PSQL v12

Getting Started with PSQL



disclaimer

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

trademarks

Btrieve, Client/Server in a Box, and Pervasive are registered trademarks of Actian Corporation.

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

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

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

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

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

© Copyright 2014 Actian Corporation. All rights reserved. Reproduction, photocopying, or transmittal of this publication, or portions of this publication, is prohibited without the express prior written consent of the publisher.

This product includes software developed by Powerdog Industries. © Copyright 1994 Powerdog Industries. All rights reserved. This product includes software developed by KeyWorks Software. © Copyright 2002 KeyWorks Software. All rights reserved. This product includes software developed by DUNDAS SOFTWARE. © Copyright 1997-2000 DUNDAS SOFTWARE LTD., all rights reserved. This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product uses the free unixODBC Driver Manager as written by Peter Harvey (pharvey@codebydesign.com), modified and extended by Nick Gorham (nick@easysoft.com), with local modifications from Actian Corporation. Actian Corporation will donate their code changes to the current maintainer of the unixODBC Driver Manager project, in accordance with the LGPL license agreement of this project. The unixODBC Driver Danager 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 PSQL
December 2014

Contents

About This Manual ix

Who Should Read This Manual x
Conventions xi

1 Welcome to PSQL 1

A Basic Introduction to PSQL v12

About PSQL 2
Competitive Advantages 2
Transactional (Direct Data) Access or Relational Access 2
The PSQL MicroKernel Engine 4
Benefits 4
Features 4
The PSQL Relational Engine 5
Benefits 5
Features 5
About the PSQL Database Editions 6
PSQL Server and Vx Server 6
PSQL Workgroup 6
Feature Comparison 6
PSQL SDK 8
Development Environment 8

2 Preparing to Install PSQL 9

Preparation Needed for PSQL Installation

Installation Requirements 10
Access Rights 10
No Other PSQL Database Engine Installed 10
Upgrading to PSQL v12 10
Operating Requirements 10
Installation Options 11
Complete Installation 11
Custom Installation 11
PSQL Products 12
Server and Vx Server 12
Workgroup 12
Client 12
PSQL Optional Features 13
PSQL Access Methods 13
Utilities 14
Documentation 14
Java Runtime Environment (JRE) 15
Installation Review 16
Quick Checklist 16
Common Preinstallation Questions 16

3 Upgrading Your PSQL Installation for Windows 21

Considerations for Users Upgrading to PSQL v12

Upgrading to PSQL v12 from a Previous Version . . . 22
Considerations When Upgrading to PSQL v12 . . . 22
Common Questions After Upgrading to PSQL . . . 23

4 Installing PSQL Server and Vx Server for Windows . . . 25

Instructions for Installing the PSQL Server and Vx Server on Windows

Before You Install . . . 26
Platform Notes . . . 26
Installation Tips . . . 26
Installing PSQL for Windows . . . 27

5 Installing PSQL Clients for Windows . . . 31

Instructions for Installing the PSQL Client on Windows

Before You Install the Windows Client Engine . . . 32
Installing the PSQL Client for Windows . . . 33
Installing the BTRBOX Requester . . . 36
Win32 DOS Box Support . . . 36
Understanding Client Requesters . . . 37
Types of Windows Requesters . . . 37
Where To Go From Here . . . 37

6 Installing PSQL Workgroup for Windows . . . 39

Instructions for Installing the PSQL Workgroup Engine on Windows

Before You Install the Windows Workgroup Engine . . . 40
Installation Tips . . . 40
Installing the PSQL Workgroup for Windows . . . 41
Authorization of Workgroup Key on Windows with UAC . . . 43

7 After Installing PSQL for Windows . . . 45

Answers to Common Post Installation Questions for PSQL on Windows

Common Questions After Installing PSQL . . . 46
Where To Go From Here . . . 47
Uninstalling PSQL . . . 48

8 Configuring the Workgroup Engine . . . 49

Understanding the Available Workgroup Engine Configurations

Overview . . . 50
Installation Requirements . . . 50
Operating System Security . . . 50
When to Use Workgroup . . . 50
What is a Gateway Engine? . . . 51
Setting Up a Small Client/Server Configuration . . . 52
Setting Up a Peer-to-Peer Configuration . . . 53
Setting Up a Gateway Configuration . . . 54
Floating or Fixed Gateway . . . 54
Working with the Gateway Locator Utility . . . 55
Running the Workgroup Engine as a Service . . . 58
Configuration . . . 58
Stopping the Service . . . 58

9 Configuring Engine Network Communications 61

How to Set Up Network Communication for PSQL Engines

Determining What Kind of Network You Have 62
Database Engine on Windows 62
Engine Network Communication Settings 63
Setting Up TCP/IP Support 64
Setting Up SPX Support 66
Setting Up NetBIOS Support (Workgroup only) 67
Avoiding Unavailable Protocols 68

10 Configuring Network Communications for Clients 69

How to Configure Network Communications for Your PSQL Clients

Client Network Communication Settings 70
Network Path Formats Supported by PSQL Requesters 71
Universal Naming Convention (UNC) Path Formats 71
Drive-based Formats 71
Linux Path Formats 71
IPv6 73
IPv6 Addresses 73
IPv6 and the MicroKernel Engine 75
IPv6 and the Relational Engine 76
IPv6 with Linux 76
Frequently Asked Questions About IPv6 77
Using TCP/IP to Connect to a Windows Server 79
Configuring a Client for the Server IP Address 79
Using SPX to Connect to a Windows Server 81
Configuring PSQL to use SPX 81
Changing the Default Communication Ports 82
Services File 82
Windows FireWalls 82
Using TCP/IP to Connect a Windows Client to a Linux Server 85
Configuring a Client for the Server's IP Address 85
Data Encoding 87
Using the DOS Requester 88
Supported Configurations 88
DOS Box Support on Windows 89
Running DOS Applications on Windows 32-bit Platforms 89

11 Application Configuration Scenarios 91

Common Scenarios for Setting up Your Database Engine

Terminal Services 92
Disabling Administrative Functions 92
Terminal Server as Network Server 92
Workgroup Engine Running as a Service 92
Active Directory Service 93
PSQL Administrative Authority 93
Active Directory Tasks 94
Multiple Client Applications 95
Settings Affected by Multiple Applications 95
Concurrent Local and Remote Applications 96
Using the Server and Workgroup Engines Concurrently 96
Accessing Data on Other Computers 98

12 Installing PSQL Server, Vx Server and Client for Linux 99

Instructions for Installing and Uninstalling PSQL Server, Vx Server and Client on Linux

Before You Install PSQL for Linux 100
Server, Vx Server 100
Client 100
Full Installations 100
Notes for 64-bit Distributions 101
Installing PSQL Using RPM 103
Installing PSQL Server, Vx Server for Linux - RPM 103
Installing PSQL Client for Linux - RPM 104
Installing PSQL Using TAR 106
Installing PSQL Sever, Vx Server for Linux - TAR 106
Installing PSQL Client for Linux - TAR 107
After Installing PSQL for Linux 109
Verifying Installed Products with RPM 109
Verifying Database Engine is Running 109
Server Configuration 109
Client Configuration 109
Licensing 110
Common Questions After Installation 110
Uninstalling PSQL for Linux 113
RPM Version 113
TAR Version 113

13 Using PSQL on Linux 115

Working With the Products on Linux

Finding What You Need 116
Accessing the User Documentation 116
Man Pages 116
PSQL Account Management on Linux 117
After Installation Behavior 117
The User Environment 117
Using Utilities from Users Other than psql 117
Configuration 119
Configuration File 119
Authentication 119
Supported Path Formats for Samba 119
Client Information 121
Authentication to Remote Machines 121
Creating a Client DSN 121
Setting Up Web-based Data Access 122
ODBC Behavior 122
Configuring Web Server 122
PHP 122
Perl 125
Using Perl and ODBC with PSQL 129
Code Snippet for Perl and DBI 129

14 Troubleshooting After Installation 131

How to Proceed When You Encounter Errors After Installation

Troubleshooting Tools . . . 132
Troubleshooting Strategies . . . 133
    Checklist for Problems . . . 133
    Troubleshoot the Problem . . . 133
Logged Messages . . . 134
Configuration for Special Installation Situations . . . 135
Diagnosing Problems with PSQL System Analyzer (PSA) . . . 136
Verifying Database Engine is Running . . . 137
    PSQL Server on Windows . . . 137
    PSQL Workgroup on Windows . . . 137
    PSQL Server on Linux . . . 138
Obtaining File, Client, and Engine Version Number . . . 139
    Determining Client and Engine Version . . . 139
    Determining a File Version . . . 140
Engine and Client Version Conflicts . . . 144
State of Key Is “Failed Validation” or “Disabled” . . . 145
Troubleshooting Common PSQL Issues . . . 146
Issues After Uninstalling PSQL on Windows . . . 148
How to Get Additional Help . . . 149
    Technical Support . . . 149

About This Manual

This manual contains information about installing the PSQL v12 database system. PSQL v12 is a complete database management system, providing the best of both worlds. It combines a MicroKernel Engine designed for high-performance data handling and improved programming productivity with an embeddable and scalable Relational Engine.

This manual also contains information about common installation issues, general network protocol information, and PSQL v12 optional features.

For information on using PSQL utilities, see *PSQL User's Guide*. For information about configuring the PSQL v12 engine, see *Advanced Operations Guide.*

Who Should Read This Manual

This manual provides information for users who install and run PSQL v12. This manual is also useful for system administrators who are responsible for maintaining databases on a network and for those who are using PSQL to develop server applications.

Actian Corporation 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 PSQL website.

Conventions

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

CASE	Commands and reserved words typically appear in uppercase letters. Unless the manual states otherwise, you can enter these items using uppercase, lowercase, or both. For example, you can type `MYPROG`, `myprog`, or `MYprog`.
Bold	Words appearing in bold include the following: menu names, dialog box names, commands, options, buttons, statements, and so forth.
`Monospaced font`	Monospaced font is reserved for words you enter, such as command syntax.
[ ]	Square brackets enclose optional information, as in [*log_name*]. If information is not enclosed in square brackets, it is required.
|	A vertical bar indicates a choice of information to enter, as in [*file name* | @*file name*].
< >	Angle brackets enclose multiple choices for a required item, as in `/D=<5|6|7>`.
variable	Words appearing in italics are variables that you must replace with appropriate values, as in *file name*.
…	An ellipsis following information indicates you can repeat the information more than one time, as in [*parameter* …].
::=	The symbol ::= means one item is defined in terms of another. For example, a::=b means the item *a* is defined in terms of *b*.

chapter

1

Welcome to PSQL

A Basic Introduction to PSQL v12

Thank you for purchasing 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 PSQL
- The PSQL MicroKernel Engine
- The PSQL Relational Engine
- About the PSQL Database Editions
- PSQL SDK

About PSQL

PSQL is a reliable, low-maintenance, high-performance database management system (DBMS). Thousands of companies around the world license 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 PSQL.

No matter whether you received 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

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 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. 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.* 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, PSQL does not lock you in to a single platform. 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, PSQL is there for you.
- *Big database features at a small price.* 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 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 Btrieve API 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.

Transactional (Direct Data) Access or Relational Access

PSQL offers an architecture that is totally unique in the database management market. Our product allows you to access the exact same data through the MicroKernel Engine and through the Relational Engine.

The transactional database engine, called the MicroKernel Engine, interacts directly with the data and does not require fixed data schema to access the data. It uses key-value store to store and access the data.

Calls to the MicroKernel Engine are made programmatically with Btrieve API rather than through the use of a query language; therefore, PSQL does not have to parse the request. This places the MicroKernel Engine in the category of NoSQL databases. Low-level API calls and memory caching of data reduce the time required to manipulate data. See The PSQL MicroKernel Engine.

The second database engine, the Relational Engine, operates in a manner similar to other relational database engines, that is, through the support of Structured Query Language queries. The Relational Engine parses SQL queries and sends them to the MicroKernel Engine to run. See The PSQL Relational Engine.

The PSQL MicroKernel Engine

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

Benefits

PSQL's MicroKernel Engine through the Btrieve API 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 PSQL. Many application developers choose PSQL for its speed, data integrity, easy scalability, and low maintenance costs.

- *Speed.* The MicroKernel Engine is capable of sub-second response rates, even when building multi-gigabyte databases for hundreds of users. The engine 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 Engine 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. PSQL provides easy scalability from workstation to large client/server environments.
- *Low Cost.* The low support costs experienced by PSQL developers translate into low maintenance costs realized by PSQL application end users. PSQL eliminates the need for sustained database administration through automatic data recovery functions and easy-to-use utilities.

Features

The MicroKernel Engine offers many features, including the following:

- Direct interaction with the data that does not require fixed data schema to access the data. It uses key-value store to store and access the data.
- Calls to the engine are made programmatically with Btrieve API rather than through the use of a query language; therefore, PSQL does not have to parse the request.
- Low-level API calls and memory caching of data reduce the time required to manipulate data.
- Access to databases distributed across multiple engines.
- Robust transactions for both single-server systems and distributed, multi-server systems.
- The engine operates in complete database transactions and guarantees full ACID (Atomicity, Consistency, Isolation, Durability).
- Records are stored in files which are roughly equivalent to the tables of a relational database engine. It supports multiple keys on a record and therefore multiple indexes in the file.

The PSQL Relational Engine

PSQL's Relational Engine offers easy installation, uncomplicated maintenance, and high levels of performance and reliability.

Benefits

Many relational database application developers choose 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.* 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 PSQL ODBC driver calls the database engine directly, without translating ODBC calls to a proprietary relational API.
- *Scalability.* PSQL allows you to scale applications from single-user to large client/server environments without changing the application or the database.
- *Maintenance-free Operation.* PSQL is simple to install and use. It requires no extensive performance setup or ongoing tuning by a database administrator.
- *Small Memory Footprint.* PSQL has a small footprint, requiring only a small amount of memory.

Features

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

- 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 and ADO.NET 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 PSQL Database Editions

This section provides some basic information about the PSQL Server and Workgroup editions. For a discussion of PSQL architecture, see The PSQL Component Architecture in *Advanced Operations Guide*.

PSQL Server and Vx Server

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

PSQL Server and Vx Server differ in that Vx Server supports capacity-based licenses for service bureau, software-as-a-service, or other multiplexed environments, while PSQL Server supports licenses for end-user client/server applications. No additional license is required for use with hypervisor features such as live migration, failover, fault tolerance (FT), high availability (HA), and disaster recovery.

PSQL Workgroup

PSQL Workgroup is designed to support single-user or small workgroup installations.

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.

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 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 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 PSQL engine. In a Workgroup environment, every machine that will access PSQL data should have a Workgroup engine installed.

Feature Comparison

All PSQL 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 a variety of access methods (see PSQL SDK)	✓	✓
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.	✓	✓

Table 1 Comparison of Server and Workgroup Features

Feature	Server	Workgroup
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	does not apply	✓
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	✓	

PSQL SDK

The 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. PSQL offers a native ODBC driver.
- 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 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. 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 PSQL Web site is an online resource that gives you access to the latest component downloads and code samples.

Development Environment

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 PSQL engine with your application is easy with a Derivative Software License.

chapter

2

Preparing to Install PSQL

Preparation Needed for PSQL Installation

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

This chapter contains the following sections:

- Installation Requirements
- Installation Options
- PSQL Products
- PSQL 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 PSQL installation. The following overview is intended to accompany the software and hardware requirements listed on the PSQL web site for PSQL.

Access Rights

You must have full administrator-level rights on the machine where you install PSQL Server, PSQL Vx Server, or PSQL Workgroup.

No Other PSQL Database Engine Installed

You cannot install an edition of PSQL v12 on the same machine with another edition of PSQL v12. You must uninstall the other PSQL edition first. For example, PSQL Workgroup and PSQL Server cannot be installed on a machine at the same time.

Upgrading to PSQL v12

If upgrading from a previous version of PSQL, you must upgrade to the same edition of PSQL v12. For example, if you have a prior version of PSQL Workgroup, you must upgrade to PSQL Workgroup and not to PSQL Server. To change editions, you must first uninstall the prior version edition. The exception is that you may upgrade from PSQL Server to PSQL Vx Server, or from PSQL Vx Server to PSQL Server.

Operating Requirements

In a VM environment that supports live migration, an authorized PSQL instance—along with your application—can move from one host to another host with no impact to your end users. (Contrast this with a traditional client/server environment in which the physical server normally remains consistent. Clients may connect and disconnect, but they communicate with a dedicated server.)

Consistent Hostname

An authorized instance of PSQL can be moved to a different physical machine, or converted into a VM, or migrated to a different VM host. However the VM hostname must remain the same. If the hostname must change, deauthorize the key for PSQL before performing the move, conversion or migration. Each instance of PSQL, including those that are cloned or copied for VMs, requires its own permanent key even if the hostname is the same. See also, License Models in *PSQL User's Guide*.

Installation Options

On Windows operating systems, PSQL offers Complete and Custom installation options. On Linux distributions, each product has its own separate installation RPM or TAR file, which do not provide custom installation options.

Complete Installation

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

Custom Installation

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

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

PSQL Products

PSQL is available in a Server, Vx Server, Workgroup, and Client installation. This section lists PSQL products and the base components for each product installation.

Server and Vx Server

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

Workgroup

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

Client

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

PSQL Optional Features

During a Custom install, PSQL features may be excluded with the 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 PSQL product is still installed by default.

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

PSQL Access Methods

PSQL Access Methods include the 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 PSQL 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 PSQL Direct Access Components (PDAC) includes a set of Visual Component Library (VCL) components that allow direct transactional and relational access to PSQL 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 PSQL Web site.

Utilities

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

- Function Executor
- Btrieve Maintenance
- Rebuild
- Query Plan Viewer
- License Administrator
- Gateway Locator (Workgroup Engine only)

Cobol Schema Executor

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

Data Dictionary File Builder

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

PSQL Control Center

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

PSQL System Analyzer

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

Documentation

The PSQL Engine and SDK user documentation is integrated into 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). The Engine and SDK user documentation is also available online at the PSQL Web site.



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

Java Runtime Environment (JRE)

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

- PCC
- DDF Builder
- Core utilities
- Documentation

PSQL features use the local version of the JRE installed by PSQL. PSQL installs a local version of the JRE only if the minimum required JRE version is not installed. If the minimum required JRE version already exists, PSQL features will use those files instead of installing another local copy of that JRE.



Note The installation of a local version of the JRE is for use only by the PSQL features listed above. The local version of the JRE does *not* affect the requirements for developing Java applications using the PSQL access methods Java Class Libraries (JCL) or JDBC. Those requirements, such as components obtained from java.sun.com, are discussed in the 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 PSQL. Each of these items should be met prior to beginning the install process.

- ❑ Your system hardware meets the minimum requirements to install PSQL.
- ❑ Your operating system and network environment is supported by PSQL.
- ❑ You have full administrator-level rights on the system where you plan to install 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_psql.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.
- ❑ If you are using a proxy server, you need to configure it to allow authorization of PSQL. Configure the proxy server before you install PSQL, or omit product authorization during installation and authorize the product after configuring the proxy server. See Authorization Access Through A Proxy Server in *PSQL User's Guide*.
- ❑ You have a good understanding of User Access Control (UAC), as well as the differences and limitations of Standard Users versus Administrators and also have the appropriate permissions to install on the target system.
- ❑ Your application vendor supports the 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 PSQL engine version and mode that you want to install.

Common Preinstallation Questions

This section contains some of the most common questions asked prior to installing 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 PSQL Server or Vx 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 PSQL Client is installed with every engine. So, if you have a PSQL engine installation, you can use your machine to connect to other remote engines as a client.

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

Where do I install the PSQL Workgroup?

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 PSQL Client 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 PSQL in a Microsoft Cluster Services environment?

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

Running virtual machines (VMs) within an MSCS environment is typically quite involved. Consequently, running PSQL on VMs within an MSCS environment is more complicated than when not using VMs. Refer to the documentation from the various hypervisor vendors for running VMs within MSCS.

How do I install PSQL in a Microsoft Terminal Services or XenApp environment?

If you plan to install PSQL to a Microsoft Terminal Server or XenApp 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.

If you install the Workgroup or the Client Cache Engine on a Terminal Services Environment, install the Engine to run as a service. For Workgroup, the default installation for a fresh install is to run as a service. For Client, the default installation is to run as an application, so you need to specify "run as a service" instead. If installed as a service, the Client Cache Engine service is set by default to auto-start. However, even though the Cache Engine service is running, an application does not use the Cache Engine unless the configuration setting for it is turned "on" (it is "off" by default). See Use Cache Engine in *Advanced Operations Guide*.

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 PSQL Web site for the list of Terminal Server Environments supported by PSQL Server, Vx Server, Workgroup and Client.



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 MicroKernel Engine.

If it is desirable to have multiple local users accessing a local engine, ensure that the Workgroup or Client Cache Engine runs as a service. See information above about installing Workgroup Engine or Cache Engine in a Terminal Services environment.

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

The installation of the 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 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 PSQL on a server that is not a domain controller.

Where do I install my 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 PSQL Clients that are of a different version than that of the database engine?

Actian Corporation 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. Actian Corporation does not guarantee that newer versions of Clients will function correctly with older versions of the engine. Therefore, Actian Corporation recommends that you avoid the use of newer version Clients with an older engine.

Does it matter where I download the PSQL v12 install file?

Yes, it does matter. If you are installing a downloaded version of 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 PSQL v12?

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 PSQL configuration settings, creating DSNs, and using the Monitor utility?

PSQL clients running within Terminal Services client sessions can perform PSQL administrative functions by default. For example, a user with such a client can change configuration settings for 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 PSQL that need to be adjusted if your configuration includes certain situations. 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.

chapter 3

Upgrading Your PSQL Installation for Windows

Considerations for Users Upgrading to PSQL v12

This chapter contains information about upgrading to PSQL v12 from a supported previous version. The configuration settings that are migrated during an upgrade from a previous version to PSQL v12 are also detailed in this chapter.

The following sections are included in this chapter:

- Upgrading to PSQL v12 from a Previous Version
- Common Questions After Upgrading to PSQL

Throughout this document, when an explicit version number is not specified (for example: Pervasive.SQL 7, Pervasive.SQL 2000, or PSQL v12), all versions are included.

Upgrading to PSQL v12 from a Previous Version

If you are upgrading a previous version of PSQL to PSQL v12, follow the same procedure as you would if you were installing for the first time. See Chapter 4, Installing PSQL Server and Vx Server for Windows for detailed installation procedures.

If you are upgrading a version of PSQL *prior* to PSQL v10, PSQL v12 archives the previous version before removing it.

If you are upgrading from much 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.

An upgrade migrates configuration settings. For a version of PSQL *prior* to PSQL v10, only the configuration settings defined in ptksetup.ini are migrated. For PSQL v10 and later, an upgrade disregards any settings defined in ptksetup.ini and simply migrates all configuration settings

Note that keys from previous PSQL products are no longer valid if you upgrade to a new major release of PSQL. (For example, an upgrade from PSQL v10 to PSQL v11.) You must have a key for the new major release to authorize PSQL after upgrading.

Considerations When Upgrading to PSQL v12

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

- *PSQL Applications* - Be aware of what applications you have currently using previous versions of Btrieve or 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 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 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 PSQL Web site.

How to Handle Data Source Names (DSNs)

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

Table 2 DSN Considerations After a PSQL Upgrade

Scenario	Discussion
You have existing DSNs created with a version of PSQL **prior to** Pervasive.SQL 2000*i* SP4	You must delete all existing DSNs before you upgrade. Once you have upgraded, re-create the DSNs to access the existing databases. New DSNs should connect to a named database, **not** to an Engine DSN because Engine DSNs are deprecated. See ODBC Connection Strings in *ODBC Guide*.
You have existing DSNs created with Pervasive.SQL 2000*i* SP4 or a later version of PSQL	You should be able to access your databases by connecting to the existing DSNs. 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. If you want to port your 32-bit application to 64-bit, then make the following changes. • If the application uses DSN-less connections that connect using “Pervasive ODBC Client Interface,” change the connection string to “PSQL 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.) See ODBC Connection Strings in *ODBC Guide*.
You do not have any PSQL DSNs defined	For details about creating DSNs, see DSN Setup and Connection Strings in *ODBC Guide*. 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 “PSQL ODBC Interface” to connect to a named database.

How Do I Convert My Files From Previous PSQL 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 PSQL v12 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 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 *PSQL User's Guide* for more information on the License Administrator utility.

chapter

4

Installing PSQL Server and Vx Server for Windows

Instructions for Installing the PSQL Server and Vx Server on Windows

This chapter contains procedures for installing and running PSQL v12. The chapter contains the following sections:

- Before You Install
- Installing PSQL for Windows

Before You Install

Before installing PSQL v12, begin by reviewing the following documents:

- *Chapter 2, Preparing to Install 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_psql.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 PSQL for Windows, you must have full administrator-level rights on the machine where you will install PSQL.

Installing the Engine on Terminal Server

To install 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 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 PSQL v12 for the first time on a system, the installer checks whether all of the needed system files meet the minimum requirements. In some cases, these files are locked by the operating system and a restart is required before the installer can continue.



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

- If you have any trouble with the following installation, see Chapter 14, Troubleshooting After Installation.

Installing PSQL for Windows

You must install PSQL 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 PSQL from its media using the default interactive installation.

➢ To install PSQL for Windows

1. Launch the installation program from your Windows machine.
   a. Insert the PSQL CD into the CD-ROM drove of your Windows server, or run the self-extracting executable.
   b. If the installation does not start automatically, run the autorun executable from the operating system: drive:`\autorun\autorun` where *`drive`* is the drive letter of your CD-ROM device.

   The installation selection dialog displays.
2. Click Server installation, which will install the components appropriate to your 32- or 64-bit Windows system.

   The installer begins its initial preparation, after which the Welcome screen appears.
3. If prompted, close or uninstall any running applications that may interfere with the 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 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 PSQL v12 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 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 PSQL optional features 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
    - PSQL Control Center
    - Documentation
    - Data Definition File Builder
    - Notification Viewer
  - Other Utilities
    - Cobol Schema Exec
    - PSQL System Analyzer

10 Click Next to continue.

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 *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 *PSQL User's Guide*.)

13 Enter your license key and click the button to authorize 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 *PSQL User's Guide*.)

14 A message box displays 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 restart your system, please do so in order to ensure proper operation of your PSQL v12 product.



Note The installation program modifies certain environment variables. On Windows systems, environment variables can be viewed in Advanced System Settings on the Advanced tab.

chapter

5

Installing PSQL Clients for Windows

Instructions for Installing the PSQL Client on Windows

This chapter contains the following topics:

- Before You Install the Windows Client Engine
- Installing the 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 PSQL. If you have not already, review the following documents before installing PSQL client requesters:

- *Chapter 2, Preparing to Install 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_psql.htm on the distribution media and contain late-breaking news that could not be included in the user documentation.

Installing the PSQL Client for Windows

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

The PSQL Client is installed by default with PSQL Server, PSQL Vx Server, and PSQL Workgroup.

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

The steps in this section explain how to install PSQL Client from its media using the default interactive installation.

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

➢ To install PSQL Client for Windows

1. Launch the installation selection program from your Windows machine.
   a. Insert the PSQL product CD in the CD-ROM drive of your Windows server.
   b. If the installation does not start automatically, run the autorun executable from the operating system: `drive:\autorun\autorun` where *drive* is the drive letter of your CD-ROM device.

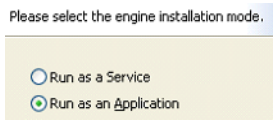
   The installation selection dialog appears.
2. Click Client, which will install the components appropriate for your 32- or 64-bit Windows system.

   The installer begins its initial preparation, after which the Welcome screen appears.
3. If prompted, close or uninstall any running applications that may interfere with the 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 PSQL installation if you ignore programs that may interfere.

4. At the Welcome screen, click Next.
5. Select the installation mode: Run as an Application (the default) or Run as a Service.

Figure 1 Client Cache Engine Installation Mode



Select Run as an Application if:

You plan to enable the use of Client Cache Engine and want the Client Cache Engine to auto-load as needed. After the Client Cache Engine starts, a tray icon appears, allowing you to control the Client Cache Engine from the Windows task bar.

Select Run as a Service if:

- You run the Client on a machine within a Terminal Services environment or in a multi-user environment where fast-user switching is used.
- You plan to enable the use of Client Cache Engine and want it to load when the machine starts.



Note Running the Client as a service requires the Log On as Service privilege. If you select to run the Client as a service under a user account other than the default Local System account, ensure that you modify the Log On Properties for the Service using the Windows Control Panel. If installed as a service, the Client cache service is set by default to auto-start. However, even though the Client cache service is running, an application does not use the Client cache unless the configuration setting for it is turned "on." See the following.

The Client Cache Engine is a specialized version of the MicroKernel Engine that caches data for reading purposes and runs as a separate process. For simplicity, Client Cache Engine is often referred to as the "Client cache."

By default, whether installed as a service or as an application, the use of the Client Cache Engine is disabled (turned "off"). "Off" means that nothing is cached on the Client side. READ requests from an application retrieve data from the remote database Engine. "On" means that the Client Cache Engine acts as an intermediary between the Client and the remote database Engine to cache data.

See Use Cache Engine in *Advanced Operations Guide*.

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 PSQL v12 components using the default options and locations.

- If you choose a Complete install, click Next and continue with step 12.

The Custom setup (recommended for advanced users) allows you to specify the installation location.

- 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 Select the components and associated subfeatures you want to exclude from the installation and click Next. All of the 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
    - PSQL Control Center
    - Documentation
    - Data Definition File Builder
  - Other Utilities
    - Cobol Schema Exec
    - PSQL System Analyzer

11 Click Next to continue.

12 Click Install to begin installation.

13 Once the installation is complete, the final dialog of the Installation Wizard displays. Click Finish.

If you are prompted to restart your system, please do so to ensure proper operation of your PSQL 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

PSQL v12 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 PSQL v12 engine installation. In the case of a Custom installation, you must select the Btrieve DOS optional feature in the PSQL 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 PSQL v12 Workgroup engine, or a remote PSQL v12 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 PSQL Server. A piece of software called a client requester, or requester for short, is required to access database files from a PSQL database server. Your application's PSQL calls go through the requester, which sends them to the 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 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

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 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 PSQL 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 PSQL solve most troubleshooting issues. For example, you would run the network connectivity tests in PSQL System Analyzer to verify network connectivity. Also at your disposal is the Knowledge Base at the PSQL Web site, 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 PSQL requesters.

chapter

6

Installing PSQL Workgroup for Windows

Instructions for Installing the PSQL Workgroup Engine on Windows

This chapter contains procedures for installing the PSQL v12 Workgroup engine. The chapter contains the following sections:

- Before You Install the Windows Workgroup Engine
- Installing the PSQL Workgroup for Windows

Before You Install the Windows Workgroup Engine

Before installing PSQL v12 Workgroup, begin by reviewing the following documents for important information:

- *Chapter 2, Preparing to Install 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_psql.htm on the distribution media and contain late-breaking news that could not be included in the user documentation.

Installation Tips

- When installing PSQL v12 for the first time on a system, the installer checks whether all of the needed system files meet the minimum requirements. In some cases, these files are locked by the operating system and a restart is required before the installer can continue. Click Yes to restart the system. The installer is then automatically restarted.
- It is strongly recommended that you restart your system if you encounter this message. If you do not restart your system, the installer encounters failures during engine and utilities configuration.
- If you have any trouble with the following installation, see Chapter 14, Troubleshooting After Installation.

Installing the 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 PSQL Workgroup from its media using the default interactive installation.

➤ To install PSQL Workgroup for Windows

1. Launch the installation program from your Windows workstation:
   a. Insert the PSQL v12 Workgroup CD in the CD-ROM drive of your Windows workstation.
   b. If the installation does not start automatically, run the autorun executable from the operating system: `drive:\autorun\autorun` where *`drive`* is the drive letter of your CD-ROM device.

   The installation selection dialog displays.
2. Click Workgroup installation.

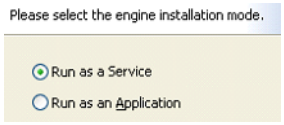
   The installation program begins its initial preparation. After the preparation completes, the Welcome screen appears.
3. If prompted, close or uninstall any running applications that may interfere with the PSQL installation.



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 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 a Service or Run as an Application.

Figure 2 Workgroup Engine Installation Mode



By default for a fresh installation, PSQL Workgroup installs to run as a service. For an upgrade installation, PSQL Workgroup is installed to run in the same manner as the previous installed version. That is, if it was already installed to run as an application, the upgrade installation installs it to run as an application, and vice versa.

Select Run as a Service if:

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

Select Run as an Application if:

The data files only reside on a remote machine, are accessed via drive mappings, and no database Engine is running on the remote machine.

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

The Complete setup (recommended for most users) installs all the PSQL v12 components using the default options and locations.

- If you choose a Complete install, click Next and continue with step 12.

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

11 Click Next to continue.

12 Click Install to begin installation.

13 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 14. (If you have no Internet connection, click Next then click Finish. See Alternative Authorization Tasks in *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 *PSQL User's Guide*.) See also Authorization of Workgroup Key on Windows with UAC.

14 Enter your license key and click the button to authorize 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 *PSQL User's Guide*.)

15 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 14, ensuring that you enter a valid license key.

16 Register your product (recommended) as explained on the Registration page that displays, then close the Registration page.

If you are prompted to restart your system, please do so in order to ensure proper operation of your PSQL v12 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 Windows with UAC

You can encounter difficulty authorizing a permanent key for PSQL Workgroup on Windows with User Account Control (UAC) if the following conditions are all true:

- 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 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 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\Actian\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 operating system Start menu or Apps screen (or use the command line interface licensing utility if you prefer).
5. Type, or paste, the permanent key for PSQL Workgroup in the Key field, then click the button to authorize 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.

chapter

7

After Installing PSQL for Windows

Answers to Common Post Installation Questions for PSQL on Windows

The chapter contains the following sections:

- Common Questions After Installing PSQL
- Uninstalling PSQL

Common Questions After Installing PSQL

This section contains information that you should read after running the installation program. The questions apply to all of the database engine products unless specific products are mentioned. 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 PSQL Web site.

What happened to PVSW\BIN on Windows platforms?

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

Where are the PSQL files installed?

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

Table 3 PSQL Default Windows Installation Locations

Platform	File Types	Default Installation Location
Windows (64-bit)	PSQL Application Data	<*drive:*>\ProgramData\Actian\PSQL\
	Program Files (64-bit)	<*drive:*>\Program Files\Actian\PSQL\
	Program Files (32-bit)	<*drive:*>\Program Files (x86)\Actian\PSQL\
Windows (32-bit)	PSQL Application Data	<*drive:*>\ProgramData\Actian\PSQL\
	Program Files	<*drive:*>\Program Files\Actian\PSQL\
Note: The DOS Requester files are installed by default only on 32-bit Windows platforms at <*drive:*>\%WINDIR%\SYSTEM32\		

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- and 64-bit Program Files?

Microsoft guidelines recommend that 64-bit components are installed in a separate location from 32-bit components. 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. PSQL 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?

The Workgroup installer installs all components appropriate to your 32- or 64-bit Windows system. No separate installation is required for PSQL v12.

What happened to the Client install image?

Previous versions of PSQL contained an image used for installing the client software needed to access the PSQL database. This image is no longer needed, since PSQL now installs the Client components with the Server and the Workgroup editions. Use the Client installation packages 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.

How Do I Read the Online Documentation?

The viewer for the documentation library is integrated into 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 PSQL installation media in the Books directory.

How Do I Verify or Update My User License?

Licenses from previous versions of PSQL are not migrated or transferable to PSQL v12. You must have a license applicable for PSQL v12, unless you choose to install using the trial version of the product.

The License Administrator utility allows you to view and adjust the keys used to license the PSQL products. For example, you can verify which keys have been authorized, check the state of a key, and check the number of authorizations remaining for a key. For a detailed discussion of the License Administrator utility, and a list of the tasks that you can perform, see License Administration in *PSQL User's Guide*.

What User License Was Installed with 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 *PSQL User's Guide* for more information on the License Administrator utility.

Where To Go From Here

If you had trouble during installation, or have questions regarding troubleshooting, see Troubleshooting After Installation.

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

Uninstalling PSQL

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

The uninstall program does not remove the following:

- PSQL keys.
- Databases that you create under the PSQL installation directories.
- DSNs and database names associated with those databases.
- Databases in locations other than the installation directory for PSQL installation directory.
- Other products related to PSQL such as AuditMaster, Backup Agent, or DataExchange. Such programs have their own uninstall programs. Refer to the documentation for each product.

➤ To uninstall PSQL

1. Use a PSQL license utility (GUI or CLI) to deauthorize the key that was used to authorize the product. This allows you to use the key for a different installation if you so choose. (See To Deauthorize a Key in *PSQL User's Guide*, for example.)
2. In the Windows Control Panel, select Programs and Features.
3. Select the installed PSQL product from the list.
4. Click Change then Next.
5. Click Remove then Next and follow any prompts during the uninstall.

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



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

Restart your system if prompted to do so.

chapter

8

Configuring the Workgroup Engine

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. A Workgroup Engine starts by default if installed as a server. By default, Workgroup Engine is installed to run as a service for a fresh install. If the Workgroup Engine was installed as an application, ensure that the application starts if the computer is restarted. 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 ~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 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. A Workgroup Engine starts by default if installed as a server. By default, Workgroup Engine is installed to run as a service for a fresh install. If the Workgroup Engine was installed as an application, ensure that the application starts if the computer is restarted. 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 Access Gateway Locator from the operating system Start menu or Apps screen.

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.

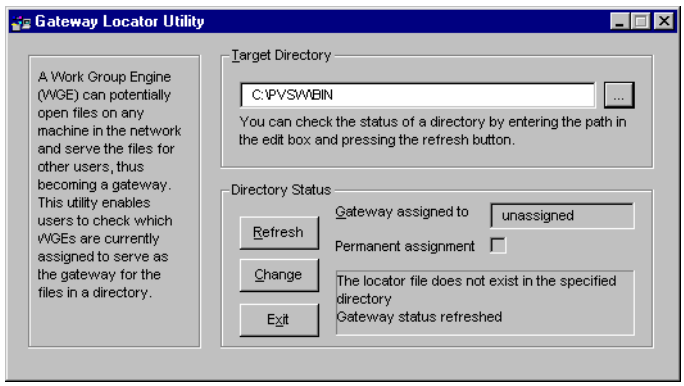
This utility enables users to determine or change the Workgroup Engine which is being used as the gateway for the data files in a particular directory. The Gateway Locator utility is used only with PSQL v12 Workgroup Engine.

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 Access Gateway Locator from the operating system Start menu or Apps screen.

Figure 3 Gateway Locator Main Dialog Box



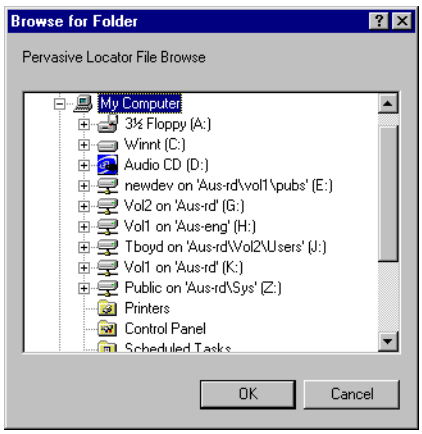


Note The Gateway Locator can be used to set the gateway for any data directory. Data directory locations are not stored with the tool. Consequently, you must always set the directory path before you click Change.

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



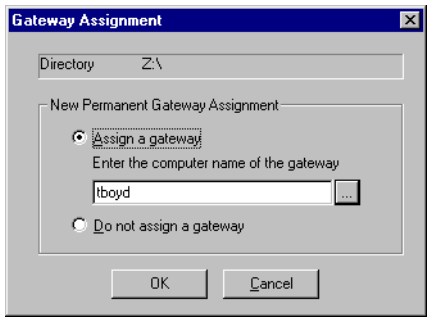
➤ Locating the Gateway Workgroup Engine

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

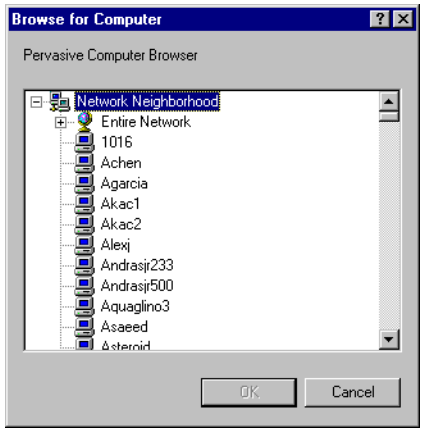
Once the target directory is selected, click **Change** to choose the Workgroup engine which you wish to serve as Gateway for a particular directory (this button is disabled if the locator file for that directory is locked.) The following dialog box appears:

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 a service for a fresh install. 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.

During installation, you can select to run the Workgroup Engine as a console application. Or, if you upgraded your Workgroup installation and you had the previous version installed to run as an application, the upgrade installation is also set to run as an application. In either case, if you decide that you would rather run the Workgroup Engine as a service, you change how it is run.

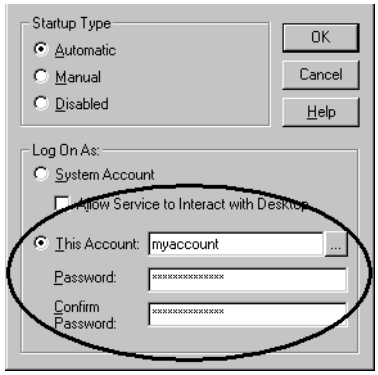
Configuration

To configure your Workgroup Engine to run as a Service as opposed to an application, you must reinstall the PSQL v12 Workgroup Engine. Choose Run As Service on the Engine Settings dialog box during install. See Installing PSQL Workgroup for Windows.

Note 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 PSQL databases on another machine. "System Account" has no rights on other machines to PSQL databases.

If you want the PSQL 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 PSQL Workgroup available locally on the other machine that can be used to access the databases instead. Access the properties for the PSQL 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 (but do not permanently remove the service), then simply restart 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.

chapter

9

Configuring Engine Network Communications

How to Set Up Network Communication for PSQL Engines

This chapter reviews the types of network communication protocols supported with 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 PSQL MicroKernel Engine (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 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 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. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the 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. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the PCC 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 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 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 PSQL MicroKernel Engine. Applications that use only the MicroKernel Engine do not require name resolution with SPX.

1. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the PCC 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 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. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the 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. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the PCC 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.

chapter

10

Configuring Network Communications for Clients

How to Configure Network Communications for Your PSQL Clients

To access network files from a machine using a PSQL application, you must use the appropriate client requester at that machine. Your application's 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 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 PSQL Requesters
- IPv6
- 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 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 PSQL Requesters

When using your Requester, you connect to the PSQL 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 DTI applications.

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 *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 Addresses.



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.

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 PSQL v12, see Using PSQL on Linux.

IPv6

A PSQL Client using any of the access methods supported by PSQL connects using IPv6 to an IPv6 host running the PSQL database engine the same way as it does for IPv4. That is, the client specifies a server and connects through one of the supported access methods. The server can be either the name or IP address of the machine running PSQL Server or Workgroup.

All of the PSQL access methods except for ActiveX support IPv6 connections. Some access methods are Windows only, but can be used with a client/server combination of Windows and Linux.

This section addresses the following aspects of IPv6:

- IPv6 Addresses
- IPv6 and the MicroKernel Engine
- IPv6 and the Relational Engine
- IPv6 with Linux
- Frequently Asked Questions About IPv6

IPv6 Addresses

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. Within this framework, several forms of IPv6 names are possible.

Recommendations for Numeric IPv6 Addresses

In general, numeric IPv6 addresses are more complex and difficult to manipulate than IPv4 addresses. Numeric Link Local addresses are notably problematic, especially with regard to Zone IDs.

For that reason, Actian Corporation recommends the use of host names through DNS servers, LLMNR, host files, or other means of address resolution, particularly for production environments. PSQL support for numeric IPv6 addresses is intended mainly for development environments. Because many contexts require square brackets for IPv6 numeric addresses, when in doubt, add them if numeric addresses are necessary.

Unicast Addresses

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

Table 4 IPv6 Unicast Address Formats Supported by 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.

Table 4 IPv6 Unicast Address Formats Supported by PSQL

Unicast Address Format	Description
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 IPv6 and the MicroKernel Engine.

IPv6 Address Modifiers

IPv6 includes address modifiers that can act as shortcuts or can specify the destination in more detail. 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 Zone ID or interface of a destination node. On Windows, a Zone ID is an integer that specifies the zone of the destination for IPv6 traffic. Zone IDs are primarily used with Link Local addresses to disambiguate those addresses. Linux distributions support only the text form of a Zone ID in a link local address. For example, you must use the “eth0” form of a Zone ID: eth0 fe80:0:0:0:713e:a426:d167:37ab. See also Limitations and Conditions. See IPv6 and the MicroKernel Engine.

Address Presentations

PSQL supports IPv6-literal.net names on Windows and bracketed IPv6 addresses on Windows and Linux.

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

Note Linux distributions do not support IPv6-literal.net names.

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 Uniform Naming Convention (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]

Note Linux distributions do not natively support UNC paths and UNC-safe addresses. However, the PSQL interfaces that accept a file path, a numeric address, or both as input support both UNC paths and UNC-safe addresses on Linux for compatibility with Windows.

IPv6 and the MicroKernel Engine

The following table lists the restrictions on the use of IPv6 with the MicroKernel Engine.

Table 5 IPv6 Restrictions for MicroKernel Engine

Restriction	Discussion
The PSQL Server Engine in an IPv6-only environment	The Server Engine is supported in an IPv6-only environment on Windows Vista and later operating systems, and in IPv6-only environments on the most current Linux distributions.
UNC paths do not allow certain special characters, such as colons, that are part of IPv6 addresses	Avoid raw IPv6 addresses. Wherever possible, use host names. See Address Presentations and Recommendations for Numeric IPv6 Addresses.
Square brackets are required for raw IPv6 addresses when the address is used in a URI or UNC	Raw IPv6 addresses, abbreviated or not, must be enclosed by square brackets if the address is used in a URI or UNC. Examples: • btrv://czjones@[2001:b1::23]/demodata • btrv://abanderas@[2001:12:34:56:78:90:12:23]/demodata • \\[2001:12:34:56:78:90:12:23]\acctsvr1\Domestic\file.mkd Failure to bracket the IPv6 address results in status code 3014 or 3103 for Btrieve calls using a URI, or status code 11, 94, or 170 for Btrieve calls using a UNC.

Table 5 IPv6 Restrictions for MicroKernel Engine

Restriction	Discussion
In a URI, if you include a ZoneID to a server address, the "%" ZoneID character must be escaped with "%25"	If you use a btrv:// connection with an IPv6 address, you must escape the ZoneID for the host name. Zone IDs are usually required with IPv6 Link Local numeric addresses. Example: A UNC-safe addresses like btrv://@[fe80::20c:29ff:fe67:2ee4%4] must be changed to btrv://@[fe80::20c:29ff:fe67:2ee4**%25**4]
Operating system limitations	See Limitations and Conditions.

IPv6 and the Relational Engine

The inclusion of a port number with an IP address is necessary only when you need to override the default port. Generally, a port number can be appended to an IP address using either a colon (:*port number*) or a dot (.*port number*).

UNC-safe names (see Address Presentations) support the appending of a port number.

IPv6 with Linux

All of the previous content about IPv6 applies to Linux distributions, with specifics noted. This subsection discusses some additional aspects of IPv6 when used with Linux.

Suggested Distributions

Full IPv6 support is a relatively recent feature of operating systems. For this reason, Actian Corporation recommends that you use the most current Linux distribution. Refer to the PSQL Web site for a list of the Linux distributions supported by PSQL and use the most current distribution of the latest version listed.

Limitations and Conditions

Note the following limitations and conditions when you use IPv6 with Linux.

- Linux ping commands are specific for IPv4 (ping) and IPv6 (ping6). If you use ping6 with a Link Local address, the Zone ID is a separate argument, not a part of the address. For example, "eth0" is the Zone ID in the following ping6 command: `ping6 -I eth0 fe80::1574:833f:b328:30ab`.
- As previously mentioned, the PSQL interfaces running on Linux support UNC paths and UNC-safe addresses even though Linux itself does not. This could be a source of confusion when you perform operations at the operating system level. For example, assume that you receive a network error for some reason when you run the following command for butil: `butil -stat // [fe80::1574:833f:b328:30ab%eth0]/DemodataShare/BILLING.MKD`.

  You may attempt to diagnose the error with Linux commands such as the following:

  ```
  ping [fe80::1574:833f:b328:30ab%eth0]
  ping6 [fe80::1574:833f:b328:30ab%eth0]
  ping6 fe80::1574:833f:b328:30ab%eth0
  ```

However, all of those commands fail with “unknown host” or “invalid argument” because of the brackets in the address, the appended Zone ID, or both. You must use `ping6 -I eth0 fe80::1574:833f:b328:30ab` to determine if the network connection actually exists.

- The mount command does not accept a Zone ID with an IPv6 numeric address. A Link Local address cannot be used with mount. You can use a host name that resolves to an IPv6 address.

Best Practices

Actian Corporation recommends the following practices when you use IPv6 with Linux.

- Use pvnetpass to ensure that the PSQL Client platform provides the required credentials on a server when the Client connects to the database engine on that server. Pvnetpass is required for each different server address connection, even equivalent server addresses. For example, suppose that a host name and an IPv6 numeric address refer to the same server. A connection to each one requires the use of pvnetpass. See pvnetpass in *PSQL User's Guide*.

Frequently Asked Questions About IPv6

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

Table 6 FAQs About IPv6 Support and PSQL

Question	Answer
Does a network environment that includes both IPv4 and IPv6 affect PSQL user counts?	No. PSQL Server or Workgroup uses one user count for each *unique* incoming protocol from the same client computer session (such as TCP/IP and SPX). IPv4 and IPv6 are just different address formats of TCP/IP.
Because PSQL Clients can have both IPv4 and IPv6 addresses will confusion result over how many licenses are required?	No. This is not a new situation. In the past, Clients have sometimes had more than one IPv4 address. PSQL handles the situation by recognizing all of the addresses of a machine to identify it. PSQL follows the same logic for multiple addresses using more than one IP version.
Any restrictions with IPv6 and the use of License Administrator?	Yes. The PSQL *licensing server* does not yet support IPv6. Because of this, you can use License Administrator over IPv6 to administer licenses. However, to *authorize* a license with License Administrator, the utility must be running on a machine that uses IPv4 to communicate with the PSQL licensing server.
Does the use of IPv6 change the use of DSNs?	No, continue to use DSNs as usual. You can use IPv6 numeric address, with or without brackets, when creating DSNs. See DSN Setup and Connection Strings in *ODBC Guide*.
Can PSQL System Analyzer still test connections?	Yes, you can test IPv6 addresses. The transaction engine test requires a mapped drive or a UNC path to the samples folder. An IPv6 numeric address used in a UNC path must include brackets. See PSQL System Analyzer (PSA) in *PSQL User's Guide*.
Can PSQL Auto Reconnect (PARC) be used with IPv6?	Yes. See PSQL Auto-Reconnect in *Advanced Operations Guide*.
Does PSQL support IPv6 in virtual machine environments?	Yes.
Can the Listen IP Address configuration setting be set to multiple addresses?	Yes. See Listen IP Address in *Advanced Operations Guide*.
Which GUI utilities support IPv6?[1]	DDF Builder, Function Executor, License Administrator, Monitor, PCC, PSA, and Rebuild.

Table 6 FAQs About IPv6 Support and PSQL

Question	Answer
Which command line interface (CLI) utilities support IPv6?[1]	Bcfg, bmon, butil, clilcadm, clilcadm64, pvdbpass, pvddl, pvnetpass, rbldcli[2], and w64clilcadm.
1 Some utilities accept UNC path formats that may not be valid for the operating system, file system, or file sharing protocol. 2 Rbldcli partly relies on direct file access for proper operation. Any file paths specified on a rbldcli command must also be directly accessible through the file system of the operating system or through a file sharing protocol such as Windows File Sharing, NFS, Samba, and so forth. One method to verify valid file paths for rbldcli is to ensure that the paths work with operating system commands such as “dir” or “ls.”	

Using TCP/IP to Connect to a Windows Server

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

Configuring a Client for the Server IP Address

When 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 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.

   ```
   # 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 PSQL 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 PSQL 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 PSQL server running on a Windows machine.

Configuring 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 PSQL.

Changing the configuration of PSQL to use SPX with a Windows platform

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

1. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the 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 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

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 MicroKernel Engine
- 1583 for the Relational Engine
- 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 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 PSQL Explorer and To create a new user using PSQL Explorer, both in *Advanced Operations Guide*.

For the Relational Engine, port assignment 1583 is configurable on the server through the 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 PSQL 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 PSQL engine for the port assignment changes to take effect. See the chapter Using PSQL in *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 PSQL Server and its clients. Be sure that the applicable ports listed in the services file are in agreement with the ports set by 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 PSQL database engine for the changes to take effect.See Starting and Stopping the Database Engine in *PSQL User's Guide*.

Windows FireWalls

The installation of PSQL Server and PSQL Workstation performs certain actions pertaining to firewalls. Starting with Vista, Windows 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 PSQL installation actions pertaining to the active profile(s).

Table 7 Installation Actions for Windows Operating Systems

Active Firewall Profile[1]	Rules Added for PSQL Services	State of Rules After Installation[2]
Multiple, such as • Domain • Private • Public	• Domain—Yes • Private—Yes • Public—Yes	• 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 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 PSQL rules are added for the Public profile but *disabled*. Neither the interactive nor the silent installation of 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 PSQL rules for the Public profile.

➢ To enable 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 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 PSQL, 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 PSQL communications, refer to the steps in To enable PSQL rules for the Public profile. Use the steps as a general guideline for how to enable the 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 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 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 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.

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 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 system.

   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 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 PSQL 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

The use of encoding methods is intended for advanced users. In general, the default encoding settings are sufficient and do not require changing. See Database Code Page and Client Encoding in *Advanced Operations Guide*.

Using the DOS Requester

PSQL v12 supports DOS Btrieve applications with the DOS Requestor. The DOS requester supports Btrieve applications only, not ODBC applications. This section explains how to use the DOS requester to run 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 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

chapter

11

Application Configuration Scenarios

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.

Disabling Administrative Functions

In prior releases, the ability to perform administrative functions was prohibited from the client. In PSQL v12, PSQL clients running within Terminal Services client sessions can perform PSQL administrative functions by default. For example, a user with such a client can change configuration settings for 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 PSQL Server engines are supported for use with Microsoft Terminal Server and Citrix XenApp 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

By default, Workgroup Engine is installed to run as a service for a fresh install. This allows the engine to start automatically when the operating system starts. A user is not required to log in to start the engine.

During a Custom installation you can select to run the Workgroup Engine as a console application. Or, if you upgraded your Workgroup installation and you had the previous version installed to run as an application, the upgrade installation is also set to run as an application. In either case, if you decide that you would rather run the Workgroup Engine as a service, you change how it is run. See Running the Workgroup Engine as a Service.

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 the conceptual steps to configure 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 PSQL into the environment.

Server and Client Support

PSQL Server runs on supported Windows Servers within an Active Directory environment. The PSQL client runs 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 PSQL table file, you will be unable to write to the table.

Microsoft Terminal Services Support

PSQL Server engines are supported for use with Microsoft Terminal Server running within an Active Directory environment. For more information about Terminal Services, seeTerminal Services.

PSQL 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 PSQL administrative privileges.

See Active Directory Tasks for the general 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 can grant the Log on locally authority directly to a user or to the Pervasive_Admin group (and add the user to the group).

You can 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 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

Use the following steps to create a Pervasive_Admin group in Active Directory to grant users PSQL administrative privileges in a Windows environment. The steps assume that you are setting privileges on the domain controller for the machine running the database engine.

➤ To add the Pervasive_Admin group as a default group policy

1. Create the Pervasive_Admin group on the domain controller for the domain of the machine running the database engine.
2. Specify Pervasive_Admin for the group name.
3. Set the group scope to Domain local. Do not use Global or Universal.
4. Add users to the Pervasive_Admin group.
5. Confirm that the users appear as members of the group.
6. Add the Pervasive_Admin group to the Log on locally privileges for the domain.

   Note: If the Log on locally option is grayed out, skip step 6 and use the next task to continue administrative group setup as a local policy.

➤ To add the Pervasive_Admin group as a local policy

These steps continue from step 5 in the previous task when you cannot use the Log on locally option in a group policy.

1. Click Start, enter `gpmc.msc`, and press Enter.
2. Double-click the name of the forest to open it.
3. Open Domains.
4. Open the name of the domain to which you want to join a computer.
5. Right-click Default Domain Controllers Policy and select Edit.
6. In the console tree, expand Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies, and then select User Rights Assignment.
7. In the details pane, open Allow Logon Locally.
8. Confirm that Define these policy settings is selected.
9. Click Add User or Group.
10. Do one of the following:
    - Enter the user account to allow to log on locally.
    - Click Browse to find the account with the Select Users, Computers, or Groups dialog box.
11. Click OK in each dialog box until you have closed them all.

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. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the PSQL Explorer, expand Engines to display the engines registered with 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 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 PSQL server, you must change the settings in your MicroKernel Router. Use the PSQL Control Center to change MicroKernel Router settings.

➢ To configure local and remote access for the MicroKernel Router

1. Access Control Center (PCC) from the operating system Start menu or Apps screen.
2. In the 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 PSQL server.
5. Click OK.



Note See *Advanced Operations Guide* for more information on changing settings using the 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

chapter

12

Installing PSQL Server, Vx Server and Client for Linux

Instructions for Installing and Uninstalling PSQL Server, Vx Server and Client on Linux

This chapter explains how to install and uninstall PSQL:

- Before You Install PSQL for Linux
- Installing PSQL Using RPM
- Installing PSQL Using TAR
- After Installing PSQL for Linux
- Uninstalling PSQL for Linux

Before You Install PSQL for Linux

Before installing or upgrading PSQL Server, Vx Server, or Client, review the following information:

- System requirements listed on the PSQL Web site.
- *Chapter 2,* Preparing to Install PSQL– This chapter provides important information including platform specific notes.
- *Release Notes* – The release notes are located in readme_psql.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, Vx Server

If you are planning to access the PSQL MicroKernel Engine 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 Web site, http://www.samba.org, for installation and configuration instructions.

After installing PSQL, 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 PSQL products currently installed, or on a Linux machine with a PSQL v12 Server or Vx 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 pvsw (if they do no exist)
- Sets user:group ownership to psql:pvsw for the installed files (if not already set)

Full Installations

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 lists the installation packages.

Table 9 Full and Client Linux Installations

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

Table 9 Full and Client Linux Installations

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

Notes for 64-bit Distributions

Note that, by default, certain 64-bit Linux distributions support only 64-bit components and executables. The distribution media for such distributions include the 32-bit libraries, but *they are not installed by default*.

PSQL supports 64-bit editions of Linux distributions, but also requires certain 32-bit libraries. If you use a 64-bit Linux distribution, verify whether the 32-bit libraries are installed by default. If not, use the following table to determine the 32-bit libraries and other components required for PSQL.

PSQL Product	32-bit Components Required From Linux 64-bit Distributions
Server 64-bit Server 32-bit Vx Server 64-bit Vx Server 32-bit Client 32-bit	The 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 PSQL installation stops and displays a message. To install PSQL, you must first install glibc.i686 and libstdc++.i686.
	The PSQL installation RPM and TAR files also check for the existence of 32-bit GTK. If GTK is not present, 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 PSQL installation continues and completes after displaying the message. To ensure that PCC, DDF Builder and GUI Phone Authorization utilities work, install the following Linux packages: • gtk2.i686 • PackageKit-gtk-module.i686 • libcanberra-gtk2.i686 • gtk2-engines.i686. For example: yum install gtk2.i686 PackageKit-gtk-module.i686 libcanberra-gtk2.i686 gtk2-engines.i686 Note that, except for the three utilities just mentioned, 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 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.
Server 64-bit Server 32-bit Vx Server 64-bit Vx Server 32-bit	The PSQL installation attempts to configure Samba. However, by default on certain 64-bit Linux distributions, Samba is only partially installed and is not enabled. You may ignore installation errors pertaining to creating the PSQLDATA Samba share (the smb.conf file exists, but nothing is there to support it). The errors are benign and PSQL functions normally.

Installing PSQL Using RPM

The RPM format allows you to install 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 PSQL products using RPM:

- Installing PSQL Server, Vx Server for Linux - RPM
- Installing PSQL Client for Linux - RPM

If you have any trouble with installation, see the chapter Troubleshooting After Installation.

Installing PSQL Server, Vx 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	PSQL-*yy.yy-zzz.zzz*.x86_64.rpm
Server 32-bit	PSQL-*yy.yy-zzz.zzz*.i486.rpm
Vx Server 64-bit	PSQL-Vx-*yy.yy-zzz.zzz*.x86_64.rpm
Vx Server 32-bit	PSQL-Vx-*yy.yy-zzz.zzz*.i486.rpm

In all cases, *yy.yy* is a release number and *zzz.zzz* is 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 `pvsw`
- Sets `user:group` ownership to `psql:pvsw` for the installed files
- Authorizes 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 PSQL daemons
- Launches the PSQL daemon `(mkded)`

➤ To install PSQL Server, Vx Server using RPM

Note If you have a previous version of 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 PSQL already installed, you must uninstall that product and then install the PSQL v12 product.

See Uninstalling PSQL for Linux for information on uninstalling PSQL.

Installing PSQL Client for Linux - RPM

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

Table 11 Linux Client Package Names - RPM

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

In all cases, *yy*.yy is a release number and *zzz.zzz* is 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

➤ To install 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 PSQL already installed, you must uninstall that product and then install the PSQL v12 product.

See Uninstalling PSQL for Linux for information on uninstalling PSQL.

Installing PSQL Using TAR

The tape archive (TAR) format allows you to install 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 PSQL products using TAR:

- Installing PSQL Sever, Vx Server for Linux - TAR
- Installing PSQL Client for Linux - TAR

If you have any trouble with installation, see the chapter Troubleshooting After Installation.

Installing PSQL Sever, Vx Server for Linux - TAR

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

Table 12 Linux Server Package Names - TAR

Installation Type	Package Name
Server 64-bit	PSQL-*yy.yy-zzz.zzz*.x86_64.tar.gz
Server 32-bit	PSQL-*yy.yy-zzz.zzz*.i486.tar.gz
Vx Server 64-bit	PSQL-Vx-*yy.yy-zzz.zzz*.x86_64.tar.gz
Vx Server 32-bit	PSQL-Vx-*yy.yy-zzz.zzz*.i486.tar.gz

In all cases, *yy*.yy is a release number and *zzz.zzz* is 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 PSQL Server, Vx 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 preinstallation script:

```
sh preinstall.sh
```

7 Run the post installation script:

```
sh postinstall.sh
```

Your tar installation is complete. For additional information, see After Installing PSQL for Linux.

Upgrade Installation

If you have a previous version of PSQL already installed, you must uninstall that product and then install the PSQL product. See Uninstalling PSQL for Linux for information on uninstalling PSQL.

Installing PSQL Client for Linux - TAR

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

Table 13 Linux Client Package Names - TAR

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

In all cases, *yy*.yy is a release number and *zzz.zzz* is 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

➤ To install PSQL Client 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_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 PSQL installation scripts reside.

```
cd psql/etc
```

6 Run the preinstallation script:

```
sh clientpreinstall.sh
```

7 Run the post installation script:

```
sh clientpostinstall.sh
```

Your tar installation is complete. For additional information, see After Installing PSQL for Linux and Configuring Network Communications for Clients.

Upgrade Installation

If you have a previous version of PSQL already installed, you must uninstall that product and then install the PSQL product. See Uninstalling PSQL for Linux for information on uninstalling PSQL for more information.

After Installing PSQL for Linux

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

- Verifying Installed Products with RPM
- Server Configuration
- Client Configuration
- Licensing
- 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 PSQL Packages Installed

PSQL Package	RPM Command to Verify Installation
Server	rpm -q PSQL
Vx Server	rpm -q PSQL-Vx
Client	rpm -q PSQL-Client Returns the specific client version installed (PSQL-Client-*release-build*).
All installed	rpm -qa | grep PSQL

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. To do this, enter the following at the command line:

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

Server Configuration

Generally, the default configuration settings for PSQL Server and Vx 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 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 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 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 name and IP to the hosts file on the Linux client, the client name is displayed correctly in the Monitor utility.

Licensing

Once you have completed installation, you may need to update your license to increase the user count for PSQL Server, or to increase the session count or data in use limits for PSQL Vx Server. See Increase User Count or Data In Use in *PSQL User's Guide*

Common Questions After Installation

If you are have problems with your installation, see Troubleshooting After Installation or get help online from the PSQL Knowledge Base at the PSQL website. The following are common questions after installation of the products:

- Where Do Files Reside After Installing PSQL?
- How Do I Access the Documentation?
- What If I Get Errors Trying To Start the Utilities?

Where Do Files Reside After Installing PSQL?

The following table lists the primary directories and files that result from installing the 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- and 64-bit products.

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

Table 15 Primary Directories and Files for 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 PSQL database	Server
./data/samples		Sample Btrieve files, alternate collating sequence file and the DefaultDB system database	Server
./etc			Server and Client

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

Path from $PVSW_ROOT	Primary Files	Description	Applies to Installation of
	.PSRegistry	PSQL 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: ◆ 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
./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 PSQL Server includes the following:

- Man pages for the command-line utilities
- PSQL Documentation Library
- 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 PSQL Server, Vx Server, and with PSQL Client. They are not installed as part of the user documentation.

Documentation Library

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

➤ To view the PSQL Documentation Library

1 Open a terminal window.

2 Run one of the following:

- As root user

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

- As the psql user

  ```
  pcc
  ```

Note that the viewer for the documentation library is integrated into PSQL Control Center (PCC). You can access the documentation library in PCC on the Welcome view, in the Help menu, or by pressing F1 (Windows) or Shift F1 (Linux).

Release Notes

The release notes in readme_psql.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 *PSQL User's Guide* and Starting DDF Builder from a Command Line in *DDF Builder User's Guide*.

Uninstalling PSQL for Linux

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

RPM Version

The following table lists the RPM commands to uninstall the various 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 PSQL Packages

To Uninstall This Package	Use This RPM Command
Server 32- or 64-bit	rpm -e PSQL
Vx Server 32- or 64-bit	rpm -e PSQL-Vx
Client 32- or 64-bit	rpm -e PSQL-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 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 PSQL Packages

Package To Uninstall	Script(s) To Execute[1,2]
Server 32- or 64-bit Vx Server 32- or 64-bit	sh preuninstall.sh sh postuninstall.sh **Note:** The scripts must be executed in sequence: preuninstall first, followed by postuninstall.
Client 32- or 64-bit Note: If you have both 32- and 64-bit Clients installed, see the next topic.	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 preunistall.sh rm postunistall.sh rm client*.sh	

Uninstalling Both 32- and 64-bit Clients

If you have installed both the 32- 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- and 64-bit clients, you would run the following:

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

chapter

13

Using PSQL on Linux

Working With the Products on Linux

The chapter contains the following sections:

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

Finding What You Need

Accessing the User Documentation

See How Do I Access the Documentation?.

Man Pages

The man pages are installed with PSQL Server or Client. Refer to the directory $PVSW_ROOT/man/man1for 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 *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.

PSQL Account Management on Linux

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

After Installation Behavior

- User psql has no password and can be accessed only through the root account using su.
- You can access the `.bash_profile` for user `psql` with `~psql/.bash_profile`.
- All PSQL files have `user:group` ownership `psql:pvsw`
- You must be logged in as `root` to run the start and stop scripts for the 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 `pvsw` to perform functionality with the following utilities:
  - 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.
  - PSQL 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 `$PVSW_ROOT/bin` to your `PATH` environment variable, and `$PVSW_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 PSQL utilities. This does not give the users administrative privileges or access to PSQL data.

Here is an example of a modified profile:

```
PVSW_ROOT=/usr/local/psql
PATH=$PATH:$PVSW_ROOT/bin:/bin:/usr/bin
LD_LIBRARY_PATH=$PVSW_ROOT/lib:$PVSW_ROOT/bin:/usr/lib
MANPATH=$MANPATH:$PVSW_ROOT/man
```

Ensure that you export all variables specific to PSQL.

Configuration

Generally, the default configuration settings for PSQL Server and Client are sufficient. You typically do not have to configure any settings for the database engine and clients to communication 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 *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 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 PSQL Client on Linux can connect to any of the 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 PSQL registry on the client machine. If you do not specify user names and passwords, your applications can receive status code 3119.

See pvnetpass in *PSQL User's Guide*.

Creating a Client DSN

A client data source name (DSN) is required if applications on the client use the PSQL Relational Engine through ODBC. To create a client DSN, you use the `dsnadd` utility included with the PSQL Client for Linux. See dsnadd in *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 PSQL data and provides connection snippets and samples for connecting to PSQL data from web applications on Linux.

ODBC Behavior

When you first install 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 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 `pvsw`. 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 `pvsw` group, substituting the name used in your Apache configuration file.

```
/usr/bin/gpasswd -a nobody pvsw
```

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.

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 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>PVSW PHP Sample</TITLE>
</HEAD>
<BODY>

<H1>PSQL 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 PSQL Web site.

This sample uses the Pvideo database that is included with the 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 PSQL.

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 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("PSQL Hello World Sample - Perl");

print <<ENDOFMENU;
<H1>PSQL 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 PSQL



Note This procedure assumes you have a working installation of PSQL v12, 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 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"
}
```

chapter

14

Troubleshooting After Installation

How to Proceed When You Encounter Errors After Installation

PSQL provides several features and tools 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
- Logged Messages
- Configuration for Special Installation Situations
- Diagnosing Problems with PSQL System Analyzer (PSA)
- Verifying Database Engine is Running
- Obtaining File, Client, and Engine Version Number
- Engine and Client Version Conflicts
- State of Key Is “Failed Validation” or “Disabled”
- Troubleshooting Common PSQL Issues
- Issues After Uninstalling PSQL on Windows
- How to Get Additional Help

Troubleshooting Tools

The following table describes tools to help you avoid or solve problems.

Table 18 PSQL Tools that Assist in Installation and Problem Determination

Feature/Component	Function	For More Information
PSQL System Analyzer	Analyzes system components and runs communication tests.	See Diagnosing Problems with PSQL System Analyzer (PSA).
PSQL Message Logging	Logged messages can be of type status, information, warning, or error, and can originate from any PSQL component.	See Reviewing Message Logs in *Advanced Operations Guide*
Gateway Locator	Determines or changes the Gateway being used for a particular data dictionary (only in PSQL v12 Workgroup Edition.)	See Configuring the Workgroup Engine.

Troubleshooting Strategies

Typically, your installation process completes with no 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, PSQL 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 the PVSW.LOG file (Windows) or in the SysLog (Linux)?

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 PSQL 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 PSQL can help you troubleshooting problems. Messages can be of type status, information, warning, or error, and can originate from any PSQL component. Certain messages specific to licensing issues originate only from the license administration components. In either case, PSQL logs messages to the following repositories:

- Notification Viewer
- Operating System Event Log
- PSQL Event Log (PVSW.LOG) (Windows only)

Any licensing message logged to Notification Viewer is also logged to the Operating System Event Log and to PSQL Event Log. Similarly, any licensing message logged to the Operating System Event Log is also logged to PSQL Event Log. Note that the Operating System Event Log and PSQL Event Log may contain licensing messages not logged to Notification Viewer.

Messages not specific to licensing are logged to the Operating System Event Log and to 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 PSQL Event Log. If you suspect a problem *not* related to licensing, check the Operating System Event Log and PSQL Event Log.

For logging details, see Reviewing Message Logs 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 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 *Advanced Operations Guide*, see: • 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 • In *Advanced Operations Guide*, see PSQL Auto-Reconnect.
Database file names that must not include embedded spaces	Enable a configuration setting that instructs PSQL to reject files with embedded spaces in the name. • In *Advanced Operations Guide*, see Embedded Spaces and Long File Names and Embedded Spaces Support.

Diagnosing Problems with PSQL System Analyzer (PSA)

PSQL System Analyzer is a diagnostic utility included with PSQL. It is integrated into the product installation and available as a stand-alone diagnostic tool to help you with the following tasks:

- Troubleshoot network problems
- Detect previous installations of Btrieve or PSQL on your system
- Note other factors that influence your networking environment
- View current component set and versions



Note For detailed in formation on using PSA, see PSQL System Analyzer (PSA) in *PSQL User's Guide*.

Verifying Database Engine is Running

To verify that the PSQL engine is running, see the procedure for your platform and engine:

- PSQL Server on Windows
- PSQL Workgroup on Windows
- PSQL Server on Linux

PSQL Server on Windows

You can use the Services function of the Windows control panel.

➢ To check PSQL Services on Windows servers using the Control Panel

1 At the operating system, open Services under Administrative Tools.

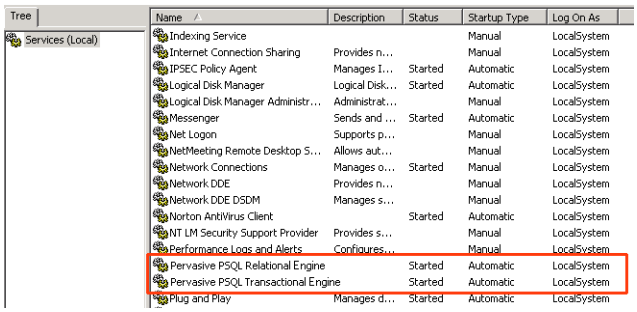
2 Scroll the list of services until you reach the following services.

- PSQL Transactional Engine
- PSQL Relational Engine

Both of these services must be started if 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



3 If a service is not started, right-click the service name, then click Start.

PSQL Workgroup on Windows

To verify that the PSQL Workgroup engine is running:

➢ To start the PSQL Workgroup engine

1 Click Start Workgroup Engine from the operating system Start menu or Apps screen.

By default, the MicroKernel allocates resources and is ready to service local application database requests.

➢ To stop the PSQL Workgroup engine

1 Click Stop Workgroup Engine from the operating system Start menu or Apps screen.



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.

PSQL Server on Linux

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 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 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 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

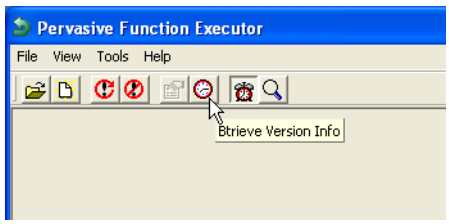
1 Access Function Executor from the operating system Start menu or Apps screen.

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 PSQL v12 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 PSQL Control Center

You can use the PSQL Control Center to determine a file version.

➢ To Determine the File Version of a Table Using PSQL Control Center

1 Access Control Center (PCC) from the operating system Start menu or Apps screen.

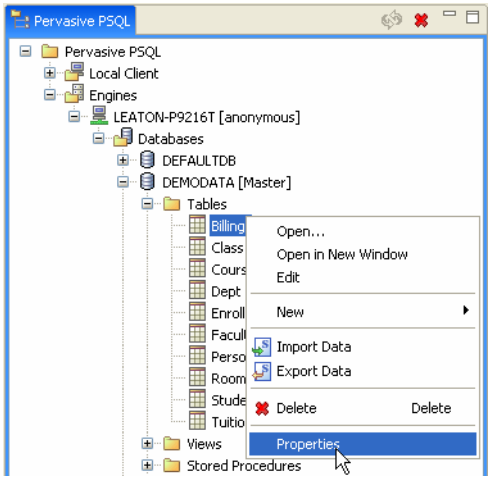
2 Find the database by expanding its name in the PSQL Explorer on the left.

3 Do one of the following:

a. Click File ▸ Properties from the File menu.

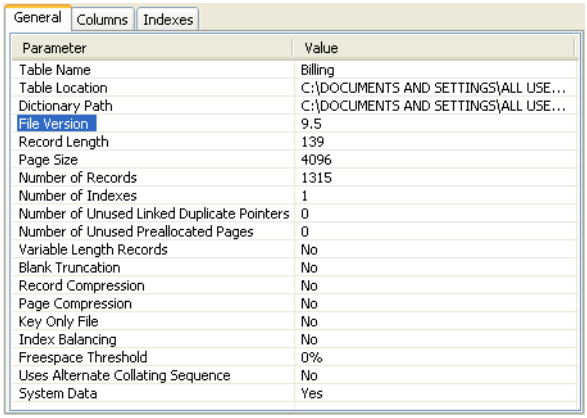
b. Right-click a table name and select Properties as shown in Figure 10.

Figure 10 Obtaining a File Version with the 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



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. Access Maintenance from the operating system Start menu or Apps screen.
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.

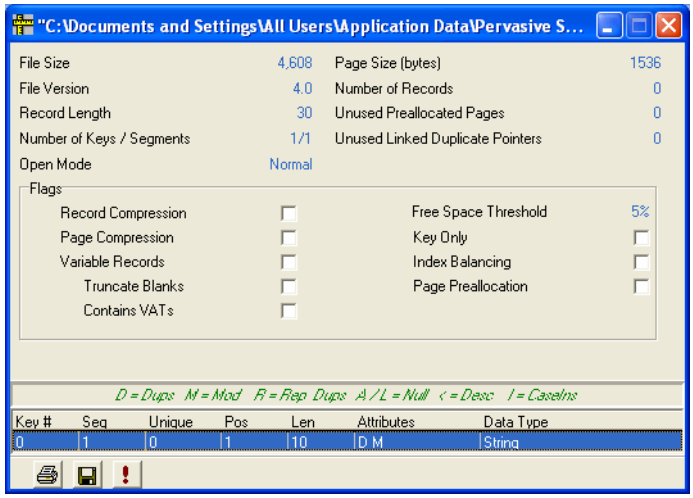
➤ To Determine the File Version of a Table Using Function Executor

1. Access Function Executor from the operating system Start menu or Apps screen.
2. From the File menu, click File then 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 then 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 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

Actian Corporation 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. Actian Corporation does not guarantee that newer versions of client requesters will function correctly with older versions of the engine. Therefore, Actian Corporation recommends that you avoid the use of newer version client requesters with an older engine.



Note See also Does it matter if I use PSQL Clients that are of a different version than that of the database engine?, particularly if you are using PSQL Vx Server.

State of Key Is “Failed Validation” or “Disabled”

PSQL licensing components may determine that the product key for the database engine has become invalid. If so, the state of the key changes from Active to Failed Validation. Usually, you discover the invalidation after one of the following actions:

- You manually run the command line utility `clilcadm` with the `-t` option to validate the key.
- You restart PSQL services, automatically prompting key validation.

Message logs and other notifications can alert you to a failed validation and possible reasons. The most common reason is a change to the host name of the system where PSQL is running.

After validation failure, the database engine continues to run normally for a set number of days, called a *failed validation period*. This time period, set within product key itself, allows ample time for you to correct the failure and move the key state from failed back to active.

You have three ways to find out how many days you have to do this:

- In License Administrator, the product key shows a state of Failed Validation and an expiration date for the last day of the grace period.
- PSQL Notification Viewer gives a history that includes licensing events. It lists any key validation failure, the number of days left until expiration, and gives steps you can take for correction.
- The file pvsw.log lists an entry for the validation failure that includes the expiration date.

If you do not correct the causes of the failed validation before the expiration date, the key changes state to Disabled. A disabled key prevents the database engine from accessing data files.

For more information, see Reviewing Message Logs in *Advanced Operations Guide* and License Administration Concepts in *PSQL User's Guide*.

Troubleshooting Common 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 PSQL Workgroup is installed as an application, you may experience this situation. Applications do not automatically inherit the user's administrative rights. You can stop the engine, run it as administrator, and then run the command line license administrator or GUI License Administrator as administrator to authorize the license. Another alternative is to install Workgroup Engine to run as a service. See Running the Workgroup Engine as a Service.

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?

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 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, 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 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 PSQL.

How do I verify that my DOS components are functioning properly?

PSQL 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 PSQL 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 PSQL requesters and later installed the PSQL v12 Workgroup engine but your application is only using the requesters, you may have an outdated configuration that sets Local Access to Off. The PSQL v12 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 PSQL Online Manuals?

➤ **To access the online documentation:**

1. Access Control Center & Documentation from the operating system Start menu or Apps screen.
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 information on how to increase the environment space defined in config.sys, refer to Microsoft knowledge base articles.

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.

Issues After Uninstalling PSQL on Windows

When you uninstall PSQL using Programs and Features in Windows, you should not have any database engine files remaining on your system. However, some actions such as 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 PSQL uninstall program is not functioning correctly.

How to Get Additional Help

If you encounter problems during or after the installation that are not covered in the user documentation, please contact Actian Corporation and we will address your problem promptly.

Technical Support

If you still have questions or problems relating to your PSQL installation, you can obtain help from the PSQL Customer Support department.