

Pervasive PSQL v11 SP1

Getting Started with Pervasive PSQL

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>

The logo for Pervasive Software, featuring the word "PERSVASIVE" in a blue, sans-serif font. The letter "V" is stylized with a diagonal line through it. A registered trademark symbol (®) is located at the top right of the word.

PERSVASIVE®

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 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 2011 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/licensing/licenses/lgpl.html.

Getting Started with Pervasive PSQL

General Release June 2011

138-004428-002

Contents

About This Manual	xi
Who Should Read This Manual	xii
Manual Organization	xiii
Conventions	xv
1 Welcome to Pervasive PSQL	1
<i>A Basic Introduction to Pervasive PSQL v11 SP1</i>	
About Pervasive PSQL	2
Competitive Advantages	2
Relational or Transactional Access	3
The Pervasive PSQL Transactional Interface	4
Benefits of the Transactional Interface	4
Transactional Features	5
The Pervasive PSQL Relational Interface	6
Benefits of the Relational Interface	6
Relational Features	6
About the Pervasive PSQL Engines	8
Pervasive PSQL Server	8
Pervasive PSQL Workgroup	8
Engine Feature Comparison	9
Pervasive PSQL SDK	10
Development Environment	11
2 Preparing to Install Pervasive PSQL	13
<i>Preparation Needed for Pervasive PSQL v11 SP1 Installation</i>	
Installation Requirements	14
Installation Options	15
Complete Installation	15
Custom Installation	15
Pervasive PSQL Products	16
Server Engine (64-bit)	16
Server Engine (32-bit)	16
Workgroup Engine (32-bit)	16
Client (64-bit)	17
Client (32-bit)	17
Pervasive PSQL v11 SP1 Optional Features	18
Xtreme I/O (Server 32-bit Only)	18
Pervasive Access Methods	18
Utilities	19
Documentation	20

Java Runtime Environment (JRE)	21
Installation Review	22
Quick Checklist	22
Common Pre-Installation Questions	23
3 Upgrading Your Pervasive PSQL Installation for Windows	29
<i>Considerations for Users Upgrading to Pervasive PSQL v11 SP1</i>	
Upgrading to Pervasive PSQL v11 SP1 From a Previous Version	30
Considerations When Upgrading to Pervasive PSQL v11 SP1.	30
Common Questions After Upgrading to Pervasive PSQL	32
4 Installing Pervasive PSQL Server for Windows	35
<i>Instructions for Installing the Pervasive PSQL Server Engine on Windows</i>	
Before You Install the Windows Server Engine	36
Platform Notes	36
Installation Tips	36
Installing Pervasive PSQL Server for Windows	38
5 Installing Pervasive PSQL Clients for Windows	43
<i>Instructions for Installing the Pervasive PSQL Client on Windows</i>	
Before You Install the Windows Client Engine	44
Installing the Pervasive PSQL Client for Windows	45
Installing the BTRBOX Requester	49
Win32 DOS Box Support	49
Understanding Client Requesters.	50
Types of Windows Requesters	50
Where To Go From Here	51
6 Installing Pervasive PSQL Workgroup for Windows.	53
<i>Instructions for Installing the Pervasive PSQL Workgroup Engine on Windows</i>	
Before You Install the Windows Workgroup Engine	54
Installation Tips	54
Installing the Pervasive PSQL Workgroup for Windows	55
Authorization of Workgroup Key on Vista and Windows 7.	58
7 After Installing Pervasive PSQL for Windows	61
<i>Answers to Common Post Installation Questions for Pervasive PSQL on Windows</i>	
Common Questions After Installing Pervasive PSQL	62
Where To Go From Here	65
Uninstalling Pervasive PSQL	66

8	Configuring the Workgroup Engine	67
	<i>Understanding the Available Workgroup Engine Configurations</i>	
	Overview	68
	Installation Requirements	68
	Operating System Security	68
	When to Use Workgroup	68
	What is a Gateway Engine?	69
	Setting Up a Small Client/Server Configuration	71
	Setting Up a Peer-to-Peer Configuration	73
	Setting Up a Gateway Configuration	75
	Floating or Fixed Gateway	75
	Working with the Gateway Locator Utility	77
	Running the Workgroup Engine as a Service	81
	Configuration	81
	Stopping the Service	82
9	Configuring Engine Network Communications	83
	<i>How to Set Up Network Communication for Pervasive PSQL Engines</i>	
	Determining What Kind of Network You Have	84
	Database Engine on Windows	84
	Engine Network Communication Settings	85
	Setting Up TCP/IP Support	86
	Setting Up SPX Support	88
	Setting Up NetBIOS Support (Workgroup only)	90
	Avoiding Unavailable Protocols	91
10	Configuring Network Communications for Clients	93
	<i>How to Configure Network Communications for Your Pervasive PSQL Clients</i>	
	Client Network Communication Settings	94
	Network Path Formats Supported by Pervasive Requesters	95
	Universal Naming Convention (UNC) Path Formats	95
	Drive-based Formats	100
	Linux Path Formats	100
	Using TCP/IP to Connect to a Windows Server	101
	Configuring a Client for the Server IP Address	101
	Using SPX to Connect to a Windows Server	103
	Configuring Pervasive PSQL to use SPX	103
	Changing the Default Communication Ports	104
	Services File	104
	Windows FireWalls	105
	Windows Operating Systems Prior to Vista	107
	Using TCP/IP to Connect a Windows Client to a Linux Server	108
	Configuring a Client for the Server's IP Address	108
	Data Encoding	110

Database Code Page	110
Client Encoding	111
Encoding Interaction	111
Legacy Conversion Methods for OEM Data	112
Using the DOS Requester	117
Supported Configurations	117
DOS Box Support on Windows.	118
Running DOS Applications on Windows 32-bit Platforms	118
11 Application Configuration Scenarios	119
<i>Common Scenarios for Setting up Your Database Engine</i>	
Terminal Services	120
Disabling Administrative Functions	120
Terminal Server as Network Server	120
Workgroup Engine Running as a Service	121
Active Directory Service	122
Pervasive Administrative Authority	122
Active Directory Tasks	123
Multiple Client Applications	130
Settings Affected by Multiple Applications	130
Concurrent Local and Remote Applications.	132
Using the Server and Workgroup Engines Concurrently	133
Accessing Data on Other Computers.	135
12 Installing Pervasive PSQL for Linux	137
<i>Instructions for Installing and Uninstalling Pervasive PSQL on Linux</i>	
Before You Install Pervasive PSQL for Linux	138
Server.	138
Client.	138
Full Installations.	139
Notes for Specific Distributions	139
Installing Pervasive PSQL Using RPM	141
Installing Pervasive PSQL Server for Linux - RPM	141
Installing the Pervasive PSQL Client for Linux - RPM.	143
Installing Pervasive PSQL Using TAR	145
Installing Pervasive PSQL Server for Linux - TAR	145
Installing Pervasive PSQL Client for Linux - TAR	147
After Installing Pervasive PSQL for Linux	149
Verifying Installed Products With RPM.	149
Verifying Database Engine is Running	149
Server Configuration	149
Client Configuration	149
User Count License	150

Common Questions After Installation	150
Uninstalling Pervasive PSQL for Linux	154
RPM Version	154
TAR Version	154
13 Using Pervasive PSQL on Linux	157
<i>Working With the Products on Linux</i>	
Finding What You Need	158
Accessing the User Documentation.	158
Man Pages	158
Exclusions.	158
Pervasive PSQL Account Management on Linux.	160
After Installation Behavior	160
The User Environment	160
Using Utilities from Users Other than psql	161
Configuration	162
Configuration File	162
Authentication	162
Supported Path Formats for Samba	162
Client Information	164
Authentication to Remote Machines	164
Creating a Client DSN	164
Setting Up Web-based Data Access	165
ODBC Behavior	165
Configuring Web Server.	165
PHP	166
Perl	169
Using Perl and ODBC with Pervasive PSQL	173
Code Snippet for Perl and DBI	173
14 Troubleshooting After Installation	175
<i>How to Proceed When You Encounter Errors During Installation</i>	
Troubleshooting Tools	176
Troubleshooting Strategies.	177
Checklist for Problems.	177
Troubleshoot the Problem	177
Logged Messages	178
Configuration for Special Installation Situations.	179
Diagnosing Problems with Pervasive System Analyzer (PSA)	180
Verifying Database Engine is Running	181
Windows Server (Non-Vista)	181
Windows Workgroup	182
Linux Server	182
Obtaining File, Client, and Engine Version Number.	183

Determining Client and Engine Version	183
Determining a File Version.	184
Engine and Client Version Conflicts	188
Troubleshooting Common Pervasive PSQL Issues	189
Issues After Uninstalling Pervasive PSQL on Windows.	193
How to Get Additional Help	194
Technical Support.	194

Figures

1	Engine Installation Mode Dialog Box	46
2	Engine Installation Mode Dialog Box	56
3	Gateway Locator Main Dialog Box	78
4	Gateway Locator Browse Dialog Box	79
5	Gateway Assignment Dialog Box.	79
6	Browse for Computer Dialog Box	80
7	Displaying the Services Status	181
8	Selecting the Btrieve Version Info button	183
9	Btrieve Version Info Display	184
10	Obtaining a File Version with the Pervasive PSQL Control Center	185
11	Table Properties Page	185
12	File Statistics in Function Executor	187

Tables

1	Comparison of Server and Workgroup Features.	9
2	DSN Considerations After a Pervasive PSQL Upgrade	33
3	Pervasive PSQL v11 SP1 Default Windows Installation Locations	62
4	IPv6 Unicast Address Formats Supported by Pervasive PSQL	96
5	IPv6 Restrictions With Pervasive PSQL	99
6	Installation Actions for Vista or Later Operating Systems	105
7	Interaction Between Database Encoding and Client Encoding	111
8	Summary of Network Configurations	135
9	Full and Client Linux Installations	139
10	Linux Server Package Names - RPM	141
11	Linux Client Package Names - RPM	143
12	Linux Server Package Names - TAR	145
13	Linux Client Package Names - TAR.	147
14	RPM Commands To Verify Pervasive PSQL Packages Installed	149
15	Primary Directories and Files for Pervasive PSQL Products Installed on Linux.	151
16	RPM Commands to Uninstall the Pervasive PSQL Packages	154
17	TAR Commands to Uninstall the Pervasive PSQL Packages	154
18	Pervasive Tools that Assist in Installation and Problem Determination	176

About This Manual

This manual contains information about installing the Pervasive PSQL v11 SP1 database system. Pervasive PSQL v11 SP1 is a complete database management system, providing the best of both worlds. It combines a transactional interface designed for high-performance data handling and improved programming productivity with an embeddable and scalable relational interface.

This manual also contains information about common installation issues, general network protocol information, and Pervasive PSQL v11 SP1 optional features.

For information on using Pervasive PSQL utilities, see *Pervasive PSQL User's Guide*. For information about configuring the Pervasive PSQL v11 SP1 engine, see *Advanced Operations Guide*.

Who Should Read This Manual

This manual provides information for users who install and run Pervasive PSQL v11 SP1. This manual is also useful for system administrators who are responsible for maintaining databases on a network and for those who are using Pervasive PSQL to develop server applications.

Pervasive Software 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 Software Web site.

Manual Organization

This manual is arranged in the order of the main installation sequence. You complete the installation by following the chapters in order (skipping the chapters that do not apply to your installation platform). *Getting Started With Pervasive PSQL* is divided into the following sections:

- Chapter 1—[Welcome to Pervasive PSQL](#)
This chapter provides a basic introduction to Pervasive PSQL v11 SP1.
- Chapter 2—[Preparing to Install Pervasive PSQL](#)
This chapter discusses important preparations that you should undertake before attempting to install Pervasive PSQL v11 SP1.
- Chapter 3—[Upgrading Your Pervasive PSQL Installation for Windows](#)
This chapter describes how to upgrade a previous version of Pervasive PSQL on Windows.
- Chapter 4—[Installing Pervasive PSQL Server for Windows](#)
This chapter describes how to install Pervasive PSQL Server for the first time.
- Chapter 5—[Installing Pervasive PSQL Clients for Windows](#)
This chapter describes how to install Pervasive PSQL Client for the first time.
- Chapter 6—[Installing Pervasive PSQL Workgroup for Windows](#)
This chapter describes how to install Pervasive PSQL Workgroup for the first time.
- Chapter 7—[After Installing Pervasive PSQL for Windows](#)
This chapter answers post installation questions you may have about Pervasive PSQL for Windows.
- Chapter 8—[Configuring the Workgroup Engine](#)
This chapter describes how to configure the Pervasive PSQL Workgroup engine.

- [Chapter 9—Configuring Engine Network Communications](#)
This chapter describes how to configure your network for use with the Server engine on Windows.
- [Chapter 10—Configuring Network Communications for Clients](#)
This chapter describes how to configure the client network settings for use with the PSQL engine. It also offers implementation notes for specific operating systems.
- [Chapter 11—Application Configuration Scenarios](#)
This chapter describes different application configurations for special installation scenarios.
- [Chapter 12—Installing Pervasive PSQL for Linux](#)
This chapter describes how to install or upgrade the Pervasive PSQL Server and Client on Linux.
- [Chapter 13—Using Pervasive PSQL on Linux](#)
This chapter provides information for using Pervasive PSQL on Linux after you have completed installation.
- [Chapter 14—Troubleshooting After Installation](#)
This chapter provides information on Pervasive PSQL tools that aid in diagnosing problems. This chapter also gives contact information for Pervasive PSQL support for the case that you do not find the answer to your problem.

This manual also contains an index.

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 <code>MYPROG</code> , <code>myprog</code> , or <code>MYprog</code> .
Bold	Words appearing in bold include the following: menu names, dialog box names, commands, options, buttons, statements, etc.
Monospaced font	Monospaced font is reserved for words you enter, such as command syntax.
[]	Square brackets enclose optional information, as in <code>[log_name]</code> . If information is not enclosed in square brackets, it is required.
	A vertical bar indicates a choice of information to enter, as in <code>[file name @file name]</code> .
< >	Angle brackets enclose multiple choices for a required item, as in <code>/D=<5 6 7></code> .
<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 <code>[parameter ...]</code> .
::=	The symbol ::= means one item is defined in terms of another. For example, <code>a::=b</code> means the item <code>a</code> is defined in terms of <code>b</code> .

Welcome to Pervasive PSQL

1

A Basic Introduction to Pervasive PSQL v11 SP1

Thank you for purchasing Pervasive PSQL. We are confident that you will find this release to be the very best, high performance, low maintenance database engine on the market.

This chapter contains the following topics:

- [About Pervasive PSQL](#)
- [The Pervasive PSQL Transactional Interface](#)
- [The Pervasive PSQL Relational Interface](#)
- [About the Pervasive PSQL Engines](#)
- [Pervasive PSQL SDK](#)

About Pervasive PSQL

Pervasive PSQL is a reliable, low-maintenance, high-performance database management system (DBMS). Thousands of companies around the world license Pervasive PSQL and distribute it as the underlying data storage program for their data-intensive software products. These companies see no reason to build their own DBMS or license from a competitor once they experience the ease-of-use, reliability, and value offered by Pervasive PSQL.

No matter whether you received Pervasive PSQL with another product or purchased it yourself, this section explains a little about the product and why it is right for you.

Competitive Advantages

Pervasive PSQL provides a number of advantages over other products available on the market. Here are just a few:

- *Lowest total cost of ownership.* An independent study conducted by Aberdeen Group concluded that no major database product can match Pervasive PSQL's low total cost of ownership. How do we do it? See the next bullet.
- *No Database Administrator (DBA) required.* You can look in the newspaper any day of the week and see classified ads for Oracle, Sybase, or SQL Server database administrators, with sky-high salaries. Pervasive PSQL offers the unique Zero Database Administrator, or Z-DBA™, architecture. Its easy-to-use tools, bulletproof installation, and set-it-and-forget-it simplicity make it the perfect workhorse for desktop, workgroup, and departmental applications.
- *Scalable from the desktop to the Web.* Pervasive PSQL is available in two editions: the Ultra-light™ Workgroup database engine supports single-user configurations up to small workgroup configurations. The Server engine comes with a six-user license and scales to hundreds of concurrent users, including intranet and extranet applications. Upgrading to another configuration requires no changes to the supported application, just plug and play with the new database engine.

- *Cross-platform support.* Unlike some competitors, Pervasive PSQL does not lock you in to a single platform. Pervasive PSQL databases are binary-compatible and supported across Microsoft Windows and several varieties of Linux. No matter where your data is or where it is going to be, Pervasive PSQL is there for you.
- *Big database features at a small price.* Pervasive PSQL offers full security, encryption, management and monitoring tools, and a host of other features you would expect to see in more expensive DBMS products.
- *Legendary stability and reliability.* There's no doubt why the Windows desktop accounting market uses Pervasive PSQL as the underlying database of choice. When you've got to manage important data, you go for the database engine that won't let you down.
- *Multiple access methods.* Your application vendor can use the transactional interface for blazing performance on bulk data operations, while offering the richness of ODBC, OLE-DB, pure Java, and JDBC interfaces for data reporting, security, analysis, and standard compatibility. No other database management system offers all these access methods.

Relational or Transactional Access

Pervasive PSQL offers an architecture that is totally unique in the database management market. Our product allows you to access the exact same data through ODBC and OLE DB, supporting applications like Microsoft ASP, Excel, and Access, or through the lightning-fast transactional interface called Btrieve. ODBC allows you to do complex reporting and data mining, while Btrieve provides massive throughput when you need the ability to view, update, or create millions of records a day.

Each application vendor chooses which interfaces are used. If you want to know which access methods are used in your application, contact your application vendor.

The Pervasive PSQL Transactional Interface

The Pervasive PSQL transactional interface, built on Btrieve, offers easy installation, uncomplicated maintenance, and high levels of performance and reliability. Pervasive PSQL provides a foundation on which you can run transactional applications or migrate to a relational database system.

Benefits of the Transactional Interface

Pervasive PSQL's transactional interface is Btrieve, which has been the data management system of choice for tens of thousands of applications around the world for more than 25 years now. In the highly competitive accounting software market—where reliability and performance are paramount—many of the top 10 vendors choose Pervasive PSQL. Many application developers choose Pervasive PSQL for its speed, data integrity, easy scalability, and low maintenance costs. As part of Pervasive PSQL, Btrieve's transactional interface offers:

- *Speed.* Pervasive PSQL uses the highly-evolved MicroKernel Database Engine, capable of sub-second response rates, even when building multi-gigabyte databases for hundreds of users. The MicroKernel achieves these high speeds through features such as internal indexing algorithms that cache pages for fast data retrieval and updates, and automatic index balancing to maintain fast data access, even as your files grow.
- *Data Integrity.* The MicroKernel guarantees data integrity through rich transaction processing support, referential integrity controls, and automatic file recovery. In the event of a server or system failure, logging features and roll forward utilities allow you to recover data up to your last completed transaction.
- *Scalability.* Many client/server database applications begin on the desktop and scale with corporate growth. Pervasive PSQL provides easy scalability from workstation to large client/server environments.
- *Low Cost.* The low support costs experienced by Pervasive PSQL developers translate into low maintenance costs realized by Pervasive PSQL application end users. Pervasive PSQL eliminates the need for sustained database administration through automatic data recovery functions and easy-to-use utilities.

Transactional Features

Pervasive PSQL provides a comprehensive transactional database management system that offers many features, including the following:

- MicroKernel Database Engine as the underlying data manager.
- Access to databases distributed across multiple engines.
- Robust transactions for both single-server systems and distributed, multi-server systems.

The Pervasive PSQL Relational Interface

Pervasive PSQL's relational interface, built on the SQL Relational Database Engine (SRDE), offers easy installation, uncomplicated maintenance, and high levels of performance and reliability.

Benefits of the Relational Interface

Many relational database application developers choose Pervasive PSQL because it provides scalability, maintenance-free operation, and a small memory footprint:

- *Standard Interface.* SQL and ODBC provide a well-known and standardized foundation upon which to build useful applications.
- *Speed.* Pervasive PSQL offers direct ODBC access to the database engine. Many competitive products use a translation layer to translate ODBC calls to proprietary "native" relational API calls that then access the database engine. In contrast, the Pervasive PSQL ODBC driver calls the database engine directly, without translating ODBC calls to a proprietary relational API.
- *Scalability.* Pervasive PSQL allows you to scale applications from single-user to large client/server environments without changing the application or the database.
- *Maintenance-free Operation.* Pervasive PSQL is simple to install and use. It requires no extensive performance setup or ongoing tuning by a database administrator.
- *Small Memory Footprint.* Pervasive PSQL has a small footprint, requiring only a small amount of memory.

Relational Features

The Pervasive PSQL relational interface provides a flexible architecture that helps you easily scale your database applications from large client/server systems to single-user environments without additional coding. Pervasive PSQL offers easy installation, uncomplicated maintenance, high levels of performance and reliability, and a smooth migration path for data. In addition, bundling Pervasive PSQL with your application is easy with the Pervasive PSQL distribution component, which provides multi-user and single-user run-time support.

Pervasive PSQL has a comprehensive relational database management system interface that offers many features, including the following:

- Application scalability from standalone to client/server.
- Fully functional Workgroup and client/server engines.
- Declarative Referential Integrity.
- Bi-directional, updateable, and scrollable cursors.
- Named database support providing location transparency for applications.
- Comprehensive, industry standard data type support.
- Programming extensions such as triggers and stored procedures.
- Cost-based optimization from statistical analysis and enhanced fetch algorithms.
- Transaction processing enhancements such as full transactional logging.
- Standards enhancements, including ODBC support.
- Other features include additional Windows utilities, large file support (up to 256 GB), and additional data type variables such as `TIMESTAMP`, `UNSIGNED`, and `CURRENCY`.

About the Pervasive PSQL Engines

This section provides some basic information about the Pervasive PSQL Server and Workgroup engines. For a discussion of Pervasive PSQL architecture, see [Understanding the Pervasive Component Architecture](#) in *Advanced Operations Guide*.

Pervasive PSQL Server

The Pervasive PSQL Server database engine is designed to support up to many hundreds of concurrent network users when installed on the required hardware. It is capable of supporting web, corporate, departmental, and other client/server or web-based applications where reliability and performance are critical.

The Pervasive PSQL Server engine may be installed with license blocks for 6, 10, 20, 50, 100, 250, 500, or an unlimited number of users, depending on the number of user licenses you purchased. The server engine is capable of scaling to hundreds of concurrent users with the purchase of additional licenses.

Pervasive PSQL Workgroup

The Pervasive PSQL Workgroup database engine is designed to support single-user or small workgroup installations.

Pervasive PSQL Workgroup offers the same level of reliability and features as the Server engine. The only differences lie in networking and performance in mid- and large-size environments.

Pervasive PSQL Workgroup offers a flexible approach to accessing data on remote servers, allowing a variety of small network configurations. If you have data files on a remote file system with no database engine present, you can configure Pervasive PSQL Workgroup so that a particular engine is always used to access the remote data, or you can set it up so that the first engine to access the files then “serves” those files until there are no more requests for data. After this point, again the first engine to access the files then owns the files while requests are coming in.

A Pervasive PSQL engine cannot be installed on more than one machine. Your user count license refers to the number of client connections allowed to that engine, not to the number of machines to which you are allowed to install the Pervasive PSQL engine. In a Workgroup environment, every machine that will access Pervasive PSQL data should have a Workgroup engine installed.

Engine Feature Comparison

All Pervasive database engines offer the same powerful feature set and full-functioned support for programming interfaces. The chart below shows the major differences between the different editions of the product.

Table 1 Comparison of Server and Workgroup Features

Feature	Server	Workgroup
Supports Btrieve, ODBC, OLE DB, Java, JDBC, PDAC and ActiveX interfaces	✓	✓
Full-featured relational support (online backup, security, referential integrity, management tools, and so on)	✓	✓
Binary compatible data files across all platforms and engine editions	✓	✓
Easy plug and play upgrading, no application changes required to change engines.	✓	✓
Includes complete online documentation	✓	✓
Can access data on a file server where no database engine is installed		✓
Supports remote ODBC client connections	✓	✓
Requires a Workgroup engine on all computers expected to access remote data	N/A	✓
Engine runs on Windows	✓	✓
Engine runs on Linux	✓	
Multi-user for small groups	✓	✓
Scales to thousands of users	✓	
Extranet license available	✓	
Enforces Operating System Security	✓	

Pervasive PSQL SDK

The Pervasive PSQL SDK includes many features to ease the burden of application development. These include:

- Low-level APIs. Direct programming to the Btrieve API gives you the fastest possible data access and the most control over the way in which your application reads and writes data. If these considerations are important to you and you are willing to develop the code that incorporates your business rules, you may find direct API programming highly useful. For relational access to data, you may also code directly to the Microsoft ODBC API.
- ODBC. Pervasive PSQL offers a native ODBC driver.
- Pervasive PSQL ADO.NET data provider. The data provider provides support for the Microsoft .NET Framework, and is an ADO.NET managed data provider, built with 100% managed code.
- Java. The Java Interface gives you the option of developing Btrieve applications in an object-oriented, platform-independent manner. It includes support for true null and Unicode values as well as for Binary Large Objects (BLOBs).
- Distributed Tuning Interface and Objects. These two related interfaces allow applications to tune and manage the database engine itself, including configuration parameters and aspects of security.
- OLE-DB. The OLE-DB provider offers access to both the relational and transaction interfaces.
- The ActiveX Interface. The ActiveX Interface allows you to leverage the power and speed of the Pervasive PSQL engine with a minimum of manual coding. These controls are designed for easy use with third-party grid controls as well.
- Complete sample application. Pervasive PSQL SDK includes a complete sample application designed to run a video rental store. Full sample code in Visual Basic, Delphi, Java, and C/C++ is supplied. Examples using ODBC, ActiveX RDO, third party controls, and direct API calls are shown.
- The Pervasive PSQL Web site is an online resource that gives you access to the latest component downloads and code samples.

***Development
Environment***

Pervasive PSQL provides an open interface that allows you to develop many front-end applications, all of which can share a common, transactional, or relational database. You can use popular programming languages and environments such as Java, Delphi, BASIC, Visual BASIC, .NET, C, C++, COBOL, Pascal, ODBC, PowerBuilder (through ODBC), and FoxPro (through ODBC). In addition, bundling a Pervasive PSQL engine with your application is easy with a Derivative Software License.

Preparing to Install Pervasive PSQL

Preparation Needed for Pervasive PSQL v11 SP1 Installation

This chapter prepares you to install Pervasive PSQL by providing an overview of the requirements, the major components included in Pervasive PSQL, the installation options available, a detailed checklist to help you gauge your readiness to proceed with the Pervasive PSQL installation.

This chapter contains the following sections:

- [Installation Requirements](#)
- [Installation Options](#)
- [Pervasive PSQL Products](#)
- [Pervasive PSQL v11 SP1 Optional Features](#)
- [Installation Review](#)

Installation Requirements

This section provides an overview of any special requirements you may need to know about in order to complete the Pervasive PSQL installation. The following overview is intended to accompany the software and hardware requirements listed on the Pervasive Software web site for Pervasive PSQL.

Installation Options

On Windows operating systems, Pervasive PSQL offers Complete and Custom installation options. On Linux distributions, each major component has its own separate installation.

Complete Installation

The Complete installation, which is recommended for most users, takes default actions for the operations performed during the installation and installs Pervasive PSQL and all optional features to the default installation location.

Custom Installation

The Custom installation is recommended for users that need control over their Pervasive PSQL installation. The Custom installation allows you to install Pervasive PSQL, along with only the features you need, in directory locations you specify.

The following sections describe the Pervasive PSQL products and optional features you can install using either of the installation options described here.

Pervasive PSQL Products

Pervasive PSQL is available in a Server, Workgroup (32-bit only) and Client installation. This section lists each of the Pervasive PSQL products and the base components inherent to each specific product installation.

Server Engine (64-bit)

- 64-bit MicroKernel Database Engine (MKDE), which provides Btrieve/MicroKernel API support for Pervasive PSQL applications.
- 64-bit SQL Relational Database Engine (SRDE), which provides ODBC/SQL API support for Pervasive PSQL applications.
- 64-bit Client Requesters and required components to access a MicroKernel engine for Windows or Linux.
- 32-bit Client Requesters and components needed to access a MicroKernel engine for Windows or Linux.
- Pervasive Distributed Tuning Interface (DTI) is used to configure and monitor the Pervasive components from low-level (compiled) applications.

Server Engine (32-bit)

- 32-bit MicroKernel Database Engine (MKDE), which provides Btrieve/MicroKernel API support for Pervasive PSQL applications.
- 32-bit SQL Relational Database Engine (SRDE), which provides ODBC/SQL API support for Pervasive PSQL applications.
- 32-bit Client Requesters and components needed to access a MicroKernel engine for Windows or Linux.
- Pervasive Distributed Tuning Interface (DTI) is used to configure and monitor the Pervasive components from low-level (compiled) applications.

Workgroup Engine (32-bit)

- 32-bit MicroKernel Database Engine (MKDE), which provides Btrieve/MicroKernel API support for Pervasive PSQL applications.
- 32-bit SQL Relational Database Engine (SRDE), which provides ODBC/SQL API support for Pervasive PSQL applications.
- 32-bit Client Requesters and components needed to access a MicroKernel engine for Windows or Linux.

- Pervasive Distributed Tuning Interface (DTI) is used to configure and monitor the Pervasive components from low-level (compiled) applications.



Note If you have a Workgroup engine running on a 64-bit machine, and you have 64-bit Btrieve or DTI applications, you may install both the Workgroup (32-bit) and Client (64-bit) engines on the same machine.

Client (64-bit)

- 64-bit Pervasive PSQL Client Requesters and required components to access a MicroKernel engine for Windows or Linux.



Note The Client (64-bit) installation does not include any utilities or documentation. To install documentation and utilities, you need to install both the Client (64-bit) and Client (32-bit) products.

Client (32-bit)

- 32-bit Pervasive PSQL Client Requesters and components needed to access a MicroKernel engine for Windows or Linux.
- Pervasive Distributed Tuning Interface (DTI) is used to configure and monitor the Pervasive components from low-level (compiled) applications.
- Pervasive PSQL Cache Engine

Pervasive PSQL v11 SP1 Optional Features

During a Custom install, Pervasive PSQL features may be excluded with the Pervasive PSQL product you are installing. The features listed here are optional, so if all the features in this section are excluded from the install, the Pervasive PSQL product is still installed by default.

The following lists the optional features available with each Pervasive PSQL installation, unless noted otherwise.

Xtreme I/O (Server 32-bit Only)

Xtreme I/O (XIO) is a database accelerator included with Pervasive PSQL. XIO increases database performance by accelerating disk access time for Pervasive PSQL data files. XIO can be installed only on Windows Server 32-bit platforms meeting the minimum system requirements for XIO. See [System Requirements](#) in *Advanced Operations Guide*.

Pervasive Access Methods

Pervasive Access Methods include the Pervasive PSQL Software Developer's Kit (SDK) and the DOS Requester.

ActiveX Interface Controls

A set of nine custom controls that enable development environments that support ActiveX to easily access Btrieve data. The interface includes a data source control and eight bound data controls.

ADO.NET Provider

ADO.NET is a .NET managed data provider, built with 100% managed code. The data provider is a native wire protocol provider, which means that the data provider will not have to call out to unmanaged code-code outside of the .NET Framework-in the form of a database client.

Btrieve DOS

The DOS VxD (Virtual eXtended Driver) (DOS client requester) is the Btrieve requester used for running DOS based applications via a Windows Command window. (Transactional access only)

DTO

The Pervasive Distributed Tuning Objects (DTO) are used from visual development environments.

JCL

The Java Class Library (JCL) is used for direct transactional access to data files via Java.

JDBC Driver

The JDBC driver is used for relational access to data files using the Java programming language.

OLE DB

The OLE DB access method includes runtime binaries used for transactional and relational access to data files.

PDAC

The Pervasive Direct Access Components (PDAC) includes a set of Visual Component Library (VCL) components that allow direct access to Pervasive Database Engines from within the Borland Delphi and C++ Builder Environments.



Note Design time component and sample downloads will be available for each access method on the Pervasive Software web site.

Utilities

The minimum set of utilities used to manage, configure and maintain the various components of the Pervasive PSQL database engines. The utilities included in this base set are installed as a set for all Pervasive PSQL products and may not be individually excluded from installation.

- Monitor
- Function Executor
- Btrieve Maintenance
- Rebuild
- Query Plan Viewer

- License Administrator
- Gateway Locator (Workgroup Engine only)

Cobol Schema Executor

The Pervasive Cobol Schema Executor utility is used for providing SQL access to COBOL based applications.

Data Dictionary File Builder

Pervasive PSQL Data Dictionary File Builder is used for creating and modifying Data Dictionary Files (DDFs).

Pervasive Control Center

The Pervasive Control Center is used for creating and manipulating database objects and accessing database tables via SQL.

Pervasive System Analyzer

The Pervasive System Analyzer utility is used for testing and troubleshooting network connectivity, viewing loaded modules and performing a component search.

Documentation

The Pervasive PSQL Engine and SDK user documentation is integrated into Pervasive PSQL Control Center (PCC). The documentation library is accessed through the PCC interface on the Welcome view, in the Help menu, by pressing F1 (Windows) or Shift F1 (Linux). Printed copies of the Pervasive PSQL Engine documentation may be purchased from Pervasive Software. The Pervasive PSQL SDK documentation titles are only available online.



Note If you choose to not install the documentation, context sensitive (F1) help will be unavailable from all of the Pervasive PSQL utility graphical user interfaces.

Java Runtime Environment (JRE)

The components of the JRE needed by the following features are installed as part of Pervasive PSQL:

- PCC
- DDF Builder
- Core utilities
- Documentation

The PSQL features use the local version of the JRE installed by Pervasive PSQL.



Note The installation of a local version of the JRE is for use only by the Pervasive PSQL features listed above. The local version of the JRE does *not* affect the requirements for developing Java applications using the Pervasive PSQL access methods Java Class Libraries (JCL) or JDBC. Those requirements, such as components obtained from java.sun.com, are discussed in the Pervasive PSQL software development kit (SDK) documentation. See *Java Class Library Guide* and *JDBC Driver Guide*.

Installation Review

This section provides you with a checklist to prepare you for installation and a set of commonly asked questions you should consider prior to installation. Please use this section as a review and a guide for a successful installation.

Quick Checklist This checklist provides a review of the requirements needed in order to install Pervasive PSQL. Each of these items should be met prior to beginning the install process.

- Your system hardware meets the minimum requirements to install Pervasive PSQL.
- Your operating system and network environment is supported by Pervasive PSQL.
- You have full administrator-level rights on the system where you plan to install Pervasive PSQL.
- You understand the different options available in the Complete and Custom installation setup so you can install only the set of components you need.
- You have reviewed the release notes in `readme.htm` on the installation media for important, late-breaking warnings and information that could not be included as part of the user documentation but may be essential to your installation and use of the product.
- Your application vendor supports the Pervasive PSQL engine.



Tip If you are uncertain, contact your application vendor or review the documentation provided by your vendor to ensure that they support the Pervasive PSQL engine version and mode that you want to install.

Windows Vista Checklist

In addition to the items in the Quick Checklist, Windows Vista users should also have a clear understanding of the differences and limitations of Standard Users versus Administrators and have the appropriate permissions to install on Windows Vista.

Common Pre-Installation Questions

This section contains some of the most common questions asked prior to installing Pervasive PSQL. These questions represent special case scenarios that could possibly prevent a successful first-time installation. Before you begin installation, consider the situations represented by these questions, along with the Quick Checklist to determine if you have met all the requirements and if there are situations that need special attention.

Where do I install the Pervasive PSQL Server?

The Server engine must be installed on the same computer where the database files are located.

What about the Server's client software?

The Pervasive PSQL Client is installed with every engine. So if you have a Pervasive PSQL engine installation, you can use your machine to connect to other remote engines as a client.

The Pervasive PSQL Client software must be installed on every computer that is expected to access the database.

Where do I install the Pervasive PSQL Workgroup?

Pervasive PSQL Workgroup can be installed on the same computer where the database files are located, or it can be installed on other computers to access the data over the network.

What about the Workgroup's Client software?

The Pervasive PSQL Client (32-bit) is installed with every engine. If you access remote files through another Workgroup engine, the Client software is already installed, so you do not need to install the Client separately.

How do I install Pervasive PSQL in a Microsoft Cluster Services environment?

If you plan to install Pervasive PSQL to a clustered environment using Microsoft Cluster Service, you should first refer to [Failover Cluster Support](#) in the *Advanced Operations Guide*. That chapter provides information about installing Pervasive PSQL in a clustering environment.

How do I install Pervasive PSQL in a Microsoft Terminal Services or Citrix MetaFrame environment?

If you plan to install Pervasive PSQL to a Microsoft Terminal Server or Citrix MetaFrame environment, you must be logged on to the console of the server as a user with system administrator rights to install. This can either be the physical console on the server or a remote console session.



Caution If you are installing the Workgroup or Client Engine on a Terminal Services Environment, the engine is installed by default to run as a service.

Only one instance of the database engine may run on any terminal server platform. You cannot run separate copies of the database engine within two or more terminal sessions. Refer to the Pervasive PSQL Web site for the list of Terminal Server Environments supported by Pervasive PSQL Server, Workgroup and Client (<http://www.pervasedb.com>).



Note If a user starts the Workgroup Engine or Cache Engine in a Terminal Services session or in a multi-user environment where fast-user switching is used, other users on the system cannot access that engine nor can they start their own copy of the engine.

Status code 3032 results if a second user attempts to access another user's engine through the transactional interface.

If it is desirable to have multiple local users accessing a local engine, install the Workgroup or Cache Engine as a service.

How do I install my Pervasive PSQL database engine in a Microsoft Active Directory environment?

The installation of the Pervasive PSQL database engine in an Active Directory environment requires no special steps. Follow the installation steps as described in this manual for the product you have purchased.

You may install the Pervasive PSQL database engine on a domain controller if you choose. Be aware, however, that activity on the domain controller may affect the performance of the database engine. For this reason, you may prefer to install Pervasive PSQL on a server that is not a domain controller.

Where do I install my Pervasive PSQL Clients accessing Web applications?

For Web applications, the Client must be installed on the same computer as the Web server. Multiple Web server platforms require a client on each platform.

Does it matter if I use Pervasive PSQL Clients that are of a different version than that of the database engine?

Pervasive recommends that you use Clients that are the *same* version as the database engine. If you choose, you may use a Client that is an older version than the database engine with which it interacts. In some situations, depending on the type of SDK access method used by your application, an older version requester will not work with the database engine. Your application will be unable to communicate with the database engine. For those situations, you must use Clients that are the same version as the database engine.

Clients that are a newer version than the database engine may or may not function correctly. Pervasive does not guarantee that newer versions of Clients will function correctly with older versions of the engine. Therefore, Pervasive recommends that you avoid the use of newer version Clients with an older engine.

Does it matter where I download the Pervasive PSQL v11 SP1 install file?

Yes, it does matter. If you are installing a downloaded version of Pervasive PSQL, do not place the install file in a location that is listed

in the PATH environment variables, as this can cause issues with file copying during install. Place the setup files in a location such as the Windows TEMP directory.

My system runs 24/7, is any time better than another for installing or upgrading to Pervasive PSQL v11 SP1?

The installation and upgrade should be performed during a period when all users are logged off the system and all data files are closed. As with any significant software installation, be sure to back up any important files on the target hard drive, including data files, before you begin the installation.

If you are performing an upgrade, keep the installation media and instructions from the old installation, in the unlikely event that you need to fall back to the previous version of the product.

How can I restrict users running in Terminal Services from changing Pervasive PSQL configuration settings, creating DSNs, and using the Monitor utility?

Pervasive PSQL clients running within Terminal Services client sessions can perform Pervasive PSQL administrative functions by default. For example, a user with such a client can change configuration settings for Pervasive PSQL, create DSNs, and use the Monitor utility. In prior releases, the ability to perform administrative functions was prohibited from the client.

To restrict this capability, a system administrator should follow these steps:

- 1 From **PCC**, open the properties for the **MicroKernel Router** under **Local Client**.
- 2 On the **Properties** dialog, check the option **Restrict Administrative Functions from a WTS Client**.
- 3 Click **OK**, then exit PCC and start it again for the setting to take effect.

Are there any special settings I need to make for my configuration that aren't listed here?

Yes, there are some default settings in Pervasive PSQL that need to be adjusted if your configuration includes certain qualities. For example, the default settings need adjustment if you have:

- Multiple network interfaces
- Database Files that must not include Embedded Spaces
- Microsoft Active Directory Service
- A Network that is subject to outages

Please review [Configuration for Special Installation Situations](#) for these or other relevant issues, especially if you encounter problems after installation.

Upgrading Your Pervasive PSQL Installation for Windows

Considerations for Users Upgrading to Pervasive PSQL v11 SP1

This chapter contains information about upgrading to Pervasive PSQL v11 SP1 from a supported previous version. The configuration settings that are migrated during an upgrade from a previous version to Pervasive PSQL v11 SP1 are also detailed in this chapter.

The following sections are included in this chapter:

- [Upgrading to Pervasive PSQL v11 SP1 From a Previous Version](#)
- [Common Questions After Upgrading to Pervasive PSQL](#)

Throughout this document, when an explicit version number is not specified (for example: Pervasive.SQL 7, Pervasive.SQL 2000, or Pervasive PSQL v11 SP1), all versions are included.

Upgrading to Pervasive PSQL v11 SP1 From a Previous Version

If you are upgrading a previous version of Pervasive PSQL to Pervasive PSQL v11 SP1, follow the same procedure as you would if you were installing for the first time. Pervasive PSQL v11 SP1 archives the previous version of Pervasive PSQL during installation. See Chapter 4, [Installing Pervasive PSQL Server for Windows](#) for detailed installation procedures.

If you are upgrading from older versions such as Pervasive.SQL 2000*i* and wish to make use of all the new version features, you must rebuild your data files so they use the v9.5 file formats. See [Converting Data Files](#) in *Advanced Operations Guide* for detailed information on how to use the Rebuild Utilities to convert your data files.

Licenses from any previous Pervasive products will not be migrated to Pervasive PSQL v11 SP1.

Considerations When Upgrading to Pervasive PSQL v11 SP1

Once you have reviewed the latest product information, review this list of considerations to complete your upgrade installation preparation.

- Pervasive PSQL Applications* - Be aware of what applications you have currently using previous versions of Btrieve or Pervasive PSQL in your environment. Don't forget to include both client and server-based applications, such as ArcServe.
- Vendor-Specific Information* - Check with your application vendors for any specific information regarding their product with Pervasive PSQL.
- TCP/IP Protocol* - Make sure that your TCP/IP network is configured correctly (you should be able to PING the server by name), and that any firewalls between the clients and server (including firewalls on those computers) are configured to pass database traffic. See [Windows FireWalls](#).

- ❑ *DOS Requester* - DOS applications are only supported via the BTRBOX requester. Native DOS machines should be migrated to a Win32 platform before upgrading. DOS applications are not supported on 64-bit Windows platforms. Therefore, BTRBOX is not supported on 64-bit Windows platforms.
- ❑ *New Features and File Rebuilding* - In order to make use of all the new version features, you must rebuild your data files so that they use the newest version file format. *Advanced Operations Guide* includes a chapter that details using the Rebuild Utility to rebuild your data files.
- ❑ *Back Up Data Files* - Make sure you have a current backup of all your data, database engine files and configuration prior to beginning upgrade installation

Common Questions After Upgrading to Pervasive PSQL

This section contains information that you should read after running the installation program. If you are having problems with your installation, go to Chapter 14, [Troubleshooting After Installation](#), or get help online from the Knowledge Base at the Pervasive Software website.

How to Handle Data Source Names (DSNs)

The following table describes the procedures for upgrading your DSNs after you have installed the Pervasive PSQL upgrade.

Table 2 DSN Considerations After a Pervasive PSQL Upgrade

Scenario	Discussion
<p>You have existing DSNs created with a version of Pervasive PSQL prior to Pervasive.SQL 2000i SP4</p>	<p>You must delete all existing DSNs before you upgrade. Once you have upgraded, re-create the DSNs to access the existing databases.</p> <p>New DSNs should connect to a named database, not to an Engine DSN because Engine DSNs are deprecated.</p> <p>See ODBC Connection Strings in <i>SQL Engine Reference</i>.</p>
<p>You have existing DSNs created with Pervasive.SQL 2000i SP4 or a later version of Pervasive PSQL</p>	<p>You should be able to access your databases by connecting to the existing DSNs.</p> <p>Note, however, the recommendation is that new or revised 32-bit applications should connect to a named database, not to an Engine DSN because Engine DSNs are deprecated.</p> <p>If you want to port your 32-bit application to 64-bit, then make the following changes.</p> <ul style="list-style-type: none"> • If the application uses DSN-less connections that connect using “Pervasive ODBC Client Interface,” change the connection string to “Pervasive ODBC Interface.” • If the application uses Engine or Client DSNs, you must create 64-bit DSNs that connect to a named database. (Also note that on 64-bit Windows operating systems, 64-bit system DSNs are distinct from 32-bit system DSNs because of the registry design.) <p>See ODBC Connection Strings in <i>SQL Engine Reference</i>.</p>
<p>You do not have any Pervasive PSQL DSNs defined</p>	<p>For details about creating DSNs, see DSNs and ODBC Administrator in <i>SQL Engine Reference</i>.</p> <p>If you want to use DSN-less connections for 32-bit applications, use the connection string “Pervasive ODBC Client Interface” and connect to a named database. For 64-bit DSNs, use the connection string “Pervasive ODBC Interface” to connect to a named database.</p>

How Do I Convert My Files From Previous Pervasive Products?

Converting your data files to 9.x format is not required, but you must convert them if you wish to take advantage of new features offered by the Pervasive PSQL v11 SP1 engine.

Users of previous Btrieve versions: Use the Rebuild utility to convert your existing pre-6.0 or 6.x files to 7.x, 8.x, or 9.x format. For more information, refer to the *Advanced Operations Guide*.

What User License Was Installed with Pervasive PSQL?

A trial license is installed if you leave the license number blank during installation.

There is no configuration necessary for the license. After installation, you can use the License Administrator utility to view your installed licenses. See the *Pervasive PSQL User's Guide* for more information on the License Administrator utility.

Installing Pervasive PSQL Server for Windows

Instructions for Installing the Pervasive PSQL Server Engine on Windows

This chapter contains procedures for installing and running Pervasive PSQL v11 SP1. The chapter contains the following sections:

- [Before You Install the Windows Server Engine](#)
- [Installing Pervasive PSQL Server for Windows](#)

Before You Install the Windows Server Engine

Before installing Pervasive PSQL v11 SP1, begin by reviewing the following documents:

- *Chapter 2, Preparing to Install Pervasive PSQL* - This chapter provides important information, including system requirements and platform specific notes, relevant to your operation.
- *Release Notes* - The release notes are located in readme.htm on the distribution media and contain late-breaking news that could not be included in the user documentation.

Platform Notes

This section contains installation information specific to the Windows platform.

- To install Pervasive PSQL for Windows, you must have full administrator-level rights on the machine where you will install Pervasive PSQL.

Installing the Engine on Terminal Server

To install Pervasive PSQL on a terminal server, you must be logged on to the console of the server as a user with system administrator rights to install. This can either be the physical console on the server or a remote console session.

Install Pervasive PSQL as you normally would, using the steps discussed in this manual. The operating system automatically handles the changing of terminal server modes.

Running the Engine on Terminal Server

Only one instance of the database engine may run on any terminal server platform. You cannot run separate copies of the database engine within two or more terminal sessions.

Installation Tips

- When installing Pervasive PSQL v11 SP1 for the first time on a system, Setup checks if all of the needed system files meet the minimum requirements. In some cases, these files are locked by the operating system and a reboot is required before Setup can continue.



Caution You must reboot your system if you encounter the reboot message. If you do not reboot your system, Setup encounters failures during engine and utilities configuration.

- If you have any trouble with the following installation, see Chapter 14, [Troubleshooting After Installation](#).

Installing Pervasive PSQL Server for Windows

You must install the Pervasive PSQL Server for Windows at the server itself; you cannot install it remotely from a client machine.



Note If the installation fails for any reason, the installation log file can be found in the Windows %Temp% directory.

The following steps explain how to install Pervasive PSQL Server from its media using the default interactive installation.

➤ To install Pervasive PSQL Server for Windows

- 1 Launch the installation program from your Windows machine.
 - a. Insert the Pervasive PSQL Server for Windows CD in the CD-ROM drive of your Windows server.
 - b. If the installation does not start automatically, click **Start** then **Run**, and type `drive:\autorun\autorun` where *drive* is the drive letter of your CD-ROM device.

The installation selection dialog displays.

- 2 Click **Server** installation for the desired bit architecture (32-bit or 64-bit).

The installation program begins its initial preparation. After the preparation completes, the **Welcome** screen appears.

- 3 If prompted, close or uninstall any running applications that may interfere with the Pervasive PSQL installation.



Note If you wish to leave one or more programs running that *may* interfere, you must click **Ignore** to continue. Unpredictable results may occur during the Pervasive PSQL installation if you ignore programs that may interfere.

- 4 At the **Welcome** screen, click **Next**.

5 On the **License Agreement** page, read and accept the Software License Agreement, then click **Next**.

6 Select the setup type: **Complete** (default) or **Custom**.

The **Complete** setup (recommended for most users) installs all the Pervasive PSQL v11 SP1 components using the default options and locations.

- If you choose a **Complete** install, click **Next** and continue with step 10.

The **Custom** setup (recommended for advanced users) allows you to specify the installation location, select the optional features and associated subfeatures to install, and determine the space requirements for the components.

- If you choose **Custom**, click **Next** and continue with the following steps.

7 To specify different installation locations, click **Change** for any of the folders listed, then enter or browse for a different folder. Click **OK** to accept the location.

8 Click **Next** to continue.

9 Select the optional features and associated subfeatures you want to exclude from the installation and click **Next**. All of the Pervasive PSQL optional features and subfeatures, except for Xtreme I/O, are selected for installation by default.

- Xtreme I/O
(This feature is only available on Windows 32-bit Server platforms meeting system requirements)
- Data Access
 - ActiveX Interface Controls
 - ADO.NET Providers
 - Btrieve DOS (32-bit only)
 - DTO
 - JCL
 - JDBC Driver
 - OLE DB
 - PDAC
- Utilities

- Java Utilities
 - Pervasive Control Center
 - Documentation
 - Data Definition File Builder
 - Notification Viewer
- Other Utilities
 - Cobol Schema Exec
 - Pervasive System Analyzer

10 Click **Install** to begin installation.

11 A dialog displays when the installation wizard completes. The product has been installed with a trial key that expires at the end of its trial period.

You have two choices at this point: continue and authorize the product with a permanent key, or end the installation (and later authorize the product with a permanent key).

- If you choose to continue and authorize the product, an Internet connection is required. Click **Next** and continue with step 12. (If you have no Internet connection, click **Next** then click **Finish**. See [Alternative Authorization Tasks](#) in *Pervasive PSQL User's Guide*.)
- If you choose to end the installation at this point, click **Next** then click **Finish**. (You may run the License Administrator utility at a later time to apply a key. See [License Administration](#) in *Pervasive PSQL User's Guide*.)

12 Enter your license key and click the button to apply the key.

(If you decide not to authorize the product at this point, click **Finish**. You may run the License Administrator utility at a later time to apply a key. See [License Administration](#) in *Pervasive PSQL User's Guide*.)

13 A message box displays with the status of the authorization action. Perform one of the following actions depending on the status:

- If the authorization status message is “**key is authorized**,” click **OK**, then click **Finish** to complete the installation.

- If the authorization status message reports an error or warning, click **OK**, and repeat step 12, ensuring that you enter a valid license key.
- 14** Register your product (recommended) as explained on the Registration page that displays, then close the Registration page.
- If you are prompted to reboot your system, please do so in order to ensure proper operation of your Pervasive PSQL v11 SP1 product.



Note The installation program modifies some of the environment variables. On Windows platforms, environment variables are stored in the Control Panel under System information.

Installing Pervasive PSQL Clients for Windows

Instructions for Installing the Pervasive PSQL Client on Windows

This chapter contains the following topics:

- [Before You Install the Windows Client Engine](#)
- [Installing the Pervasive PSQL Client for Windows](#)
- [Installing the BTRBOX Requester](#)
- [Understanding Client Requesters](#)
- [Where To Go From Here](#)

Before You Install the Windows Client Engine

This section contains information with which you need to be familiar to successfully install Pervasive PSQL. If you have not already, review the following documents before installing Pervasive PSQL client requesters:

- *Chapter 2, [Preparing to Install Pervasive PSQL](#)* - This chapter provides important information, including system requirements and platform specific notes, relevant to your operation.
- *Release Notes* - The release notes are located in `readme.htm` on the distribution media and contain late-breaking news that could not be included in the user documentation.

Installing the Pervasive PSQL Client for Windows

You must install the Pervasive PSQL Client for Windows at the client machine itself; you cannot install it remotely from a server machine.

The Pervasive PSQL Client is installed by default with the Server and Workgroup engines; a separate installation is no longer necessary.



Note If the installation fails for any reason, the installation log file can be found in the Windows %Temp% directory.

The following steps explain how to install Pervasive PSQL Client from its media using the default interactive installation.

➤ **To install Pervasive PSQL Client for Windows**

- 1 Launch the installation selection program from your Windows machine.
 - a. Insert the Pervasive PSQL product CD in the CD-ROM drive of your Windows server.
 - b. If the installation does not start automatically, click **Start** then **Run**, and type `drive:\autorun\autorun` where *drive* is the drive letter of your CD-ROM device.

The installation selection dialog displays.

- 2 Click **Client** installation for the desired bit architecture (32-bit or 64-bit).

The installation program begins its initial preparation. After the preparation completes, the **Welcome** screen appears.

- 3 If prompted, close or uninstall any running applications that may interfere with the Pervasive PSQL installation.

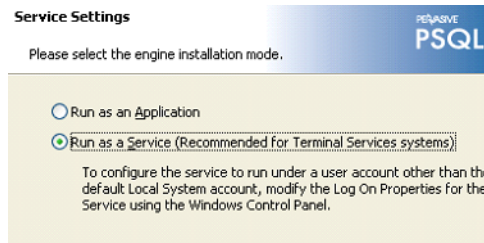


Note If you wish to leave one or more programs running that *may* interfere, you must click **Ignore** to continue. Unpredictable results may occur during the Pervasive PSQL installation if you ignore programs that may interfere.

- 4 At the **Welcome** screen, click **Next**.

- 5 For the 32-bit Client only, select the engine installation mode (): **Run as an Application** (default) or **Run as a Service**.

Figure 1 Engine Installation Mode Dialog Box



Caution Running the engine as a service requires the **Log On as Service** privilege. If you select to run the engine as a service under a user account other than the default Local System account, you will need to modify the Log On Properties for the Service using the Windows Control Panel.

- 6 On the **License Agreement** page, read and accept the Software License Agreement, and then click **Next**.

- 7 Select the setup type: **Complete** (default) or **Custom**.

The **Complete** setup (recommended for most users) installs all the Pervasive PSQL v11 SP1 components using the default options and locations.

- If you choose a **Complete** install, click **Next** and continue with step 11.

The **Custom** setup (recommended for advanced users) allows you to specify the installation location. For the 32-bit client only, you may also select the components and associated subfeatures to install, and determine the space requirements for the components.

- If you choose **Custom**, continue with the following steps.

- 8 To specify different installation locations, click **Change** for any of the folders listed, click **Change** for any of the folders listed, then enter or browse for a different folder. Click **OK** to accept the location.

- 9 Click **Next** to continue.
- 10 For the 32-bit client only, select the components and associated subfeatures you want to exclude from the installation and click **Next**. All of the Pervasive PSQL components and subfeatures are selected for installation by default.
 - Data Access
 - ActiveX Interface Controls
 - ADO.NET Providers
 - Btrieve DOS (32-bit only)
 - DTO
 - JCL
 - JDBC Driver
 - OLE DB
 - PDAC
 - Utilities
 - Java Utilities
 - Pervasive Control Center
 - Documentation
 - Data Definition File Builder
 - Other Utilities
 - Cobol Schema Exec
 - Pervasive System Analyzer



Note The Client 64-bit installation does not include the utilities, documentation, or SDK components listed above. To install them, you need to install both the Client 64-bit and Client 32-bit products.

- 11 Click **Install** to begin installation.
- 12 Once the installation is complete, the final dialog of the Installation Wizard displays. Click **Finish**.

If you are prompted to reboot your system, please do so to ensure proper operation of your Pervasive PSQL v11 SP1 product.



Note The installation program modifies some of the environment variables. On Windows platforms, environment variables are stored in the Control Panel under System information.

Installing the BTRBOX Requester

Pervasive PSQL v11 SP1 supports DOS Btrieve applications with the BTRBOX requester for Windows platforms. Use this Requester for legacy DOS applications.

A separate installation is no longer needed for the DOS Requester. The DOS Requester is automatically installed during a complete Pervasive PSQL v11 SP1 engine installation. In the case of a Custom installation, you must select the Btrieve DOS optional feature in the Pervasive Access Methods group to install the DOS Requester.



Note Clients using the DOS operating system will have only transactional access to the data files. No relational access is available for this platform.

Win32 DOS Box Support

BTRBOX allows a DOS application to run in a DOS box on a Windows workstation. This enables direct communication to the Windows 32-bit workstation components rather than to the database engine. This configuration can be used with either a local Pervasive PSQL v11 SP1 Workgroup engine, or a remote Pervasive PSQL v11 SP1 server engine. The TCP/IP or SPX protocol supported for client/server access depends on the configuration of the Windows 32-bit components.

DOS applications are not supported on 64-bit Windows platforms. Therefore, BTRBOX is not supported on 64-bit Windows platforms.

Understanding Client Requesters

A workstation that needs to access database files is considered a client to the machine running the Pervasive PSQL Server. A piece of software called a client requester, or requester for short, is required to access database files from a Pervasive PSQL database server. Your application's Pervasive PSQL calls go through the requester, which sends them to the Pervasive PSQL Server for processing and then returns the reply to your application.

Refer to the release notes provided with the product for a list of the platforms on which Pervasive PSQL requesters are supported. The requesters use the TCP, SPX or NetBIOS protocols to communicate with the server MicroKernel, depending on the type of server you have. Ensure that your workstation has the appropriate network protocol software installed.



Note Clients using DOS operating systems will have only transactional access to the data files. No relational access is available for this platform.

Types of Windows Requesters

Pervasive PSQL includes the following types of requesters for Windows:

- DOS
- Trace

You do not load or unload the Requester explicitly; the system loads the Requester with the first application call to Pervasive PSQL and unloads the Requester when you exit your application.

DOS Requesters

This type of requester is used for applications that run under the DOS operating system.

Trace Requesters

Trace requesters are used for troubleshooting (tracing) client problems at a low level. Generally, you will never need to perform this type of tracing. The low-level tracing is intended for use by

trained support staff. Your product vendor or Pervasive Software Support will direct you on how to conduct low-level client tracing, which includes how to use the trace requesters.

Note that the tools provided with Pervasive Software solve most troubleshooting issues. For example, you would run the network connectivity tests in Pervasive System Analyzer to verify network connectivity. Also at your disposal is the Knowledge Base at the Pervasive Software website, through which you may search for information about particular client issues.

***Where To Go
From Here***

A proper configuration is essential to smooth operation of your requester software. See Chapter 10, [Configuring Network Communications for Clients](#) for detailed information on how to configure Pervasive PSQL requesters.

Installing Pervasive PSQL Workgroup for Windows

Instructions for Installing the Pervasive PSQL Workgroup Engine on Windows

This chapter contains procedures for installing the Pervasive PSQL v11 SP1 Workgroup engine. The chapter contains the following sections:

- [Before You Install the Windows Workgroup Engine](#)
- [Installing the Pervasive PSQL Workgroup for Windows](#)

Before You Install the Windows Workgroup Engine

Before installing Pervasive PSQL v11 SP1 Workgroup, begin by reviewing the following documents for important information:

- ❑ *Chapter 2, [Preparing to Install Pervasive PSQL](#)* - This chapter provides important information including system requirements and platform specific notes that are relevant to your operation.
- ❑ *Release Notes* - The release notes are located in `readme.htm` on the distribution media and contain late-breaking news that could not be included in the user documentation.

Installation Tips

- When installing Pervasive PSQL v11 SP1 for the first time on a system, Setup will check if all of the needed system files meet the minimum requirements. In some cases, these files are locked by the operating system and a reboot is required before Setup can continue. Click **Yes** to reboot the system. Setup is then automatically restarted.
- It is strongly recommended that you reboot your system if you encounter this message. If you do not reboot your system, Setup will encounter failures during engine and utilities configuration.
- If you have any trouble with the following installation, see Chapter 14, [Troubleshooting After Installation](#).

Installing the Pervasive PSQL Workgroup for Windows



Note If the installation fails for any reason, the installation log file can be found in the Windows %Temp% directory.

The following steps explain how to install Pervasive PSQL Workgroup from its media using the default interactive installation.

➤ **To install Pervasive PSQL Workgroup for Windows**

- 1 Launch the installation program from your Windows workstation:
 - a. Insert the Pervasive PSQL v11 SP1 Workgroup CD in the CD-ROM drive of your Windows workstation.
 - b. If the installation does not start automatically, click **Start**, select **Run**, and type `drive:\autorun\autorun` where *drive* is the drive letter of your CD-ROM device.

The installation selection dialog displays.

- 2 Click **Workgroup** installation.

The installation program begins its initial preparation. After the preparation completes, the **Welcome** screen appears.

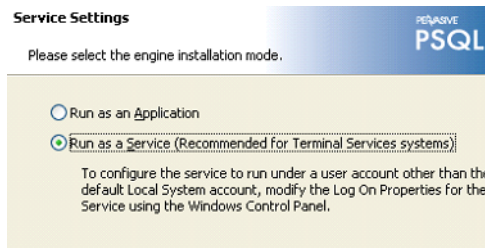
- 3 If prompted, close or uninstall any running applications that may interfere with the Pervasive PSQL installation.



Note If you wish to leave one or more programs running that *may* interfere, you must click **Ignore** to continue. Unpredictable results may occur during the Pervasive PSQL installation if you ignore programs that may interfere.

- 4 At the **Welcome** screen, click **Next**.
- 5 On the **License Agreement** page, read and accept the Software License Agreement, and then click **Next**.
- 6 Select the Workgroup Engine installation mode: **Run as an Application** (default) or **Run as a Service**.

Figure 2 Engine Installation Mode Dialog Box



Caution Running the engine as a service requires the **Log On as Service** privilege. If you select to run the engine as a service under a user account other than the default Local System account, you will need to modify the Log On Properties for the Service using the Windows Control Panel.

7 Select the setup type: **Complete** (default) or **Custom**.

The **Complete** setup (recommended for most users) installs all the Pervasive PSQL v11 SP1 components using the default options and locations.

- If you choose a **Complete** install, click **Next** and continue with step 11.

The **Custom** setup (recommended for advanced users) allows you to specify the installation location, select the components and associated subfeatures to install, and determine the space requirements for the components.

- If you choose **Custom**, click **Next** and continue with the following steps.

8 To specify different installation locations, click **Change** for any of the folders listed, then enter or browse for a different folder. Click **OK** to accept the location.

9 Click **Next** to continue.

10 Select the components and associated subfeatures you want to exclude from the installation and click **Next**. All of the Pervasive PSQL components and subfeatures are selected for installation by default.

- Data Access

- ActiveX Interface Controls
- ADO.NET Providers
- Btrieve DOS (32-bit only)
- DTO
- JCL
- JDBC Driver
- OLE DB
- PDAC
- Utilities
 - Java Utilities
 - Pervasive Control Center
 - Documentation
 - Data Definition File Builder
 - Notification Viewer
 - Other Utilities
 - Cobol Schema Exec
 - Pervasive System Analyzer

11 Click **Install** to begin installation.

12 A dialog displays when the installation wizard completes. The product has been installed with a trial key that expires at the end of its trial period.

You have two choices at this point: continue and authorize the product with a permanent key, or end the installation (and later authorize the product with a permanent key).

- If you choose to continue and authorize the product, an Internet connection is required. Click **Next** and continue with step 13. (If you have no Internet connection, click **Next** then click **Finish**. See [Alternative Authorization Tasks](#) in *Pervasive PSQL User's Guide*.)
- If you choose to end the installation at this point, click **Next** then click **Finish**. (You may run the License Administrator utility at a later time to authorize a key. See [License Administration](#) in *Pervasive PSQL User's Guide*.) See also [Authorization of Workgroup Key on Vista and Windows 7](#).

13 Enter your license key and click the button to apply the key.

(If you decide not to authorize the product at this point, click **Finish**. You may run the License Administrator utility at a later time to authorize a key. See [License Administration](#) in *Pervasive PSQL User's Guide*.)

- 14 A message box displays with the status of the authorization action. Perform one of the following actions depending on the status:
 - If the authorization status message is “**key is authorized**,” click **OK**, then click **Finish** to complete the installation.
 - If the authorization status message reports an error or warning, click **OK**, and repeat step 13, ensuring that you enter a valid license key.
- 15 Register your product (recommended) as explained on the Registration page that displays, then close the Registration page.

If you are prompted to reboot your system, please do so in order to ensure proper operation of your Pervasive PSQL v11 SP1 product.



Note The installation program modifies some of the environment variables. On Windows platforms, environment variables are stored in the Control Panel under System information.

Authorization of Workgroup Key on Vista and Windows 7

You can encounter difficulty authorizing a permanent key for Pervasive PSQL Workgroup on Windows Vista or Windows 7 if the following conditions are all true:

- Pervasive PSQL Workgroup was installed as an application.
- The Workgroup database engine is running without administrative privileges. Note that, by default, applications run with privileges of a standard user unless the privileges are elevated. *That is, even if you are a member of the administrator's group and you start the Workgroup database engine without using Run as Administrator to elevate privileges, the engine runs with privileges of a standard user.*

- A permanent key for Pervasive PSQL Workgroup was not supplied during the installation process. That is, you chose to authorize the permanent key *after* installation by using a licensing utility.

Complete the following steps to ensure a permanent key is correctly authorized:

- 1 If the Pervasive PSQL Workgroup application is running, close the application (stop the database engine by right-clicking on the engine tray icon then clicking stop).
- 2 From File Explorer, locate the file w3dbsmgr.exe. Look for the file under <install_drive>\Program Files\Pervasive Software\PSQL\bin.
- 3 Right-click w3dbsmgr.exe then click **Run as Administrator**. You must have administrative rights or know the password and name of a user with administrative rights. You need to elevate the privileges of the database engine before authorizing a key with a licensing utility.
- 4 Start License Administrator from the Pervasive group in the Start menu (or use the command line interface licensing utility if you prefer).
- 5 Type, or paste, the permanent key for Pervasive PSQL Workgroup in the **Key** field, then click the button to apply the key.
- 6 Optionally, stop the database engine and re-start it without elevated privileges.

Note that elevating the privileges for a license administrator utility is *not* the solution. The database engine itself, w3dbsmgr.exe, is what requires elevated privileges.

After Installing Pervasive PSQL for Windows

Answers to Common Post Installation Questions for Pervasive PSQL on Windows

The chapter contains the following sections:

- [Common Questions After Installing Pervasive PSQL](#)
- [Uninstalling Pervasive PSQL](#)

Common Questions After Installing Pervasive PSQL

This section contains information that you should read after running the installation program. If you are having problems with your installation, go to Chapter 14, [Troubleshooting After Installation](#), or get help online from the Knowledge Base at the Pervasive Software website.

What happened to PVSWBIN on Windows platforms?

Starting with Pervasive PSQL v10, files are no longer installed to <drive>:\pvs\bin on Windows platforms. This change comes as Pervasive PSQL adapts to suggested guidelines from Microsoft.

Where are the Pervasive PSQL files installed?

Table 3 lists the default locations where Pervasive PSQL installs the program and application data files on Windows platforms. Table 15 lists similar information for Linux platforms.

Table 3 Pervasive PSQL v11 SP1 Default Windows Installation Locations

Platform	File Types	Default Installation Location
Windows Vista and later ¹ (64-bit)	Application Data	<drive:>\ProgramData\Pervasive Software\PSQL\
	Program Files (64-bit)	<drive:>\Program Files\Pervasive Software\PSQL\
	Program Files (32-bit)	<drive:>\Program Files (x86)\Pervasive Software\PSQL\
Windows pre-Vista ² (64-bit)	Application Data	<drive:>\Documents and Settings\All Users\Application Data\Pervasive Software\PSQL\
	Program Files (64-bit)	<drive:>\Program Files\Pervasive Software\PSQL\
	Program Files (32-bit)	<drive:>\Program Files (x86)\Pervasive Software\PSQL\
Windows Vista and later ¹ (32-bit)	Application Data	<drive:>\ProgramData\Pervasive Software\PSQL\
	Program Files	<drive:>\Program Files\Pervasive Software\PSQL\

Table 3 Pervasive PSQL v11 SP1 Default Windows Installation Locations

Platform	File Types	Default Installation Location
Windows pre-Vista ² (32-bit)	Application Data	<drive:>\Documents and Settings\All Users\Application Data\Pervasive Software\PSQL\
	Program Files	<drive:>\Program Files\Pervasive Software\PSQL\
<p>Note: The DOS Requester files are installed by default on all Windows platforms at <drive:>\%WINDIR%\SYSTEM32\</p> <p>¹ Windows Vista and later refers to Windows Vista and any Windows operating system released after Windows Vista that is currently supported by Pervasive PSQL.</p> <p>² Windows pre-Vista refers to any Windows operating system currently supported by Pervasive PSQL that was released prior to Windows Vista.</p>		

What is an Application Data file?

Application data files are typically files to which the system can write. Examples of Application Data files include log files, tutorial files, and sample database files, such as DEMODATA and TEMPDB.

What is a Program File?

Program files are typically files the system requires in order to function. Examples of program files include binary system files, executable files, dynamic link libraries and JAR files.

What is the difference between 32-bit and 64-bit Program Files?

Microsoft guidelines recommend that 64-bit components are installed in a separate location to 32-bit components. Pervasive PSQL 64-bit components are installed in the 64-bit program files location and are registered in the Windows registry under the 64-bit hive. 32-bit components are installed in the 32-bit program files location and are registered in the Windows registry under the 32-bit (x86) hive.

What if I need a 64-bit Client in a Workgroup environment?

By default, the 32-bit Client is installed with the Workgroup engine. If you have a Workgroup engine running on a 64-bit machine, and you have 64-bit applications (Btrieve or DTI) that you need to access

with a client, you may install both the Workgroup (32-bit) and Client (64-bit) engines on the same machine. Install each product as you would normally; no special configuration is required.

What happened to the Client install image?

Previous versions of Pervasive PSQL contained an image used for installing the client software needed to access the Pervasive PSQL database. This image is no longer needed since Pervasive PSQL v11 SP1 now installs the Client (32-bit) components with the Server and the Workgroup editions. Use the Pervasive PSQL v11 SP1 Client (32-bit or 64-bit) to install on individual Client machines.

Do I need to install the Client with a Workgroup engine?

If you are installing the Workgroup engine, you must have a license for and install the software on every computer that is expected to share data within your workgroup. Because every computer must have the Workgroup engine installed and the client software is installed with every engine by default, there is no need to install the client software separately.



Note Only the 32-bit Client components are installed.

How Do I Read the Online Documentation?

The viewer for the documentation library is integrated into Pervasive PSQL Control Center (PCC). Access the documentation library through the PCC interface on the Welcome view, in the Help menu, by pressing F1 (Windows) or Shift F1 (Linux).

You can also view the documentation in the form of Adobe Acrobat (PDF) files. These PDF files are available on the Pervasive PSQL installation media in the **Books** directory.

How Do I Verify or Update My User License?

Licenses from previous versions of Pervasive PSQL are not migrated or transferable to Pervasive PSQL v11 SP1. You must have a license applicable for Pervasive PSQL v11 SP1, unless you choose to install using the trial version of the product.

The License Administrator utility is documented in *Pervasive PSQL User's Guide* in the section [License Administration](#). Please refer to that document for information on user licenses.

What User License Was Installed with Pervasive PSQL?

A trial license is installed if you leave the license number blank during installation.

There is no configuration necessary for the license. After installation, you can use the License Administrator utility to view your installed licenses. See the *Pervasive PSQL User's Guide* for more information on the License Administrator utility.

Where To Go From Here

If you had trouble during installation, see [Troubleshooting After Installation](#).

If you completed installation successfully, continue with your Pervasive PSQL deployment by installing and configuring the clients for the machines that will connect to your servers. Review Chapter 5, [Installing Pervasive PSQL Clients for Windows](#).

Uninstalling Pervasive PSQL

The uninstall program removes the Pervasive PSQL engine, and all related components from your system that were added by the installation program, including registry settings, configurations and Pervasive PSQL system and sample databases.

The uninstall program does **not** remove the following:

- Databases that you create under the Pervasive PSQL Server installation directory.
- DSNs and database names associated with those databases.
- Databases in locations other than the Pervasive PSQL Server installation directory.
- DSNs and database names associated with those databases.

➤ **To uninstall Pervasive PSQL**

- 1 In the Windows Control Panel, select **Add/Remove Programs**.
- 2 Select the installed **Pervasive PSQL v11 SP1** product from the list.
- 3 Click **Change** then **Next**.
- 4 Click **Remove** then **Next** and follow any prompts during the uninstall.

If prompted, close or uninstall any running applications that may interfere with uninstalling Pervasive PSQL.



Caution Unpredictable results may occur during the uninstall if you ignore programs that may interfere.

Reboot your system, if prompted to do so.

Configuring the Workgroup Engine

8

Understanding the Available Workgroup Engine Configurations

This chapter discusses the concepts behind using the Workgroup engine. The configurations available for the Workgroup engine are covered, as well as the procedures for setting up those configurations. Instructions for setting up a Gateway configuration using the Gateway Locator Utility are included.

The sections in this chapter include:

- [Overview](#)
- [Setting Up a Small Client/Server Configuration](#)
- [Setting Up a Peer-to-Peer Configuration](#)
- [Setting Up a Gateway Configuration](#)
- [Running the Workgroup Engine as a Service](#)

Overview

This section explains the basic concepts and requirements of Workgroup engines. If you need more in-depth information about the Workgroup engine, refer to the *Advanced Operations Guide*. The *Advanced Operations Guide* contains detailed technical information about the Workgroup engine, setting up a Gateway configuration, and re-directing locator files.

Installation Requirements

Every computer that may be used to access the same data at the same time must have a Workgroup engine installed on it.

Operating System Security

Only database server engines can enforce OS level file security based on the privileges assigned to the login user name. The Workgroup engine does not attempt to do this. In a small office, where Workgroup engines are most common, this can be considered a plus because they are usually short on networking experts, and the fewer barriers to successful data access the better.

When to Use Workgroup

There are three main configurations in which you would want to use the Workgroup engine.

Small Client/Server Configuration

The first configuration takes place when all the data is located on a single computer with a Workgroup engine installed, and there is limited sharing of data. This configuration is roughly equivalent to a small client/server configuration.

Peer-to-Peer Configuration

Another situation when you would want to use the Workgroup engine is when the data is distributed among the workstations. This is called a peer-to-peer topology. This configuration is used when each application typically stores much of its own data on the local hard drive, but periodically needs to access data from other workstations or share its own data with others.

In this configuration, each computer shares its data directory or directories. Any computer that needs access to that data maps one or more drives to the shared data directories. Then the Workgroup

engine on each computer acts as a mini-server engine to read/write all changes to the data files on that machine.

Gateway Configuration

The third topology requiring the use of the Workgroup engine is when the data is stored on a file server where there is no MicroKernel engine. This can be a UNIX server or other type of network file server that gets backed up regularly, but cannot support a MicroKernel engine. In this situation, the first Workgroup engine that opens files in a directory on the server becomes the Gateway to each file in that directory. The other workstations access the data in a client-server fashion through that Gateway engine.

The Gateway engine for a given directory identifies itself by creating a file named `~PVSW~.LOC` in that directory. This file is called a *Gateway locator file* and contains the network name of the computer where the Gateway engine is located. Other Workgroup engines attempting to access this data read the locator file to find the name of the engine they must communicate with in order to access the data.

You can ensure that the same engine always services the files in a given directory by making the locator file read-only. This is called a *static gateway*, also referred to as a *fixed gateway*. See [To Set up a Fixed Gateway](#) for more information.

The Gateway engine acts as a server engine as it reads and writes pages to the data files, allowing it to make the most use out of its cache. The Gateway feature is designed so that the ownership of any particular directory can change whenever the current gateway engine has no more client applications with any files open in that directory. When the last data file is closed in a directory by a given database engine, the engine releases and deletes the locator file. When the next engine opens a data file, that engine becomes the new gateway to the directory where the data file(s) resides.

What is a Gateway Engine?

A *Gateway engine* is a Workgroup engine that acts as the sole point of access to all data files in a particular directory on a remote file server. If several Workgroup engines are accessing the same database at the same time, they do not all open the files simultaneously, nor do they share the files. Rather, the first Workgroup engine to access that database becomes the temporary “owner” of those files, and all other Workgroup engines must access the data by contacting the Gateway engine. Only the Gateway engine has the files open and

reads/writes the files. The other Workgroup engines act as clients, making requests to the Gateway engine acting as a mini-server engine.



Caution Make certain the Gateway computer is NOT shut down while users are still using it as a Gateway, or data loss can occur.

A Gateway engine only comes into play when no database engine is installed on the machine where the data files are, or when the database engine on that machine is not operating.

Setting Up a Small Client/Server Configuration

As explained in [Small Client/Server Configuration](#), you should use this set up when you have only a few workstations sharing data located on a central computer where you have a Workgroup engine installed.

If you have data located on many computers, or if you do not or cannot install a database engine on the computer where the data is located, you should use one of the other configurations.

➤ To Set Up a Small Client/Server Configuration

- 1 You must have the Workgroup engine installed both on the central computer where the data is and on all computers expected to access the data.

Ensure that the Workgroup engine on the central computer where the data is located is operational each time the computer is started, before any other database engines attempt to access the data. If the Workgroup engine was installed as an application, ensure that the application starts if the computer is restarted. A Workgroup engine starts by default if installed as a server. See [Running the Workgroup Engine as a Service](#).



Note You may inadvertently fall into a Gateway configuration if the database engine on the machine where the data is located is not started when the computer is started. If another Workgroup engine attempts to access the data and the local database engine is not running, the other database engine may establish itself as the Gateway for those data files.

You can resolve this situation by shutting down the computer where the data is located, and starting it again, while ensuring that no other computers request data before the local Workgroup engine is restarted. You may need to remove the file `~PVS~.LOC` from the data directory to ensure the Gateway is not re-established.

The best way to ensure that only the Gateway services the file is to set a static gateway locator file using the **Gateway Locator Utility**.

- 2 Share the directory where the data is located so that other computers can map a drive to the data directory.
- 3 Ensure that each workstation expected to access the data can access the named database on the central computer. See [Named Database](#) in *Advanced Operations Guide*.

Setup is complete. The Workgroup engine on the machine where the data is located now acts as a mini-server, to fulfill all requests for data on that machine.

Setting Up a Peer-to-Peer Configuration

As explained in [Peer-to-Peer Configuration](#), you should use this set up when you have workstations sharing local data as well as data located on many different machines, and each machine has the Workgroup engine installed.

This configuration is similar to the small client/server configuration discussed above, except that now every Workgroup engine is sharing data as a server.

If you have data located on only one computers, or if you do not or cannot install a database engine on the computer where the data is located, you should use one of the other configurations.

➤ To Set Up a Peer-to-Peer Configuration

- 1 You must have the Workgroup engine installed on each computer where data is located, and installed on all computers expected to access the data.

Ensure that the Workgroup engine on each computer where the data is located is operational each time the computer is started, before any other database engines attempt to access the data. If the Workgroup engine was installed as an application, ensure that the application starts if the computer is restarted. A Workgroup engine starts by default if installed as a server. See [Running the Workgroup Engine as a Service](#).



Note You may inadvertently fall into a Gateway configuration if the database engine on a machine where data is located is not started when the computer is started. If another Workgroup engine attempts to access the data and the local database engine is not running, the other database engine may establish itself as the Gateway for those data files.

You can resolve this situation by shutting down the computer where the data is located, and starting it up again, while ensuring that no other computers request data before the local Workgroup engine is restarted. You may need to remove the file ~PVSW~.LOC from the data directory to ensure the Gateway is not re-established.

The best way to ensure that only the Gateway services the file is to set a static gateway locator file using the **Gateway Locator Utility**.

- 2 On each computer where data is located, share the directory where the data is located so that other computers can map a drive to the data directory.
- 3 Ensure that each workstation expected to access the data can access the named database to which the data belongs. See [Named Database](#) in *Advanced Operations Guide*.

Also, ensure that each Workgroup engine can access any local data from its own physical drive.

- 4 Set up is complete. The Workgroup engine on each machine where data is located now acts as a mini-server, to fulfill all requests for data on that machine.

Each Workgroup engine also handles any local data access, that is, database requests from applications on that machine for data that resides on the same machine.

Setting Up a Gateway Configuration

As explained in [Gateway Configuration](#), you should use this set up only when you have data files on a computer where no database engine is installed.

If you have database engines installed on all machines, you should use one of the other configurations.



Note You may inadvertently fall into a Gateway configuration if the database engine on a machine where data is located is not started when the computer is started. If another Workgroup engine attempts to access the data and the local database engine is not running, the other database engine may establish itself as the Gateway for those data files.

You can resolve this situation by specifying a permanent Gateway as described in this section, or by shutting down the computer where the data is located, and starting it up again, while ensuring that no other computers request data before the local Workgroup engine is restarted. You may need to remove the file `~PVSW~.LOC` from the data directory to ensure the Gateway is not re-established.

The best way to ensure that only the Gateway services the file is to set a static gateway locator file using the **Gateway Locator Utility**.

Floating or Fixed Gateway

You can set up two different Gateway configurations. The default behavior is a *floating* Gateway configuration. In this configuration, the first engine to open the remote data files becomes the Gateway engine for that directory until all files in the directory are closed. Then the next engine to open the data files becomes the new Gateway. This configuration is the most flexible, but also can entail delays upon initial connection to the database, as the engine tries the different network protocols and checks for an existing Gateway engine.



Note Using a floating Gateway in a peer-to-peer configuration with multiple shared data sources is not recommended. This configuration is supported and it operates as designed, however, with multiple engines shuffling ownership among multiple data locations, connection delays may be significant. It is also possible to create a situation where a Workgroup engine on a remote machine serves as the Gateway for data located on your local hard drive. Obviously there is no reason to endure this delay when your local Workgroup engine can serve this data with higher performance.

You can avoid this situation by ensuring that the Workgroup engine on every computer is started when the computer is started. You must also ensure that someone logs on to each computer, because normally the Workgroup engine doesn't start until a user logs on.

You can also avoid this situation by permanently assigning each machine as the Gateway for the data files located on it. See [To Set up a Fixed Gateway](#) for information on how to perform this task.

The second configuration is called a *fixed* or *permanent* Gateway configuration. In this configuration, a specific engine is permanently assigned as the Gateway engine for a specific directory. If that engine is not running when another engine attempts to access the data, an error code results and the data is not available.

➤ **To Set up a Floating Gateway**

- 1 This is the default behavior. Ensure that each Workgroup computer can access the named database (and therefore its data). See [Named Database](#) in *Advanced Operations Guide*.

The Gateway assignment now floats dynamically as different Workgroup engines access the remote data.

➤ **To Set up a Fixed Gateway**

To specify a permanent Gateway engine for a given directory, you need only change the attributes of the ~PVSW~.LOC file to read-only, once it contains the name of the desired engine. There are several ways to perform this task.

Use the Gateway Locator Utility

- 1 On the **Start** menu select **Gateway Locator** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the **Target Directory** area, type in or browse for the directory containing the data files for which you wish to set up a permanent Gateway.
- 3 In the **Directory Status** area, click **Change**. In the dialog box that appears, click **Assign a Gateway**, then type in or browse for the network name of the computer that you want to be the Gateway. Click **OK**.
- 4 Back in the main Gateway Locator window, check **Permanent assignment**. Click **Exit**.

Use the DOS Command Line

- 1 Use the **ATTRIB +R** command at a DOS command prompt to change the attributes of the `~PVSW~.LOC` file.

For example, if your current directory is the directory where the file is located, you can type the following command:

```
ATTRIB +R ~PVSW~.LOC
```

Use the Windows Explorer

- 1 Right-click the `~PVSW~.LOC` file in the directory for which you want to make a permanent Gateway assignment. Choose **Properties** from the pop-up menu.
- 2 In the **Properties** window, on the **General** tab, click **Read-only** in the section labeled **Attributes**. Click **OK**.

Working with the Gateway Locator Utility

The Gateway Locator Utility provides control of and insight into any Gateway configuration you have on your network. This section explains how to use the utility for a variety of purposes.

This utility enables users to determine or change the Workgroup Engine which is being used as the gateway for the data files in a particular directory. The Gateway Locator utility is used only with Pervasive PSQL v11 SP1 Workgroup Engine.

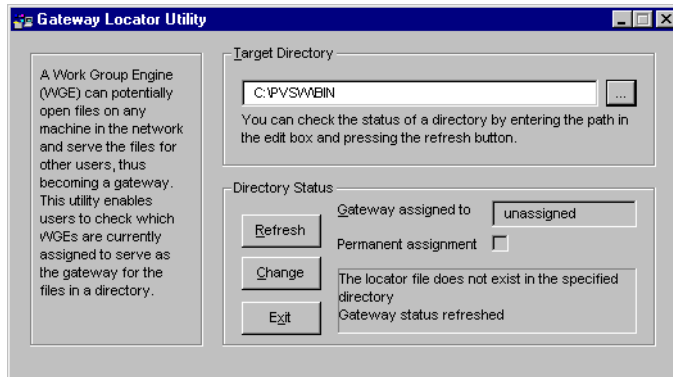
The Gateway Locator operates by reading and manipulating the locator file, `~PVSW~.LOC`, which resides in any directory which is assigned a Gateway engine. If this file is locked (in use), the Gateway

Locator can only locate, not change, the Workgroup engine being used as a Gateway for that particular directory.

➤ **To start the Gateway Locator Utility**

- 1 On the **Start** menu select **Gateway Locator** from the **Pervasive PSQL v11 SP1** program group.

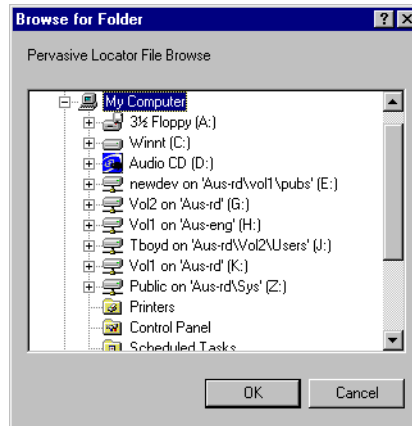
Figure 3 Gateway Locator Main Dialog Box



Note The Gateway Locator can be used to set the gateway for any data directory. Data directory locations are not stored with the tool. Consequently, you must always set the directory path before you click **Change**.

- 2 In **Target Directory**, enter or browse for the directory path which contains the data files for which you wish to locate or change the Gateway engine.
- 3 The default target directory is the current working directory. Clicking the browse (...) button allows you to browse for the target directory, by bringing up the following dialog box:

Figure 4 Gateway Locator Browse Dialog Box



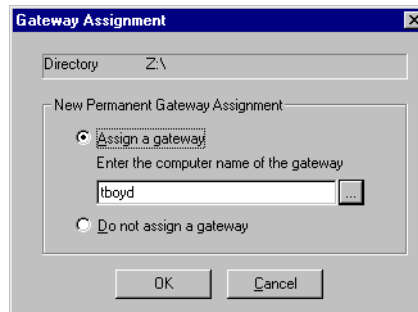
➤ Locating the Gateway Workgroup Engine

Once the target directory is selected, clicking the **Refresh** button causes the name of the Gateway engine for that directory (if such exists) to appear in the **Gateway Assigned To** box. If no Gateway exists for a particular directory, the box reads “unassigned.”

➤ Changing the Gateway Workgroup Engine

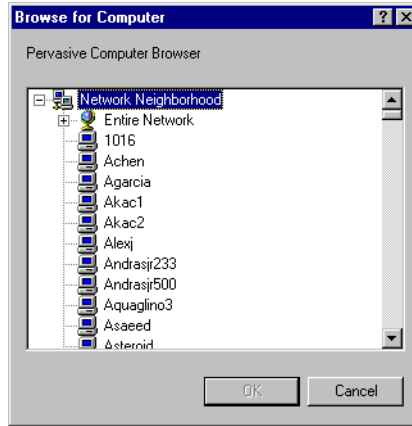
Once the target directory is selected, click **Change** to choose the Workgroup engine which you wish to serve as Gateway for a particular directory (this button is disabled if the locator file for that directory is locked.) The following dialog box appears:

Figure 5 Gateway Assignment Dialog Box



Enter or browse for the machine name you wish to serve as gateway.

Figure 6 Browse for Computer Dialog Box



Running the Workgroup Engine as a Service

By default, the Workgroup engine is installed to run as an application. During a Custom installation you may configure your workstation to run the Workgroup engine as a service rather than as a console application. Running the engine as a service allows the engine to start automatically when the operating system starts. A user is not required to log in to start the engine.

No tray icon appears when you run the Workgroup engine as a service.

Configuration

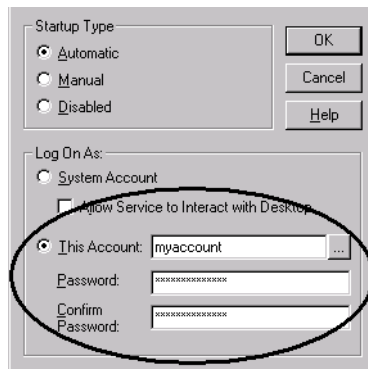
In order to configure your Workgroup engine to run as a Service as opposed to an application, you must reinstall the Pervasive PSQL v11 SP1 Workgroup engine. Choose the **Custom** install and select **Run As Service** on the Engine Settings dialog box during install.



Caution Running the engine as a service requires the **Log On as Service** privilege. If you select to run the engine as a service under a user account other than the default Local System account, you will need to modify the Log On Properties for the Service using the Windows Control Panel.

Note that on Windows platforms, a valid user name and password are required to access Pervasive PSQL databases on another machine. "System Account" has no rights on other machines to Pervasive PSQL databases.

If you want the Pervasive Workgroup service to access databases on another machine, then you must specify a valid user name and password for the other machine, unless there is a running Pervasive Workgroup available locally on the other machine that can be used to access the databases instead. Access the properties for the Pervasive Workgroup service. (Double-click the service on the Services dialog.) In the **Log On As** section, check **This Account** and specify a valid user name and password.



Stopping the Service

If you want to stop and then restart the service (and not permanently remove the service), then just reboot the machine.

You stop the service on Windows platforms just as you would any other service.

➤ **Stopping the Engine as a Service on Windows**

- 1 In the Windows **Control Panel**, click **Administrative Tools**, then double-click **Services**.
- 2 Right-click the service name assigned to the Workgroup engine and click **Stop**.

Configuring Engine Network Communications

How to Set Up Network Communication for Pervasive PSQL Engines

This chapter reviews the types of network communication protocols supported with Pervasive PSQL and how to set protocol support for your network, if you require different settings.

- [Determining What Kind of Network You Have](#)
- [Engine Network Communication Settings](#)
- [Setting Up TCP/IP Support](#)
- [Setting Up SPX Support](#)
- [Setting Up NetBIOS Support \(Workgroup only\)](#)
- [Avoiding Unavailable Protocols](#)

Determining What Kind of Network You Have

This section explains how to determine the network protocol that you should use with the database engine. If you already know what protocol or protocols are supported on your network, you can skip this section.

Database Engine on Windows

If your network is 100% Microsoft, and you have a database Server engine, then your network probably uses TCP/IP. The Server engine does not support NetBIOS.

You can run applications over SPX on Microsoft networks if the applications use only the Pervasive PSQL transactional interface (Btrieve or ODBC).

If your network is 100% Microsoft, and you are using Workgroup engines, then you can use either NetBIOS or TCP/IP.

Engine Network Communication Settings

This section lists the configuration settings used by the Pervasive PSQL engines for network communication. These settings may be changed using a command line utility or from within PCC on the engine properties.

The *Advanced Operations Guide* provides detailed information about each of the settings. See the following configuration settings in *Advanced Operations Guide* for network communication:

- [Auto Reconnect Timeout](#)
- [Enable Auto Reconnect \(Windows only\)](#)
- [Listen IP Address](#)
- [Supported Protocols](#)
- [TCP/IP Multihomed](#)
- [TCP/IP Port](#)
- [NetBIOS Port \(Workgroup engines only\)](#)

Setting Up TCP/IP Support

By default, TCP/IP is supported between Pervasive PSQL clients and remote database engines or between multiple Workgroup engines. If you have modified the default settings or need to verify that TCP/IP support is available, refer to this section.



Note To perform any of the tasks in this section, you must have full administrator-level rights on the machine where the database engine is running, or be a member of the Pervasive_Admin group defined on the machine where the database engine is running.

➤ To Enable TCP/IP Support

Complete the following steps to ensure that the database engine can communicate with clients over TCP/IP networks.

- 1 On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the Pervasive PSQL Explorer, double-click **Engines** to display a list of the engines registered with PCC.
- 3 Right-click the target engine and click **Properties**. Login if prompted.
- 4 Click **Communication Protocols**, and the list of **Supported protocols** displays. If the list of Supported protocols shows the value **TCP/IP** checked, then TCP/IP is already supported.
- 5 Click **TCP/IP** then restart the database engine for the changes to take effect.



Tip Remember that you also need to confirm that your client computers or the client software on your other Workgroup computers are configured to use TCP/IP, as well. See Chapter 10, [Configuring Network Communications for Clients](#).

➤ **To Enable Multihomed TCP/IP Support**

Completing the following steps configures your Windows server to use two installed network cards.

- 1 On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the PCC Pervasive PSQL Explorer, double-click **Engines** to display the list of registered engines with PCC.
- 3 Right-click the target engine and click **Properties**. Login if prompted.
- 4 Click **Communication Protocols** and click **TCP/IP Multihomed** to configure the server engine to listen for client connections on multiple network interfaces.

If you only have one network interface, this setting is ignored.

- 5 Restart the server engine for the changes to take effect. You do not need to make any changes to client settings.



Note If your server computer has two network interfaces, and you set the value of **TCP/IP Multihomed** to **Off**, you must edit the setting **Listen IP Address** and specify the TCP/IP address of the interface on which you want the database engine to listen.

If you do not specify an IP address, the database engine will receive communications only from the first network interface to bind to the operating system. Because this can vary with driver installation, a working system can easily break after receiving driver updates. To avoid this problem, always set the **Listen IP Address**.

Setting Up SPX Support

SPX is supported between Pervasive PSQL clients and servers. If you have modified the default settings or need to verify that SPX support is available, refer to this section.

Your network's SPX Frame Type setting does not have any effect on Pervasive PSQL. All computers communicating over SPX should be configured for the same SPX Frame Type. The Ethernet_802.2 frame type is the default and is recommended.



Note In order to perform any of the tasks in this section, you must be a member of the Pervasive_Admin group defined on the server.

➤ To Enable SPX Support

Complete the following steps to ensure that the database server engine can communicate with clients over SPX networks.



Note In an all-Microsoft environment, SPX can be used with applications that use only the Pervasive PSQL transactional interface. Applications that use only the transactional interface do not require name resolution with SPX.

- 1 On the **Start** menu select **Control Center** (PCC) from the **Pervasive PSQL v11 SP1** program group.
- 2 In the PCC Pervasive PSQL Explorer, double-click **Engines** to display a list of the engines registered with PCC.
- 3 Right-click the target engine then click **Properties**. Login if prompted.
- 4 Click **Communication Protocols**, and the list of **Supported protocols** displays. If SPX is checked, then SPX is already supported.
- 5 Click **SPX** then restart the database engine for the changes to take effect.



Tip Remember that you also need to confirm that your client computers are configured to use SPX, as well. See Chapter 10, [Configuring Network Communications for Clients](#).

Setting Up NetBIOS Support (Workgroup only)

By default, NetBIOS is supported among Pervasive PSQL Workgroup engines. If you have modified the default settings or need to verify that NetBIOS support is available, refer to this section.



Note In order to perform any of the tasks in this section, you must be seated at the console of the machine running the Workgroup engine. You cannot remotely configure the Workgroup engine.

➤ To Enable NetBIOS Support

Complete the following steps to ensure that the database engine can communicate with clients over NetBIOS networks.

- 1 On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the Pervasive PSQL Explorer, double-click **Engines** to display a list of the engines registered with PCC.
- 3 Right-click the target engine then click **Properties**. Login if prompted.
- 4 Click **Communication Protocols**, and the list of **Supported protocols** displays. If **NetBIOS** is checked, then NetBIOS is already supported.
- 5 Click **NetBIOS** then restart the database engine for the changes to take effect.



Tip Remember that you also need to confirm that the client software on your other Workgroup computers are configured to use NetBIOS, as well. Please refer to Chapter 10, [Configuring Network Communications for Clients](#).

Avoiding Unavailable Protocols

It may be possible to improve performance on the initial connection to the database by disabling database communications support for any protocols that are not available on your network or that you do not wish to use.

In order to perform any of the procedures in this section you must have one of the following:

- full administrator-level rights on the machine where the database engine is running
- membership in the Pervasive_Admin group defined on the machine where the database engine is running.



Note In order to perform any of the tasks in this section, you must be seated at the console of the machine running the Workgroup engine. You cannot remotely configure the Workgroup engine.

➤ To Remove Support for a Network Protocol



Note This procedure does not affect your operating system configuration in any way. This procedure only prevents the database communications system from attempting communications on unavailable or undesired protocols.

- 1 On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the PCC Pervasive PSQL Explorer, double-click **Engines** to display a list of the engines registered with PCC.
- 3 Right-click the target engine then click **Properties**. Login if prompted.
- 4 Click **Communication Protocols**, and the list of **Supported protocols** displays. Selected protocols are considered available for use by the engine.
- 5 Clear the check box for any of the selected protocols that are not supported on your network or that you do not wish to use.

You must leave at least one protocol selected.

- 6 Click **OK** then restart the database engine for the changes to take effect.



Tip Remember that you also need to confirm that your client computers are configured to use the protocol remaining in the Supported protocols list. Please refer to Chapter 10, [Configuring Network Communications for Clients](#).

Configuring Network Communications for Clients

10

How to Configure Network Communications for Your Pervasive PSQL Clients

To access network files from a machine using a Pervasive PSQL application, you must use the appropriate client requester at that machine. Your application's Pervasive PSQL calls go through the client requester, which sends them to the server for processing and then returns the reply to your application.

Generally, the default configuration settings for Pervasive PSQL Server and Client are sufficient. You typically do not have to configure any settings for the database engine and clients to communicate and function together correctly.

This chapter contains the following sections.

- [Client Network Communication Settings](#)
- [Network Path Formats Supported by Pervasive Requesters](#)
- [Using TCP/IP to Connect to a Windows Server](#)
- [Using SPX to Connect to a Windows Server](#)
- [Changing the Default Communication Ports](#)
- [Using TCP/IP to Connect a Windows Client to a Linux Server](#)
- [Data Encoding](#)
- [Using the DOS Requester](#)
- [DOS Box Support on Windows](#)

Client Network Communication Settings

This section lists the configuration settings used by the Pervasive PSQL Clients for network communication. These settings may be changed using a command line utility or from within PCC on the engine properties.

The *Advanced Operations Guide* provides detailed information about each of the settings. See the following configuration settings in *Advanced Operations Guide* for network communication:

- [Enable Auto Reconnect \(Windows only\)](#)
- [Supported Protocols](#)
- [Connection Timeout](#)

Network Path Formats Supported by Pervasive Requesters

When using your Requester, you connect to the Pervasive server engine to access data files. This section shows the variations on network file syntax you can use to access files on your network using Btrieve or SQL applications.

Pervasive PSQL supports the Universal Naming Convention (UNC) and Drive path formats (explicit and current) across the majority of operating environments.

For more information on the path formats, see the sections that follow:

- [Universal Naming Convention \(UNC\) Path Formats](#)
- [Drive-based Formats](#)
- [Linux Path Formats](#)

If you are an application developer, also note that the certain access methods, such as the Btrieve API, support URI connection strings. For details about URI strings, see [Database URIs](#) in *Pervasive PSQL Programmer's Guide*. In *Btrieve API Guide*, see [Create \(14\)](#), [Open \(0\)](#), and [Login/Logout \(78\)](#).

Universal Naming Convention (UNC) Path Formats

The following UNC path formats are supported on all clients to all servers:

```
\\ServerName or IP address\share\path\file
\\ServerName or IP address\share:[\]path\file
```

UNC syntax is resolved correctly regardless of the actual type of network operating system (NOS) running on the target server. If you use an IP address, it must be a dotted IPv4 address or one of the two formats supported for IPv6. See [IPv6 With UNC Paths and URI Connections](#).



Note In all instances above, backslashes (\) can be interchanged with forward slashes (/) except for the double backslash (\\). The syntax [\] indicates that the backslash is optional.

IPv6

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.

UNC paths do not allow certain special characters, such as colons, that are part of IPv6 addresses. The use of IPv6 with UNC paths is discussed below. First, however, a brief summary of IPv6 formats supported is useful to orient you to IPv6.

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 4 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. 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

Iipv6-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](#).

Restrictions

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

Table 5 IPv6 Restrictions With Pervasive PSQL

Restriction	Discussion
The Pervasive PSQL Server Engine in an IPv6-only environment	<p>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 may be used with IPv6 on Windows Vista, Windows 7, and Windows Server 2008.</p> <p>With Windows Server 2003 or Windows XP, status code 170 is returned when a PSQL Client attempts to connect to the PSQL Server. NetBIOS over TCP (NetBT) is not implemented for IPv6 on Windows Server 2003 or Windows XP.</p> <p>If you want to use PSQL Server on Windows Server 2003 or Windows XP, the network environment must be IPv4 or a combination of IPv4 and IPv6.</p> <p>This restriction applies only to the Server Engine. Pervasive PSQL Workgroup may be used with IPv6 on Windows Server 2003, Windows XP, Windows Vista, Windows 7, and Windows Server 2008.</p>
Square brackets are required for raw IPv6 addresses when the address is used in a URI or UNC	<p>Raw IPv6 addresses, abbreviated or not, must be enclosed by square brackets if the address is used in a URI or UNC.</p> <p>Examples:</p> <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 <p>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.</p>
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 btrv:// connection with an IPv6 address, you must escape the ZoneID for the host name.</p> <p>Example:</p> <p>A UNC-safe addresses like</p> <p>btrv://@[fe80::20c:29ff:fe67:2ee4%4]</p> <p>must be changed to</p> <p>btrv://@[fe80::20c:29ff:fe67:2ee4%254]</p>

Table 5 IPv6 Restrictions With Pervasive PSQL

Restriction	Discussion
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 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.

Drive-based Formats

The following drive representations are supported on all clients to all servers:

```
drive:file
drive:[\]path\file
file
[\\]path\file
..\file
```

Linux Path Formats

Incoming paths on a Linux server using Samba will be processed as follows in order of relative priority:

Share names

```
\\<server>\<sharename>\<path>
```

The smb.conf file must be configured to accept <sharename>, otherwise it will default to the following:

Absolute paths

```
\\<server>\<absolute_path>
```

If the smb.conf file is not configured properly or not found on the target server, the absolute path is used.

For more information on the Linux version of Pervasive PSQL v11 SP1, see [Using Pervasive PSQL on Linux](#).

Using TCP/IP to Connect to a Windows Server

This section documents the use of TCP/IP when connecting to a Pervasive PSQL server running on a Windows server platform.

Configuring a Client for the Server IP Address

When Pervasive PSQL operates in a TCP/IP network, your client must be able to obtain the IP address of your Windows server from the name given to that server. There are two mechanisms that enable this address-to-name translation:

- DNS (Domain Naming Service)
- Editing the Hosts file

The following procedures provide an overview for how to set up the IP address using each method. For details about network configuration and set up, refer to the documentation for the operating system.

Using DNS to Configure the Server IP Address

When you use DNS, you specify settings that allow your computer to look up the address of the server in a database of servers. Your network administrator can provide the information you need to configure DNS.

➤ To configure your Windows clients to use DNS to resolve the server IP address

- 1 From the operating system, access the **Network** information.
- 2 Select the **Properties** for **Local Area Connection**.
- 3 Select the **Properties** of the **TCP/IP** connection being used (IPv4 or IPv6, for example).
- 4 Enable DNS and enter the appropriate server information.

Using the Hosts File to Configure the Server IP Address

The Hosts file is a way to manually enter a relationship between a name and an IP address. Use this method if DNS is not used in your organization.

➤ **To Edit the Hosts file on your Windows client**

- 1 Find the Hosts file on your Windows machine.

For example, on certain Windows platforms it is located here:

```
%WINDIR%\SYSTEM32\DRIVERS\ETC\HOSTS
```

- 2 Edit the Hosts file with a text editor such as Notepad.
- 3 Enter your server's IP address and name in the hosts file as a new line as shown in the following example. Your network administrator can provide you with the IP address of your server.

```
# the following is an example of a Hosts file entry  
for IPv4 address
```

```
146.23.45.2 acctserver
```

Preventing the Windows Dial-Up Network Dialog Box from Displaying When Using a Pervasive Application with TCP/IP

Depending on the settings for your browser, the **Windows Dial-Up Networking** dialog box can display when a TCP/IP request is made. Usually, this is to make an Internet connection, but this feature can be an annoyance when using Pervasive applications and TCP/IP.

➤ **To Prevent the Dial-Up Networking Dialog Box from Displaying Automatically**

- 1 In Control Panel, double-click **Internet Options**.
- 2 Click the **Connections** tab.
- 3 Clear the **Dial whenever a network connection is not present** option (select one of the other options, such as **Never Dial a Connection**).

Using SPX to Connect to a Windows Server

This section documents the use of SPX when connecting to a Pervasive PSQL server running on a Windows machine.

Configuring Pervasive PSQL to use SPX

SPX is not a native protocol on the Windows platforms. If you want to use SPX, perform the following procedures to ensure proper operation with Pervasive PSQL.

Changing Pervasive's configuration to use SPX with a Windows platform

If you have both TCP/IP and SPX installed, you must remove TCP/IP from the Pervasive PSQL Client configuration to make SPX function with Pervasive applications

- 1 On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 In the Pervasive PSQL Explorer, expand **Local Client**.
- 3 Right-click **MicroKernel Router** and select **Properties**. Login if prompted.
- 4 Click **Communication protocols**. In the window to the right, a list of **Supported protocols** displays.
- 5 Clear **TCP/IP** from the list of selected protocols and click **OK**.

Changing Windows Configuration to Make SPX Run with Pervasive PSQL

➤ To Ensure that your SPX settings are correct

- 1 From the operating system, access the **Network** information.
- 2 Select the **Properties** for **Local Area Connection**.
- 3 Select the **Properties** of the **SPX** connection.
- 4 In the **Frame Type** field, ensure that the correct frame type for your network is selected. Do **not** use **Auto Detect**.
- 5 In the **Network number** field, enter a non-zero value for your network address. For information about what your network address should be, contact your system administrator.

Changing the Default Communication Ports

Pervasive PSQL communicates through three ports. Your firewall(s) and routers need to allow access to the following ports for remote access with the Server database engine:

- 3351 for the transactional interface
- 1583 for the relational interface
- 139 for named pipes (see note)

Typically, you do not need to modify the ports unless you have a conflict with them.



Note The Windows operating system uses port 139 for authentication to the operating system. An alternative to allowing access to port 139 through a firewall is to enable security on the Pervasive PSQL database. Once security is enabled, users such as “Master” are authenticated to the database through the database’s own security features. See [To turn on security using Pervasive PSQL Explorer](#) and [To create a new user using Pervasive PSQL Explorer](#), both in *Advanced Operations Guide*.

Port assignment 1583 is configurable for the server through the Pervasive PSQL utilities. This port is manually configurable for clients. See [TCP/IP Port](#) in *Advanced Operations Guide*.

It is recommended that port assignment 3351 **not** be changed. If you must change it, contact Pervasive Support.

Ensure that the port configurations match on both the server machine and all clients.

After changing your server listening port, you must stop and restart your Pervasive PSQL engine for the port assignment changes to take effect. See the chapter [Using Pervasive PSQL](#) in *Pervasive PSQL User's Guide*.

Services File

The services file is a text file used by the operating system for network communications. In the services files, you can manually assign the ports used by Pervasive PSQL Server and its clients. Be sure that the applicable ports listed in the services file are in agreement with the

ports set by Pervasive PSQL in the utilities and with the associated Windows Firewall rules.

After changing port assignments in the services file, you must stop then start the Pervasive PSQL database engine for the changes to take effect. See [Starting and Stopping the Database Engine](#) in *Pervasive PSQL User's Guide*.

Windows FireWalls

The installation of Pervasive PSQL Server and Pervasive PSQL Workstation performs certain actions pertaining to firewalls depending on the Windows operating system.

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). These operating system enable the firewall by default. The following table summarizes the Pervasive PSQL 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](#).

➤ **To enable Pervasive PSQL rules for the Public profile**

- 1 Open the console for Windows Firewall with Advanced Security.
- 2 Click **Inbound Rules** in the left pane.
- 3 Locate the desired Pervasive PSQL rule in the list in the center pane.

Note that the rules are listed twice. The enabled rules (indicated by a check mark on a green circle) apply to profiles other than Public. The disabled rules apply to the Public profile.
- 4 Right-click on the disabled rule you want then click **Properties**.
- 5 Click the **Advanced** tab. Ensure that the Public profile is selected. If not, select it.
- 6 Click the **General** tab, then click the **Enabled** option.
- 7 Click **OK**.
- 8 Exit the console for Windows Firewall with Advanced Security.

Profile Changes After Installation

If you change a network profile *after* installation of Pervasive PSQL, Pervasive PSQL may no longer be able to accept communications. For example, assume that only the Private network profile was active during installation. At some point after installation, the active profile is changed to Domain (assume its settings are very different from those of Private). The database engine will no longer be able to communicate across the network.

If you change profiles or firewall rules in a way that prevents Pervasive PSQL communications, refer to the steps in [To enable Pervasive PSQL rules for the Public profile](#). Use the steps as a general guideline for how to enable the Pervasive PSQL rules for the active profile(s). This will allow the database engine to communicate again across the network.

Notes About Policies

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, the installation of Pervasive PSQL Server and Pervasive PSQL Workstation adds each Pervasive PSQL service to the Windows Firewall Exception List. If the operating system security prompts you to unblock or allow communication with a Pervasive PSQL component, select OK (yes).

Installation of Pervasive PSQL 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.

If you encounter problems with your client/server applications not working correctly after installation, check the firewall access list or the ports. You may need to adjust some of the security settings to enable client/server applications. You can add Pervasive PSQL executable files to the access list or open ports. You do not need to do both. Note that opening a port opens it for all access, not just for Pervasive PSQL.

If you want to add Pervasive PSQL components to the firewall access list, add the following:

- For Pervasive PSQL 32-bit Server, ntdbsmgr.exe.
- For Pervasive PSQL Workgroup, w3dbsmgr.exe.
- For Pervasive PSQL 64-bit Server, ntdbsmgr.exe and ntdbsmgr64.exe.

Using TCP/IP to Connect a Windows Client to a Linux Server

Your Samba must be properly configured on your Linux server to properly network with Windows-based clients when using mapped drives.

Configuring a Client for the Server's IP Address

When Pervasive PSQL operates in a TCP/IP network, your client must be able to obtain the IP address of your Linux server from the name given to that server. There are two mechanisms that enable this address to name translation:

- DNS (Domain naming service)
- Editing the Hosts file

The following procedures provide an overview for how to set up the IP address using each method. For details about network configuration and set up, refer to the documentation for the operating system.

Using DNS to Configure the Server IP Address

When you use DNS, you specify settings that allow your computer to look up the address of the server in a database of servers. Your network administrator can provide the information you need to configure DNS.

- **To configure your clients to use DNS to resolve the server IP address**

For clients on Windows platforms:

- 1 From the operating system, access the **Network** information.
- 2 Select the **Properties** for **Local Area Connection**.
- 3 Select the **Properties** of the **TCP/IP** connection being used (IPv4 or IPv6, for example).
- 4 Enable DNS and enter the appropriate server information from your network administrator.

Using the Hosts File to Configure the Server IP Address

The Hosts file is a way to manually enter a relationship between a name and an IP address. Use this method if DNS is not used in your organization.

➤ To Edit the Hosts file on your Windows client

- 1** Find the Hosts file on your Windows machine.

For example, on certain Windows platforms it is located here:

```
%WINDIR%\SYSTEM32\DRIVERS\ETC\HOSTS
```

- 2** Edit the Hosts file with a text editor such as Notepad.
- 3** Enter your server's IP address and name in the hosts file as a new line as shown in the following example. Your network administrator can provide you with the IP address of your server.

```
# the following is an example of a Hosts file entry  
for IPv4 address
```

```
146.23.45.2  acctserver
```

Preventing the Windows Dial-Up Network Dialog Box from Displaying

The **Windows Dial-Up Networking** dialog box can display when a TCP/IP request is made to Windows. Usually, this is to make an Internet connection, but this feature can be an annoyance when using Pervasive applications and TCP/IP.

➤ To Prevent the Dial-Up Networking Dialog Box from Displaying Automatically

- 1** In Control Panel, double-click **Internet Options**.
- 2** Click the **Connections** tab.
- 3** Clear the **Dial whenever a network connection is not present** option (select one of the other options, such as **Never Dial a Connection**).

Data Encoding

An encoding is a standard for representing character sets. Character data must be put in a standard format, that is, encoded, so that a computer can process it digitally. An encoding must be established between the Pervasive PSQL database engine (server) and a Pervasive PSQL client application. A compatible encoding allows the server and client to interpret data correctly.

Pervasive PSQL v11 SP1 better handles the complexity of the encoding between client and server and the various combinations of operating system, languages, and access method. The encoding enhancements are divided into database code page and client encoding. The two types of encoding are separate but interrelated (see Table 7).

The use of the two encoding methods is intended for advanced users. In general, the default encoding settings are sufficient and do not require changing.

Database code page and client encoding apply only to the relational interface. The transactional interface is not affected.

This section contains the following topics:

- [Database Code Page](#)
- [Client Encoding](#)
- [Encoding Interaction](#)
- [Legacy Conversion Methods for OEM Data](#)

Database Code Page

Database code page is specified with a new property called **database code page**, which identifies the encoding to use for data and metadata. The default database code page is “server default,” meaning the operating system (OS) code page on the server where the database engine is running. (The OS code page is generally referred to as the “OS encoding,” which is the phrase used throughout the rest of this chapter.)

Database code page is particularly handy if you need to manually copy Pervasive PSQL DDFs to another platform with a different OS encoding and still have the metadata correctly interpreted by the database engine.

Client Encoding

Client encoding is the data encoding used by an application on a Pervasive PSQL client. An application can store data in any encoding it chooses. But, as mentioned earlier, a compatible encoding must be established between the database engine and the client application. Previous versions of Pervasive PSQL provided methods to ensure compatible encoding between the database engine and clients.

Those methods have been enhanced to take advantage of database code page. An application can now specify that it wants the Pervasive PSQL client to translate data automatically between the database code page and the client application. This is referred to as *automatic translation*. Note, however, that automatic translation can translate characters only if they are present in both code pages (the code page on the server machine and the code page on the client machine).

Automatic translation is specified when the client application connects to the database engine. See [ODBC Connection Strings](#) in *SQL Engine Reference*.

Data translation, if required, occurs at the client. (Translation is not always required—for example, when the client operating system (OS) encoding matches the server OS encoding.)

Encoding Interaction

The following table explains the interaction between database code page and client encoding.

Table 7 Interaction Between Database Encoding and Client Encoding

If Database Encoding Is	And the Client Application Specifies	The Pervasive PSQL Client
Server Default	Automatic Translation	Translates data and metadata from the default operating system (OS) encoding on the server to the OS encoding on the client.
A specific code page	Automatic Translation	Translates data and metadata from the database code page to the OS encoding on the client.

Table 7 Interaction Between Database Encoding and Client Encoding

If Database Encoding Is	And the Client Application Specifies	The Pervasive PSQL Client
Server Default <i>or</i> A specific code page	Nothing (no encoding specified) (No encoding specified is the default behavior in versions prior to Pervasive PSQL v10.10.)	Sends data to the database engine in the encoding of the client machine and ignores database code page. For compatible data interpretation, the encoding used by the client machine must match the encoding of the data and metadata in the database.
Server Default <i>or</i> A specific code page	A specific encoding	Sends data to the server in the encoding specified by the client application and ignores database code page. For compatible data interpretation, the encoding specified by the client application must match the encoding of the data and metadata in the database.

When a database has OEM character data in it, the legacy solution was for the access method, such as ODBC using a DSN, to specify OEM/ANSI conversion. Now it is possible to set the OEM code page for the database and have the access method specify automatic translation. See also [Encoding Translation](#) in *SQL Engine Reference*.



Note The database engine does **not** validate the encoding of the data and metadata that an application inserts into a database. The engine assumes that all data was entered using the database code page as explained in Table 7.

Legacy Conversion Methods for OEM Data

If a database has OEM character data in it, a legacy solution is to specify OEM/ANSI conversion in the access method. This topic discusses some legacy methods for Linux clients using OEM character data.



Note While the legacy methods are still supported, the recommendation is to specify the OEM code page for the database and have the access methods use automatic translation as discussed above.

Btrieve and DTI

When using the Btrieve API or the Distributed Tuning Interface (DTI), you must provide file names and paths in the local encoding used in your application. The Btrieve API and DTI handle the differences between OS encoding on the server and client.

ODBC

See also [OEM/ANSI Conversion](#) in *SQL Engine Reference*.

When using ODBC, Win32 encoding is expected to be SHIFT-JIS.

Japanese versions of Linux by default have their encodings typically set to EUC-JP or UTF-8.

When using Japanese versions of Linux, a client can connect to another Linux server (for example, locally), or to a Win32 SHIFT-JIS server. It is also possible to connect to a database encoded in SHIFT-JIS but located on a Linux server.

Use the following instructions for your listed configuration. In each case, it is assumed that the application itself does not do any conversion and uses the encoding that is native for the machine.

- [Connecting a Linux EUC-JP Client to a Win32 SHIFT-JIS Server](#)
- [Connecting a Linux UTF-8 Client to a Win32 SHIFT-JIS Server](#)
- [Connecting a Linux EUC-JP Client to a Linux EUC-JP Server](#)
- [Connecting a Linux UTF-8 Client to a Linux UTF-8 Server](#)
- [Connecting a Linux UTF-8 Client to a Linux EUC-JP Server](#)
- [Connecting a Linux EUC-JP Client to a Linux EUC-JP Server, with SHIFT-JIS Encoding Used to Store Data on the Server](#)

Connecting a Linux EUC-JP Client to a Win32 SHIFT-JIS Server

The server requires that everything is received as SHIFT-JIS. The client requires that the server send everything as EUC-JP.

To accomplish this, the client DSN settings in ODBC.INI (located by default in `/usr/local/psql/etc`) used to connect to the given database should be set up as follows:

```
[dbclient]
Driver=/usr/local/psql/lib/libodbccci.so
Description=Pervasive ODBC Client Interface: JPN-
2000SERVER:1583/dbclient
ServerDSN=DEMODATA
```

```
ServerName=JPN-2000SERVER:1583  
TranslationDLL=/usr/local/psql/lib/libxlate.so.10  
TranslationOption=90000932
```

The `TranslationDLL` line specifies the translation library that the ODBC client interface should use.

The `TranslationOption` line specifies that the translation needs to occur from 9000 (representing EUC-JP) to 0932 (representing SHIFT-JIS).

Using this example, all data coming from the client will be translated into SHIFT-JIS before it gets to the server, and to EUC-JP before the data is received by the client.

Connecting a Linux UTF-8 Client to a Win32 SHIFT-JIS Server

The server requires that everything is received as SHIFT-JIS. The client requires that the server send everything as UTF-8.

To accomplish this, the client DSN settings in ODBC.INI (by default in `/usr/local/psql/etc`) used to connect to the given database should be set up as follows:

```
[dbclient]  
Driver=/usr/local/psql/lib/libodbccli.so  
Description=Pervasive ODBC Client Interface: JPN-  
    2000SERVER:1583/dbclient  
ServerDSN=DEMODATA  
ServerName=JPN-2000SERVER:1583  
TranslationDLL=/usr/local/psql/lib/libxlate.so.10  
TranslationOption=90010932
```

The `TranslationDLL` line specifies the translation library that the ODBC client interface should use.

The `TranslationOption` line specifies that the translation needs to occur from 9001 (representing UTF-8) to 0932 (representing SHIFT-JIS).

Using this example, all data coming from the client will be translated into SHIFT-JIS before it gets to the server, and to UTF-8 before the data is received by the client.

Connecting a Linux EUC-JP Client to a Linux EUC-JP Server

Using this configuration, no changes to the DSN description are needed. Use the DSN as it was created by the `dsnadd` utility.

Connecting a Linux UTF-8 Client to a Linux UTF-8 Server

Using this configuration, no changes to the DSN description are needed. Use the DSN as it was created by the `dsnadd` utility. See `dsnadd` in *Pervasive PSQL User's Guide*.

Connecting a Linux UTF-8 Client to a Linux EUC-JP Server

The server requires that everything is received as EUC-JP. The client requires that server send everything as UTF-8.

To accomplish this, the client DSN settings in ODBC.INI (by default in `/usr/local/psql/etc`) used to connect to the given database should be set up as follows:

```
[dbclient]
Driver=/usr/local/psql/lib/libodbccli.so
Description=Pervasive ODBC Client Interface: JPN-
  2000SERVER:1583/dbclient
ServerDSN=DEMODATA
ServerName=JPN-2000SERVER:1583
TranslationDLL=/usr/local/psql/lib/libxlate.so.10
TranslationOption=90019000
```

The `TranslationDLL` line specifies the translation library that the ODBC client interface should use.

The `TranslationOption` line specifies that the translation needs to occur from 9001 (representing UTF-8) to 9000 (representing EUC-JP).

Using this example, all data coming from the client will be translated into EUC-JP before it gets to the server, and to UTF-8 before the data is received by the client.

Connecting a Linux EUC-JP Client to a Linux EUC-JP Server, with SHIFT-JIS Encoding Used to Store Data on the Server

This situation is possible if you have a SHIFT-JIS database on a Win32 engine, and you want to move all the files to the Linux EUC-JP server. In this case, the database resides on a EUC-JP Linux machine, but all the data inside the DDF files and data files are in SHIFT-JIS.

In this case, your DSN should be set up as follows:

```
[dbclient]
Driver=/usr/local/psql/lib/libodbccli.so
Description=Pervasive ODBC Client Interface: JPN-
  2000SERVER:1583/dbclient
```

Configuring Network Communications for Clients

```
ServerDSN=DEMADATA  
ServerName=JPN-2000SERVER:1583  
TranslationDLL=/usr/local/psql/lib/libxlate.so.10  
TranslationOption=90000932  
CodePageConvert=932
```

The last line specifies that even though the server uses EUC-JP encoding, it should treat the data on the server as SHIFT-JIS.

Using the DOS Requester

Pervasive PSQL v11 SP1 supports DOS Btrieve applications with the DOS Requester. The DOS requester supports Btrieve applications only, not ODBC applications. This section explains how to use the DOS requester to run Pervasive PSQL-based DOS applications in Windows.

DOS Box support allows a DOS application to run in a DOS box on a Windows platform. This enables direct communication to the Windows 32-bit workstation components rather than to the database engine. This configuration can be used with either a local Pervasive PSQL Workgroup engine, or a remote engine. The TCP/IP, SPX, or NetBIOS protocol supported for client/server access depends on the configuration of the Windows 32-bit components.

Supported Configurations

The DOS requester supports both Workgroup and Client to remote Server engine configurations.

DOS Box Support on Windows

The Requester for Windows is BTRBOX. You can use this Requester for DOS applications.

Running DOS Applications on Windows 32-bit Platforms

All of the components needed to run DOS applications using BTRBOX are installed with your client. After the Windows client component installation, you have everything you need to run a DOS or Windows 32-bit application. The default DOS application support installed is the Win32 DOS Box configuration.

DOS applications are not supported on 64-bit Windows platforms. Therefore, BTRBOX is not supported on 64-bit Windows platforms.

Using DOS Box Support

On Windows platforms, the DOS Box install configures the drivers to be completely transparent. Thus, you are able to immediately open a command prompt and run a DOS Btrieve application. The CONFIG.NT file, located in the %WINDIR%\SYSTEM32 directory, contains the command that enables DOS application support. This file is similar to CONFIG.SYS in DOS. The Windows operating system loads the driver for each DOS session opened. In the configuration file, the install places the following path to load the DOS Box driver:

```
DEVICE = C:\WINDOWS\SYSTEM32\BTRDRVR.SYS
```

Application Configuration Scenarios

chapter

11

Common Scenarios for Setting up Your Database Engine

This chapter explains the engine configuration settings necessary for some common environment scenarios. The topics discussed in this chapter include the following:

- [Terminal Services](#)
- [Active Directory Service](#)
- [Multiple Client Applications](#)
- [Concurrent Local and Remote Applications](#)
- [Accessing Data on Other Computers](#)

Terminal Services

Microsoft Terminal Services is a multi session environment that provides remote computers access to Windows-based programs running on a server. Citrix MetaFrame extends Windows Terminal Services with additional client and server functionality.

Disabling Administrative Functions

In prior releases, the ability to perform administrative functions was prohibited from the client. In Pervasive PSQL v11 SP1, Pervasive PSQL clients running within Terminal Services client sessions can perform Pervasive PSQL administrative functions by default. For example, a user with such a client can change configuration settings for Pervasive PSQL, create DSNs, and use the Monitor utility.

If you want to restrict this capability, intervention is necessary from a system administrator.

► **To disable remote administrative functions for Terminal Services clients**

- 1 From PCC, open the properties for the **MicroKernel Router** under **Local Client**.

See [To access configuration settings in PCC for a local client](#) in Advanced Operations Guide.

- 2 On the **Properties** dialog, check **Restrict Administrative Functions from a WTS Client**.
- 3 Click **OK**, then exit PCC and start it again for the setting to take effect.



Note Pervasive PSQL Server engines are supported for use with Microsoft Terminal Server and Citrix MetaFrame running within an Active Directory environment.

Terminal Server as Network Server

You may use your terminal server as your main network server and database server. However, if you have high usage of the server as a file server as well as many terminal sessions running at the same time, you may find the performance less than satisfactory.

Another concern is having all of your mission critical services on the same machine. If it goes down, all of your services go down at once.

For these reasons, you may wish to consider distributing your mission critical services on two or more computers.

Workgroup Engine Running as a Service

You may configure your server to run the Workgroup engine as a service. This allows the engine to start automatically when the operating system starts. A user is not required to log in to start the engine. Refer to [Running the Workgroup Engine as a Service](#).



Caution Running the engine as a service requires the **Log On as Service** privilege. If you select to run the engine as a service under a user account other than the default Local System account, you will need to modify the Log On Properties for the Service using the Windows Control Panel.

Active Directory Service

Active Directory is a central component of the network architecture on certain Windows operating systems. Active Directory provides a directory service specifically designed for distributed networking environments.

This section describes how to configure Pervasive PSQL in an environment that has Microsoft Active Directory service installed and functioning correctly.

Ensure that Active Directory service is installed and functioning correctly before you install Pervasive PSQL into the environment.

Server and Client Support

Pervasive PSQL Server runs on supported Windows Servers within an Active Directory environment. The Pervasive PSQL clients run on all supported Windows platforms within an Active Directory environment.

Directory and File Permissions

The database engines enforce directory and file permissions set at the operating system level. An Active Directory environment does not change this behavior. For example, if you set “read only” permission on a Pervasive PSQL table file, you will be unable to write to the table.

Microsoft Terminal Services Support

Pervasive PSQL Server engines are supported for use with Microsoft Terminal Server and Citrix MetaFrame running within an Active Directory environment. For more information about Terminal Services and Citrix MetaFrame, see [Terminal Services](#).

Pervasive Administrative Authority

Active Directory service manages the security of the network. You must grant the correct access authority at the operating system level to users who need Pervasive administrative privileges.

See [Active Directory Tasks](#) for the steps to set access authority. Users must have the following authority on the machine running the database engine:

- Log on locally
- Administrator privileges or belong to the Pervasive_Admin group

You may grant the Log on locally authority directly to a user or to the Pervasive_Admin group (and add the user to the group).

You may create the Pervasive_Admin group on the machine running the database engine (the local machine), on the domain controller for the local machine, or on both. The database engine checks privileges first on the domain controller for the local machine then on the local machine.

An example helps illustrate this. Suppose you have two servers in your domain that run the Pervasive PSQL database engine, Server A and Server B. You could create a Pervasive_Admin group on each server and on the domain controller. You then add User 1 to the group on Server A, User 2 to the group on Server B, and User 3 to the group on the domain controller. User 1 has administrative privileges for the database engine only on Server A. Similarly, User 2 has administrative privileges only on Server B. User 3, however, has administrative privileges for the database engines on both Server A and Server B.

If you create the Pervasive_Admin group on a domain controller, then the group must be a domain local group. If you create the Pervasive_Admin group on a machine that is not a domain controller, then the Pervasive_Admin group must be a local group.

Active Directory Tasks

This section explains the tasks needed to ensure users have Pervasive administrative privileges. The tasks assume the following:

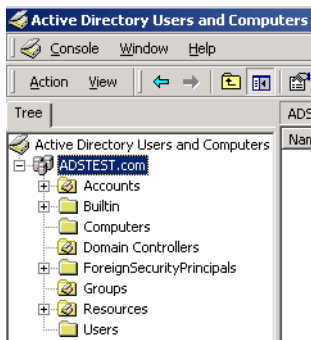
- Network user IDs have been added for users who need Pervasive administrative privileges
- A Pervasive_Admin group has been created on the domain controller and users added to the group
- Windows Server 2003 is the operating system on the domain controller.

➤ To Create the Pervasive_Admin Group on a Domain Controller

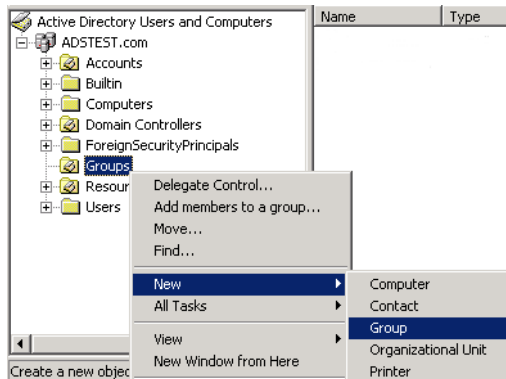
- 1 Click Start ▶ Programs ▶ Administrative Tool ▶ Active Directory Users and Computers.

- 2 Expand the tree for the domain to which you want to add the Pervasive_Admin group.

For example, the following image shows the expanded tree for the ADSTEST.com domain.



- 3 Right-click the Organizational Unit or folder that you are using in your Active Directory environment to house groups, then click **New** ► **Group**. For example, the following image shows an Organizational Unit named “Groups,” but your Organizational Unit may be named differently.



Note If your Active Directory environment does not have an Organizational Unit to house groups, you need to create one. Click the domain root (for example, in the figure above, you would right-click ADSTEST.com), then click **Action** ► **New** ► **Organizational Unit**. Type a meaningful name for the unit, then click **OK**.

- 4 For **Group name**, type **Pervasive_Admin**. Click **Domain local** for group scope.

Group name:
Pervasive_Admin

Group name (pre-Windows 2000):
Pervasive_Admin

Group scope:
 Domain local
 Global
 Universal

Group type:
 Security
 Distribution

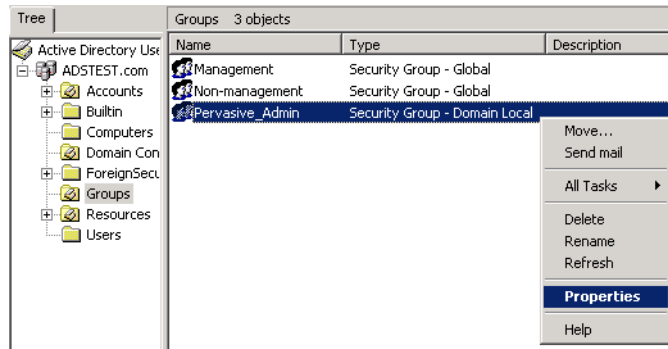


Note The **Pervasive_Admin** group must have a scope of **Domain local**. Do not use **Global** or **Universal**.

- 5 Click **OK**.

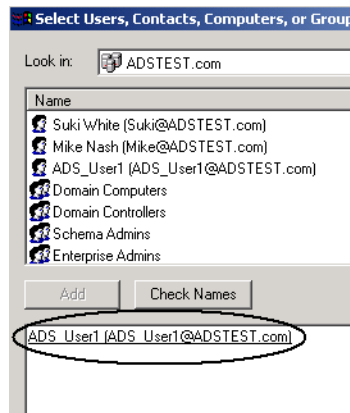
Now that the **Pervasive_Admin** group exists, you need to add users to it.

- 6 On the Active Directory Users and Computers window, right-click the **Pervasive_Admin** group, then click **Properties**. (You may also double-click the group.)



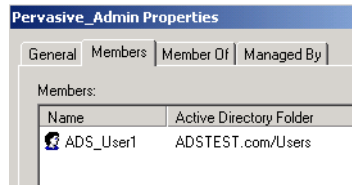
- 7 Click the **Members** tab on the **Properties** dialog box.
- 8 Click **Add** on the **Members** tab.
- 9 Click the user in the **Name** list that you want to add to the **Pervasive_Admin** group, then click **Add**.

The user is added to the list on the bottom. For example, the following image shows that user ADS_USER1 has been added.



10 Click OK.

The user you added now appears as a member of the Pervasive_Admin group.

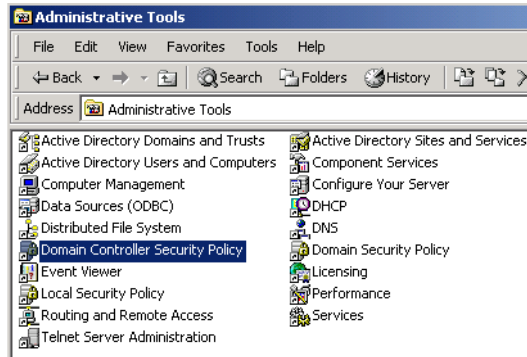


11 Click OK to exit the properties dialog box.

12 Add the Pervasive_Admin group to the Log on locally privileges (complete the task [To Grant Log On Locally Privileges to the Pervasive_Admin Group](#)).

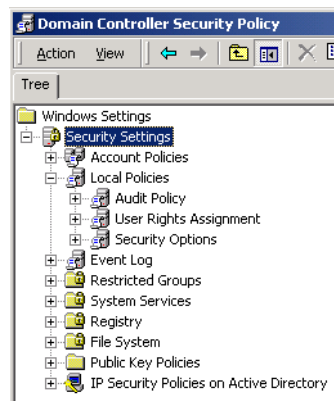
➤ **To Grant Log On Locally Privileges to the Pervasive_Admin Group**

1 In the Windows Control Panel double-click Administrative Tools, then double-click Domain Controller Security Policy.

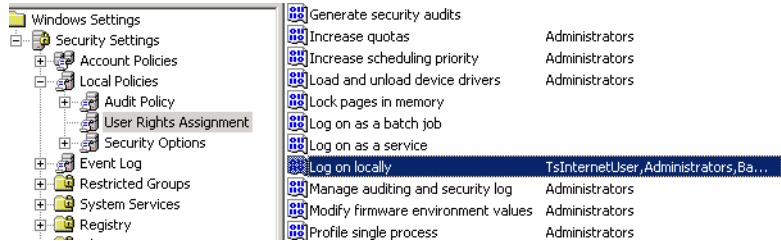


Note Ensure that you open Domain *Controller* Security Policy and not Domain Security Policy.

- 2 Expand the following security settings:
 - Security Settings
 - Local Policies

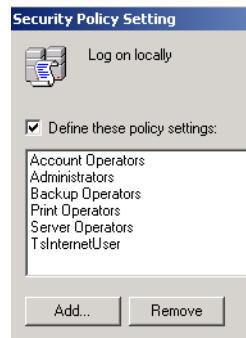


- 3 Click **User Rights Assignment**.
- 4 Scroll the policies in the right pane until you locate **Log on locally**.



- 5 Double-click the **Log on locally** policy (or right-click the policy then click **Security**).

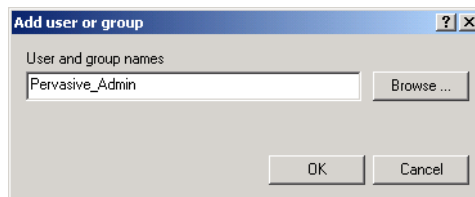
The **Security Policy Setting** dialog box appears.



- 6 Click **Add**.

The **Add user or group** dialog box displays.

- 7 Type **Pervasive_Admin** in the **Users and group names** field.



You may also specify the group by clicking **Browse** and navigating to the group through dialogs.

- 8 Click **OK**.

The **Security Policy Setting** dialog appears with **Pervasive_Admin** added.

- 9** Click **OK** to exit the Security Policy Setting dialog.
- 10** Exit the Domain Controller Security Policy window.

Multiple Client Applications

Sometimes, two or more client/server applications may use the same database engine. You will need to configure the database engine differently depending on whether the applications are used at the same time.

If your vendors supply configuration guidelines for engine configuration parameters, you will need to adjust your configuration based on these guidelines.

If the applications run concurrently (that is, if two or more applications are using the database server at the same time) ...

You should configure the engine by adding together all the recommended values for each parameter. For example, if one application vendor suggests **Performance Tuning | Number of Input/Output Threads** should be set to 4, and another application vendor suggests this parameter should be set to 8, then you should set it to 12.

If the default value is higher than the sum of the recommended settings, then do not change the default value.

Do not add up the recommended values for any buffer size settings, or log file size settings. Use the largest recommended setting. Again, do not change the default if it is larger than any vendor recommendation.

If the applications do not run concurrently (that is, if only one application is running at any given point in time) ...

You should configure the server by using the largest recommended value for each parameter. For example, if one application vendor suggests **Performance Tuning | Number of Input/Output Threads** should be set to 4, and another application vendor suggests this parameter should be set to 8, then you should set it to 8.

If the default value is higher than the largest recommended setting, then do not change the default value.

Settings Affected by Multiple Applications

Most engine settings are not affected when you are running multiple applications. This section explains the settings that may need to be adjusted for multiple applications.

Compatibility | Create File Version

Some applications may require that new files be created with version 7.x file format, while other applications may require version 9.x file format (the default).

These applications can run concurrently only if new files are not created during runtime. There is no way to toggle the setting back and forth for each application, unless you wish to do it by hand or write a program to do so using the Distributed Tuning Objects.

If the applications do not create new files during runtime, then this setting is not relevant for multiple applications.

Data Integrity | Transaction Durability

Some applications may require durable transactions, while others may not. If you have two application vendors recommending different values for this parameter, then you should set it to **On**. Generally, having transaction durability turned on does not affect applications that do not use transactions, but may slow performance.

Concurrent Local and Remote Applications

The Server engine allows both remote client requests as well as communications from applications running on the same computer as the server.



Note To perform these steps, you must have full administrator-level rights on the machine where the database engine is running, or be a member of the Pervasive_Admin group defined on the machine where the database engine is running.

➤ **To configure database connections from both remote and local applications**



Tip When changing the Server engine settings, you must be at the Windows server computer where the database server runs.

- 1 Click **Control Center** from the **Pervasive** program on the **Start** menu.
- 2 In the Pervasive PSQL Explorer, expand **Engines** to display the engines registered with **Pervasive PSQL Control Center**.
- 3 Right-click the target engine and click **Properties**. Login if prompted.
- 4 Click **Access**. In the right-hand pane, select the **Accept Remote Requests** check box.

If you wish to prevent the server from accepting client connections from other computers, clear the check box.

- 5 Click **OK**.
This configures the server to accept remote requests.
- 6 In the Pervasive PSQL Explorer, expand **Local Client**.
- 7 Right-click **MicroKernel Router** and click **Properties**. Login if prompted.
- 8 Click **Access**. In the right-hand pane, select the following check boxes:

- **Use Local MicroKernel Engine.** Select this check box to configure the local engine for local file access.
- **Use Remote MicroKernel Engine.** Select this check box to access databases on other computers.

If you plan to only access data on this computer, clear this check box.

9 Click OK.

This configures the server to accept local requests.

10 Restart the server engine to implement the changes.

**Using the
Server and
Workgroup
Engines
Concurrently**

The Workgroup engine can be configured to access files on a remote file server through a mapped drive on a Windows server.

The client software installed with your Workgroup engine can be used to connect to other server engines on a remote machine.

If you want to use your local engine for local file access and a remote server for access to files being serviced by the remote Pervasive server, you must change the settings in your MicroKernel Router. Use the Pervasive PSQL Control Center to change MicroKernel Router settings.

➤ **To configure local and remote access for the MicroKernel Router**

- 1** On the **Start** menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2** In the Pervasive PSQL Explorer window, expand **Local Client**.
- 3** Right-click **MicroKernel Router** and click **Properties**. Login if prompted.
- 4** Click **Access**. In the right-hand pane, select the following check boxes:
 - **Use Local MicroKernel Engine.** Select this check box to configure the local engine for local file access.
 - **Use Remote MicroKernel Engine.** Select this check box to configure the remote server for access to files being serviced by the remote Pervasive server.
- 5** Click **OK**.



Note See *Advanced Operations Guide* for more information on changing settings using the Pervasive PSQL Control Center.

Accessing Data on Other Computers

The Workgroup engine provides great flexibility for a variety of small networked environments. The table below explains the most common configurations and where to look for more information. In any of the configurations below, a Workgroup engine must be installed on every computer that is expected to access data.

Table 8 Summary of Network Configurations

Configuration	Where to look for more information
Small client/server: Data resides on a single computer where a Workgroup engine is installed.	Setting Up a Small Client/Server Configuration
Peer-to-Peer: Data resides on two or more computers where Workgroup engines are installed.	Setting Up a Peer-to-Peer Configuration
Gateway: Data resides on a file server where no database engine is installed, or it is not running.	Setting Up a Gateway Configuration

Installing Pervasive PSQL for Linux

Instructions for Installing and Uninstalling Pervasive PSQL on Linux

This chapter explains how to install and uninstall Pervasive PSQL:

- [Before You Install Pervasive PSQL for Linux](#)
- [Installing Pervasive PSQL Using RPM](#)
- [Installing Pervasive PSQL Using TAR](#)
- [After Installing Pervasive PSQL for Linux](#)
- [Uninstalling Pervasive PSQL for Linux](#)

Before You Install Pervasive PSQL for Linux

Before installing or upgrading any of the Pervasive PSQL products, review the following information:

- System requirements listed on the Pervasive Software Web site for Pervasive PSQL v11 SP1.
- *Chapter 2, [Preparing to Install Pervasive PSQL](#)*– This chapter provides important information including platform specific notes.
- *Release Notes* – The release notes are located in `readme.htm` on the distribution media and contain late-breaking news that may not be included in the user documentation.

You must be logged in as root to install any of the products. If you are installing from the CD, you must be at the CD root directory.

If you have any trouble with an installation, see [Troubleshooting After Installation](#).

Server

If you are planning to access the Pervasive PSQL transactional interface across a network from a Windows-based client using drive mappings, we recommend that the Samba package be installed on the server. Please refer to the Samba website, <http://www.samba.org>, for installation and configuration instructions.

After installing Pervasive PSQL Server, review [Supported Path Formats for Samba](#) for information regarding Samba's path configuration.

Client

The Linux client can be installed on a Linux machine with no Pervasive PSQL products currently installed, or on a Linux machine with a Pervasive PSQL v11 SP1 Server engine installed.

If your database server engine does not match certain installation requirements, your applications may receive the following status code: “status **3031: Linux requester cannot connect to this server.**” This status code indicates client/server incompatibility. In some cases, you may receive a permissions error status instead: “**94: The application encountered a permission error.**”

The installation scripts perform the following tasks:

- Verify necessary permissions to complete install

- Create user `psql` and group `pvsu` (if they do not exist)
- Sets user:group ownership to `psql:pvsu` for the installed files (if not already set)

Full Installations

Pervasive PSQL offers full installations of both the RPM and TAR Linux packages. A full installation includes the necessary engine and client files, utilities, and the complete user documentation. A full installation does not include the word “full” in the package name.

The following table outlines the installation packages.

Table 9 Full and Client Linux Installations

Pervasive PSQL Product	Installation Type	Contains
Server 64-bit	Full	Engine and client files, utilities, and documentation
Server 32-bit	Full	Engine and client files, utilities, and documentation
Client 64-bit	Client	Client files ¹
Client 32-bit	Client	Client files, utilities, and documentation
¹ Because of the minimal files included in the 64-bit client, the installation package name includes the word “core.”		

Notes for Specific Distributions

Note that, by default, the 64-bit edition of Red Hat Enterprise Linux 6 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*.

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 <code>glibc.i686</code> and <code>libstdc++.i686</code>.</p> <hr/> <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"> • <code>gtk2.i686</code> • <code>PackageKit-gtk-module.i686</code> • <code>libcanberra-gtk2.i686</code> • <code>gtk2-engines.i686</code>. <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 PSQLDAPATA Samba share (the <code>smb.conf</code> file exists, but nothing is there to support it). The errors are benign and Pervasive PSQL functions normally.</p>

Installing Pervasive PSQL Using RPM

The RPM format allows you to install Pervasive PSQL if your Linux distribution contains the Red Hat Package Manager (RPM). Version 4 or greater of RPM is required.

This section explains how to install the following Pervasive PSQL products using RPM:

- [Installing Pervasive PSQL Server for Linux - RPM](#)
- [Installing the Pervasive PSQL Client for Linux - RPM](#)

If you have any trouble with installation, see the chapter [Troubleshooting After Installation](#).

Installing Pervasive PSQL Server for Linux - RPM

Determine the package name to use for the installation using the following table and the distribution media.

Table 10 Linux Server Package Names - RPM

Installation Type	Package Name
Server 64-bit	Pervasive.SQL-yy.yy-zzz.zzz.x86_64.rpm
Server 32-bit	Pervasive.SQL-yy.yy-zzz.zzz.i486.rpm

In all cases, the *yy.yy* designates a release number and *zzz.zzz* designates a build number. Refer to the distribution media for the actual name of the package.

Installation is categorized as either first time, in which no previous version exists, or upgrade, in which a previous version exists.

- [First Time Installation](#)
- [Upgrade Installation](#)

First Time Installation

The package manager copies all necessary files onto disk (default location is `/usr/local/psql`) and runs a post-installation script which performs the following tasks:

- Creates user `psql` and group `pvs`
- Sets `user:group` ownership to `psql:pvs` for the installed files

- Applies a trial license
- Creates a new ODBC DSN (data source name) for the DEMODATA test database
- If Samba configuration file is found
 - Creates a new Samba share PSQLDATA
 - Creates a new Samba share PVPIPE\$
- Creates startup/shutdown scripts for Pervasive PSQL daemons
- Launches the Pervasive PSQL daemon (`mkded`)

➤ **To install Pervasive PSQL Server using RPM**



Note If you have a previous version of Pervasive PSQL on your Linux machine, see [Upgrade Installation](#).

- 1 Log in as the root user.
- 2 Assuming the RPM package is in the current directory, enter the following command:

```
rpm -ivh <Linux_Server_Package_Name>
```



Note Refer to [Linux Server Package Names - RPM](#) for the package name to use. You must include the appropriate release and build number information to perform the installation. Verify the complete package name from the distribution media.

If the RPM package is in another directory, preface the package name with a path.

Upgrade Installation

If you have a previous version of Pervasive PSQL already installed, you must uninstall that product and then install the Pervasive PSQL v11 SP1 product.

See [Uninstalling Pervasive PSQL for Linux](#) for information on uninstalling Pervasive PSQL.

Installing the Pervasive PSQL Client for Linux - RPM

The name of the Pervasive PSQL Client installation package conforms to the following conventions:

Table 11 Linux Client Package Names - RPM

Installation Type	Package Name
Client 64-bit	Pervasive.SQL-Client-Core-yy.yy-zzz.zzz.x86_64.rpm
Client 32-bit	Pervasive.SQL-Client-yy.yy-zzz.zzz.i486.rpm

In all cases, the *yy.yy* designates a release number and *zzz.zzz* designates a build number. Refer to the distribution media for the actual name of the package.

Installation is categorized as either first time, in which no previous version of the product exists, or upgrade, in which a previous version of the product exists.

- [First Time Installation](#)
- [Upgrade Installation](#)



Note The Client 64-bit installation does not include certain utilities, documentation, or SDK components. To install them, you need to install both the Client 64-bit and Client 32-bit products. Because this is the same as for the Pervasive PSQL Client for Windows, see [Installing the Pervasive PSQL Client for Windows](#) for details.

First Time Installation

► To install Pervasive PSQL Client Using RPM

- 1 Log in as the root user.
- 2 Assuming the RPM package is in the current directory, execute the following command.

```
rpm -ivh <Linux_Client_Package_Name>
```



Note Refer to [Linux Client Package Names - RPM](#) for the package name to use. You must include the appropriate release and build number information to perform the installation. Verify the complete package name from the distribution media.

If the RPM package is in another directory, preface the package name with a path.

Upgrade Installation

If you have a previous version of Pervasive PSQL already installed, you must uninstall that product and then install the Pervasive PSQL v11 SP1 product.

See [Uninstalling Pervasive PSQL for Linux](#) for information on uninstalling Pervasive PSQL.

Installing Pervasive PSQL Using TAR

The tape archive (TAR) format allows you to install Pervasive PSQL if you have a Linux distribution that does not support the RPM format or if you prefer not to use RPM.

This section explains how to install the following Pervasive PSQL products using TAR:

- [Installing Pervasive PSQL Server for Linux - TAR](#)
- [Installing Pervasive PSQL Client for Linux - TAR](#)

If you have any trouble with installation, see the chapter [Troubleshooting After Installation](#).

Installing Pervasive PSQL Server for Linux - TAR

The name of the Pervasive PSQL Server installation package conforms to the following conventions.

Table 12 Linux Server Package Names - TAR

Installation Type	Package Name
Server 64-bit	Pervasive.SQL-yy.yy-zzz.zzz.x86_64.tar.gz
Server 32-bit	Pervasive.SQL-yy.yy-zzz.zzz.i486.tar.gz

In all cases, the *yy.yy* designates a release number and *zzz.zzz* designates a build number. Refer to the distribution media for the actual name of the package.

Installation is categorized as either first time, in which no previous version exists, or upgrade, in which a previous version exists.

- [First Time Installation](#)
- [Upgrade Installation](#)

First Time Installation

➤ To install Pervasive PSQL Server using TAR

- 1 Log in as the root user.
- 2 Change to the `/usr/local` directory.

```
cd /usr/local
```

- 3 Enter the following command to copy the tar into `/usr/local`.

```
cp <path_to_tar>/<Linux_Server_Package_Name> .
```



Note Refer to [Linux Server Package Names - TAR](#) for the package name to use. You must include the appropriate release and build number information to perform the installation. Verify the complete package name from the distribution media.

For example, if you downloaded the installation package into the `/home/bholly` directory:

```
cp /home/bholly/<Linux_Server_Package_Name> .
```

- 4 Unpack the tar using the following command.

```
tar -xzf <Linux_Server_Package_Name>
```

- 5 Change directories to the `/usr/local/psql/etc` folder where the installation scripts reside.

```
cd psql/etc
```

- 6 Run the pre-installation script:

```
sh preinstall.sh
```

- 7 Run the post installation script:

```
sh postinstall.sh
```

Your tar installation is complete. For additional information, see [After Installing Pervasive PSQL for Linux](#).

Upgrade Installation

If you have a previous version of Pervasive PSQL already installed, you must uninstall that product and then install the Pervasive PSQL v11 SP1 product.

See [Uninstalling Pervasive PSQL for Linux](#) for information on uninstalling Pervasive PSQL.

Installing Pervasive PSQL Client for Linux - TAR

The name of the Pervasive PSQL Client installation package conforms to the following conventions:

Table 13 Linux Client Package Names - TAR

Installation Type	Package Name
Client 64-bit	Pervasive.SQL-Client-Core-yy.yy-zzz.zzz.x86_64.tar.gz
Client 32-bit	Pervasive.SQL-Client-yy.yy-zzz.zzz.i486.tar.gz

In all cases, the *yy.yy* designates a release number and *zzz.zzz* designates a build number. Refer to the distribution media for the actual name of the package.

Installation is categorized as either first time, in which no previous version exists, or upgrade, in which a previous version exists.

- [First Time Installation](#)
- [Upgrade Installation](#)



Note The Client 64-bit installation does not include certain utilities, documentation, or SDK components. To install them, you need to install both the Client 64-bit and Client 32-bit products. Because this is the same as for the Pervasive PSQL Client for Windows, see [Installing the Pervasive PSQL Client for Windows](#) for details.

First Time Installation

➤ To install Pervasive PSQL Client using TAR

- 1 Log in as the root user.
- 2 Change to the `/usr/local` directory.
- 3 Enter the following command to copy the tar into `/usr/local`.

```
cp path_to_tar/<Linux_Client_Package_Name>.
```

For example, if the installation package resides in the `/home/bholly` directory:

```
cp /home/bholly/<Linux_Client_Package_Name>.
```



Note Refer to [Linux Client Package Names - TAR](#) for the package name to use. You must include the appropriate release and build number information to perform the installation. Verify the complete package name from the distribution media.

- 4 Unpack the tar using the following command.

```
tar -xzf <Linux_Client_Package_Name>
```

The unpacking action creates a directory named “psqlclient.”

- 5 Change directories to the `/usr/local/psql/etc` folder where the Pervasive PSQL installation scripts reside.

```
cd psql/etc
```

- 6 Run the pre-installation script:

```
sh clientpreinstall.sh
```

- 7 Run the post installation script:

```
sh clientpostinstall.sh
```

Your tar installation is complete. For additional information, see [After Installing Pervasive PSQL for Linux and Configuring Network Communications for Clients](#).

Upgrade Installation

If you have a previous version of Pervasive PSQL already installed, you must uninstall that product and then install the Pervasive PSQL v11 SP1 product.

See [Uninstalling Pervasive PSQL for Linux](#) for information on uninstalling Pervasive PSQL for more information.

After Installing Pervasive PSQL for Linux

The following topics are useful to review after you install Pervasive PSQL:

- [Verifying Installed Products With RPM](#)
- [Server Configuration](#)
- [Client Configuration](#)
- [User Count License](#)
- [Common Questions After Installation](#)

Verifying Installed Products With RPM

The following table provides commands with which you can verify which packages the RPM packager installed. The commands are case sensitive.

Table 14 RPM Commands To Verify Pervasive PSQL Packages Installed

Pervasive PSQL Package	RPM Command to Verify Installation
Server	<code>rpm -q Pervasive.SQL</code>
Client	<code>rpm -q Pervasive.SQL-Client</code> The command returns the specific client version installed (<code>Pervasive.SQL-Client-release-build</code>).
All installed	<code>rpm -qa grep Pervasive</code>

Verifying Database Engine is Running

Optionally, after the installation script finishes, you can verify that the database engine is running with the Linux `ps` utility. Type the following at the command line:

```
ps -e | egrep mkded
```

Server Configuration

Generally, the default configuration settings for Pervasive PSQL Server are sufficient. See [Configuration](#) for settings that you may want or need to set.

If you want to explore all of the configuration settings, see the chapter [Configuration Reference](#) in *Advanced Operations Guide*.

Client Configuration

All configuration settings for the Linux client are discussed in [Linux Client Configuration Parameters](#) in the *Advanced Operations Guide*.

In this guide, see also [Installing Pervasive PSQL Clients for Windows](#) and [Configuring Network Communications for Clients](#) for additional information about clients.

Linux Clients and the Monitor Utility

This information applies only to Linux clients that use a static IP address. Ignore this subsection if you use DHCP and have a DSN to resolve named addresses.

When you monitor Linux clients using the Pervasive PSQL Monitor utility, the client IP address that gets transmitted across the network originates from the “hosts” file. If the system name and IP have not been added to the “hosts” file, network communication uses the local host's IP address, which is 127.0.0.1 or ::1 (a loopback address).

If you change the loopback address to the correct IP, or if you add the system's name and IP to the “hosts” file on the Linux client, the client name correctly displays when in the Monitor utility.

User Count License

Once you have completed installation, you may need to update your user count license by using the `clilcdm` utility. The update can be done anytime before using Pervasive PSQL from a client. Information about how to do this can be found in *Pervasive PSQL User's Guide* (see [License Administration](#)). Detailed information about `clilcdm` can also be found in the Linux man pages. The *Pervasive PSQL User's Guide* also explains `clilcdm` (see [clilcdm](#)).



Note You must be a member of group `pvsu` to run the `clilcdm` utility. See [Pervasive PSQL Account Management on Linux](#) for more information.

Common Questions After Installation

If you have problems with your installation, see [Troubleshooting After Installation](#) or get help online from the Pervasive Knowledge Base at the Pervasive Web site. The following are common questions after installation of the products:

- [Where Do Files Reside After Installing Pervasive PSQL?](#)
- [How Do I Access the Documentation?](#)
- [What If I Get Errors Trying To Start the Utilities?](#)

Where Do Files Reside After Installing Pervasive PSQL?

The following table lists the primary directories and files that result from installing the Pervasive PSQL products on Linux.

\$PVSW_ROOT refers to the root directory where the files are installed. By default it is set to /usr/local/psql. Unless otherwise noted, the primary directories and files are the same for 32-bit and 64-bit products.

For an upgrade installation, your existing Pervasive PSQL files were updated to the latest versions.

Table 15 Primary Directories and Files for Pervasive PSQL Products Installed on Linux

Path from \$PVSW_ROOT	Primary Files	Description	Applies to Installation of
./	LICENSE	License information	Server
./bin		Binary files, executable utilities and so forth	Server and Client
./bin/plugins		A directory pertaining to files for the utilities and documentation	Server and Client
./data/DEMODATA		Sample Pervasive PSQL database	Server
./data/samples		Sample Btrieve files, alternate collating sequence file and the DefaultDB system database	Server
./etc			Server and Client
	.PSRegistry	Pervasive registry of configuration settings (this directory and its subordinate directories)	Server
	btpasswd	User passwords file	Server
	dbnames.cfg	Master table of database names	Server
	odbc.ini	ODBC settings	Server and Client
		Shell scripts for the following: <ul style="list-style-type: none"> ◆ pre-product installation ◆ post-product installation ◆ pre-product uninstall ◆ post-product uninstall 	Server and Client
./lib		Library of 32-bit shared objects	Server and Client

Table 15 Primary Directories and Files for Pervasive PSQL Products Installed on Linux

Path from \$PVSW_ROOT	Primary Files	Description	Applies to Installation of
./lib64		Library of 64-bit shared objects Note: This directory exists only if you install the 64-bit Server or Client	Server and Client
./log		Transaction log files directory	Server and Client
./man/man1		Man pages for the command-line utilities	Server and Client

How Do I Access the Documentation?

The documentation installed with Pervasive PSQL Server includes the following:

- Man pages for the command-line utilities
- Pervasive PSQL Documentation Library
- Pervasive PSQL Release Notes

Man Pages

Man pages are provided for the command-line utilities. To make these man pages available, add `$PVSW_ROOT/man` to your `MANPATH` environment variable.

Note that the man pages are installed with Pervasive PSQL Server and with Pervasive PSQL Client. They are **not** installed as part of the user documentation.

Documentation Library

The Pervasive PSQL Documentation Library contains the complete set of user documentation, including the user documentation for the Pervasive PSQL engine and software developer's kit, as well as a glossary of database terminology.

➤ To view the Pervasive PSQL Documentation Library

- 1 Open a terminal window.
- 2 Run one of the following:
 - a. As root user


```
/usr/local/psql/bin/pcc
```

b. As the psql user

pcc

Note that the viewer for the documentation library is integrated into Pervasive PSQL Control Center (PCC). The documentation library is accessed through the PCC interface on the Welcome view, in the Help menu, by pressing F1 (Windows) or Shift F1 (Linux).

Release Notes

The release notes in readme.htm contain late-breaking news that could not be included as part of the user documentation. The release notes file is located in the /usr/local/psql/docs/ directory.

What If I Get Errors Trying To Start the Utilities?

See [Starting PCC On Linux](#) in *Pervasive PSQL User's Guide* and [Starting DDF Builder from a Command Line](#) in *DDF Builder User's Guide*.

Uninstalling Pervasive PSQL for Linux

This section explains how to uninstall the RPM and TAR distributions of Pervasive PSQL.

RPM Version

The following table lists the RPM commands to uninstall the various Pervasive PSQL packages. You must log in as the root user using the “su” command before executing any of the commands.

Table 16 RPM Commands to Uninstall the Pervasive PSQL Packages

To Uninstall This Package	Use This RPM Command
32-bit Server or 64-bit Server	rpm -e Pervasive.SQL
32-bit Client or 64-bit Client	rpm -e Pervasive.SQL-Client



Note The uninstall program does **not** remove the system databases DEFAULTDB and SYSTEMDB.

TAR Version

The following table lists the shell scripts used to uninstall the various Pervasive PSQL packages. You must log in as the root user using the su command before executing any of the commands.

Table 17 TAR Commands to Uninstall the Pervasive PSQL Packages

Package To Uninstall	Script(s) To Execute ^{1,2}
32-bit Server or 64-bit Server	sh preuninstall.sh sh postuninstall.sh Note: The scripts must be executed in sequence: preuninstall first followed by postuninstall.

Table 17 TAR Commands to Uninstall the Pervasive PSQL Packages

Package To Uninstall	Script(s) To Execute ^{1,2}
32-bit Client or 64-bit Client Note: If you have both 32-bit and 64-bit Clients installed, see the following section.	sh clientpreuninstall.sh sh clientpostuninstall.sh Note: The scripts must be executed in sequence: clientpreuninstall first followed by clientpostuninstall.
¹ Assumes that the scripts are executed from the directory where they reside: /usr/local/psql/etc ² You may want to remove the uninstall scripts themselves after the product is uninstalled. For example: rm preuninstall.sh rm postuninstall.sh rm client*.sh	

Uninstalling Both 32-bit and 64-bit Clients

If you have installed both the 32-bit and 64-bit clients on your machine, you may uninstall one or both by passing the architecture option with the uninstall script. Running the scripts without any architecture option removes both clients, as the default option is to remove both clients.

Example

To uninstall only the 64-bit client you would run the following:

```
/usr/local/psql/etc/clientpreuninstall.sh -a x86_64
/usr/local/psql/etc/clientpostuninstall.sh -a x86_64
```

The 32-bit client remains fully operational.

To uninstall only the 32-bit client you would run the following:

```
/usr/local/psql/etc/clientpreuninstall.sh -a x86
/usr/local/psql/etc/clientpostuninstall.sh -a x86
```

The 64-bit client remains fully operational.

To uninstall both the 32-bit and 64-bit clients you would run the following:

```
/usr/local/psql/etc/clientpreuninstall.sh
/usr/local/psql/etc/clientpostuninstall.sh
```


Using Pervasive PSQL on Linux

chapter

13

Working With the Products on Linux

The chapter contains the following sections:

- [Finding What You Need](#)
- [Pervasive PSQL Account Management on Linux](#)
- [Configuration](#)
- [Client Information](#)
- [Setting Up Web-based Data Access](#)
- [Using Perl and ODBC with Pervasive PSQL](#)

Finding What You Need

Accessing the User Documentation See [How Do I Access the Documentation?](#)

Man Pages The man pages are installed with Pervasive PSQL Server or Client. Refer to the directory `$PVSW_ROOT/man/man1` for the man pages available.

To make these man pages easily accessible, add `$PVSW_ROOT/man` to your `MANPATH` environment variable. If you need more detailed information on a utility or application, see the chapter [Command Line Interface Utilities](#) *Pervasive PSQL User's Guide*.



Note Check the man pages for the most current information. Every effort is made to ensure that the information in this guide matches that in the man pages. On occasion, last-minute changes may be included in the man pages after this guide has been published.

Exclusions

Because the Linux platform is unique, the following areas in the Pervasive PSQL documentation do not apply to Linux.

- The section [Pervasive PSQL Message Logging](#) in *Advanced Operations Guide* regarding differs for Pervasive PSQL v11 SP1 on Linux.

Pervasive PSQL v11 SP1 uses the standard Linux logging system. Depending on the configuration of `/etc/syslog.conf`, messages are sent to the `syslogd` daemon, which does one of the following:

- logs it in an appropriate system log
- writes it to the system console
- forwards it to a list of users
- forwards it to `syslogd` on another host over the network

More information can be found in the man pages for `syslogd` and `syslog.conf`.

- The chapter, [Manipulating Btrieve Data Files with Maintenance](#) of the *Advanced Operations Guide* works only on the Windows client for Pervasive PSQL v11 SP1.

Pervasive PSQL Account Management on Linux

This section discusses information on Linux user accounts with respect to operation of Pervasive PSQL.

After Installation Behavior

- User `psql` has no password and can only be accessed through the `root` account by using the `su` command.
- You can access the `.bash_profile` for user `psql` with `~psql/.bash_profile`.
- All Pervasive files have `user:group` ownership `psql:pvs`
- You must be logged in as `root` to run the start and stop scripts for the Pervasive PSQL engines.
- You can run utilities on other user accounts if you add the necessary environment variables to the user `.bash_profile` or system `/etc/profile` as described in [Using Utilities from Users Other than psql](#).
- In addition to the instructions outlined in [Using Utilities from Users Other than psql](#), users other than `ROOT` must be a member of the group `pvs` to perform functionality with the following utilities:
 - Pervasive PSQL Control Center (PCC) to administer the local server.
 - License Administrator utility (`clilcadm`) for functions other than displaying current licenses.
 - Named Database Maintenance utility (`dbmaint`) for functions other than displaying current databases.
 - Pervasive Services Registry Editor (`psregedit`) for functions other than displaying the registry.
 - Linux command-line configuration (`bcfg`).

The User Environment

The single environment variable `$PVSW_ROOT` is used to determine the location of installed components. The generic location for configuration files are `$PVSW_ROOT/etc`. For executable files, the location is `$PVSW_ROOT/bin`. For shared libraries (32-bit) the location is `$PVSW_ROOT/lib`; for shared libraries (64-bit) the location is `$PVSW_ROOT/lib64`.

It is recommended that you add `$PVSQ_ROOT/bin` to your `PATH` environment variable, and `$PVSQ_ROOT/lib` to `LD_LIBRARY_PATH` as described in the following section.

***Using Utilities
from Users
Other than psql***

To use utilities from user accounts other than `psql`, you must first make modifications to the user account configuration. Add the following to either the profile for a specific user or to the profile that all users inherit.

```
/home/username/  
.bash_profile
```

Profile for the user. Similar to the `/etc/profile` file but only for the current user.

Look in `/home/username` for this file.

```
/etc/profile
```

Default profile for all user accounts on the system. Copy the lines below into this text file if you want all user accounts on the machine to have access to Pervasive PSQL utilities.

This does not give the users administrative privileges or access to Pervasive PSQL data.

Here is an example of a modified profile:

```
PVSQ_ROOT=/usr/local/psql  
PATH=$PATH:$PVSQ_ROOT/bin:/bin:/usr/bin  
LD_LIBRARY_PATH=$PVSQ_ROOT/lib:$PVSQ_ROOT/bin:/usr/lib  
MANPATH=$MANPATH:$PVSQ_ROOT/man
```

Ensure that you export all variables specific to Pervasive PSQL.

Configuration

Generally, the default configuration settings for Pervasive PSQL Server and Client are sufficient. You typically do not have to configure any settings for the database engine and clients to communicate and function together correctly. This subsection discusses two settings that you may want or need to configure:

- [Configuration File](#)
- [Authentication](#)

If you want to explore all of the configuration settings, see the chapter [Configuration Reference](#) in *Advanced Operations Guide*:

Configuration File

The Server configuration setting “Configuration File” defines the path to the Samba configuration file (smb.conf), which is parsed on engine startup to determine mapping between share names and server directory locations. See [Configuration File \(Linux engines only\)](#) in *Advanced Operations Guide*.

Authentication

This option specifies which type of authentication to use for access to the server engine. The available options are:

- **Emulate Workgroup Engine.** Use this value when Samba is used to authenticate user access on the system.
- **Proprietary Authentication (using btpasswd).** Use this value when not using Samba and the user does not have an account on the server. This allows a separate password file to be maintained when connecting to the Linux system.
- If you are using BTPASSWD or PAM authentication on your Linux server, user names and passwords must be set up using the pvnetpass utility from clients connecting to this server. See [pvnetpass](#) in the *Pervasive PSQL User's Guide*.
- **Standard Linux Authentication.** Use this value when not using Samba but users have accounts on the Linux system.

Supported Path Formats for Samba

From a Pervasive PSQL Client on a Windows platform, the order of path parsing is as follows:

- `\\server\share\relative\path`

share denotes a valid Samba share, made accessible to a Windows client.

server reads `smb.conf` to determine the absolute path to the shared directory, then combines it with the relative path to get a full UNIX path. The location of `smb.conf` is essential for valid resolution of the file path supplied in this format on the client. If the relative path is not correct, status 12 is returned.

- Drive:\path

drive must be a Samba drive mapped on the client. It is the client responsibility to convert it into the latter format and pass to a server, which never knows a drive mapping on the client.



Note Client users must be advised that share names on a Linux server are case sensitive. When mapping drives to a Linux server they must pay careful attention to the case of the share name if they want all their utilities to work properly.

- If neither `smb.conf` nor the share name are found, the path defaults to `\\server\absolute\path` format. If the absolute path is not correct, status 12 is returned.

Client Information

A Pervasive PSQL Client on Linux can connect to any of the Pervasive PSQL Servers provided the client and server machines can communicate with a shared protocol.

Authentication to Remote Machines

To connect to a remote machine using the Linux client, you need to have authentication to the remote machines. This is accomplished by entering a specific username and password for the server using the `pvnetpass` utility. This utility stores the username and password in an encrypted format for that particular server in the Pervasive registry on the client machine. If you do not specify user names and passwords, your applications can receive status code 3119.

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

Creating a Client DSN

A client data source name (DSN) is required if applications on the client use the Pervasive PSQL relational interface through ODBC. To create a client DSN, you use the `dsnadd` utility included with the Pervasive PSQL Client for Linux. See [dsnadd](#) in *Pervasive PSQL User's Guide* and the man page for `dsnadd` located in `/usr/local/psql/man/man1`.

Setting Up Web-based Data Access

This section contains information about configuring web servers to provide access to Pervasive PSQL data and provides connection snippets and samples for connecting to Pervasive PSQL data from web applications on Linux.

ODBC Behavior When you first install Pervasive PSQL, the `odbc.ini` file is written to `/usr/local/psql/etc`

If you have other ODBC driver managers such as `unixODBC`, they might be using a different `odbc.ini` file located, for example, at `/etc/odbc.ini`.

One way to unify the ODBC setup is to add soft links from where `unixODBC` expects the `odbc.ini` file to be located over to the Pervasive PSQL directories.

```
su
cd /etc
ln -s /usr/local/psql/etc/odbc.ini
```

Configuring Web Server

This section shows how you should set up the machine where the web server such as Apache resides.

You should make the user account under which you run any web server such as Apache a member of the group `pvs`. These user accounts run under restricted accounts such as `nobody`

To find the user account, see your Apache configuration file, typically located at `/etc/httpd/conf/httpd.conf`

In this file, the following lines show what user the Apache server uses to operate under.

```
User nobody
Group nobody
Options ExecCGI Indexes
```

You should add this user to the `pvs` group, substituting the name used in your Apache configuration file.

```
/usr/bin/gpasswd -a nobody pvs
```

PHP

PHP allows for easy development of web applications, using a style that is similar to both ASP in the Microsoft world and JSP in the Java world. Using PHP, you enclose database calls in special tags and format the output using HTML.

Pervasive PSQL PHP Requirements

- PHP - obtain from <http://www.php.net>
- DSN pointing to the database (use dsnadd)

PHP Connection Snippet

This code segment shows the essential part of connecting to a Pervasive PSQL database using PHP.

```
// connect to DEMODATA database no uid or password
$connect = odbc_connect("demodata", "", "");

// set the query variable to your SQL
$query = "SELECT * from Department";

// obtain a result object for your query
$result = odbc_exec($connect, $query);
```

PHP Sample

This complete sample presents the user a choice of three DEMODATA tables and then displays the table.

```
<HTML>
<HEAD>
<TITLE>PVSU PHP Sample</TITLE>
</HEAD>
<BODY>

<H1>Pervasive Hello World Samples - PHP using PHP
ODBC APIs</H1>
<P>
This sample will display the DEMODATA database
tables in the following drop-down
by using PHP.
</p>

<?

// -----MAIN MENU-----

// if there is no function specified in the URL
```

```

if (!(isset ($HTTP_GET_VARS["_function"]))):

// -----

?>

<p>Please select from the following tables</p>
<form method=post
action='<?=$PHP_SELF?>?_function=showtable'>
<select name="selecttable">
<option SELECTED value="Department">Department
<option value="Course">Course
<option value="Room">Room
</select>

<p>
<input type=submit value="Show table">
</p>
</form>

<?

// -----SHOWTABLE-----

Elseif ($HTTP_GET_VARS["_function"] ==
"showtable"):

// -----

print("<p>Return to <a href='$PHP_SELF'>Sample 1
Main menu</a></p>");

    $thetable = $HTTP_POST_VARS["selecttable"];
// determine from FORMS data which table to open

    $connect = odbc_connect("demodata", "", "");
// connect to DEMODATA database no uid or password
    $query = "SELECT * from $thetable";
// set the query variable to contain the SQL you
// want to execute
    $result = odbc_exec($connect, $query);
// perform the query

// print out the entire resultset as HTML table
// (uncomment following line)
// odbc_result_all($result);

// or format the output yourself and display

```

```
// a nicer table (but more code required)

// initialize row counter
$i = 0;

// determine number of columns
$numcols = odbc_num_fields($result);

// start HTML table
print("<table border=1 cellpadding=5>");

// PRINT COLUMN HEADINGS

print("<tr>"); // start of row

while ($i < $numcols)
{
    $i++;
    $colname = odbc_field_name($result, $i);
    print("<th>$colname</th>");
}
$i=0;

print("</tr>"); // end of row

// PRINT TABLE DATA

// while there are still rows
while(odbc_fetch_row($result))
{
    print("<tr>"); // start row

    while ($i < $numcols)
    {
        $i++;
        $tablecell = odbc_result($result, $i);
        print("<td>$tablecell</td>");
    }

    print("</tr>"); // end row
    $i = 0; // reset counter
} // end odbc_fetch_row

print("</table>"); // end HTML table

odbc_close($connect); // CLOSE THE CONNECTION
```

```

// END OF SHOWTABLE

// ---CATCH INVALID MENU OPTIONS-----

Else:

// -----

print("<p>An Invalid function was entered. Please
<a href='\$PHP_SELF'>try again</a>.</p>");

Endif;

?>
</BODY>
</HTML>

```

Additional PHP Sample

A more comprehensive PHP sample application that simulates the operations of a video store is available online at the Pervasive Software Web site.

This sample uses the Pvideo database that is included with the Pervasive PSQL SDK. If you do not have the SDK installed, you can download the Pvideo database separately with the sample application.

Perl

Perl allows for both command line and web-based applications using Pervasive PSQL.

Pervasive PSQL Perl Requirements

- Perl
- ODBC-DBD library
- CGI library
- DSN pointing to the database

Perl Connection Snippet

This code segment shows the essential part of connecting to a Pervasive PSQL database using Perl.

```
# specify use of Perl's database interface (DBI)
```

```
use DBI;

# connect to DEMODATA database no uid or password
$dbInfo = "DBI:ODBC:DEMODATA";
$dbUserName = "";
$dbPassword = "";

# set the query variable to your SQL
$query = "SELECT * FROM Department";

# Connect to the server
$connect = DBI->connect($dbInfo, $dbUserName,
$dbPassword);

# Prepare the SQL query
$myRecordSet = $connect->prepare($query);

# Execute the query and obtain a recordset
$myRecordSet->execute();
```

Perl Sample

This complete sample presents the user a choice of three DEMODATA tables and then displays the table.

```
# Perl sample

use CGI":cgi-lib";
$cgiquery = new CGI;

$functionreq = $cgiquery->url_param('_function');
# use 'url_param' for GET and 'param' for POST

print &PrintHeader;
print &HtmlTop("Pervasive PSQL Hello World Sample -
Perl");

print <<ENDOFMENU;
<H1>Pervasive Hello World Samples - Perl</H1>

<P>
This sample will display the DEMODATA database
tables in the following drop-down
by using Perl/DBI.
</p>
ENDOFMENU

# -----MAIN MENU-----
```

```

# if there is no function specified in the URL

if (!$functionreq) {

# -----

print <<ENDOFTEXT;

<p>Please select from the following tables</p>
<form method=post
action="$ENV{'SCRIPT_NAME'}?_function=showtable">
<select name="selecttable">
<option SELECTED value="Department">Department
<option value="Course">Course
<option value="Room">Room
</select>
<p>
<input type=submit value="Show table">
</p>
</form>

ENDOFTEXT

} # !($function)

# -----SHOWTABLE-----

elsif ($functionreq eq "showtable") {

print("<p>Return to <a
href='$ENV{'SCRIPT_NAME'}'>Perl Hello World Sample
- Main Menu</a></p>");

# determine from FORMS data which table to open
$thetable = $cgiquery->param('selecttable');

use DBI;

$dbInfo = "DBI:ODBC:DEMODATA";
$dbUserName = "";
$dbPassword = "";

$query = "SELECT * FROM $thetable";
$connect = DBI->connect($dbInfo, $dbUserName,
$dbPassword);

$myRecordSet = $connect->prepare($query);
$myRecordSet->execute();

```

```
# start HTML table
print "<table border=1 cellpadding=5>";

# PRINT COLUMN HEADINGS

$num_fields = $myRecordSet->{NUM_OF_FIELDS};
$count = 0;

print "<tr >";
while ($count < $num_fields) {
    $column_name = $myRecordSet->{NAME}->[$count];
    print "<th>$column_name</th>";
    $count++;
}
print "</tr>\n";

$count = 0;

# PRINT TABLE DATA

while(@row=$myRecordSet->fetchrow_array) {
    print "<tr>\n";
    while ($count < $num_fields) {
        print "<td>$row[$count]</td>\n";
        $count++;
    }
    print "</tr>\n";
    $count = 0;
}

print "</table>";          # end HTML table
# END OF SHOWTABLE
}

# -----CATCH INVALID MENU OPTIONS-----

else {

print "<p>An Invalid function was entered. Please
<a href='$ENV{'SCRIPT_NAME'}'>try again</a>.</p>";

}

print &HtmlBot;
```

Using Perl and ODBC with Pervasive PSQL



Note This procedure assumes you have a working installation of Pervasive PSQL v11 SP1, Perl, and an ODBC distribution. A free version of ODBC is available at <http://www.iODBC.org>. Perl can be found at <http://www.perl.org>

➤ To Get Pervasive PSQL to work with Perl's ODBC Interface

- 1 Download the DBI (database interface) support for Perl.
Read the Readme or INSTALL for instructions.
- 2 Download the ODBC DBD database driver for Perl.
Please see the installation instructions in the Readme or INSTALL file.
- 3 Make sure you have the proper environment variables set, as shown in the following example. Note, this is also explained in the iODBC docs.

Code Snippet for Perl and DBI

```
print "using odbc...\n";
use DBI;
$dbName = "DBI:ODBC:DEMODATA";
$dbUserName = "";
$dbPassword = "";
print "connecting...\n";
$sql = "SELECT * FROM class";
$dbh = DBI->connect($dbName, $dbUserName,
    $dbPassword);
$dataObject = $dbh->prepare($sql);
$dataObject->execute();
while(@row=$dataObject->fetchrow_array)
{
print "$row[0]\t$row[1]\t$row[2]\n\n"
}
}
```


Troubleshooting After Installation

How to Proceed When You Encounter Errors During Installation

Pervasive Software provides several features and tools in Pervasive PSQL v11 SP1 that help to prevent configuration and installation problems.

Some of these utilities are installed and run as part of the installation process and all can be run later to evaluate configuration and registry settings and to troubleshoot problems. They are shown in Table 18.

This chapter contains the following sections:

- [Troubleshooting Tools](#)
- [Troubleshooting Strategies](#)
- [Configuration for Special Installation Situations](#)
- [Diagnosing Problems with Pervasive System Analyzer \(PSA\)](#)
- [Verifying Database Engine is Running](#)
- [Obtaining File, Client, and Engine Version Number](#)
- [Engine and Client Version Conflicts](#)
- [Troubleshooting Common Pervasive PSQL Issues](#)
- [Issues After Uninstalling Pervasive PSQL on Windows](#)
- [How to Get Additional Help](#)

Troubleshooting Tools

The following table describes some tools that can help you avoid or solve problems.

Table 18 Pervasive Tools that Assist in Installation and Problem Determination

Feature/ Component	Function	For More Information
Pervasive System Analyzer	Analyzes system components and runs communication tests.	See Diagnosing Problems with Pervasive System Analyzer (PSA) .
Pervasive Message Logging	Logged messages can be of type status, information, warning, or error, and can originate from any Pervasive PSQL component.	See Pervasive PSQL Message Logging
Gateway Locator	Determines or changes the Gateway being used for a particular data dictionary (only in Pervasive PSQL v11 SP1 Workgroup Edition.)	See Configuring the Workgroup Engine .
Knowledge Base	Provides information about many Pervasive software configurations and common environments.	Search the Pervasive Knowledge Base at: www.pervasivedb.com

Troubleshooting Strategies

Pervasive Software hopes that your installation process completes without experiencing any problems. However, this depends on a number of factors, including proper network support, and operating system configuration.

If something does go wrong during an installation, Pervasive offers some tools that can help in diagnosing the problem. This chapter explores some of the troubleshooting techniques that you can use.



Note If the installation fails for any reason, the installation log file can be found in the Windows %Temp% directory.

Checklist for Problems

- Did you see any error messages displayed during installation?
- Does the Network function correctly?
- Do you have the appropriate administrator-level privileges?
- Is the Engine running?
- Is the Client software correctly functioning?
- Are there errors in your PVSW.LOG file?

Troubleshoot the Problem

The rest of this section contains procedures that you can use in verifying your installation.

- [Logged Messages](#)
- [Configuration for Special Installation Situations](#)
- [Diagnosing Problems with Pervasive System Analyzer \(PSA\)](#)
- [Verifying Database Engine is Running](#)
- [Obtaining File, Client, and Engine Version Number](#)
- [How to Get Additional Help](#)

Logged Messages

Messages logged by Pervasive PSQL can help you troubleshooting problems. Messages can be of type status, information, warning, or error, and can originate from any Pervasive PSQL component.

Certain messages specific to licensing issues originate only from the license administration components. In either case, Pervasive PSQL logs messages to the following repositories:

- Pervasive Notification Viewer
- Operating System Event Log
- Pervasive PSQL Event Log (PVSU.LOG) (Windows only)

Any licensing message logged to Pervasive Notification Viewer is also logged to the Operating System Event Log and to Pervasive PSQL Event Log. Similarly, any licensing message logged to the Operating System Event Log is also logged to Pervasive PSQL Event Log. Note that the Operating System Event Log and Pervasive PSQL Event Log may contain licensing messages not logged to Pervasive Notification Viewer.

Messages not specific to licensing are logged to the Operating System Event Log and to Pervasive PSQL Event Log.

Follow these guidelines for using messages to troubleshooting problems:

If you suspect a problem related to licensing, first check Notification Viewer, then check the Operating System Event Log and Pervasive PSQL Event Log. If you suspect a problem *not* related to licensing, check the Operating System Event Log and Pervasive PSQL Event Log.

For logging details, see [Pervasive PSQL Message Logging](#) in *Advanced Operations Guide*. For status code details, see [Status Codes](#) in *Status Codes and Messages*.

Configuration for Special Installation Situations

This section lists some scenarios where the *default* configuration settings for Pervasive PSQL need adjusting for proper database operation.

The following table summarizes some of these situations. If you find that your configuration matches an issue, please see the reference included for more information.

If your computing environment includes...	Then you need to:
Microsoft Active Directory Service	Read the following section: Active Directory Service
Multiple network interfaces	Enable a configuration setting for Multi-homed setting In <i>Advanced Operations Guide</i> , see: <ul style="list-style-type: none"> • TCP/IP Multihomed • Listen IP Address
A network that is subject to outages	Enable a configuration setting that tries to auto-reconnect to a server when a network outage occurs <ul style="list-style-type: none"> • In <i>Advanced Operations Guide</i>, see Pervasive Auto-Reconnect.
Database filenames that must not include embedded spaces	Enable a configuration setting that instructs Pervasive PSQL to reject files with embedded spaces in the name. <ul style="list-style-type: none"> • In <i>Advanced Operations Guide</i>, see Embedded Spaces and Long File Names and Embedded Spaces Support.

Diagnosing Problems with Pervasive System Analyzer (PSA)

Pervasive System Analyzer is a diagnostic utility included with Pervasive PSQL v11 SP1.

Pervasive System Analyzer (PSA) is conveniently integrated into the product installation and available as a stand-alone diagnostic tool to help you with a the following tasks:

- Troubleshoot network problems
- Detect previous installations of Btrieve or Pervasive PSQL on your system
- Note other factors that influence your networking environment
- View current component set and versions

PSA replaces the features that were previously offered by SmartScout and InstallScout.



Note For detailed information on using Pervasive System Analyzer, refer to the *Pervasive PSQL User's Guide*.

Verifying Database Engine is Running

To verify that the Pervasive PSQL engine is running, see the procedure for your platform and engine:

- [Windows Server \(Non-Vista\)](#)
- [Windows Workgroup](#)
- [Linux Server](#)

Windows Server (Non-Vista)

You can use the Services function of the Windows control panel.

➤ To check Pervasive Services on Windows servers using the Control Panel

- 1 At the operating system, open **Services** under **Administrative Tools**.
- 2 Scroll the list of services until you reach the following services.
 - Pervasive PSQL Transactional Engine
 - Pervasive PSQL Relational Engine

Both of these services must be started if Pervasive PSQL is to function correctly.

The **Status** column displays whether or not the service is currently running. The **Startup** column indicates whether the service is set to automatically start on system startup or start manually.

Figure 7 *Displaying the Services Status*

Name	Description	Status	Startup Type	Log On As
Indexing Service			Manual	LocalSystem
Internet Connection Sharing	Provides n...		Manual	LocalSystem
IPSEC Policy Agent	Manages I...	Started	Automatic	LocalSystem
Logical Disk Manager	Logical Disk...	Started	Automatic	LocalSystem
Logical Disk Manager Administr...	Administrat...		Manual	LocalSystem
Messenger	Sends and ...	Started	Automatic	LocalSystem
Net Logon	Supports p...		Manual	LocalSystem
NetMeeting Remote Desktop S...	Allows aut...		Manual	LocalSystem
Network Connections	Manages o...	Started	Manual	LocalSystem
Network DDE	Provides n...		Manual	LocalSystem
Network DDE DSDM	Manages s...		Manual	LocalSystem
Norton AntiVirus Client		Started	Automatic	LocalSystem
NT LM Security Support Provider	Provides s...		Manual	LocalSystem
Performance Logs and Alerts	Configures...		Manual	LocalSystem
Pervasive PSQL Relational Engine		Started	Automatic	LocalSystem
Pervasive PSQL Transactional Engine		Started	Automatic	LocalSystem
Plug and Play	Manages d...	Started	Automatic	LocalSystem

- 3 If a service is not started, right-click the service name, then click **Start**.

Windows Workgroup

To verify that the Pervasive PSQL v11 SP1 workgroup engine is running:

➤ **To start the Pervasive Workgroup engine**

- 1 From the **Start** menu, select **Engines** from the **Pervasive** program.
- 2 Click **Start Workgroup Engine**.

By default, the MicroKernel allocates resources and is ready to service local application database requests.

➤ **To stop the Pervasive Workgroup engine**

- 1 From the **Start** menu, select **Engines** from the **Pervasive** program.
- 2 Click **Stop Workgroup Engine**.



Note You will receive a warning message when trying to stop the engine if any of the following is true:

- There are active clients.
 - No activity took place since the engine loaded.
 - 10 seconds has not elapsed since the last operation took place.
-

Linux Server

You can verify that the engine (`mkded`) is running with the Linux `ps` utility:

Type the following at a command line:

```
ps -e | egrep 'mkded'
```

➤ **To start the Pervasive PSQL services in Linux**

Enter the following at the command line under the `root` user account:

```
etc/init.d/psql start
```

Obtaining File, Client, and Engine Version Number

You can use Pervasive PSQL utilities to verify that the client and engines have the version number you expect, or to check the version of a particular file.

Determining Client and Engine Version

You can check the engine and client versions using Function Executor on Windows platforms or using the BUTIL command-line utility on all platforms. Function Executor is a utility that simulates Btrieve client operations using the Pervasive PSQL requesters.

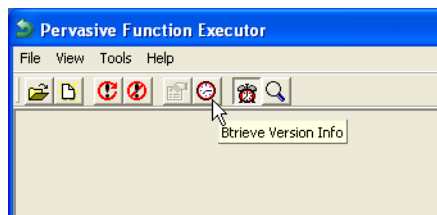
Using Function Executor

Use Function Executor to determine the version of the client, local and remote engines.

➤ **To Determine the Engine Version using Function Executor**

- 1 On the **Start** menu select **Function Executor** from the **Pervasive PSQL v11 SP1 Utilities** program group.
- 2 Do one of the following:
 - a. Click **View** ▶ **Version** from the **File** menu.
 - b. Select the **Btrieve Version Info** toolbar button, as shown in Figure 8.

Figure 8 Selecting the Btrieve Version Info button



- 3 After choosing either of the **Version** options, a dialog box displays that indicates the version of the client requesters and the local engine. If a file is open when the **Version** option is selected, the remote engine version displays as well.

Figure 9 Btrieve Version Info Display



Using the BUTIL Utility

From a command prompt, enter the following:

```
BUTIL -VER
```

The requester and engine versions are then displayed. You cannot determine the version of a remote server engine with this tool.

Determining a File Version

You can determine the file version of a MicroKernel data file using the Pervasive PSQL v11 SP1 utilities. On the Windows platform, use Control Center, Function Executor, DDF Builder, or Btrieve Maintenance. On any platform, use the BUTIL command-line utility. The following provides information on using a few of these methods.

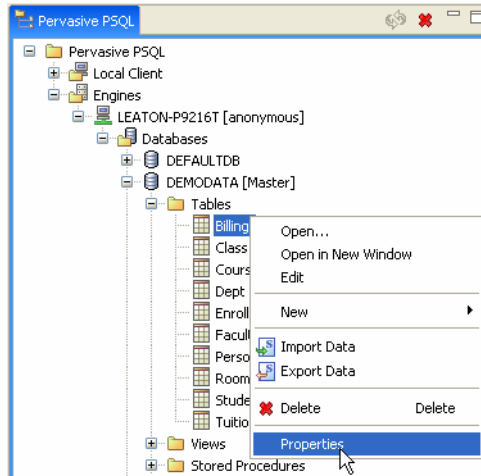
Using the Pervasive PSQL Control Center

You can use the Pervasive PSQL Control Center to determine a file version.

➤ To Determine the File Version of a Table Using Pervasive PSQL Control Center

- 1 On the Start menu select **Control Center (PCC)** from the **Pervasive PSQL v11 SP1** program group.
- 2 Find the database by expanding its name in the Pervasive PSQL Explorer on the left.
- 3 Do one of the following:
 - a. Click **File** ▶ **Properties** from the File menu.
 - b. Right-click a table name and select **Properties** as shown in Figure 10.

Figure 10 Obtaining a File Version with the Pervasive PSQL Control Center



- 4 The table properties are displayed, which includes the file version of the underlying MicroKernel data file version.

Figure 11 Table Properties Page

Parameter	Value
Table Name	Billing
Table Location	C:\DOCUMENTS AND SETTINGS\ALL USE...
Dictionary Path	C:\DOCUMENTS AND SETTINGS\ALL USE...
File Version	9.5
Record Length	139
Page Size	4096
Number of Records	1315
Number of Indexes	1
Number of Unused Linked Duplicate Pointers	0
Number of Unused Preallocated Pages	0
Variable Length Records	No
Blank Truncation	No
Record Compression	No
Page Compression	No
Key Only File	No
Index Balancing	No
Freespace Threshold	0%
Uses Alternate Collating Sequence	No
System Data	Yes

Using Btrieve Maintenance

If you are unfamiliar with the command line, you can use the GUI-based Btrieve Maintenance tool.

➤ To Determine the File Version of a Table Using Btrieve Maintenance Utility

- 1 From the **Start** menu click **Maintenance** from the **Pervasive Utilities** program group.
- 2 From the **File** menu, click **Options** and select **Show Information Editor**.

The **File Information Editor** dialog box displays.

- 3 Click **Load Information** and the **Select File** dialog box displays.
- 4 Enter or browse for the file for which you need to determine the version.

The version displays in the upper right-hand corner of the dialog box.

Using Function Executor

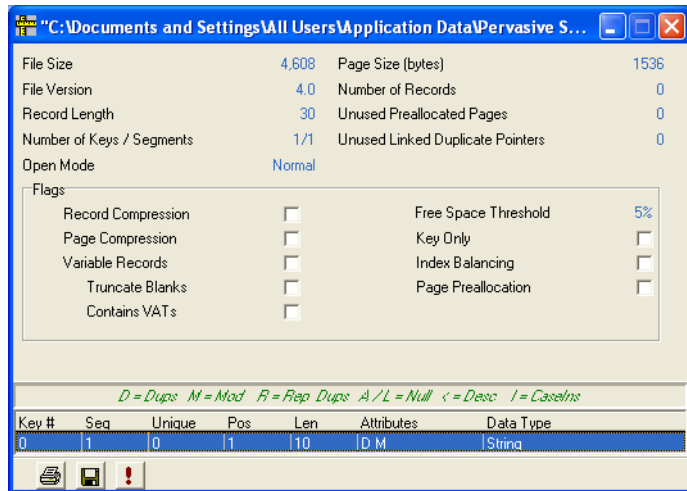
The Function Executor utility can simulate Btrieve operations and can be used to determine the file version by performing a statistics report against the file.

➤ To Determine the File Version of a Table Using Function Executor

- 1 From the **Start** menu click **Function Executor** from the **Pervasive Utilities** program group.
- 2 From the **File** menu, click **File ▶ Open**.
The **Open Btrieve File** dialog box displays.
- 3 Enter or browse for the file for which you need to determine the version.
- 4 With the file open in Function Executor, click **View ▶ File Statistics**.

The **File Statistics** dialog box displays the file version in the top portion of the screen, as seen in Figure 12.

Figure 12 File Statistics in Function Executor



The Function Executor utility is documented in more detail in *Advanced Operations Guide*.

Using BUTIL command-line utility

Use the `-stat` parameter of BUTIL to query the file statistics, which includes information about:

- File version
- Pages
- Records
- Keys

Type the following at a command prompt:

```
butil -stat <filename>
```

For example, to query the statistics of the file `DEPT.MKD` of the `DEMODATA` database included with Pervasive PSQL:

```
butil -stat dept.mkd
```

The BUTIL utility (available on Windows and Linux) is documented in more detail in *Advanced Operations Guide*.

Engine and Client Version Conflicts

If you update your engine to the latest Pervasive PSQL version without also updating your client requesters, you may encounter warning messages from Pervasive PSQL indicating the version conflict. The message displayed is:

```
An engine to client component mismatch was found
```

When you receive such a message, it is also logged to your Pervasive Event Log (PVSUW.LOG).

This message is a warning. The client is not prevented from connecting to the engine in this situation. Note, however, that Pervasive recommends that you use client requesters that are the same version as the database engine. If you choose, you may use a client requester that is an *older* version than the database engine with which it interacts. In some situations, depending on the type of SDK access method used by your application, an older version requester will not work with the database engine. Your application will be unable to communicate with the database engine. For those situations, you must use client requesters that are the *same* version as the database engine.

Client requesters that are a *newer* version than the database engine may or may not function correctly. Pervasive does **not** guarantee that newer versions of client requesters will function correctly with older versions of the engine. Therefore, Pervasive recommends that you **avoid** the use of newer version client requesters with an older engine.

If circumstances in your organization dictate that you cannot upgrade the clients for some time, you may want to disable the dialog boxes that appear when your client components are authorized. However, you cannot disable the entries in the Pervasive Event Log, and you should note that over time this log could grow to a large size as these entries are logged.

To permanently solve the problem, update your client requesters to the same version as your server engine.

Troubleshooting Common Pervasive PSQL Issues

This section outlines problems you may encounter during the installation or when first using the Workgroup product.

I receive Status 7224 or my license is no longer listed in the License Administrator utility.

When the Pervasive PSQL Workgroup is installed as an application on Vista and Windows 7 operating systems, you may experience this situation. Applications do not inherit the user's administrative rights on Vista and Windows 7.

The Workgroup Engine can be installed as a service or you may stop the engine, run it as administrator, and then run the command line license administrator or GUI License Administrator as administrator to apply the license.

I fail to see the effects of my configuration changes.

Try stopping and then restarting the database engine. Whenever you make a change to engine configuration components, you must stop and restart the database engine for the changes to take effect. For information on how to start and stop the database engine, see [Verifying Database Engine is Running](#).

Why do I receive Status 7012 when trying to create a new database with the Workgroup Engine using PCC on Windows Vista?

When PCC creates a new database, the new database name is added to dbnames.cfg and an entry is added to the ODBC.INI registry in order to create a corresponding System DSN.

Due to Microsoft Vista operating system constraints on registry access, the Workgroup Engine should be run in an elevated mode, so that the database System DSN can be created.

Once the System DSN is created successfully, any user may start the Workgroup Engine and use the DSN.



Note In Windows Vista, standard users may create *User DSNs* without this restriction.

Why do I (now) receive Status 95, after running my application successfully?

Your application has lost its session with the database engine. This can happen if you make changes to your configuration settings and must restart the database engine, as in the troubleshooting example given above. At the moment the database engine is stopped, any application that is running loses its session with the database engine. You must stop all those utilities and restart them in order to reestablish communication.

See the *Status Codes and Messages* manual for more cases in which this status code can be returned.

Installing a Pervasive PSQL application has rendered another application unusable.

If the latest DLLs have been overwritten, it is possible to restore the overwritten DLLs using a backup directory that is automatically created when you install Pervasive PSQL v11 SP1.

How do I verify that my DOS components are functioning properly?

Pervasive provides a DOS version of BUTIL.EXE for purposes of verifying that your DOS components are functioning properly. This file is installed in the PSQL\BIN folder of the Pervasive PSQL v11 SP1 Program Files default installation directory.

Why can't I restart my application after an improper program exit?

Database engine components may remain in memory if the engine is interrupted improperly.

➤ **If you cannot restart your program after improperly aborting the application by using Ctrl-C or stopping the process:**

- 1 Shut down and restart your system.
- 2 Avoid terminating applications in an abnormal manner.

Why isn't my application using the Workgroup engine?

If you previously installed Pervasive PSQL requesters and later installed the Pervasive PSQL v11 SP1 Workgroup engine but your application is only using the requesters, you may have an outdated configuration that sets Local Access to Off. The Pervasive PSQL v11 SP1 Workgroup engine's installation does not overwrite existing settings. To reset Local Access to On, see [Using the Server and Workgroup Engines Concurrently](#).

How Do I Access the Pervasive PSQL v11 SP1 Online Manuals?

➤ **To access the online documentation:**

- 1 Click **Control Center & Documentation** from the Pervasive program group off the Start menu.
- 2 Click the desired manual on the PCC Welcome page. (If the Welcome page has been closed, click **Help** then **Welcome**.)

I received an error message during installation that begins: "Setup did not update the PATH statement in autoexec.bat because the new path would be too long for Windows."

This message appears when the installation program cannot update the PATH environment variable because the resulting PATH definition would be too long (exceeds the environment space). For info on how to increase the environment space defined in config.sys, see the Microsoft knowledge base article:

<http://support.microsoft.com/?kbid=230205>

If you get this error message, then a REM statement (a comment) has been added to your autoexec.bat file. The REM statement contains the PATH value that would have been entered. You can change the PATH statement manually.

The best approach, if possible, is to install the product at a location with a shorter installation directory so that the value of PATH does not exceed the environment space.

My license is now disabled

Pervasive PSQL licenses are intended to reside on one machine only be it physical or virtual. Moving the license from a physical machine to a virtual machine without deauthorizing the license first will result in a disabled license. Other changes that can disable your license include:

- Mac address changes
- CPU name or type changes
- Machine name changes

These types of changes often occur when licenses are moved from a physical to a virtual machine or one virtual machine to another virtual machine without deauthorizing the license first.

Issues After Uninstalling Pervasive PSQL on Windows

When you uninstall Pervasive PSQL using the Add/Remove Programs mechanism in Windows, you should not have any database engine files remaining on your system. However, some actions such as installing multiple times to the same machine or restoring archived components can cause a significant number of files to be left on your system. This is a side effect of how the installation process works with the Windows operating system.

In the situations described previously, the files are left because Windows has the files marked with usage counts that indicate that they are being used by more than one program, and therefore the uninstallation program does not remove them from your system. This is expected behavior, but it may lead you to conclude that the Pervasive PSQL uninstall program is not functioning correctly.

How to Get Additional Help

Pervasive Software strives to ensure that your product installation is easy and successful. If you encounter problems during or after the installation that are not covered in the user documentation, please contact Pervasive Software and we will address your problem promptly.

See Table 19 in *Pervasive PSQL User's Guide* for a list of resources to help you get answers to your questions, troubleshoot problems, and interact with the Pervasive team as well as with other customers.

Technical Support

If you still have questions or problems relating to your Pervasive PSQL v11 SP1 installation, you can obtain help from the Pervasive Customer Support department.

Index

A

- About Pervasive PSQL
 - engines 8
 - relational or transactional access 3
- Accessing
 - data on other computers 135
 - documentation in Linux 152, 191
 - documentation in Windows 64
 - remote server engines 133
- Active Directory 122
 - administrative rights required for 122
 - create Pervasive_Admin group on domain controller 123
 - directory and file permissions 122
 - grant log on privileges to Pervasive_Admin group 126
 - installation of Pervasive PSQL with 25
 - Pervasive PSQL clients used with 122
 - tasks 123
 - use of Terminal Services with 122
- ActiveX Interface Controls
 - Pervasive Access Methods 18
- Administrative rights
 - for Active Directory 122
- ADO.NET Provider
 - Pervasive Access Methods 18
- After Installing Pervasive PSQL for Linux 149
 - behavior 160
- Application data files
 - in Pervasive PSQL 63
- Applications
 - configuration scenarios 67
 - configuring for multiple 132
 - developing with Pervasive PSQL 10
 - not using Workgroup engine 191
 - restarting after improper program exit 190
- Authentication
 - for Linux 162
- Authentication to Remote Machines 164
- Automatic encoding 110

B

- Before you install
 - common questions 23
 - Pervasive PSQL for Linux 138
 - Windows client 44
 - Windows server 36
 - Windows workgroup 54
- BTRBOX Requester 117
 - installation 49
- Btrieve
 - file conversion 34
 - older versions, problems with 190
- Btrieve DOS
 - Pervasive Access Methods 18
- BUTIL
 - determining file version 184

C

- Changes, configuration
 - effects not seen 189
- Changing
 - default communication ports 104
- Checklist
 - for installation 22
 - for installation troubleshooting 177
- Citrix MetaFrame
 - installing Pervasive PSQL on 24
- Client
 - determining version 183
 - DOS support 117
 - encoding 111
 - engine for Windows 16
 - installation 45
 - installed with Workgroup engine 64
 - network communication settings 94
 - SPX support 103
 - TCP/IP support 101, 108
- Client configuration
 - in Linux 149
 - on Linux 149
- Client encoding 110
 - interaction with database code page 111
- Client engine
 - 32-bit for Windows 17
 - 64-bit for Windows 17
 - install image 64

- installation 43
 - installation location 23
 - installed after server 25
 - using 64 bit with Workgroup engine 63
 - Client Information
 - for Linux 164
 - Client installation
 - on Linux 138
 - Client requesters
 - concepts 50
 - Client version conflicts
 - troubleshooting 188
 - Client/server configuration
 - workgroup engine 68, 71
 - Cobol Schema Executor
 - Utilities 20
 - Code page 110
 - Code Snippet for Perl and DBI 173
 - Common Questions After Installation 150
 - Communication ports
 - changing defaults 104
 - Communication, network
 - client settings 94
 - Communications
 - testing 180
 - troubleshooting 180
 - Complete Setup
 - installation option 15
 - Configuration
 - database code page 110
 - on Linux 162
 - Configuration changes
 - effects not seen 189
 - Configuration File
 - for Linux 162
 - Configuration parameters
 - for server authentication 162
 - Configuration settings
 - affected by multiple applications 130
 - migration of existing, during upgrade 32
 - other 27
 - to restrict access on Terminal Services 26
 - Configuring
 - application scenarios 67
 - for special installation situations 179
 - multiple applications, for 132
 - server engine with two network cards 87
 - SPX support for Windows server 88
 - TCP/IP on Windows server 86, 90
 - web server 165
 - workgroup engine 81
 - Connectivity
 - testing 175
 - Create File Version
 - multiple applications and 130
 - Creating a Client DSN 164
 - Creating, new database
 - with Workgroup engine using PCC (Windows Vista) 189
 - Custom Setup
 - installation option 15
- ## D
- Data
 - accessing on other computers 135
 - Data Dictionary File Builder
 - Utilities 20
 - Data Source Names
 - handling 33
 - Data translation 110
 - Database code page 110
 - interaction with client encoding 111
 - Database encoding 110
 - Database engine
 - checking status 181
 - uninstall on Linux 154
 - uninstalling on Windows 66
 - Default communication ports
 - changing 104
 - Determining
 - type of network 84
 - Developing
 - applications with Pervasive PSQL 10
 - Diagnosing
 - system problems 180
 - Disabled license
 - troubleshooting 192
 - Disabling administrative functions
 - in Terminal Services 120
 - DNS
 - used to configure Server IP address (Linux) 108

used to configure Server IP address (Windows)

101

Documentation

accessing 191

accessing in Linux 191

accessing in Windows 64

accessing on Linux 152

installing as an optional feature 20

DOS

components 190

components, troubleshooting 190

utilities 190

DOS box 117

preferred for Windows platforms 118

DOS requesters 50

ODBC not supported 117

using 117

with Active Directory 122

Downloading

Pervasive PSQL installation file 25

DTO

Pervasive Access Methods 19

E

Encoding 110

ANSI 112

automatic 110

client 110, 111

interaction between client and database code page
111

OEM 112

Engine

checking status 181

determining version 183

features compared 9

network communication settings for Pervasive
PSQL 85

relational 6

running on Terminal Services 36

running Workgroup as service 81

stopping Workgroup as service 82

transactional 4, 5

troubleshooting version conflicts 188

Environment space 191

Exclusions

in Linux 158

F

Features

optional in Pervasive PSQL 18

File Conversion 34

Btrieve 34

rebuild utility 34

scalable SQL 34

File locations

for Pervasive PSQL 62

File version

determining 184

Files

difference between 32-bit and 64-bit 63

installed on Linux 151

services 104

Finding what you need

using Pervasive PSQL on Linux 158

Fixed gateway engine 75

setting up 76

Floating gateway engine 75

setting up 76

Frame type 88

Full installations for Linux 139

G

Gateway engine

configuration 69, 75

definition of 69

fixed, definition of 75

fixed, setting up 76

floating, definition of 75

floating, setting up 76

locator file, definition of 69

Gateway Locator File

definition of 69

Gateway Locator Utility 77, 176

changing workgroup engine 79

locating workgroup engine 79

GLIBC library 140

Group

pervasive_admin TCP/IP support on Windows
85

GTK package 140

H

Handling

- Data Source Names (DSNs) 33

Hosts file

- used to configure Server IP address in Linux 109
- used to configure Server IP address in Windows 101

I

install.log file 38, 45, 177

- for workgroup engine 55

Installation

- before you begin on Linux 138
- BTRBOX Requester 49
- checklist 22
- client 45
- client for Windows 43
- image for Client engine 64
- location for client software 23
- location for clients accessing web applications 25
- location for downloaded file 25
- location for server 23
- location for workgroup 23
- log file location 38, 45, 55, 177
- on Terminal Services 36
- options 15
- over existing Pervasive products 30
- overview 14
- problems during 175
- requirements 14
- requirements for Workgroup engine 68
- review 22
- Samba 138
- scheduling upgrade 22, 26
- server for Windows 35, 38
- server or client first 25
- tips for Server engine on Windows 36
- tips for Workgroup engine on Windows 54
- troubleshooting 176, 177, 189
- troubleshooting checklist 177
- Workgroup 55
- workgroup engine for Windows 53

Installation, special situations

- troubleshooting 179

Installing

- Linux client using RPM 143

- Linux client using TAR 147

- Linux server using RPM 141

- Linux server using TAR 145

- server on Linux 137

- Windows client 44

- Windows server 36

- Windows workgroup 54

- with Active Directory 25

- with Citrix MetaFrame 24

- with Microsoft Cluster Services 24

- with Terminal Services 24

- workgroup for Windows 55

IP address, for server

- configuring using DNS in Linux 108

- configuring using DNS in Windows 101

- configuring using hosts file in Linux 109

- configuring using hosts file in Windows 101

IPX/SPX 103

J

Java Runtime Environment (JRE) 21

JCL

- Pervasive Access Methods 19

JDBC Driver

- Pervasive Access Methods 19

L

Libraries

- Pervasive PSQL libraries needed by certain Linux distributions 140

LIBSTDC++ library 140

License

- installed with Pervasive PSQL 34

license no longer listed

- troubleshooting 189

License, disabled 192

Licenses

- installed with Pervasive PSQL 65

- updating or verifying 64

Linux

- accessing documentation 152

- authentication 162

- before you install 138

- client configuration 149

- client information 164

- client installation 138

- configuration 162
- configuration file 162
- engine status 181
- exclusions 158
- files installed 151
- installation 137
- man (manual) pages 158
- path formats 100, 162
- platform notes 138
- pre-installation notes 138
- readme 153
- server installation 138
- user count licenses 150
- user environment 160

Locations

- for installing client engine 23
- for installing clients accessing web applications 25
- for installing server engine 23
- for installing workgroup engine 23
- no longer used by Pervasive PSQL 62
- of files installed 62

Locator file, see Gateway locator file

M

- Man pages 158
- MicroKernel Database Engine 5
- Microsoft Cluster Services
 - installing on 24
- Migrating
 - configuration settings 32
- Multi-engine systems 133
- Multiple applications
 - configuring for 132
 - for clients 130
- Multiple network cards
 - configuring server engine for 87

N

- NetBIOS
 - not supported by Server engine 84
- Network
 - determining what type 84
 - path formats 95
 - path formats, drive-based 100
 - path formats, Linux 100

- path formats, UNC 95
- removing unused protocols 91
- setting up SPX for Windows server 88
- setting up TCP/IP for Windows server 86, 90
- Network cards
 - configuring server engine for multiple 87
- Network communications
 - settings for clients 94
 - settings for Pervasive PSQL engines 85
 - testing 180
- Network server
 - from terminal services 120

O

- ODBC
 - Administrator 6
 - not supported with DOS requesters 117
- ODBC Behavior 165
- OEM/ANSI 112
- OLE DB
 - Pervasive Access Methods 19
- Optional features
 - ActiveX Interface Controls 18
 - ADO.NET Provider 18
 - Btrieve DOS 18
 - Cobol Schema Executor 20
 - Data Dictionary File Builder 20
 - Documentation 20
 - DTO 19
 - in Pervasive PSQL 18
 - JCL 19
 - JDBC Driver 19
 - OLE DB 19
 - PDAC 19
 - Pervasive Access Methods 18
 - Pervasive Control Center 20
 - Pervasive System Analyzer (PSA) 20
 - Utilities 19
 - Xtreme I/O (Server 32-bit Only) 18
- Options
 - for Complete installation 15
 - for Custom installation 15
 - for installation 15
- Overview
 - for installation 14
 - Pervasive PSQL 2

Pervasive PSQL engines 8

P

Path formats

drive-based 100

Linux 100

network 95

UNC 95

PATH too long error 191

PDAC

Pervasive Access Methods 19

Peer-to-peer configuration

workgroup engine 68, 73

Perl 169

Pervasive Access Methods

ActiveX Interface Controls 18

ADO.NET Provider 18

Btrieve DOS 18

DTO 19

JCL 19

JDBC Driver 19

OLE DB 19

optional feature 18

PDAC 19

Pervasive Control Center

Utilities 20

Pervasive PSQL

about 2

about relational or transactional access 3

about the engines 8

application data files 63

engines 16, 17

engines, status of 181

files location 62

installation 53

license installed 34

licenses installed 65

optional features 18

products 16

program files 63

PVSW\BIN location 62

relational engine 6

server engine overview 8

Software Developer Kit (SDK) 10

transactional engine 4

troubleshooting 189

Pervasive PSQL Account Management on Linux 160

Pervasive PSQL Workgroup engine

overview 8

Pervasive Software Website 194

Pervasive System Analyzer (PSA) 180

Utilities 20

Pervasive_admin

and TCP/IP support on Windows 85

Pervasive_Admin security group

use with Active Directory 122, 123, 126

PHP 166

Platform notes

Linux 138

Windows 36

Ports, communication

changing default 104

Post installation

common questions 62

Pre-installation notes, Linux 138

Previous version

upgrading from 29, 30

Program files

difference between 32-bit and 64-bit 63

in Pervasive PSQL 63

Protocol

determining correct 84

IPX/SPX 103

NetBIOS not supported by server 84

removing unused 91

PVSW\BIN

no longer used 62

R

Readme

for Linux 153

Rebuild utility 34

Relational engine 6

ODBC Interface 6

Release notes

as part of installation 36, 44

Remote

server engine access 133

Remote applications

concurrent with local 132

Removing

unused network protocols 91

- Requesters
 - preferred for Windows platforms 118
 - trace 50
 - types for Windows 50
 - used with Active Directory 122
- Requesters, client
 - concepts 50
- Requesters, DOS 50
- Requirements
 - for installing 14
- Restarting applications
 - after improper program exit 190
- Restricting user access
 - on Terminal Services 26
- Review
 - for installation 22
- Rights
 - administrative authority for Active Directory 122
 - Terminal Services 36
- RPM Version 154
- Running
 - Workgroup engine as a service 81

S

- Samba
 - special notes for installation on 64-bit Linux distributions 140
- Samba installation 138
- Scalable SQL
 - file conversion 34
- Security
 - pervasive_admin support for TCP/IP on Windows 85
 - workgroup engine and 68
- Server Configuration 149
- Server engine
 - 32-bit for Windows 16
 - 64-bit for Windows 16
 - compared to Workgroup 9
 - for Windows 16
 - installation 35, 38
 - installation location 23
 - installed before client 25
 - mixed engine systems 133
 - NetBIOS not supported 84
 - on Linux 137

- Server installation
 - on Linux 138
- Server IP Address
 - configuring using DNS (Linux) 108
 - configuring using DNS (Windows) 101
 - configuring using hosts file in Linux 109
 - configuring using hosts file in Windows 101
- Service
 - running Workgroup engine as 81
 - stopping Workgroup engine as 82
- Services
 - checking status 181
- Services File 104
- Setting up
 - Web-based data access 165
- Settings
 - for configuration 27
- Setup Type
 - Complete 15
 - Custom 15
- Software Developer Kit (SDK) 10
 - environment 11
 - programming languages 11
- SPX
 - frame type 88
 - setting up for Windows server 88
 - support for clients 103
- Status 7012, when creating a new database
 - with Workgroup engine using PCC (Windows Vista) 189
- Status 7224
 - troubleshooting 189
- Status 95
 - after running applications successfully 190
- Status, of database engine 181
- Stopping
 - Workgroup engine as a service 82
- Supported Path Formats for Samba 162

T

- TAR Version 154
- TCP/IP
 - for clients 101, 108
 - setting up for Windows server 86, 90
- Technical Support 194
- Terminal Server

- as network server 120
- Terminal Services 120
 - disabling administrative functions 120
 - installing on 24, 36
 - permissions 36
 - restricting user access 26
 - running the engine with 36
 - use within Active Directory environment 122
- Testing
 - network connectivity 180
- Tips
 - for installing Server engine on Windows 36
 - for installing Workgroup engine on Windows 54
- Trace Requesters 50
- Transaction Durability
 - multiple applications and 131
- Transactional engine 4, 5
- Translate data 110
- Translation
 - data encoding 110
- Troubleshooting 175
 - after installation 62
 - after uninstalling 193
 - communications 180
 - DOS components 190
 - engine and client version conflicts 188
 - installation 189
 - strategies 177
 - tools 176
- Types of installation
 - Complete 15
 - Custom 15

U

- Uninstall
 - Pervasive PSQL Server on Linux 154
- Uninstalling
 - files not removed by 193
 - Pervasive PSQL on Windows 66
- Uninstalling Pervasive PSQL for Linux 154
- Universal Naming Convention *See* Network path formats, UNC.
- Updating
 - licenses 64
- Upgrading
 - from a previous version 29, 30

- installation 22, 26
- migration of existing configuration settings 32
- User count licenses
 - Linux 150
- User environment
 - in Linux 160
- Using Perl and ODBC with Pervasive PSQL 173
- Using Utilities from Users Other than psql 161
- Utilities
 - Cobol Schema Executor 20
 - Data Dictionary File Builder 20
 - Gateway Locator Utility 77, 79, 176
 - optional features 19
 - Pervasive Control Center 20
 - Pervasive PSQL 18
 - Pervasive System Analyzer (PSA) 20
 - Rebuild 34

V

- Verifying
 - database engine is running 149
 - engine status 181
 - installed products with RPM 149
 - licenses 64
- Version
 - determining in files 184
 - engine and client 183
 - troubleshooting conflicts 188

W

- Web applications
 - client installation location 25
- Web Server
 - configuring 165
- Web site
 - Pervasive Software 194
- Win32 DOS Box Support 49
- Windows
 - accessing documentation 64
 - installation error "PATH statement too long" 191
 - pervasive_admin group and TCP/IP support 85
 - platform notes 36
 - types of requesters 50
- Workgroup engine
 - and Client (64-bit) 63
 - applications not using 191

- compared to Server engine 9
- configuration 81
- for Windows 16
- gateway configuration 69, 75
- installation 53, 55
- installation location 23
- installation requirements 68
- installing client with 64
- mixed engine systems 133
- peer-to-peer configuration 68, 73
- running as a service 81, 121
- security and 68
- small client/server configuration 68, 71
- stopping as a service 82

X

- Xtreme I/O (Server 32-bit Only)
 - optional feature 18

