Product Guide* for a discussion of the hypervisor products with which Pervasive PSQL Vx Server is compatible.

Bootable VHD allows the creation of a Hyper-V VM from which you can boot Windows 8 on another machine instead of loading the operating system from the image on the machine's physical disk. The other machine must already have Windows 8 installed.

Note the following if you intend to use DisktoVHD or Bootable VHD:

- Pervasive PSQL Vx Server requires Internet connectivity at all times and consistent host name and all network interface card (NIC) MAC addresses.
- Each instance of Pervasive PSQL Vx Server, including those that are created, cloned or copied for VMs, requires its own permanent key even if the host name and MAC addresses are the same. Therefore, deauthorize the Pervasive PSQL Vx Server key on the physical machine before creating the VM and then apply the Pervasive PSQL key for each VM instance. See [To Deauthorize a Key](#) and [To Authorize a Key](#), both in *Pervasive PSQL User's Guide*.

Internet Explorer 10

Pervasive PSQL documentation provided in an HTML format, such as the PCC documentation library, PSA network troubleshooting tips, release notes, and the Pervasive PSQL Web Library, are compatible with Internet Explorer 10.

Windows Defender

Windows Defender provides stronger protection against malware. You should be able to install, configure, and run Pervasive PSQL without interfering with Windows Defender. See installation steps in *Getting Started With Pervasive PSQL* and [Configuration Reference](#) in *Advanced Operations Guide*.

Unsupported and Inapplicable Features

This section discusses features of Windows 8 not supported and Pervasive PSQL features that are not applicable to Windows 8.

Unsupported Features

Pervasive PSQL does not support the following features of Windows 8. Refer to the Microsoft Web site if you want details about these features.

- WinRT (Windows 8 native runtime)
- ARM Processor Support
- Live Tiles
- Connected Standby
- Low Impact Startup
- Deployment through Windows Store (deployment of Pervasive PSQL through Windows Store)

Windows To Go

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

In the traditional sense of a WTG device that can be used among multiple machines, Pervasive PSQL is not supported. You can install Pervasive PSQL on a WTG device. However, the product key for Pervasive PSQL remains in the “active” state only for the machine from which you authorized Pervasive PSQL. If you boot 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 machine from which you authorized Pervasive PSQL. Using the WTG device with other machines would be limited to a duration less than the failed-validation period.

SkyDrive

In addition to the items just listed, SkyDrive is not supported if you intend to use Pervasive PSQL in a multi-user environment. SkyDrive is Microsoft's implementation of cloud storage to serve as *personal* cloud storage. SkyDrive is designed for a single user who operates only one device at a time. Multiple users accessing the data files from multiple devices does not work correctly on SkyDrive. Multiple concurrent updates from multiple devices to the same data can result in corrupted data or data loss.

SkyDrive has the same limitations as other cloud storage options intended for personal data storage, such as Dropbox.com, Apple's iCloud, Google Drive, CX.com, and SugarSync. If you use SkyDrive as a storage location for Pervasive PSQL data files it must be solely for single-user data.

Inapplicable Features

- **Pervasive PSQL as a Windows Store app.** Pervasive PSQL is not a Windows Store app. You cannot install or run the database engine or the Client as a Windows Store app.



Note However, a Windows Store app can use Pervasive PSQL as a data repository. Refer to the content on Open Data Protocol (OData) available on the Pervasive PSQL Web site.

- **Pervasive PSQL XIO.** XIO is supported only on 32-bit server-class Windows operating systems. Therefore, XIO is not available on Windows 8 editions because they are not server-class, or on Windows Server 2012 because it is only available in 64-bit.

Installation

The installation of Pervasive PSQL on Windows platforms includes the following changes.

- A single name that combines “Pervasive” with the major release designation is used to organize product shortcuts. For example, the Start menu location is now Start --> Programs --> Pervasive PSQL 11 instead of Start --> Programs --> Pervasive --> PSQL 11.
- The shortcut name for the following components now includes “PSQL” as part of the name:
 - PSQL Control Center & Documentation
 - PSQL PVSU Log
 - PSQL Function Executor
 - PSQL Maintenance
 - PSQL Monitor
 - PSQL Notification Viewer
 - PSQL Query Plan Viewer
 - PSQL Rebuild
 - Start PSQL Workgroup Engine (Pervasive PSQL Workgroup only)
 - Stop PSQL Workgroup Engine (Pervasive PSQL Workgroup only)
- The shortcut name for License Administrator is now “Pervasive License Administrator.”

See also [Start Screen and Apps Screen](#).

Additional Interface Support for IPv6

With this release, Pervasive PSQL has increased its support for IPv6, the next-generation Internet Protocol version. This section discusses the following topics:

- [Using Pervasive PSQL With IPv6](#)
- [Frequently Asked Questions](#)
- [Pervasive PSQL Utilities and IPv6](#)

For information about implications for application programmers of IPv6 support, see [IPv6 Aspects for Application Programmers](#).

For complete details about IPv6 and Pervasive PSQL, see [IPv6](#) in *Getting Started With Pervasive PSQL*.

Using Pervasive PSQL With IPv6

Pervasive PSQL v11 SP3 supports IPv6 for the following additional access methods on Windows operating systems. (See also [Support for IPv6](#) in Appendix C.)

- OCBC
- ADO.NET
- JDBC
- JCL

There is now complete IPv6 support for the following Pervasive PSQL utilities. (See also [Pervasive PSQL Utilities and IPv6](#) in Appendix C.)

- PCC
- DDF Builder

All of these access methods function correctly in an IPv4 environment, an IPv6 environment, or an environment that combines the two. No special configurations of Pervasive PSQL are required.

Client Connections

With any of the access methods listed above, a Pervasive PSQL Client connects using IPv6 to a host running the Pervasive PSQL database engine (Server or Workgroup) the same way as it connects using IPv4. That is, the Client specifies a server either by name or using its IP address and connects through a supported access method.

For a complete discussion of address formats for IPv6, please see the following topics in this document:

- [IPv6 Address Formats](#)
- [IPv6 With UNC Paths and URI Connections](#)

Certain restrictions that applied to IPv6 addresses have changed. These are summarized in the following table:

Table 3 IPv6 Restrictions With Pervasive PSQL

Restriction	Discussion
The Pervasive PSQL Server Engine in an IPv6-only environment	The Pervasive PSQL Server Engine is not supported in an IPv6-only environment on Windows Server 2003 or Windows XP operating systems. The Server Engine is supported in an IPv6-only environment on Windows Vista and later operating systems.
If you wish to append a port number to an IPv6 address, you must use UNC-safe notation	The inclusion of a port number with an IP address is necessary only when you need to override the default port. Generally, a port number can be appended to an IP address using either a colon (:port number) or a dot (.port number). UNC-safe names (see IPv6 With UNC Paths and URI Connections) support the appending of a port number.
Square brackets are required for raw IPv6 addresses when the address is used in a URI or UNC	Raw IPv6 addresses, abbreviated or not, must be enclosed by square brackets if the address is used in a URI or UNC. Examples: <ul style="list-style-type: none"> • btrv://czjones@[2001:b1::23]/demodata • btrv://abanderas@[2001:12:34:56:78:90:12:23]/demodata • \\[2001:12:34:56:78:90:12:23]acctsvr1\Domesticfile.mkd Failure to bracket the IPv6 address results in status code 3014 or 3103 for Btrieve calls using a URI, or status code 11, 94, or 170 for Btrieve calls using a UNC.
In a URI, if you include a ZoneID to a server address, the “%” ZoneID character must be escaped with “%25”	If you use a btrv:// connection with an IPv6 address, you must escape the ZoneID for the host name. Zone IDs are usually required with IPv6 Link Local numeric addresses. Example: A UNC-safe addresses like btrv://@[fe80::20c:29ff:fe67:2ee4%4] must be changed to btrv://@[fe80::20c:29ff:fe67:2ee4%254]
License Administrator (and cliicadm)	The Pervasive licensing server does not yet support IPv6. Because of this, you can use License Administrator over IPv6 to administer licenses but you cannot authorize a license with the utility. To authorize a license, you must use an IPv4 network, remote authorization, or telephone authorization.

Frequently Asked Questions

The following table answers some frequently asked questions (FAQs) about IPv6 support for interfaces in Pervasive PSQL v11 SP3.

Table 4 FAQs About IPv6 Support for Interfaces

Question	Answer
Is the fact that PSQL Clients will now have both IPv4 and IPv6 addresses going to lead to confusion over how many licenses I should have?	No. This is not a new situation. In the past, Clients have sometimes had more than one IPv4 address. Pervasive PSQL has handled the situation by recognizing all of the addresses of a machine to identify it. PSQL will follow the same logic for multiple addresses using more than one IP version.
What formats should I use for IPv6 addresses?	See IPv6 Address Formats and IPv6 With UNC Paths and URI Connections for a complete description.
Does this in any way change my use of DSNs?	No, continue to use DSNs as usual.
Can I still use Pervasive System Analyzer to test connections?	Yes, you can use IPv6 addresses.
Can I use Pervasive Auto Reconnect (PARC) with IPv6?	Yes.
Does Pervasive PSQL support IPv6 communications in virtual machine environments?	Yes.
Is IPv6 supported for Linux distributions or Macintosh OS X?	No. Only Windows platforms are supported.
Is IPv6 supported for Pervasive DataExchange, and AuditMaster?	No.
Does a network environment that includes both IPv4 and IPv6 affect Pervasive PSQL user counts?	No. Pervasive PSQL Server or Workgroup uses one user count for each <i>unique</i> incoming protocol from the same client computer session (such as TCP/IP and SPX). IPv4 and IPv6 are just different address formats of TCP/IP.
Can the Listen IP Address configuration setting be set to multiple addresses?	Yes. See Listen IP Address .

Pervasive PSQL Utilities and IPv6

As a result of the implementation of IPv6 for additional interfaces, the following changes have occurred in Pervasive PSQL utilities:

Utility	Description	See Also
Pervasive PSQL Control Center (PCC)	PCC is now completely supported and fully functional in an IPv6 environment.	Using Pervasive PSQL Control Center in <i>Pervasive PSQL User's Guide</i>
DDF Builder	DDF Builder is now completely supported and fully functional in an IPv6 environment.	<i>DDF Builder User's Guide</i>
Monitor	Network Address is one of the display fields in the "MicroKernel Active Sessions" Session Information and in "MicroKernel Active Files" Handle Information . If the calling process uses TCP/IP, the address is preceded by "T:". This applies whether the address shown is a hostname, an IPv4 address or an IPv6 address.	Monitoring Database Resources in <i>Advanced Operations Guide</i> .

Licensing

In this release, Pervasive introduces an option for users to perform the repair of a product key in the event that it fails validation or is disabled. Failed validation of a product key was often the result of changes in hardware that caused the Pervasive license server to no longer recognize the machine.

In the past, when a key failed validation or became disabled, users had to contact either Pervasive Support or the support staff of their original equipment manufacturer (OEM).

Pervasive License Administrator now provides a way to repair a product key, which also re-authorizes the key on the same machine. The repair feature provides self-service key re-authorization for situations when a key cannot be recovered from the machine and the key is not in use on another machine.

The repair process requires Internet access. It cannot be done over the telephone. The default number of allowed repairs per key is 5.

See [License Administrator](#) in this chapter and [Failed Validation and Failed-Validation Period](#) in *Pervasive PSQL User's Guide*.

Utilities

This section discusses the new and revised functionality to the Pervasive PSQL utilities.

Pervasive Control Center (PCC)

PCC includes the following changes:

- [Capacity Usage Viewer](#)
- [Monitor](#)

Capacity Usage Viewer

PCC now includes functionality to monitor graphically the concurrent sessions and data usage for all database engines. This feature, called Capacity Usage Viewer, is especially useful when you are considering migrating from PSQL Server to PSQL Vx Server, because of the difference in the way those two editions are licensed.

You access the capacity usage graphs from the right-click context menu of the desired database server in Pervasive PSQL Explorer.

See [Capacity Usage Viewer](#) in *Pervasive PSQL User's Guide*.

Monitor

PCC now integrates a Monitor utility. You access it from the right-click context menu of the desired database engine in Pervasive PSQL Explorer.

The integrated Monitor presents information in the form of a series of tabs and provides the same functionality as the GUI Monitor and CLI Monitor. In addition, the integrated Monitor allows you to display more data, select which columns of data to display, and sort the columns. You can monitor multiple engines at the same time and compare them side by side.

The GUI Monitor and CLI Monitor are still included with Pervasive PSQL. However, the GUI Monitor is deprecated and will be removed from the product in a future release. Its replacement is the Monitor integrated into PCC, which will be enhanced in future releases.

See [Monitor](#) in *Pervasive PSQL User's Guide* and [Monitoring Database Resources](#) in *Advanced Operations Guide*.

License Administrator

The following table summarizes the changes to License Administrator. See [License Administration](#) chapter in *Pervasive PSQL User's Guide* for details.

Table 5 License Administrator Changes

Change	Type	Discussion
Repair functionality	Addition	<p>The repair feature allows you to repair a product key in a "failed validation" or "disabled" state. The repair process also re-authorizes the key on the same machine. The default number of allowed repairs per key is 5.</p> <p>License Administrator prompts you to acknowledge that you are performing the repair on a genuine copy of Pervasive PSQL. Your acknowledgement affirms that the license is being used in accordance with the terms and conditions of the End User License Agreement (EULA) and that the license has not been installed on more machines than the maximum allowed by the license.</p> <p>Depending on the circumstances, some keys cannot be repaired. However, if the state change has resulted because of hardware configuration changes to a machine, and the machine is still in operation, repair is an appropriate solution.</p> <p>On the GUI, you access this functionality with the "Repair" button or from the right-click menu for a product key.</p> <p>For the CLI, you access the functionality with the "repair" option.</p>
Apply to Authorize	Revision	<p>Certain terminology has been revised in this release to enhance understanding. "Authorize" is the term now used to describe the action of associating a key with a machine signature to ensure that the copy of Pervasive PSQL is legitimate.</p> <p>On the GUI, the "Apply" button is now labeled "Authorize."</p> <p>For the CLI, the "apply" option is now "authorize." For backwards compatibility, the CLI still accepts the "apply" option but the option is no longer listed in the help information for the utility.</p>
Delete to Deauthorize	Revision	<p>Certain terminology has been revised in this release to enhance understanding. "Deauthorize" is the term now used to describe the action of disassociating a key with a machine signature.</p> <p>On the GUI, the "Delete" button is now labeled "Deauthorize."</p> <p>For the CLI, the "delete" option is now "deauthorize." For backwards compatibility, the CLI still accepts the "delete" option but that option is no longer listed in the help information for the utility.</p>
Validate functionality	Removal	<p>The validate functionality has been removed from the GUI to enhance understanding because validation was often confused with authorization. In addition, the repair functionality and the authorization functionality include key validation. A separate validate action is no longer required.</p> <p>For backwards compatibility, the CLI License Administrator still accepts the "validate" option and the help information still lists the option.</p>
Refresh functionality	Removal	<p>To streamline the License Administrator GUI, the "Refresh" button has been removed. The refresh functionality available through the Windows operating system (F5 by default) refreshes the License Administrator GUI. A separate refresh feature is not required.</p>

Status Codes

Global Changes

- Standardization of vocabulary in status code messages and explanations. Pervasive made the decision to standardize licensing-related terminology to ensure that messages were unambiguous and to support the addition of a key repair feature to License Administrator. “Authorize” (rather than “apply,” “add,” or “install”), “deauthorize” (rather than “delete”), “repair” (rather than “recover”) are the standardized terms.
- Addition or substitution of “Contact the product key vendor” where appropriate. This alerts the end user immediately that the solution of the issue lies with the product key vendor.

New Status Codes

7116: Buffer overflow

7133: An associated UCI key is preventing the deauthorization of the permanent key. The UCI key may already be in the failed or disabled state, or it may be set to transition into the failed state the next time it is validated. Repair the UCI key and try again to deauthorize the permanent key.

7254: Key cannot be authorized because it contains invalid license data. Contact your product key vendor.

7300: Local licensing component incompatible with remote licensing server

7314: This key cannot be deauthorized because it was authorized on a machine with a different machine signature. Either it is being used on a different machine or, more likely, the hardware signature of the original machine has changed. If you are cloning or copying a key licensed for use on only one machine, please purchase another valid license key. Otherwise, repair the key, then try again to deauthorize it.

7334: Key has no machine signature associated with it

7449: Key cannot be set to active because of a pending issue with the key vendor. Contact your product key vendor.

Revised Status Codes

The following status code messages were changed for clarification:

7218: Debug entry logged pertaining to acquiring the machine configuration

7125: An active permanent license already exists

The following status code explanations were corrected or expanded:

7224: User lacks the appropriate rights to authorize or deauthorize a key

7239: Internet connection cannot access authorization server

7313: Product key is already used on a different machine

7336: Authorization server prohibits authorization from within a Virtual Machine session

Relational Interface

This section discusses the new and revised functionality to support the relational interface.

ORDER BY in Table Subquery

A table subquery within a SELECT now supports the ORDER BY clause. A common use for this feature is to include a TOP clause in the SELECT list to return the first number of desired rows. ORDER BY clauses are allowed in table subqueries only, not in any other type of subquery clause.

See [SELECT](#) in *SQL Engine Reference*.

CREATE INDEX Using Specified Index Number

CREATE INDEX now supports the specification of an index number with the USING *index_number* clause. This functionality is particularly useful when the data file for a table is accessed directly through the transactional interface as well as through the relational interface. You can include the USING *index_number* clause with CREATE INDEX and CREATE INDEX IN DICTIONARY.

See [CREATE INDEX](#) in *SQL Engine Reference*.

SDK Access Methods

Pervasive PSQL v11 SP3 includes additions or enhancements to the following SDK access methods:

- [ADO.NET](#)
- [PDAC](#)
- [JPA](#)

ADO.NET

Pervasive PSQL v11 SP3 includes Pervasive PSQL ADO.NET Data Provider 4.0 and deprecates Data Provider 3.2.

Data Provider 4.0

The default installation of Pervasive PSQL Server, Vx Server, Workgroup, and Client now includes Pervasive PSQL ADO.NET Data Provider 4.0 (along with Data Provider 3.5 and 3.2). Data Provider 4.0 can run under all .NET Frameworks from 2.0 SP1 to 4.0. The Data Provider 4.0 includes support for IPv6, Entity Framework 4.0 DbContext functionality, and Visual Studio LightSwitch (and all of the Entity Framework features for 1.0 through 3.5 SP1). See *Data Provider for .NET Guide* for details.

Data Provider 3.2

Data Provider 3.2 is deprecated. It is included in this release but will not be included in future releases. Applications using Data Provider 3.2 continue to work under PSQL v11 SP3. However, we recommend that you begin your designs to move from Data Provider 3.2 to Data Provider 3.5 or 4.0. Code changes will no longer be made to Data Provider 3.2. Any necessary changes will be included in Data Provider 3.5 and 4.0 instead.

PDAC

Pervasive PSQL provides support for 32-bit PDAC through Delphi RAD Studio XE3. Install RAD Studio XE3 in the same manner as RAD Studio XE2.

New compile time PDAC binaries for RAD Studio XE3 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 *Pervasive Direct Access Components Guide*.

JPA

Java Persistence API (JPA) is an object-relational mapping (ORM) framework, that is, it maps objects to relational databases. JPA provides the standard for object-relational mapping in the Java language. JPA can use either Standard or Enterprise Edition Java.

EclipseLink is one implementation of JPA. It is an open source project, and it works with a large number of databases or platforms. Pervasive has added PSQL as an EclipseLink JPA platform.

The Pervasive database platform file has been submitted to EclipseLink and is awaiting full promotion. For the latest information about its status and download, go to <http://www.pervasivedb.com> and navigate to the SDK section of the website.

Features for Original Equipment Manufacturers

See “About This Manual” chapter in *Manufacturing Partner Handbook* for a summary of Pervasive PSQ v11 SP3 features specifically for original equipment manufacturers (OEMs).

Features for Update 4

Update 4 (May 2013) for Pervasive PSQL v11 SP3 contains the following new features:

- IPv6 support for Linux distributions and Macintosh operating system. All of the Pervasive PSQL access methods except for ActiveX support IPv6 connections. IPv6 support is also included for more utilities. See [IPv6](#) in *Getting Started With Pervasive PSQL* for complete details.
- PDAC support for 64-bit RAD Studio XE3 runtime packages. See [Building and Deployment Instructions for Delphi or C++ Builder](#) in *Pervasive Direct Access Components Guide*. The PSQL Web site also has an updated PDAC SDK available for download. See www.pervasivedb.com.
- Support for ADO.NET 4.0 SP1, which includes the following:
 - Data support for the Microsoft .NET Framework Version 4.5 with Entity Framework 1.0 functionality.
 - Microsoft Visual Studio 2012 and Microsoft Visual Studio LightSwitch 2011 and 2012.
 - Federal Information Processing Standard (FIPS) compliance.
 - The .NET Framework 4.0 Canonical Functions.

See also [Quick Start](#) in *Data Provider for .NET Guide*.

Update 4 is available for download from www.pervasivedb.com.

What Was New in Pervasive PSQL v11 SP2

An Overview of Features in Pervasive PSQL v11 SP2

The General Release includes new features and changes as summarized in the following topics:

- [Data Backup Solutions](#)
- [Licensing](#)
- [Installation](#)
- [Utilities](#)
- [Status Codes](#)
- [SDK Access Methods](#)
- [Features for Original Equipment Manufacturers](#)

Data Backup Solutions

The chapter [Logging, Backup, and Restore](#) in *Advanced Operations Guide* discusses the data backup solutions provided with Pervasive PSQL. In addition to those solutions, Pervasive PSQL v11 SP2 now includes the following:

- [Pervasive Backup Agent](#)
- [Pervasive PSQL VSS Writer](#)

If your backup software is *not* aware of the Microsoft Volume Shadow Copy Service (VSS), you can use Pervasive Backup Agent with your backup software. The VSS framework is included with Windows Server 2003 and newer operating systems.

If your backup software *is* VSS aware, Pervasive PSQL VSS Writer is automatically invoked during VSS backups. You do not need to use Pervasive Backup Agent if your backup software is already VSS aware.

Pervasive Backup Agent and Pervasive PSQL VSS Writer can be used together, but there is no advantage in doing so. Your backup process will be more streamlined if you select one method or the other. (See also [Pervasive PSQL VSS Writer and Pervasive Continuous Operations](#) in *Advanced Operations Guide*.)

Pervasive Backup Agent

Pervasive Backup Agent provides a quick and simple method for you to set and manage Continuous Operations on your Pervasive PSQL database files. Setting and managing Continuous Operations is a critical piece when backing up your Pervasive PSQL databases without using Microsoft Volume Shadow Copy Service. Backup Agent automatically handles setting and managing Continuous Operations on your open files so that your data is still available from your application during your backup. Once the backup procedure is complete, stopping Backup Agent takes the files out of Continuous Operations and rolls in all the changes captured during the backup.

Pervasive Backup Agent is an optional product included with Pervasive PSQL v11 SP2 Server. Pervasive Backup Agent is *not* included with Pervasive PSQL Workgroup, but can be purchased as a separate product.

By default, Pervasive Backup Agent is not installed. You must install it from the Pervasive PSQL v11 SP2 media after you install Pervasive PSQL Server.

No separate license for Pervasive Backup Agent v3.1 is required if it is installed on the same machine as Pervasive PSQL v11 SP2 Server. This also applies if you upgrade Pervasive PSQL v11 Server or v11 SP1 Server to Pervasive PSQL v11 SP2 Server.

For details on Pervasive Backup Agent, see the following:

- [Data Backup with Backup Agent and VSS Writer](#) in *Advanced Operations Guide*.
- *Pervasive Backup Agent Guide*, which is available on the Pervasive Software Website (<http://www.pervasivedb.com>).

Pervasive PSQL VSS Writer

The Microsoft Volume Shadow Copy Service (VSS) consists of Writer, Provider, and Requestor components. Pervasive PSQL v11 SP2 supports VSS with only a Writer component, Pervasive PSQL VSS Writer.

Pervasive PSQL VSS Writer is a feature of the database engine and is enabled for Pervasive PSQL v11 SP2 Server. Pervasive PSQL VSS Writer is available for use after that product is installed. Pervasive PSQL VSS Writer is currently not available for use with Pervasive PSQL Workgroup.

Pervasive PSQL VSS Writer is available only on Windows operating systems. For more information on Volume Shadow Copy Service, refer to the Microsoft Website document, *A Guide for SQL Server Backup Application Vendors*.

During VSS snapshots, Pervasive PSQL VSS Writer quiesces all disk I/O *write* activity to all Pervasive PSQL data and transaction log files, regardless of the volume on which they reside. After the snapshot is taken, Pervasive PSQL VSS Writer allows all disk I/O to resume; this includes any writes that were deferred during the quiesced period.

Pervasive PSQL VSS Writer never quiesces disk I/O *read* activity, allowing normal database processing to continue during the quiesced period as long as writes are not required. Pervasive PSQL VSS Writer operates normally during the backup phase, although performance may likely be reduced due to the backup activity of the VSS service and VSS Requestor.

For details about Pervasive PSQL VSS Writer, see [Data Backup with Backup Agent and VSS Writer](#) in *Advanced Operations Guide*.

Licensing

The license model for Pervasive PSQL v11 SP2 remains unchanged. A product key specifies a licensed user count.

However, a different license model—capacity based—also exists to accommodate the addition of Pervasive PSQL Vx Server. See [License Models](#) in *Pervasive PSQL User's Guide*.

We mention the different license model because certain features of Pervasive PSQL v11 SP2 have changed to accommodate both license models. For example, the wording of some status codes has changed, as have the interfaces for certain utilities. See the following in this guide:

- [Utilities](#)
- [Status Codes](#)

Installation

The installation media for Pervasive PSQL v11 SP2 Server also includes Pervasive Backup Agent v3.1. By default, Pervasive Backup Agent is not installed. You must install it from the Pervasive PSQL media after you install the Server database engine.

No separate key for Pervasive Backup Agent v3.1 is required if the product is installed on the same machine as Pervasive PSQL v11 SP2 Server. You are *not* prompted for a key if the machine is running Windows. Similarly, you are not required to apply a key if the machine is running Linux.

A separate key for Pervasive Backup Agent *is* required if the product is installed on the same machine as Pervasive PSQL v11 SP2 Workgroup.

Utilities

The following utilities have changed to accommodate both license models (user count and capacity based).

- [License Administrator \(GUI and CLI\)](#)
- [Monitor](#)
- [Query Plan Viewer](#)

License Administrator (GUI and CLI)

License Administrator now displays the limit for user count, session count, and data in use for a key. A value of “n/a” (“not applicable”) is displayed for the limit total if the type of key does not apply. For example, session count and data in use do not apply to a key for Pervasive PSQL v11 SP2, so both limits display as “n/a.” See [License Administration](#) in *Pervasive PSQL User's Guide* for details.

Monitor

Monitor has the following changes to accommodate both license models. The changes are summarized for the graphical user interface. The command line interface (bmon) contains similar changes where applicable. See [Monitoring Database Resources](#) in *Advanced Operations Guide* for details.

- The “Active Users” menu command and “MicroKernel Active Users” dialog have been renamed to “Active Sessions” and “MicroKernel Active Sessions,” respectively. The dialog has been modified to reflect session information.
- The “MicroKernel Resource Usage” dialog shows the current, peak, and maximum values for user count, session count, and data in use. “Licenses in Use” has been removed.
- The “MicroKernel Active Files” dialog lists the physical file size in kilobytes (KB) for a given file.
- The amount of time in weeks, days, hours, and minutes that the database engine has been running (referred to as “MicroKernel Uptime”) is shown on the “MicroKernel Communications Statistics” dialog and on the “MicroKernel Resource Usage” dialog.

If a resource does not apply to the type of product being monitored, “n/a” (“not applicable”) appears for each statistic. For example, “n/a” appears as the Maximum value for “Session Count” and “Data in Use MB” if Pervasive PSQL v11 SP2 is being monitored.

However, if you are considering using Pervasive PSQL Vx Server, you need the ability to estimate Current and Peak values for “Session Count” and “Data in Use MB.” Consequently, those statistics *are* displayed for Pervasive PSQL v11 SP2 but are *not* enforced. No notifications are sent about them regardless of their values.

Also note that, if a file is immediately closed after you insert a large number of records, Monitor does not immediately reflect the changes in file size. For example, the statistics for “Data in Use MB” and “Physical File Size KB” are not refreshed for that file until the next time the file is opened for reading or writing.

Query Plan Viewer

Query Plan Viewer contains the following changes.

Query Viewer Window

- The window can be scrolled vertically.
- Shortcut keys are provided for the View menu items.
- An option to export an XML file of the query is provided on the File menu.

Plan Viewer Window

- The window can be scrolled both vertically and horizontally, zoomed, and set to several fixed-percent sizes.
- The window displays the name of the query or subquery in parentheses in its title (to be consistent with the Query Viewer window); its cursor changes shape to a hand when the user mouses over “hot” nodes in the tree; and Subquery menu items are dynamically created instead of irrelevant items being disabled.

See [Query Plan Viewer](#) in *SQL Engine Reference*.

Status Codes

The wording, description, or both, for the following status codes has changed to better address both license models (user count and capacity based).

- 161: A key has reached a maximum limit for user count, session count, or data in use, or has changed state to expired or disabled
- 7064: No license obtained for the product
- 7065: A product has not been obtained
- 7102: Illegal license type is defined
- 7110: Only temporary licenses are installed on the system and all of them have expired
- 7124: An increase key for user count, session count, or data in use must have a vendor ID that matches the vendor ID of an authorized permanent key
- 7125: An active permanent license already exists
- 7126: Cannot increase the value beyond the maximum limit for user count, session count, or data in use
- 7475: Invalid value for user count, session count, or data in use
- -4838: A key has reached a maximum limit for user count, session count, or data in use, or has changed state to expired or disabled.

See also [License Models](#) in *Pervasive PSQL User's Guide*.

SDK Access Methods

Pervasive PSQL v11 SP2 includes enhancements to the following SDK access methods:

- [DTI](#)
- [DTO](#)
- [PDAC](#)

DTI

DTI includes the following changed function.

Function	Discussion
PvGetProductsInfo()	<p>The XML returned by the function has changed to accommodate both license models (user count and capacity based).</p> <p>If the product is Pervasive PSQL v11 SP2 Server or Workgroup, the XML elements returned for maximum session count and maximum data in use always return zero. If the product is Pervasive PSQL Vx Server, the XML elements returned for user count and maximum user count always return zero.</p>

DTO

DTO includes the following changed method.

DTO Object	Method	Discussion
DtoLicenseMgr	GetProductInfo	<p>The XML returned by the method has changed to accommodate both license models (user count and capacity based).</p> <p>If the product is Pervasive PSQL v11 SP2 Server or Workgroup, the XML elements returned for maximum session count and maximum data in use always return zero. If the product is Pervasive PSQL Vx Server, the XML elements returned for user count and maximum user count always return zero.</p>

PDAC

Pervasive PSQL v11 SP2 includes Pervasive Direct Access Components (PDAC) for the RAD Studio XE2 development environment. The PDAC components for RAD Studio XE2 are available only in 32 bit.

Features for Original Equipment Manufacturers

See “About This Manual” chapter in *Manufacturing Partner Handbook* for a summary of Pervasive PSQL v11 SP2 features specifically for original equipment manufacturers (OEMs).

What Was New in Pervasive PSQL v11 SP1

An Overview of Features in Pervasive PSQL v11 SP1

The General Release includes new features and changes as summarized in the following topics:

- [Performance Monitoring Counters for Pervasive PSQL](#)
- [Message Logging](#)
- [Installation](#)
- [Utilities](#)
- [Relational Interface](#)
- [SDK Access Methods](#)
- [Linux](#)
- [Macintosh](#)
- [Features for Original Equipment Manufacturers](#)

Performance Monitoring Counters for Pervasive PSQL

Pervasive PSQL v11 SP1 provides performance counters for use with the Windows Performance Monitor utility. The Pervasive PSQL performance counters measure state or activity of the database engine, which allows you to analyze performance of your application. Windows Performance Monitor requests the current value of the performance counters at specified time intervals.

The use of the performance counters applies only to Pervasive PSQL Server running on the following platforms:

- Windows Vista or later, such as Windows 7, Windows Server 2008 and Windows Server 2008 R2.
- Both 32-bit and 64-bit versions of the operating systems are supported.

By default, the Pervasive PSQL installation registers the Pervasive PSQL performance counters with Performance Monitor. The counters are available for use after installation completes.

Note that the use of performance counters is an advanced feature intended primarily for application developers and other technical staff. Refer to the Microsoft documentation for details about the Windows Performance Monitor and on the use of counters in general.

For details about the Pervasive PSQL performance counters, see [Analyzing Performance](#) in *Advanced Operations Guide*.

Message Logging

Pervasive PSQL now supports various logging repositories for messages. The logging falls into two broad categories:

- All messages. These messages include status, error, warning, and information messages. They can originate from any Pervasive PSQL component, including the license administration components.
- Licensing messages. These messages alert you about licensing issues and provide troubleshooting information. They originate from license administration components.

The following table summarizes the repositories.

Repository	Written To By	Notes
Pervasive Notification Viewer	License administration components	New for Pervasive PSQL v11 SP1
Operating System Event Log (Windows)	License administration components	Writing to the Windows Event Log is new for Pervasive PSQL v11 SP1
Operating System Event Log (Linux)	All Pervasive PSQL components	Additional messages pertaining to licensing now logged
Pervasive PSQL Event Log (Windows only)	All Pervasive PSQL components	Additional messages pertaining to licensing now logged

Licensing Messages

As the table above shows, several of the logging repositories emphasize licensing messages. The reason is that the Pervasive PSQL licensing components periodically verify that the key for the database engine is still valid. If a key is determined to be invalid, the key changes state from “active” to “failed validation.” The database engine functions normally for a certain number of days so that you have ample time to correct the failures.

If you do not correct the causes of the failed validation before the number of days ends, the key changes state again to “disabled.” The key is no longer valid and the database engine cannot access data files.

Because you need to attend to a failed validation in a timely manner, the state change of the key is brought to your attention as soon as possible through the various logging repositories.

Note that message logging follows a one-way hierarchy: any licensing message logged to Pervasive Notification Viewer is also logged to the Operating System Event Log and to the Pervasive PSQL Event Log. Similarly, any licensing message logged to the Operating System Event Log is also logged to the Pervasive PSQL Event Log.

Pervasive Notification Viewer

The Pervasive Notification Viewer is an application utility for displaying messages logged by the licensing components. The purpose of the utility is to inform you of noteworthy licensing messages in a noticeable but unobtrusive manner.

Pervasive Notification Viewer is installed by default with Pervasive PSQL Server, 32-bit and 64-bit, on Windows and Linux, and with Pervasive PSQL Workgroup.

On Windows platforms, the executable is named notifyviewer.exe. The Pervasive PSQL installation adds the utility to the Start Menu so that it re-starts when you restart Windows.

On Linux distributions, the utility is a shell script named `notifyviewer`. If you restart Linux, you must restart Pervasive Notification Viewer. The shell script is not automatically executed when Linux is restarted.

Pervasive Notification Viewer provides two interfaces: system tray icons and a graphical user interface (GUI).

See [Pervasive Notification Viewer](#) in *Pervasive PSQL User's Guide* for details.

Operating System Event Log

The Pervasive PSQL license administration components write messages to the operating system event log on Windows platforms. All Pervasive PSQL components write messages to the operating system event log on Linux distributions.

Any licensing message categorized as an “error” or “warning” is logged. In addition, certain information messages are logged.

On Windows platforms, Pervasive PSQL logs licensing messages to the Windows Application Event Log. They are a subset of the licensing messages written to `PVSW.LOG`. Pervasive PSQL Server, Workgroup, and Client 32-bit currently log messages; Pervasive PSQL Client 64-bit does not. See [Windows Platforms Event Logs](#) in *Pervasive PSQL User's Guide* for details.

On Linux distributions, all Pervasive PSQL components write messages to the standard Linux logging system, `syslog`. See [Linux Distributions Event Logs](#) in *Pervasive PSQL User's Guide* for details.

Pervasive PSQL Event Log

On Windows platforms, all Pervasive PSQL components write status, error, warning, and information messages to the Pervasive PSQL event log, `PVSW.LOG`. Any licensing message categorized as an “error” or “warning” is logged. In addition, certain information messages are logged. See [Pervasive PSQL Event Log](#) in *Pervasive PSQL User's Guide* for details.

Related Topics

See also [Utilities](#) and [SDK Access Methods](#).

Installation

Pervasive PSQL v11 SP1 includes the following changes to its installation.

- [Firewalls](#)
- [Pervasive Notification Viewer](#)

Firewalls

Installation actions pertaining to firewalls have changed for Pervasive PSQL Server and Pervasive PSQL Workstation. The change applies only to Windows operating systems. For details, see [Windows FireWalls](#) in *Getting Started With Pervasive PSQL*.

Windows Vista and Later

Windows Vista and later operating systems include Windows Firewall with Advanced Security, which provides firewall profiles (a group of security settings). The following table summarizes the installation actions pertaining to the active profile(s).

Table 6 Installation Actions for Vista or Later Operating Systems

Active Firewall Profile ¹	Rules Added for Pervasive PSQL Services	State of Rules After Installation ²
Multiple, such as <ul style="list-style-type: none"> • Domain • Private • Public 	<ul style="list-style-type: none"> • Domain—Yes • Private—Yes • Public—Yes 	<ul style="list-style-type: none"> • Domain—Enabled • Private—Enabled • Public—Disabled
Public only	Yes	Enabled
¹ “Active” means that the profile is monitoring network connections. ² An “enabled” rule means that inbound TCP and UDP traffic can communicate with the Pervasive PSQL service on all ports for any network connection managed by that firewall profile.		

As the table shows, if the Public profile is active *with one or more other active profiles*, the Pervasive PSQL rules are added for the Public profile but *disabled*. Neither the interactive nor the silent installation of Pervasive PSQL Server or Workgroup can be modified to change this behavior. If you want to enable the rules for the Public profile, you must do so manually. See [To enable Pervasive PSQL rules for the Public profile](#) in *Getting Started With Pervasive PSQL*.



Note A corporate policy may prevent a local administrator from modifying the firewall profiles on a particular machine (that is, the profile is “locked”). If so, the Pervasive PSQL installation cannot add or enable the firewall rules required for the database engine to communicate across a network connection monitored by a locked profile. For such a situation, you should contact a corporate systems administrator and request that the firewall policy be modified to allow inbound TCP and UDP traffic on all ports to communicate with all installed Pervasive PSQL services.

Also be aware that a Group Policy only prevents the installation from adding and enabling rules on

firewall profiles controlled by the Group Policy when the target system is joined to the domain. If the user installing Pervasive PSQL is logged into the target system as a local user instead of as a domain user, the installation **does** add and enable the rules on the firewall profiles. However, the rules are **disabled** if the target system is later joined to the domain controlling the Group Policy.

Windows Operating Systems Prior to Vista

Windows Server 2003 and Windows XP do not include firewall profiles. On these operating system, installation adds each Pervasive PSQL service to the Windows Firewall Exception List. This results in the addition of one or more firewall rules that allow inbound TCP and UDP traffic to communicate with the Pervasive PSQL services on all ports through the Windows Firewall. All ports are used in case you need to change the default ports used by Pervasive PSQL.

Pervasive Notification Viewer

A new utility, Pervasive Notification Viewer, is installed by default with Pervasive PSQL Server, 32-bit and 64-bit, on Windows and Linux, and with Pervasive PSQL Workgroup.

On Windows platforms, the Pervasive PSQL installation adds the utility to the Start Menu so that it restarts when you restart Windows. On Linux distributions, the utility is a shell script. (See also [Pervasive Notification Viewer](#) in “Message Logging” section.)

Ptksetup.ini contains new properties to control whether Pervasive Notification Viewer is installed. See [PTKSetup.ini](#) in *Installation Toolkit Guide*.

Utilities

Pervasive PSQL v11 SP1 includes new and changed utilities.

- [License Administrator](#)
- [Pervasive Notification Viewer](#)
- [Pervasive System Analyzer \(PSA\)](#)

License Administrator

License Administrator provides new functionality.

Feature	Discussion
Authorizations remaining for a key	A new action has been added to check the number of authorizations remaining for a key. See To Display Remaining Authorizations (GUI) and To Display Remaining Authorizations (CLI) in <i>Pervasive PSQL User's Guide</i> .
Validation action	A new action to validate keys has been added. In License Administrator, a validation action updates the state of all keys. The allows you to see if a state has changed (see next row in this table). See To Perform a Validation (GUI) and To Perform a Validation (CLI) in <i>Pervasive PSQL User's Guide</i> . See also DTI .
New state for a key and period allowed for correction of failures	A new state for a key, "failed validation," has been added. Pervasive PSQL licensing components periodically verify that the key for the database engine is still valid. If a key is determined to be invalid, the key changes state from "active" to "failed validation." The database engine functions normally for a certain number of days so that you have ample time to correct the validation failure(s). By default, the number of days is 14. If you do not correct the failures before the number of days ends, the key changes state again to "disabled." The key is no longer valid and the database engine cannot access data files. (The number of days is automatically extended, if necessary, so that a key changes state to "disabled" only from Tuesday through Thursday in the local time zone.) See Key Status, or State <i>Pervasive PSQL User's Guide</i> .

Pervasive Notification Viewer

Pervasive Notification Viewer is a new tray application utility for displaying messages logged by the licensing components. See [Pervasive Notification Viewer](#) in "Message Logging" section.

Pervasive System Analyzer (PSA)

PSA now supports numeric and named IPv6 addresses for its network test. Note that Pervasive PSQL supports only the unicast types for IPv6 loopback, global, and link local addresses. For additional details, see [Support for IPv6](#) in [What Was New in Pervasive PSQL v11](#).

Relational Interface

This section discusses the new and revised functionality to support the relational interface.

- [System Catalog Functions](#)
- [Scalar Functions](#)

System Catalog Functions

Pervasive PSQL v11 SP1 includes the following new system catalog functions:

- `dbo.fSQLColumns`
- `dbo.fSQLForeignKeys`
- `dbo.fSQLPrimaryKeys`
- `dbo.fSQLProcedures`
- `dbo.fSQLProcedureColumns`
- `dbo.fSQLSpecialColumns`
- `dbo.fSQLStatistics`
- `dbo.fSQLTables`

The system catalog functions allow you to obtain database metadata from the data dictionary files (also known as the catalog) and can be used only in a FROM clause. A temporary view schema for each system catalog function is created during SQLPrepare and data is stored in the view by calling a corresponding ODBC catalog API during SQLExecute.

See the chapter [System Catalog Functions](#) in *SQL Engine Reference*.

Scalar Functions

The following scalar functions are either new, have additional parameters, or were previously undocumented.

Function	Description	See in <i>SQL Engine Reference</i>
DAY (<i>date_exp</i>)	Returns the day of the month for the given <i>date_exp</i> .	Time and Date Functions
DATEPART (<i>datepart, date_exp</i>)	TZoffset is an addition to the values that can be used for <i>datepart</i> . TZoffset returns a time zone offset in number of minutes (signed).	Time and Date Functions
ISNUMERIC (<i>string</i>)	Returns 1 (TRUE) if <i>string</i> can be evaluated as a numeric value; otherwise returns 0 (FALSE).	String Functions
REVERSE (<i>string</i>)	Returns a character string with the order of the characters in <i>string</i> reversed.	String Functions

Function	Description	See in <i>SQL Engine Reference</i>
SYSDATETIMEOFFSET	Returns the current date and time along with the hour and minute offset between the current time zone and UTC of the computer on which the Pervasive PSQL database engine is running.	Time and Date Functions
WEEKDAY (<i>date_exp</i>)	Returns the day of the week for the given <i>date_exp</i> , where 1=Sunday and 7=Saturday.	Time and Date Functions

SDK Access Methods

Pervasive PSQL v11 SP1 includes enhancements to the following SDK access methods:

- [Btrieve API](#)
- [DTI](#)
- [DTO](#)
- [PDAC](#)

Btrieve API

The Pervasive PSQL downloads now include a sample application and files to show the use of the Btrieve API in the VB.NET environment. This samples are available on the Pervasive web site at <http://www.pervasivedb.com>.

DTI

DTI includes new and changed functions.

Functions	Description	New or Changed
PvValidateLicenses()	Initiates a validation action to check validity of keys	New
PvGetProductsInfo()	Retrieves a list of all Pervasive Software products found by the License Manager	Changed. Includes additional information about licensing in returned XML string, such as failed-validation period



Note [PvValidateLicenses](#) returns only the result from *requesting* a validation action. It does *not* return any information about the state of the keys. You must separately call [PvGetProductsInfo\(\)](#) to get the XML string of product information that includes information about the state of the keys.

DTO

The Pervasive PSQL DTO SDK now includes support for 64-bit environments, including the .NET framework. To use DTO with your 64-bit applications, you need to install Pervasive PSQL Server 64-bit or Client 64-bit to obtain the 64-bit library [W64DTO2.DLL](#). See [W64DTO2](#) in *Distributed Tuning Objects Guide*.

If you use DTO with your .NET applications compiled as “AnyCPU,” the 64-bit version of DTO allows your application to make DTO calls on both 32-bit and 64-bit operating systems. See also [Understanding How Applications and DLLs Interact](#) in *Distributed Tuning Objects Guide*.

DTO also includes new and changed methods.

DTO Object	Method	Description	New or Changed
DtoDatabase	RemoveUserFromGroup	Removes an existing user from an existing group	New
DtoLicenseMgr Object	ValidateLicenses	Initiates a validation action to check validity of all keys	New
DtoLicenseMgr Object	GetProductInfo	Retrieves a list of all Pervasive Software products found by the License Manager	Changed. Includes additional information about licensing in returned XML string, such as failed-validation period



Note `ValidateLicenses` returns only the result from *requesting* a validation action. It does *not* return any information about the state of the keys. You must separately call `GetProductInfo` to get the XML string of product information that includes information about the state of the keys.

PDAC

Pervasive PSQL v11 SP1 includes Pervasive Direct Access Components fully integrated with the RAD Studio XE development environment. See also *Pervasive Direct Access Components Guide* in the SDK documentation.

Linux

Pervasive PSQL v11 SP1 includes the following support for the Linux operating system:

- [64-bit ODBC Driver](#)
- [Red Hat Enterprise Linux 6](#)

64-bit ODBC Driver

Pervasive PSQL v11 SP1 supports the ODBC interface for 64-bit applications on Linux. The 64-bit ODBC driver is installed with Pervasive PSQL Server 64-bit and Pervasive PSQL Client 64-bit.

On Linux distributions, individual ODBC drivers are loaded through the driver manager UNIXODBC. The driver manager maintains a mapping from Data Source Names (DSNs) to the specific Pervasive PSQL ODBC drivers.

The installation of Pervasive PSQL Server 64-bit or Client 64-bit leaves user-defined, pre-existing 32-bit DSNs as is, meaning they are not immediately accessible from a 64-bit application. For new DSNs, the installation of both products assigns a 32-bit and a 64-bit ODBC driver in `odbcinst.ini`. This assignment allows a single DSN to be used by both a 32-bit application and a 64-bit application.

If you want pre-existing 32-bit DSNs to be accessible to both 32-bit and 64-bit applications, you must recreate them as new-style DSNs (see [Pervasive ODBC Interface](#) in Table 7).

ODBC and Data Source Names (DSNs)

The client application bitness does not have to match the bitness of the Pervasive PSQL Server product. For example, the 64-bit ODBC driver or the 32-bit ODBC driver can be used to connect to either Pervasive PSQL Server 64-bit or Pervasive PSQL Server 32-bit.

For the Pervasive PSQL Client, however, the application bitness does have to match the bitness of the Pervasive PSQL Client on the client machine. That is, if you want to use a 64-bit application on the client, the Pervasive PSQL 64-bit Client must be installed.

Pervasive PSQL v11 provides three ODBC drivers, as explained in the following table.

Table 7 Pervasive PSQL ODBC Driver Descriptions for Linux

ODBC Driver Description in INI Files	PSQL Product Installed With	Behavior for All Products Installed With
Pervasive ODBC Engine Interface	Server 64-bit Server 32-bit	<ul style="list-style-type: none"> • Installation assigns an ODBC driver in odbcc.ini for compatibility with pre-existing Engine DSNs • Connects to a local named database • No longer created by default with dsnadd utility (the -engdsn option must be specified). See also Utilities Affected by ODBC Changes. • Visible when viewing "Engine DSNs" in ODBC Administrator running on a Windows client • For use by 32-bit applications that are already coded to use Engine DSNs • 32-bit Engine DSNs deprecated in Pervasive PSQL v11. See What Was New in Pervasive PSQL v11 SP1.
Pervasive ODBC Client Interface	Server 64-bit Server 32-bit Client 32-bit	<ul style="list-style-type: none"> • Installation assigns an ODBC driver in odbcc.ini for compatibility with pre-existing Client DSNs • Connects to a local or remote named database or an Engine DSN • For use by 32-bit applications that are already coded to use Client DSNs • No longer created by default when -host is specified with dsnadd (the -clntdsn option must be specified). See also Utilities Affected by ODBC Changes.
Pervasive ODBC Interface	Server 64-bit Client 64-bit	<ul style="list-style-type: none"> • Installation assigns ODBC drivers in odbccinst.ini for use by new 32-bit and 64-bit DSNs • Connects to a local or remote named database • The recommended driver description to use for 32-bit and 64-bit applications • Created by default with dsnadd utility. See also Utilities Affected by ODBC Changes. • Not visible as "Engine DSNs" in ODBC Administrator running on a Windows client

Frequently Asked Questions

The following table answers some frequently asked questions (FAQs) about the ODBC and DSN support for Linux in Pervasive PSQL v11 SP1.

Table 8 FAQs About ODBC and DSN Support for Linux

Question	Answer
Have 32-bit Engine DSNs been deprecated?	Yes. Pervasive recommends, rather than using Engine DSNs, that new applications or revised 32-bit applications create a DSN that connects to a named database. See also What Was New in Pervasive PSQL v11 SP1 .
What is a so called "DSN-less" connection?	A DSN-less connection is one that connects to a named database using the ODBC driver "Pervasive ODBC Client Interface" (for pre-existing 32-bit applications) or the "Pervasive ODBC Interface" driver (for 32-bit or 64-bit applications).
What happens to my existing 32-bit Engine DSNs when I upgrade to Pervasive PSQL v11 SP1 Server or Client?	No migration steps are required. Existing user-defined 32-bit Engine DSNs remain in place and continue to work as configured. Applications on the PSQL Server or Client machine continue to work with 32-bit Engine DSNs. An exception to this is the DEMODATA sample database installed with Pervasive PSQL Server 64-bit. The DSN for DEMODATA is recreated during installation using "Pervasive ODBC Interface" to connect to a named database. If you want pre-existing 32-bit DSNs to be accessible to both 32- and 64-bit applications, you must recreate them using "Pervasive ODBC Interface" to connect to a local or remote named database.
What happens to my existing 32-bit Client DSNs when I upgrade to Pervasive PSQL v11 SP1 Client?	No migration steps are required. Existing Client DSNs continue to connect to remote Engine DSNs. Note, however, the recommendation is that new applications or revised 32-bit applications should connect to a named database, not to an Engine DSN since Engine DSNs are deprecated.
Can I still create Engine DSNs and Client DSNs?	Yes, but you must specify the <code>-engdsn</code> or <code>-clntdsn</code> option with <code>dsnadd</code> . See Utilities Affected by ODBC Changes . DSNs created with either option support only 32-bit applications.
What about connections from PSQL Clients of previous releases (such as a PSQL v10.x Client)?	Pervasive PSQL still supports remote Client DSNs, so clients from previous versions can still connect. Note, however, Engine DSNs are only 32-bit for both Pervasive PSQL Server 32-bit and 64-bit. 64-bit Engine DSNs cannot be created with Pervasive PSQL.
What are the ODBC connection strings for Pervasive PSQL DSNs?	See ODBC Connection Strings in <i>SQL Engine Reference</i> .
What do I need to do about DSNs if I port my 32-bit application to 64-bit?	You must create new DSNs that connect to a named database using "Pervasive ODBC Interface."

Table 8 FAQs About ODBC and DSN Support for Linux continued

Question	Answer
What about the DSNs for the Demodata sample database installed with the database engine?	The installation of Pervasive PSQL Server 64-bit or Client 64-bit creates the DEMODATA DSN using “Pervasive ODBC Interface” to connect to a named database. This means that clients using Windows ODBC Administrator no longer see DEMODATA in the list of Engine DSNs from the Pervasive PSQL Server. The clients have to view the Database Names list instead. A 64-bit application would not be able to connect to DEMODATA through an Engine DSN.
What if my application uses DTI to manage DSNs?	The DTI functions for DSNs manage only 32-bit Engine DSNs. Therefore, the DTI functions for DSNs are deprecated along with the 32-bit Engine Interface ODBC driver. See What Was New in Pervasive PSQL v11 SP1 . (An alternative to using DTI to manage DSNs is to use SQL and ODBC statements. For example, you could use CREATE DATABASE to create a named database and SQLConfigDataSource to configure the DSNs.)
Does Pervasive PSQL v11 SP1 include new utilities to support 64-bit ODBC and DSNs?	Yes. See Utilities Affected by ODBC Changes .
Are there any changes to existing utilities to support 64-bit ODBC and DSNs?	Yes. See Utilities Affected by ODBC Changes .
Do some descriptor fields that can be set through the various ODBC SQLSet and SQLGet functions accommodate 64-bit values while others are still 32-bit values?	Yes, if you are using the 64-bit ODBC driver. Ensure that you use the appropriate sized variable when setting and retrieving descriptor fields. For more information, see http://www.unixodbc.org/doc/ODBC64.html . A point of clarification is that <code>SQL_ROWSET_SIZE</code> is supported by both <code>SQLGetStmtOption</code> and <code>SQLGetStmtAttr</code> . If you are using the 64-bit ODBC driver and you call either <code>SQLGetStmtOption</code> or <code>SQLGetStmtAttr</code> , a 64-bit value is returned in <code>*ValuePtr</code> when that attribute parameter is set to <code>SQL_ROWSET_SIZE</code> .
Going forward, is there a recommended strategy for ODBC connections?	Yes. New applications or revised 32-bit applications, local or remote, should connect to a named database. This positions your application for the future when Engine DSNs will no longer be supported in Pervasive PSQL. (Note that you can manage DSNs through the use SQL and ODBC statements. For example, you could use CREATE DATABASE to create a named database and SQLConfigDataSource to configure the DSNs.)

ODBC Header Files

The `sql.h`, `sqltypes.h`, and `sqlxt.h` header files for ODBC contain differences for the compilation of 32-bit and 64-bit applications. Refer to the ODBC documentation on the UNIXODBC Web site for a discussion of 64-bit ODBC. For example, you may find the following information useful: <http://www.unixodbc.org/doc/ODBC64.html>.

Utilities Affected by ODBC Changes

- A 64-bit version of isql, isql64, is installed with Pervasive PSQL Server 64-bit and Client 64-bit. See [isql](#) in *Pervasive PSQL User's Guide*.
- The dsnadd utility has been changed to create a DSN using “Pervasive ODBC Interface” by default. This style DSN allows a single DSN to be used by both a 32-bit application and a 64-bit application. The DSN is not visible as an Engine DSN in ODBC Administrator, and the string "Pervasive ODBC Engine Interface" is not present in the odbc.ini file.

If you want to create a deprecated, legacy-style Engine DSN or a Client DSN, you must specify the -engdsn or -clntdsn option, respectively. See [dsnadd](#) in *Pervasive PSQL User's Guide*.

- The option to create a DSN on the New Database dialog Pervasive PSQL Control Center (PCC) is qualified for 32-bit: “Create 32-bit Engine DSN.” See [New Database GUI Reference](#) in *Pervasive PSQL User's Guide*.

PCC is a 32-bit application; a 64-bit version of it is not available. See also [What Was New in Pervasive PSQL v11 SP1](#).

- The Pervasive ODBC DSN setup GUIs have changed for Windows platforms. See [What Was New in Pervasive PSQL v11 SP1](#).

Red Hat Enterprise Linux 6

Pervasive PSQL supports Red Hat Enterprise Linux 6. Note that, by default, the 64-bit edition of this Linux distribution supports only 64-bit components and executables. The distribution media for 64-bit Red Hat Enterprise Linux 6 includes the 32-bit libraries, but *they are not installed by default*.

The following table summarizes installation notes specific to Red Hat Enterprise Linux 6, including a discussion of certain 32-bit libraries required for Pervasive PSQL.

Pervasive PSQL Product	Installation Notes for Red Hat Enterprise Linux 6 64-bit
<ul style="list-style-type: none"> • 64-bit Server • 32-bit Server • 32-bit Client 	<p>The Pervasive PSQL installation RPM and TAR files check for the existence of the 32-bit GLIBC and LIBSTDC++ libraries. If these libraries are not present, the Pervasive PSQL installation stops and displays a message. To install Pervasive PSQL, you must first install glibc.i686 and libstdc++.i686.</p> <p>The Pervasive PSQL installation RPM and TAR files also check for the existence of 32-bit GTK. If GTK is not present, Pervasive PSQL displays a warning message. The message informs you that PCC, DDF Builder and GUI Phone Authorization utilities will not function until 32-bit GTK is installed. The Pervasive PSQL installation continues and completes after displaying the message.</p> <p>To ensure that PCC, DDF Builder and GUI Phone Authorization utilities work, install the following Linux packages:</p> <ul style="list-style-type: none"> • gtk2.i686 • PackageKit-gtk-module.i686 • libcanberra-gtk2.i686 • gtk2-engines.i686. <p>For example:</p> <pre>yum install gtk2.i686 PackageKit-gtk-module.i686 libcanberra-gtk2.i686 gtk2-engines.i686</pre> <p>Note that, except for the three utilities just mentioned, Pervasive PSQL is functional even if the 32-bit GTK is not present. That is, the database engine is functional, all command line utilities work and the Pervasive PSQL libraries are present and accessible by your applications. Your client applications should still function as long as they do not require the 32-bit GTK.</p>
<ul style="list-style-type: none"> • 64-bit Server • 32-bit Server 	<p>The Pervasive PSQL installation attempts to configure Samba. However, by default, Samba is only partially installed and is not enabled on Red Hat Enterprise Linux 6.</p> <p>You may ignore installation errors pertaining to creating the PSQDATA Samba share (the smb.conf file exists, but nothing is there to support it). The errors are benign and Pervasive PSQL functions normally.</p>

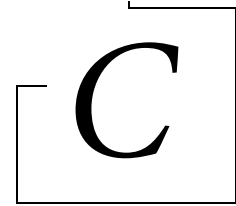
Macintosh

Pervasive PSQL v11 SP1 supports 64-bit Btrieve and 64-bit DTI on the Mac OS X 10.5 and 10.6 operating systems.

Features for Original Equipment Manufacturers

See “About This Manual” chapter in *Manufacturing Partner Handbook* for a summary of Pervasive PSQL v11 SP1 features specific for original equipment manufacturers (OEMs).

What Was New in Pervasive PSQL v11



An Overview of Features in Pervasive PSQL v11

The General Release includes the following new features and changes:

- [Multi-core Support](#)
- [Support for IPv6](#)
- [64-bit ODBC Driver](#)
- [Support for .NET Framework 3.5 SP1 and 4.0](#)
- [PDAC Development Environments](#)
- [Enhancements to Other SDK Access Methods](#)
- [Product Authorization](#)
- [Configuration Settings](#)
- [Utility Changes](#)
- [Deprecated and Discontinued Features](#)

Multi-core Support

Pervasive PSQL v11 is specifically designed to increase scalability and performance on multi-core machines. Install Pervasive PSQL v11 on a multi-core machine and the benefits are immediately available in a multiuser environment.

You may wonder “what benefits?” Increased scalability and performance are obviously desirable and assumed to be available with advances in hardware technology. Heretofore, advances in hardware technology meant advances in speed. Applications just ran faster. Today, advances in computing technology mean increased parallelism and not increased clock speeds. And that presents challenges to which your application has probably never had to contend.

The rules have not just changed because of multi-core environments, they have changed dramatically. For example, applications that share data with multiple users and use a database where transactional integrity must be maintained can run *slower* on multi-core processors.

Because the majority of applications using Pervasive PSQL fall into that category, multi-core support is a primary feature of Pervasive PSQL v11. It is of primary importance to you as you transition your multiuser applications into multi-core environments.

Why Multi-core Support

Without modifications, almost all software applications can run on multi-core machines. But consider the following scenario, which is based on real-world feedback:

You replace your antiquated production server with a current one. Your multiuser application gets installed on the new multi-core machine with a compatible operating system. Things should be humming better than ever. But response time is slower. Performance is worse than before the hardware upgrade.

What happened? Critical components of your business solution are no longer optimized for one another in the new world of multi-core.

Think of it this way. Your “application” comprises four main pieces: the code you wrote (application in its common definition), the database, the operating system, and the hardware. Changing the hardware has a significant impact if it fundamentally differs from its predecessors.

But tuned in the right way, applications that would otherwise be slowed down can take advantage of hardware changes and experience significant performance improvement. In many cases, swapping out portions of the application stack, such as the database, can address multi-core issues with no immediate changes required to the application. This approach provides as a low risk way to buy time while you plan longer-term strategies for application development.

Using Pervasive PSQL v11 as the database, you can realize increased performance and scalability on multi-core machines.

Performance

Pervasive PSQL v11 has been architected to provide parallel threads performing similar activities. The gains in increased parallel processing improve the throughput to the point that multiple processors are engaged. The result is that performance of the database engine *increases* in multi-core environments with multiple clients accessing a central server. Your multi-client application can benefit from this increased performance without requiring you to recompile or rearchitect the code.

Pervasive PSQL v11 also provides enhancements to the low-level synchronizations mechanisms in the transactional interface. Multiple users can read the same cached file pages simultaneously and their operations can proceed on independent server CPUs. Non-user activity such as checkpoints and log management can also use additional server CPUs.

Scalability

The scalability of Pervasive PSQL v11 has also been enhanced through architecture designs made specifically for multi-core hardware. For example, multiple users accessing independent files can proceed on independent server CPUs. The database engine can also handle higher user loads with less overhead, resulting in steadier throughput.

Just as with the performance improvements, all of the scalability enhancements are available without requiring you to recompile or rearchitect your code.

Configuration Settings

The majority of multi-core improvements in Pervasive PSQL v11 are transparent. You are not required to adjust any settings to further enhance the optimizations. The configuration setting “Communications Threads” has changed and can be used to fine-tune performance if you choose. See [Configuration Settings](#).

The Multi-core Dilemma

Several common problems are at play in the multi-core world of hardware and software interaction that may cause *decreased* performance with your application. Among them are multiple threads and memory contention. For a thorough discussion of these and other problems, refer to the white paper *The Multi-core Dilemma* by Dan Woods, CTO of CITO Research. The white paper is available on the Pervasive Web site.

A brief discussion in this document of multiple threads and memory contention illustrates why multi-core support is a primary feature of Pervasive PSQL v11.

Multiple Threads

A multithreaded application does not necessarily run better on a multi-core machine. In fact, you may find that your multithreaded application runs slower.

To work correctly in parallel, the threads must be synchronized. An application can be multithreaded, but the threads themselves not synchronized. This situation is actually quite common, in which older applications spin off additional threads as needed, more for convenience than based on a design to ensure efficiency. Such applications do not run better on a multi-core machines because the threads contend with one another. Multiple cores provide no benefit because thread contention inhibits throughput to the point that multiple cores are not engaged.

Also, the multi-core architecture can perceive the subtasks that spin off the multiple thread as a series of single threads. And, just as with single-threaded programs, the threads are then forced into a single queue and processed one by one. Caching does not improve the problem; it makes it worse (see [Memory Contention](#)).

Where possible, each core should process separate data. Otherwise, the overhead associated with synchronization can slow down performance significantly. Recall that Pervasive PSQL v11 has been architected to provide parallel threads that are synchronized.

Memory Contention

When most applications were written, developers did not have to decide between parallel and non-parallel processes. The majority of applications were written sequentially, meaning that they access information serially or sequentially. A problem with memory contention occurs when running a non-parallel (typical) application on a multi-core system.

Consider the slapstick comedy skit that depicts a group of people trying to get through a single doorway at the same time. This is good for laughs because the individuals just jam together at the opening, wedged into an immovable mass. Now, image that, instead of people and a doorway, it is multiple threads trying to be processed at the same time. With four to sixteen threads (or more) trying to get through the same processor at once, a jam occurs that the operating system must sort out.

If multiple cores or processors have caches that point to the same data and one core modifies the data, the cached data on the other core is no longer valid, and the caches must be synchronized. Contention also occurs as the processors repeatedly check the caches to ensure a task on one processor does not execute on outdated data produced by another task on another processor. This checking slows processing because each processor checks the memory cache individually and sequentially.

Recall that with Pervasive PSQL v11 activities of multiple users proceed on independent server CPUs as a way to reduce memory contention. Multiple users can read the same cached file pages simultaneously and access independent files.

The Role of the Operating System

You may be wondering how much the operating system (OS) assists with the problems of multiple cores. Less than you would guess, even with current 64-bit ones.

When contention for resources happens, the OS handles the resolution. For the majority of applications, the OS handles thread contention slower on multi-core systems. That is, the OS on multi-core systems take a longer time to resolve the contention points.

Why is this? An OS optimized for multi-core does not fix your problems if your applications still require the operating system to perform tasks in a single-file fashion.

When the OS gets requests from an application that do not incorporate instructions for multi-core processing, the OS is very cumbersome at sorting out the sequence in which the requests are processed. This is analogous to a traffic jam on a highway. Conceptually, the OS asks *each* waiting driver whether or not they are ready to go before allowing the vehicle to proceed. Although such processing jams are occurring at the OS level, users perceive the slowdown as an application performance problem.

An application optimized for multi-core provides instructions for the OS on how to manage shared resources and determine priority for access to those resources. Information requests are organized in such a way that they do not compete for cache lines or access to central memory.

Recall that Pervasive PSQL v11 includes architecture designs made specifically for multi-core hardware. Low-level locking has been optimized for multi-core machines.

Benefiting from the Present While Planning For the Future

Multi-core machines are the norm, so any current or future hardware upgrades will include multiple cores. Operating systems have yet to catch up with multi-core machines to assist optimal performance. How best, then, to address these conditions?

Ultimately, applications will have to be rearchitected to perform optimally on multi-core machines. This allows the application to take advantage of parallel threads on multiple processors while avoiding synchronization issues.

Rearchitecting takes thoughtful planning and time to implement, perhaps even years. Meanwhile, business continues. As mentioned at the beginning of this section, multi-core support becomes of primary importance to you as you transition your applications into multi-core environments.

Your “application” consists of your code, the database, the operating system, and hardware. Hardware systems have already addressed multi-core support. Operating system provide some assistance provided your application takes advantage of the multiple cores. That leaves the database.

The multi-core features of Pervasive PSQL v11 can help offset any performance degradation your end users might experience from your application not being optimized for multi-core environments. In most cases, you can boost application performance without having to recompile or change your application code.

Support for IPv6

Internet Protocol version 6 (IPv6) is the next-generation Internet Protocol version designated as the successor to IPv4. This section discusses the following topics:

- [Using Pervasive PSQL With IPv6](#)
- [Frequently Asked Questions About IPv6 Support](#)
- [Pervasive PSQL Utilities and IPv6](#)
- [IPv6 Aspects for Application Programmers](#)

Using Pervasive PSQL With IPv6

Pervasive PSQL v11 supports IPv6 for the following access methods on Windows operating systems:

- Transactional (also known as Btrieve)
- DTI (Distributed Tuning Interface)

Both access methods function correctly in an IPv4 environment, an IPv6 environment, or an environment that combines the two. No special configurations of Pervasive PSQL are required.

Client Connections

A Pervasive PSQL Client connects to a IPv6 host running the Pervasive PSQL database engine the same way as for IPv4. That is, the Client specifies a server and connects through DTI or by specifying a URI or UNC. The server can be either the name or IP address of the machine running Pervasive PSQL Server or Workgroup.

See also the following:

- [Database URIs](#) in *Pervasive PSQL Programmer's Guide*.
- [Universal Naming Convention \(UNC\) Path Formats](#) in *Getting Started With Pervasive PSQL*.
- [Making a Connection to a Server Using DTI](#) in *Distributed Tuning Interface Guide*.

The following topics in this subsection discuss how to specify a server using IPv6 addresses.

IPv6 Address Formats

Raw IPv6 addresses can be written as 8 colon-separated segments where each segment is a 4-digit hexadecimal value. For example, 1234:5678:90ab:cdef:1234:5678:90ab:cdef.

Pervasive PSQL supports only unicast addresses. The following are the unicast address formats that can be used with Pervasive PSQL.

Table 9 IPv6 Unicast Address Formats Supported by Pervasive PSQL

Unicast Address Format	Description
Loopback	The local loopback address, which in IPv6 is 0:0:0:0:0:0:0:1. The loopback address can be abbreviated to ::1. The IPv6 loopback address is equivalent to the IPv4 loopback address of 127.0.0.1.
Global	Global addresses have a 64-bit prefix where the first 3 bits are always 001, the next 45 bits are set to the global routing prefix, the next 16 bits are set to the subnet ID and the last 64-bits are the interface ID. Example: 2001:db8:28:3:f98a:5b31:67b7:67ef
Link Local	Link Local addresses are used by nodes when communicating with neighboring nodes on the same link. Link Local addresses have a 64-bit prefix where the first 10 bits are set to 1111 1110 10, the next 54 bits are set to 0 and the last 64 bits are the interface ID. The link local prefix is often represented as FE80::/64. Example: fe80:0:0:0:713e:a426:d167:37ab (which may also be specified as fe80::713e:a426:d167:37ab) See also Restrictions .

IPv6 Address Modifiers

IPv6 includes address modifiers which can act as shortcuts, or to specify the destination in more detail. Pervasive PSQL supports the following ones for IPv6.

Modifier	Explanation
::	Represents one or more colon-separated zeroes. For example, ::1 is equivalent to 0:0:0:0:0:0:0:1. The :: modifier can be used only once within an IPv6 address.
%	Represents the ZoneID or interface of a destination node. A ZoneID is an integer that specifies the zone of the destination for IPv6 traffic. ZoneIDs are primarily used with Link Local addresses to disambiguate those addresses See Restrictions .

IPv6 With UNC Paths and URI Connections

UNC paths do not allow certain special characters, such as colons. Since raw IPv6 addresses use colons, different methods of handling UNC paths are available. Pervasive PSQL supports the following methods:

- [IPv6-literal.net Names](#)
- [Bracketed IPv6 Addresses](#)

IPv6-literal.net Names

An ipv6-literal.net name is a raw IPv6 address with three changes:

- ":" is replaced with "-"
- "%" is replaced with "s"
- The whole address is appended with ".ipv6-literal.net"

Examples:

Initial Addresses	fe80::713e:a426:d167:37ab%4 2001:db8:28:3:f98a:5b31:67b7:67ef
Modified Addresses	fe80--713e-a426-d167-37abs4.ipv6-literal.net 2001-db8-28-3-f98a-5b31-67b7-67ef.ipv6-literal.net

Ipv6-literal.net names are allowed in a URI or UNC used with Pervasive PSQL.

Bracketed IPv6 Addresses

A bracketed IPv6 address is a raw IPv6 address with square brackets around it. This format is also referred to as a UNC-safe address.

Examples:

Initial Addresses	fe80::713e:a426:d167:37ab%4 2001:db8:28:3:f98a:5b31:67b7:67ef
Modified Addresses	[fe80::713e:a426:d167:37ab%4] [2001:db8:28:3:f98a:5b31:67b7:67ef]

The use of square brackets is required for raw IPv6 addresses used in a URI or UNC with Pervasive PSQL. See [Restrictions](#). Note that if you use an address with a ZoneID in a URI, the ZoneID character “%” must use the escape characters “%25.” See [Restrictions](#). Support for bracketed IPv6 or UNC-safe addresses in UNC paths is absent on Windows XP and Windows 2003 operating systems.

Restrictions

The following table lists the restrictions on the use of IPv6 with Pervasive PSQL.

Table 10 IPv6 Restrictions With Pervasive PSQL

Restriction	Discussion
The Pervasive PSQL Server Engine in an IPv6-only environment	The Pervasive PSQL Server Engine is not supported in an IPv6-only environment on Windows Server 2003 or Windows XP operating systems. The Server Engine is supported in an IPv6-only environment on Windows Vista and later operating systems.
Square brackets are required for raw IPv6 addresses when the address is used in a URI or UNC	Raw IPv6 addresses, abbreviated or not, must be enclosed by square brackets if the address is used in a URI or UNC. Examples: <ul style="list-style-type: none"> • btrv://czjones@[2001:b1::23]/demodata • btrv://abanderas@[2001:12:34:56:78:90:12:23]/demodata • \\[2001:12:34:56:78:90:12:23]\acctsvr1\Domestic\file.mkd Failure to bracket the IPv6 address results in status code 3014 or 3103 for Btrieve calls using a URI, or status code 11, 94, or 170 for Btrieve calls using a UNC.

Table 10 IPv6 Restrictions With Pervasive PSQL

Restriction	Discussion
In a URI, if you include a ZoneID to a server address, the “%” ZoneID character must be escaped with “%25”	<p>If you use a <code>btrv://</code> connection with an IPv6 address, you must escape the ZoneID for the host name. Zone IDs are usually required with IPv6 Link Local numeric addresses.</p> <p>Example:</p> <p>A UNC-safe addresses like</p> <pre>btrv://@[fe80::20c:29ff:fe67:2ee4%4]</pre> <p>must be changed to</p> <pre>btrv://@[fe80::20c:29ff:fe67:2ee4%254]</pre>
PCC usage in an IPv6-only environment	In an IPv6-only environment, PCC allows only the functionality supported by the transactional or DTI access methods. For example, you can connect a PSQL Client from an IPv6-only machine to a database engine on an IPv6-only server machine. PCC allows you to view and set Engine and Client properties because those features use DTI. However, you cannot browse databases or use Table Designer because those feature use other access methods, such as the relational interface, which are not yet supported for IPv6.
License Administrator (and clicadm)	The Pervasive licensing server does not yet support IPv6. Because of this, you can use License Administrator over IPv6 to administer licenses but you cannot authorize a license with the utility. To authorize a license, you must use an IPv4 network, remote authorization, or telephone authorization.

Frequently Asked Questions About IPv6 Support

The following table answers some frequently asked questions (FAQs) about IPv6 support in Pervasive PSQL v11.

Table 11 FAQs About IPv6 Support

Question	Answer
Can I use Pervasive Auto Reconnect (PARC) with IPv6?	Yes.
Does Pervasive PSQL support IPv6 communications in virtual machine environments?	Yes.
Does IPv6 support apply to the relational access method (SRDE)?	No. Only the transactional and DTI access methods are supported.
Is IPv6 supported for Linux distributions or Macintosh OS X?	No. Only Windows platforms are supported.
Is IPv6 supported for Pervasive DataExchange, AuditMaster, and Backup Agent?	No.
Does a network environment that includes both IPv4 and IPv6 affect Pervasive PSQL user counts?	No. Pervasive PSQL Server or Workgroup uses one user count for each <i>unique</i> incoming protocol from the same client computer session (such as TCP/IP and SPX). IPv4 and IPv6 are just different address formats of TCP/IP.
Can the Listen IP Address configuration setting be set to multiple addresses?	Yes. See Listen IP Address .

Pervasive PSQL Utilities and IPv6

The following Pervasive PSQL utilities support IPv6. No special configuration of them is required.

Utility	See Also
bcfg	Configuration Reference in <i>Advanced Operations Guide</i>
Function Executor	Testing Btrieve Operations in <i>Advanced Operations Guide</i>
License Administrator (and clicadm)	License Administration in <i>Pervasive PSQL User's Guide</i> See Restrictions .
Monitor (and bmon)	Monitoring Database Resources in <i>Advanced Operations Guide</i>
Pervasive PSQL Control Center (PCC)	Using Pervasive PSQL Control Center in <i>Pervasive PSQL User's Guide</i> If you are using PCC in an IPv6-only environment, see Restrictions .

IPv6 Aspects for Application Programmers

Because IPv6 has not been widely adopted, the section discusses a few aspects of it that an application programmer may want to investigate further. The intent is not to explain in detail networking concepts or IPv6, but to provide a very brief introduction to IPv6. For a complete discussion of IPv6, see the IPv6 specification at www.ipv6.org, and refer to the IPv6 documentation from the various operating system vendors and network hardware vendors.

Importance of IPv6

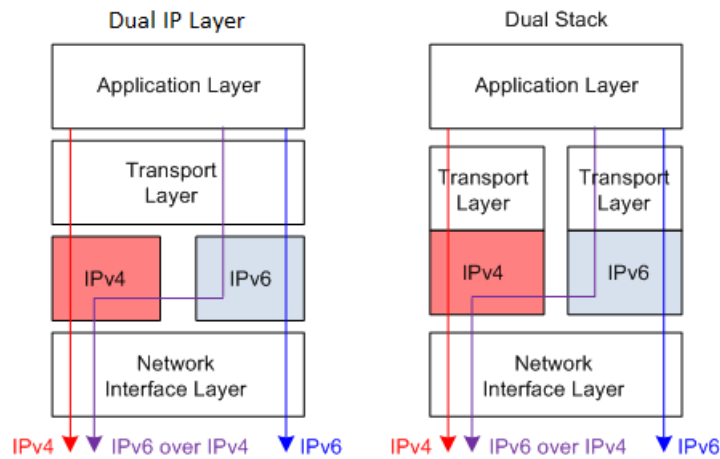
IPv6 is the next-generation Internet Protocol version designated as the successor to IPv4. IPv4 was the first implementation used in the Internet that is still in dominant use currently. Because of the age of IPv4, and the changing world environment of networking, IPv4 contains several limitations that make it unsuitable for future needs.

Perhaps the most serious limitation is that its address space will eventually be exhausted. Even today, public IPv4 addresses have become relatively scarce. In addition, world-wide networking has introduced requirements beyond what IPv4 provides, such as simpler configuration capabilities, enhanced security, and extensibility.

IPv6 addresses the shortcomings of IPv4 as well as offering a host of additional benefits. Newer hardware and operating systems provide IPv6 support. Applications for certain sectors already require IPv6 support. For example, the governments of the United States and Japan have mandated support for IPv6. Since IPv4 must eventually be replaced, the sooner that occurs, the sooner the benefits of IPv6 can be realized.

Client/Server Communications

During the transition period between IPv4 and IPv6 support, both protocols will likely be functional on certain operating systems. Depending on the operating systems, this is referred to as dual IP layer or dual stack. Note, however, that IPv4 and IPv6 traffic is independently routed. For two hosts to communicate, both must either be capable of using IPv4 or capable of using IPv6.



Dual IP Layer

Dual Stack

- Available on Windows Vista, Windows Server 2008, and Windows 7
- IPv6 automatically installed with the operating system
- IPv6 cannot be uninstalled
- IPv6 can be turned off
- IPv4 can be turned off

- Available on Windows Server 2003 and Windows XP (and Linux distributions)
- IPv6 must be installed as an add-on for Windows platforms
- IPv6 can be uninstalled on Windows platforms
- IPv6 can be turned off
- IPv4 cannot be turned off

If you want to configure the network settings at the operating system level, note the following.

Operating System	IPv6 Notes
Windows Server 2003 and Windows XP	IPv6 must be manually installed. No network GUI utilities are available, but the following command-line utilities are provided: ipconfig, netsh and nsupdate.
Windows Vista and newer	Network GUI configuration utilities available, as well as the command-line utilities ipconfig, netsh and nsupdate.

Host Files, ZoneIDs, and Name Discovery

Within the hosts file, each IP uses only lines with compatible address formats. For example, if you request an IPv4 address for a host name, the IPv6 lines are ignored. Compatible addresses also apply for localhost, so a hosts file typically has localhost lines for 127.0.0.1 (IPv4) and ::1 (IPv6).

When doing a lookup to convert a name into an address, the application programmer specifies whether to use IPv4, IPv6 or both. A networking component of the operating system uses administrator-level preferences to determine how to sequence the lookups to the local hosts file, the local DNS cache, the remote DNS server, and so forth. With IPv6, there are new auto-discovery protocols that can find remote machines without using DNS.

You can specify an IPv6 address in a hosts file with the following restrictions:

- Records in a hosts file cannot include the ZoneID
- The hosts file can have separate lines for IPv4 and IPv6 with the same node name.

The use of hosts files is most useful when ZoneIDs are not required.

ZoneID

The ZoneID maps to a network interface. With a single network interface card (NIC) and gateway, a ZoneID is not needed because the gateway is reached by only one route. Most machines enabled for IPv6 have multiple interfaces because of built-in support for transition routers like ISATAP, 6to4, or Teredo.

In netsh commands, you must use the interface name (using the interface= parameter), for example “Local Area Connection 2” or “eth0.” When using ping with an IPv6 address, you may need to use the ZoneID, for example fe80::abcd%10, in which case the decimal integer 10 is the ZoneID.

On Windows platforms, you can display the ZoneIDs for each interface with the ipconfig command.

Name Discovery

IPv6 contains auto-discovery protocols that can find remote machines without using Domain Name System (DNS). The Link Local Multicast Name Resolution (LLMNR) is a protocol based on the DNS packet format. LLMNR allows both IPv4 and IPv6 hosts to perform name resolution for hosts on a single subnet without a DNS server. Since every IPv6 machine has a link-local address, LLMNR locates the machine on the subnet, if present, before having to perform a DNS lookup for a link-global address.

64-bit ODBC Driver

Pervasive PSQL v11 now supports the ODBC interface for 64-bit applications. The 64-bit ODBC driver is installed with Pervasive PSQL Server 64-bit and Pervasive PSQL Client 64-bit.

ODBC and Data Source Names (DSNs)

On 64-bit Windows operating systems, 64-bit DSNs are distinct from 32-bit DSNs because of the Windows registry design. Windows ODBC Data Manager requires that you know the bit architecture (called “bitness”) of your application and create a DSN with that same bitness. Pervasive PSQL v11 adopts this same model. Therefore, 64-bit applications use the 64-bit ODBC driver and 32-bit applications use the 32-bit ODBC driver.

The application bitness does not have to match the bitness of the Pervasive PSQL Server product. For example, the 64-bit ODBC driver or the 32-bit ODBC driver can be used with either Pervasive PSQL Server 64-bit or Pervasive PSQL Server 32-bit.

Pervasive PSQL v11 provides three ODBC drivers, as shown in the following table.

Table 12 Pervasive PSQL ODBC Drivers for Windows

ODBC Driver	PSQL Product Installed With	Behavior for All Products Installed With
Pervasive ODBC Engine Interface	Server 64-bit Server 32-bit Workgroup	<ul style="list-style-type: none"> • Setup creates 32-bit Engine DSNs • Connects to a local named database • For use by 32-bit applications • Deprecated in Pervasive PSQL v11, as explained below
Pervasive ODBC Client Interface	Server 64-bit Server 32-bit Client 32-bit Workgroup	<ul style="list-style-type: none"> • Setup creates 32-bit Client DSNs • Connects to a local or remote named database or an Engine DSN • GUI lists both named databases and Engine DSNs • For use by 32-bit applications
Pervasive ODBC Interface	Server 64-bit Client 64-bit	<ul style="list-style-type: none"> • Setup creates 64-bit DSNs • Connects to a local or remote named database • For use by 64-bit applications

To simplify the method for connecting to a named database, Pervasive PSQL v11 includes the following enhancements:

- Deprecating 32-bit Engine DSNs. The 32-bit Engine Interface driver is still provided in this release, primarily for backwards compatibility. Pervasive recommends, rather than using Engine DSNs, that new or revised 32-bit applications connect to a named database through a Client DSN or use a DSN-less connection by specifying “Pervasive ODBC Client Interface.”
- Deprecating the DTI functions that manage 32-bit Engine DSNs. See [DTI](#).
- Providing a 64-bit Interface driver only for named databases. The 64-bit ODBC Interface can connect to a local named database, thus replacing the function of the Engine DSN, or to a remote named database. Connection to an Engine DSN is not supported.

Frequently Asked Questions

The following table answers some frequently asked questions (FAQs) about the ODBC and DSN support in Pervasive PSQL v11.

Table 13 FAQs About ODBC and DSN Changes

Question	Answer
Is the 64-bit ODBC driver supported for Linux distributions or Macintosh OS X?	No. Only Windows platforms are supported as discussed in Table 12.
What happens to my existing 32-bit Engine DSNs when I upgrade to Pervasive PSQL v11 Server or Workgroup?	No migration steps are required. Existing 32-bit Engine DSNs remain in place and continue to work as configured. Applications on the PSQL Server or Workgroup machine continue to work with 32-bit Engine DSNs.
What happens to my existing 32-bit Client DSNs when I upgrade to Pervasive PSQL v11 Client?	No migration steps are required. Existing Client DSNs continue to connect to remote Engine DSNs. If you edit a Client DSN with ODBC Administrator, you have the option to continue using a remote Engine DSN or to use a remote named database. See ODBC DSN Setup GUIs . Note, however, the recommendation is that new or revised 32-bit applications should connect to a named database, not to an Engine DSN since Engine DSNs are deprecated.
Are connections that use “Pervasive ODBC Client Interface” affected (so called “DSN-less” connections)?	No. DSN-less connections that connect using “Pervasive ODBC Client Interface” continue to work.
What about connections from PSQL Clients of previous releases (such as a PSQL v10.x Client)?	Pervasive PSQL v11 still supports remote Client DSNs, so clients from previous versions can still connect. Note, however, Engine DSNs are only 32-bit for both Pervasive PSQL Server 32-bit and 64-bit. 64-bit Engine DSNs cannot be created with Pervasive PSQL.
What are the ODBC connection strings for Pervasive PSQL DSNs?	See ODBC Connection Strings in <i>SQL Engine Reference</i> .
What do I need to do about DSNs if I port my 32-bit application to 64-bit?	If the application uses DSN-less connections that connect using “Pervasive ODBC Client Interface,” change the connection string to “Pervasive ODBC Interface.” See ODBC Connection Strings in <i>SQL Engine Reference</i> . If the application uses DSNs, you must create 64-bit DSNs that connect to a named database.
What about the DSNs for the Demodata sample database installed with the database engine?	The installation of Pervasive PSQL Server 32-bit or Pervasive PSQL Workgroup creates a Client DSN for Demodata instead of an Engine DSN. The installation of Pervasive PSQL Server 64-bit creates both a 32-bit Client DSN and a 64-bit DSN for Demodata. If you install Pervasive PSQL Client 64-bit on top of Pervasive PSQL Server 32-bit or on top of Pervasive PSQL Workgroup, no 64-bit DSNs are created. Only the DSNs created by the installation of the 32-bit database engine are present. Similarly, if you install Pervasive PSQL Server 32-bit or Pervasive PSQL Workgroup on top of Pervasive PSQL Client 64-bit, no 64-bit DSNs are created. Only the DSNs created by the installation of the 32-bit database engine are present.

Table 13 FAQs About ODBC and DSN Changes *continued*

Question	Answer
How do I run the 32-bit ODBC Administrator on a 64-bit operating system?	See ODBC Administrator in <i>SQL Engine Reference</i> .
Why do I not see my DSNs in ODBC Administrator?	<p>On 64-bit Windows operating systems, 64-bit system DSNs are distinct from 32-bit system DSNs because of the registry design.</p> <p>If you are using the 64-bit ODBC Administrator, you will not see the 32-bit system DSNs, and vice versa.</p> <p>Note that, when the relational service interface on a 64-bit operating system receives a connection from a client to an Engine DSN, the database engine looks up the requested Engine DSN only in the 32-bit registry.</p> <p>See ODBC DSN Setup GUIs.</p>
What if my application uses DTI to manage DSNs?	See DTI .
What are the changes to ODBC Administrator?	See ODBC DSN Setup GUIs .
Other than ODBC Administrator, does Pervasive PSQL v11 include new utilities to support 64-bit ODBC and DSNs?	No.
Are there any changes to existing utilities to support 64-bit ODBC and DSNs?	Yes. See Utilities Affected by ODBC Changes .
Do some descriptor fields that can be set through the various ODBC SQLSet and SQLGet functions accommodate 64-bit values while others are still 32-bit values?	<p>Yes, if you are using the 64-bit ODBC driver. Ensure that you use the appropriate sized variable when setting and retrieving descriptor fields. For more information, refer to the Microsoft ODBC documentation. See especially http://msdn.microsoft.com/en-us/library/ms716287%28VS.85%29.aspx.</p> <p>A point of clarification is that SQL_ROWSET_SIZE is supported by both SQLGetStmtOption <i>and</i> SQLGetStmtAttr. If you are using the 64-bit ODBC driver and you call either SQLGetStmtOption or SQLGetStmtAttr, a 64-bit value is returned in *ValuePtr when that attribute parameter is set to SQL_ROWSET_SIZE.</p>
Going forward, is there a recommended strategy for ODBC connections?	<p>Yes. New or revised 32-bit applications, local or remote, should connect to a named database through a Client DSN, not to an Engine DSN. Alternately, applications could use DSN-less connections by specifying "Pervasive ODBC Client Interface."</p> <p>This positions your application for the future when Engine DSNs will no longer be supported in Pervasive PSQL.</p>

DTI

The DTI functions for DSNs manage only 32-bit Engine DSNs. Therefore, the following DTI functions are deprecated along with the 32-bit Engine Interface ODBC driver:

- [PvCreateDSN\(\)](#)
- [PvCreateDSN2\(\)](#)
- [PvGetDSN\(\)](#)
- [PvGetDSNEx\(\)](#)
- [PvGetDSNEx2\(\)](#)

- [PvDeleteDSN\(\)](#)
- [PvListDSNs\(\)](#)
- [PvModifyDSN\(\)](#)
- [PvModifyDSN2\(\)](#)

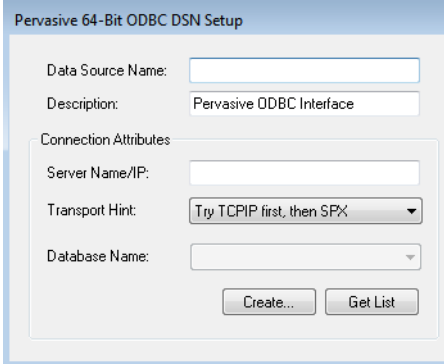
All of these functions operate only on the 32-bit registry. This applies even if a 64-bit database engine is installed on a 64-bit operating system. The 32-bit ODBC Administrator uses the DTI functions for Engine DSNs. Therefore, the list of existing Engine DSNs and newly created Engine DSNs are only for the 32-bit registry.

See *Distributed Tuning Interface Guide* for an explanation of the functions that manage DSNs.

ODBC DSN Setup GUIs

The following changes apply to setting up DSNs through ODBC Administrator.

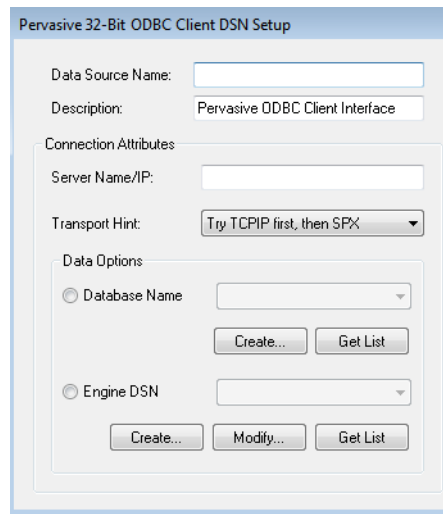
- A new graphical user interface (GUI) is available for setting up 64-bit DSNs. See also Table 12, [Pervasive PSQL ODBC Drivers for Windows](#).



The screenshot shows a dialog box titled "Pervasive 64-Bit ODBC DSN Setup". It contains the following fields and controls:

- Data Source Name:** A text input field.
- Description:** A text input field containing the text "Pervasive ODBC Interface".
- Connection Attributes:** A section containing:
 - Server Name/IP:** A text input field.
 - Transport Hint:** A dropdown menu with the selected option "Try TCP/IP first, then SPX".
 - Database Name:** A dropdown menu.
- Buttons:** Two buttons at the bottom: "Create..." and "Get List".

- The GUI for setting up 32-bit Client DSNs has been modified as follows:
 - The GUI now allows selection of a local or remote server name or IP address. See also Table 12, [Pervasive PSQL ODBC Drivers for Windows](#).



- The “Server” group box is now labeled “Connection Attributes”
- The control labeled “Address” is now labeled “Server Name/IP”
- The control labeled “Data Source Name” is now labeled “Engine DSN.”
- The “Options” button is now labeled “Advanced” and displays the advanced connection attributes. The advanced connection attributes provide the same choices as were previously available on the Options dialog.

- The GUI for setting up Engine DSNs has been modified as follows:
 - The “Database” group box is now labeled “Connection Attributes”
 - The “Options” button is now labeled “Advanced” and displays the advanced connection attributes. The advanced connection attributes provide the same choices as were previously available on the Options dialog.

See the chapter [DSNs and ODBC Administrator](#) in *SQL Engine Reference* for a discussion of the new controls on the GUIs.

ODBC Header Files

The `sql.h` and `sqltypes.h` header files for ODBC contain differences for the compilation of 32-bit and 64-bit applications. Refer to the ODBC documentation on the Microsoft Web site for a discussion of 64-bit ODBC. For example, you may find the following information useful: [http://msdn.microsoft.com/en-us/library/ms716287\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms716287(VS.85).aspx).

Utilities Affected by ODBC Changes

For Pervasive PSQ Server and Client installations on 64-bit operating systems, Pervasive PSQ Control Center (PCC) contains separate choices for 32-bit and 64-bit ODBC Administrator. The choices are available on the Tools menu. See [Additional Utilities](#) in *Pervasive PSQ User's Guide*.

In addition, the option to create a DSN on the New Database dialog is now qualified for 32-bit: “Create 32-bit Engine DSN.” See [New Database GUI Reference](#) in *Pervasive PSQ User's Guide*. (PCC is a 32-bit application. A 64-bit version of it is not available.)

The Pervasive ODBC DSN setup GUIs have changed. See [ODBC DSN Setup GUIs](#).

Support for .NET Framework 3.5 SP1 and 4.0

Pervasive PSQL v11 provides two versions of the ADO.NET Data Provider, version 3.2 and 3.5. Both versions are installed by default with the database engine and with the Pervasive PSQL Client.

The installation puts both Data Providers under the “Program Files (x86)” directory whether you are installing the 32-bit or 64-bit version of Pervasive PSQL. The Data Providers are bitness independent, however. Each Data Provider works with both the 32-bit and the 64-bit .NET Framework.

Pervasive PSQL ADO.NET Data Provider 3.2

The Pervasive PSQL ADO.NET Data Provider 3.2 has no new features from prior versions of the 3.2 Provider. It is included for application developers who want to use that version with Pervasive PSQL v11.

Pervasive PSQL ADO.NET Data Provider 3.5

Pervasive PSQL ADO.NET Data Provider 3.5 supports the new features in .NET Framework 3.5 SP1. The Provider is compliant with the .NET Framework Versions 2.0, 3.0, 3.5, 3.5 SP1, and 4.0. The 3.5 Provider does not support new features introduced in .NET Framework 4.0 but will run under the .NET Framework 4.0 with support for all the Entity Framework 1.0 features.

In addition, the Pervasive PSQL ADO.NET Data Provider 3.5 includes the following main features:

- Development using a set of methods tailored for the new Entity Framework consumers such as LINQ, EntitySQL, and ObjectServices.
- Pervasive Bulk Load. The DbBulkCopy class supports bulk loading data in the Common Programming Model. In addition, the Data Provider has a provider-specific bulk load class.
- Connection statistics support.
- A Schema Options connection string option to specify additional schema metadata to be returned.
- Native parameter marker and parameter binding support.

- Microsoft Enterprise Library 4.1 (October 2008) support, including data access application block (DAAB) support.
- Initial Command Timeout connection string option to specify an initial command timeout when a connection is established.
- Support for Microsoft Visual Studio 2008 and Visual Studio 2010.

See *Pervasive SQL Data Provider for .NET Guide* in the SDK documentation for complete details.

PDAC Development Environments

Pervasive PSQL v11 includes PDAC for these additional development environments:

- RAD Studio 2009
- RAD Studio 2010

Support for RAD Studio 2009 and 2010 refers only to the Delphi and C++ Builder components supported by Pervasive PSQL v11.

Pervasive PSQL v11 no longer provides PDAC integration for Delphi and C++ Builder development environments version 6 and older. See [Deprecated and Discontinued Features](#).

See also *Pervasive Direct Access Components Guide* in the SDK documentation.

Enhancements to Other SDK Access Methods

Pervasive PSQL v11 includes enhancements to the SDK access method Distributed Tuning Objects (DTO).

DTO

Pervasive PSQL v11 includes the following new methods.

DTO Object	Method	Description
DtoDatabase	AddUserToGroup	Adds an existing user to an existing group in the database
	AlterUserName	Changes an existing user's name in the specified database
	AlterUserPassword	Changes an existing user's password in the specified database
	CreateGroup	Creates a new user group in the existing database
	CreateUser	Creates a new user in the existing database
	DropGroup	Removes an existing group from the database
	DropUser	Removes an existing user from the database
DtoLicenseMgr	GetProductInfo	Returns an XML formatted list of all Pervasive Software products found by License Manager

See the new methods in *Distributed Tuning Objects Guide*.

Product Authorization

Product authorization is a validation process verifying that the copy of the software is legitimate, correctly licensed and on the appropriate hardware and software platform. Pervasive PSQL v11 includes the following additions to product authorization:

- [Telephone Authorization](#)
- [Product Authorization for OEMs](#)

Telephone Authorization

If Pervasive PSQL Server or Workgroup is installed on a system that has no Internet connectivity, directly or indirectly, the product can be authorized by telephone with the assistance of Technical Support. The toll free number at Pervasive is 800 287-4383.

Telephone authorization is available during regular United States office hours, Central Standard Time. Calls received during off-hours or holidays are returned the next business day.

See [Telephone Authorization](#) in *Pervasive PSQL User's Guide* for complete details.

Product Authorization for OEMs

Pervasive PSQL v11 extends the product authorization technology to our original equipment manufacturer (OEM) partners. If you are an OEM partner, refer to the following resources:

- Product authorization information on the Pervasive Web site.
- OEM Web Portal on the Pervasive Web site. The Portal allows you to generate product keys and perform various administrative functions pertaining to keys. The Portal is available 24/7 and provides an easy-to-use interface. (Your Pervasive PSQL sales representative can provide more information about the Portal.) See also on the Portal:
 - *Manufacturing Partner Handbook*, which has been extensively revised.
 - *Product Authorization for OEM Partners* white paper.
 - *Product Authorization Troubleshooting Guide for OEM Support Staff*.

Configuration Settings

Pervasive PSQL v11 includes changes to the following configuration settings:

- [Communications Threads](#)
- [Listen IP Address](#)

Communications Threads

The range and default for the Communications Threads setting have changed.

- The range is now *num_cores* to 256, where *num_cores* is the number of processors in the machine on which the database engine is running.
- The default is *num_cores*.

Previously, the range was 1 to 1,024 and the default was 16.

The Communications Threads setting can help improve scaling under certain conditions. For example, if you have many clients performing operations (typically writes) on one file, a lower setting should improve scalability. The lower number of threads prevents context switching on system resources. Another condition that this setting may improve is a slowdown caused by thrashing among large numbers of worker threads. In Pervasive PSQL v11, worker threads are dynamically created only if all the existing threads are waiting on record or file locks.

See [Communications Threads](#) in *Advanced Operations Guide*.

Listen IP Address

The Listen IP Address setting now accepts multiple IP addresses separated by a comma between each address. The string can be a combination of IPv4 and IPv6 addresses. Any of the IPv6 address formats supported can be used. See [IPv6 Address Formats](#).

The Listen IP Address setting specifies the IP address or addresses the database engine listens on when TCP/IP Multihomed is Off. See also [Listen IP Address](#) and [TCP/IP Multihomed](#) in *Advanced Operations Guide*.

Utility Changes

Pervasive PSQL v11 includes changes to the following utilities:

- [Pervasive PSQL Control Center](#)
- [ODBC Administrator](#)

Pervasive PSQL Control Center

Pervasive PSQL Control Center (PCC) contains the following change pertaining to DSNs.

- On Pervasive PSQL Server 64-bit installations, the PCC Tools menu contains separate choices for 32-bit and 64-bit ODBC Administrator.
- The option to create a DSN on the New Database dialog is now qualified for 32-bit: “Create 32-bit Engine DSN.”

See also [ODBC and Data Source Names \(DSNs\)](#).

ODBC Administrator

The Pervasive ODBC setup GUIs for 32-bit DSNs have changed. A new ODBC setup GUI for 64-bit DSNs is available. See [ODBC DSN Setup GUIs](#).

Deprecated and Discontinued Features

Deprecated Features

The following categories discuss features that are deprecated in Pervasive PSQL v11. Although the features are still available in Pervasive PSQL v11, they will be removed from the product in a future release. Plan accordingly for new application development and revisions to existing applications.

ODBC

The following ODBC features are still available in Pervasive PSQL v11 but will be removed from the product in a future release.

- 32-bit Engine DSNs (32-bit Client DSNs now provide comparable functionality). See [ODBC and Data Source Names \(DSNs\)](#).
- DTI functions that manage 32-bit Engine DSNs. See [DTI](#).

Pervasive Direct Access Components (PDAC)

The PDAC dynamic libraries for Delphi 2006 (and 2007 which is compatible with 2006) are still available in Pervasive PSQL v11 but will be removed from the product in the future.

Discontinued Features

The following features are no longer supported in Pervasive PSQL v11.

- Support for Windows 2000
- Delphi and C++ Builder development environments version 6 and older. Pervasive PSQL v11 does not provide PDAC integration with development environments version 6 and older.
- The Pervasive PSQL ADO.NET Data Provider versions 2.1 and 3.0. The installation of the Pervasive PSQL v11 database engine automatically uninstalls either Provider if the version is detected.

