Pervasive PSQL v10.10

What's New in Pervasive PSQL

An Overview of New Features and Changed Behavior

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

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

disclaimer

trademarks

What’s New in Pervasive PSQL

June 2008

100-004382-002

Contents

An Overview of New and Changed Features

Tables

About This Manual

This manual contains information about the features and enhancements that are new in this release of Pervasive PSQL. This release is referred to as Pervasive PSQL v10.10.

This manual describes the new and changed behaviors of the product relative to the most recent major release. The previous release for the Pervasive PSQL Windows products was Pervasive PSQL v10. The previous release for the Pervasive PSQL Linux products was Pervasive PSQL v9 Service Pack 2.

Who Should Read This Manual

This document is designed for any user who is familiar with Pervasive PSQL and wants to know what has changed in this release of the software.

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

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

Manual Organization

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

- Chapter 1—“What Is New in Pervasive PSQL v10.10”

  This chapter provides an overview of the changes in the current release of the software.

- Appendix A—“What Was New in Pervasive PSQL v10”

  This appendix provides an overview of the changes in the Pervasive PSQL v10 release.

This manual also contains an index.

Conventions

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

CASE

Commands and reserved words typically appear in uppercase letters. Unless the manual states otherwise, you can enter these items using uppercase, lowercase, or both. For example, you can type `MYPROG`, `myprog`, or `MYprog`.

Bold

Words appearing in bold include the following: menu names, dialog box names, commands, options, buttons, statements, etc.

`Monospaced font`

Monospaced font is reserved for words you enter, such as command syntax.

[ ]

Square brackets enclose optional information, as in [*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*.

%*string*%

A variable defined by the Windows operating system. *String* represents the variable text. Example: %ProgramFiles% is a variable for the location C:\Program Files.

$*string*

An environment variable defined by the Linux operating system. *String* represents the variable text. Example: $PATH, which contains a colon-separated list of directories that the shell searches for commands that do not contain a slash in their name.

chapter 1

What Is New in Pervasive PSQL v10.10

An Overview of New and Changed Features

The purpose of this chapter is to summarize and explain the major new features and differences in behavior between this product and the previous release. The previous release for the Pervasive PSQL Windows products was Pervasive PSQL v10. The previous release for the Pervasive PSQL Linux products was Pervasive PSQL v9 Service Pack 2.

Where applicable, links are provided to additional information for a given feature.

List of New Features and Improvements

This release includes the following new features and changes:

These features are described in the sections that follow. Also see the Readme file for additional information about this release that *What's New* may not contain.

Windows Server 2008 Certification

Pervasive PSQL v10.10 64-bit Server is certified for Windows Server 2008. As a certified Windows program, Pervasive PSQL v10.10 complies with best practices, stability, security and reliability for the Windows Server 2008 platform.

Among other features, Windows Server 2008 includes many changes to the Graphical User Interface (GUI), a new security model using User Account Controls (UAC), changes to the Windows firewall and additional Internet protection.

Because Windows Server 2008 and Windows Vista are so similar, see “Support for Windows Vista” on page A-3 for a discussion of working with Windows Server 2008.

“Works With Windows Vista” Compatibility

The Pervasive PSQL v10.10 client provides “Works with Windows Vista” compatibility. This certification allows users to adopt Windows Vista with the assurance that the Pervasive PSQL v10.10 client has meet all certification and application compatibility.

Linux

Pervasive PSQL v10.10 supports Linux distributions running Kernel 2.6.0 or higher.

Installation

For information on installing Pervasive PSQL v10.10 on Linux, see "Installing Pervasive PSQL for Linux" on page 12-1 in *Getting Started With Pervasive PSQL*.

Upgrading to Pervasive PSQL v10.10

If you are upgrading from a previous version of Pervasive PSQL on Linux, you must uninstall and then install Pervasive PSQL v10.10.

For information on upgrading to Pervasive PSQL v10.10 on Linux, see "Upgrade Installation" on page 12-6 in *Getting Started With Pervasive PSQL*.

Full and Core Installations

Full and core installation packages are available for Linux. A full installation includes the necessary engine and client files, all available utilities and the complete user documentation. The core installation contains only the engine and client files.

See "Full and Core Installations" on page 12-3 in *Getting Started With Pervasive PSQL* for detailed information about them.

PAM Authentication

If the installation of Pervasive PSQL Server detects PAM (pluggable authentication modules), the installation script completes its configuration so that PAM can be used. For more details, see "Authentication (Linux engines only)" on page 4-18 in *Advanced Operations Guide*.

Support for 64-bit Operating Systems and Applications

Pervasive PSQL supports all interfaces on 64-bit operating systems. All interfaces are available in 32-bit versions, which can run on 64-bit operating systems. In addition, the transactional interface (the Btrieve API) and the DTI interface are available in 64-bit. Using either of those interfaces, an application can be written for a native 64-bit operating system running on machines with 64-bit architecture.

Note that the relational interface is 32-bit only. With 64-bit versions of the Server Engine, the transactional interface runs as a 64-bit daemon and the relational interface runs as a separate, 32-bit daemon.

Database Products

Pervasive Software now offers 32-bit and 64-bit versions of the Server Engine and the client for Linux. The following tables summarize the platforms on which the products can be installed and the type of applications supported.

Table 1-1 Server Install Platforms and Applications Supported

Product	Install Platform		Local Applications		Remote Applications	
	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
Server Engine 32-bit	✓	✓	✓		✓	✓
Server Engine 64-bit[1]		✓	✓	✓	✓	✓
[1] Only the transactional interface is 64-bit. The relational interface is 32-bit.						

Table 1-2 Client Install Platforms and Applications Supported

Product	Install Platform		Applications Supported	
	32-bit	64-bit	32-bit	64-bit
Client 32-bit	✓	✓	✓	
Client 64-bit		✓		✓
[1] Only the transactional interface is 64-bit. The relational interface is 32-bit.				

Note that the 32-bit Pervasive PSQL products are supported on 64-bit operating systems. Application created with a 32-bit interface can be used with the 64-bit Server engine.

SDK Interfaces

The 64-bit client supports 64-bit applications that use the transactional interface (the Btrieve API) or the distributed tuning interface (DTI). Other software development kit (SDK) interfaces are not supported for 64-bit access.

The header files for the Btrieve API and DTI have been enhanced to support 64-bit applications. For a 64-bit application using the Btrieve API, you need to define the preprocessor symbol BTI_LINUX_64 and link against the psqlmif library that is located in the $(PVSW_ROOT)/lib64 directory. 64-bit DTI applications link against the psqldti library that is located in the $(PVSW_ROOT)/lib64 directory.

Chunk Operations

Chunk descriptors are larger in size when used in a 64-bit application than when used in a 32-bit application. Pointers are 64 bit (8 bytes wide) in a 64-bit application.

Application	Chunk Offset (bytes)	Chunk Length (bytes)	User Data Pointer (bytes)
32-bit	4	4	4
64-bit	4	4	8

As with previous releases of Pervasive PSQL, the User Data field is only used in indirect chunk descriptors. You should initialize it to zero in direct chunk descriptors. Refer to the various chunk operations in the “Btrieve API Operations” chapter in *Btrieve API Guide*.

Utilities

Some new utilities are available on Linux. See “Utilities” on page 1-25.

Heartbeat

Pervasive PSQL v10.10 is cluster compatible with Linux Heartbeat in a clustering environment. The Heartbeat program is one of the core components of the Linux-HA (High-Availability Linux) project. The purpose of the cluster is to provide high-availability, scalability, and manageability of resources and applications.

See “Linux Heartbeat” on page 9-26 in *Advanced Operations Guide* for how to add Pervasive PSQL v10.10 to a cluster configuration.

Licensing

The following changes in Pervasive PSQL v10.10 pertain to licensing.

Permanent Keys

In previous releases, you could apply another permanent license to an existing permanent license. In this manner, you could increase the user count. The total user count equalled the sum of user counts from all of the permanent licenses.

With this release, only one permanent license at a time can be applied. A permanent license has a maximum user count that cannot be exceeded. User counts can now be increased up to that maximum only with a special key for that purpose.

User Counts

To increase a user count up to the maximum allowed by the permanent license, you now apply a special license key specifically for increasing user count. Your permanent license stays the same, only the user count increases.

The new key is referred to as a user count increase license key. It can be issued by Pervasive Software or by your application vendor if the Pervasive PSQL database engine is embedded in an application.

In License Administrator, the license type displays as “User Count Increase.” A user count increase license key requires that a permanent license key already be present on the system.

Note that you can increase the user count only to the maximum allowed by the permanent license. If you are at the maximum but require additional user counts, you need to obtain a new permanent license key with an increased maximum, delete the existing permanent key, and apply the new one.

See also “Increasing the User Count” on page 4-4 in the *Pervasive PSQL User's Guide*.

Encoding

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

Pervasive PSQL v10.10 better handles the complexity of the encoding between client and server and the various combinations of operating system, languages, and access method. The encoding enhancements are divided into database code page and client encoding. The two methods of encoding are separate but interrelated (see Table 1-3).

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

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

This section contains the following topics:

- "Database Code Page" on page 1-9
- "Client Encoding" on page 1-10
- "Encoding Interaction" on page 1-11
- "Utilities" on page 1-11
- "Access Methods" on page 1-15
- "ODBC Administrator and DSNs" on page 1-17

Database Code Page

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

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

Note For SQL statements that involve the use of more than one database, ensure that the database code page is the same for all of the databases. Otherwise, string data can be returned incorrectly.

Client Encoding

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

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

Automatic translation is specified when the client application connects to the database engine. See “Access Methods” on page 1-15 and “ODBC Administrator and DSNs” on page 1-17.

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

Encoding Interaction

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

Table 1-3 Interaction Between Database Code Page and Client Encoding

If Database Code Page Is	And the Client Application Specifies	The Pervasive PSQL Client
Server Default	Automatic Translation	Translates data and metadata from the default operating system (OS) encoding on the server to the OS encoding on the client.
A specific code page	Automatic Translation	Translates data and metadata from the database code page to the OS encoding on the client.
Server Default *or* A specific code page	Nothing (no encoding specified) (No encoding specified is the default behavior in versions prior to Pervasive PSQL v10.10.)	Sends data to the database engine in the encoding of the client machine and ignores database code page. For compatible data interpretation, the encoding used by the client machine must match the encoding of the data and metadata in the database.
Server Default *or* A specific code page	A specific encoding	Sends data to the server in the encoding specified by the client application and ignores database code page. For compatible data interpretation, the encoding specified by the client application must match the encoding of the data and metadata in the database.

When a database has OEM character data in it, the legacy solution was for the access method, such as ODBC using a DSN, to specify OEM/ANSI conversion. Now it is possible to set the OEM code page for the database and have the access method specify automatic translation. See also “DSN Configuration” on page 1-17.

Note The database engine does not validate the encoding of the data and metadata that an application inserts into a database. The engine assumes that all data was entered using the database code page as explained in Table 1-3.

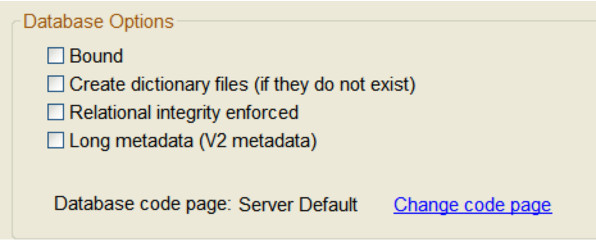
Utilities

New encoding functionality has been added to Pervasive PSQL Control Center (PCC), Dbmaint, and Dsnadd. (See also “ODBC Administrator and DSNs” on page 1-17.)

PCC

New Database Dialog

A new option, Database code page, has been added to the “New Database” dialog to set a database code page.



Database Options

□ Bound

□ Create dictionary files (if they do not exist)

□ Relational integrity enforced

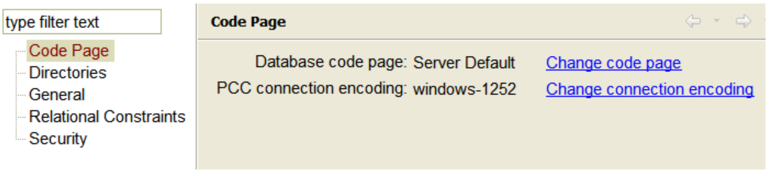
□ Long metadata (V2 metadata)

Database code page: Server Default

The default is “Server Default,” meaning the server OS code page. Generally, for a new database, “Server Default” is the setting you want. The link “Change code page” provides additional information about the setting and lets you select a specific code page.

Code Page Property Sheet

A new sheet for database properties, “Code Page,” has been added to PCC. The sheet contains two properties: Database code page and PCC connection encoding.



Database code page: Server Default

PCC connection encoding: windows-1252

The links “Change code page” and “Change connection encoding” provide additional information about the setting and let you select a specific encoding.

Database Code Page

This property applies to the database, which means that it potentially affects *all* client applications that exchange data with that database. A compatible encoding must be established between the Pervasive PSQL database engine and a client application. See Table 1-3 on page 1-11 for the various ways in which this can be accomplished.

Note Changing the database code page does not convert existing data or metadata in the database. To avoid data corruption, ensure that the code page setting matches the current encoding of any pre-existing data or metadata in the database.

PCC Connection Encoding

PCC is, itself, a client application to the database engine. As a client, PCC lets you specify the encoding to use for each database session when PCC reads and inserts metadata and data. The default for an existing database is to use the encoding of the machine where PCC is running. This is the legacy behavior of PCC. The default for a new database is to use automatic translation.

The following explains the interaction between the settings for “PCC connection encoding” and “Database code page.”

PCC Connection Encoding Set to "Automatic Translation"	PCC Connection Encoding Set to a Specific Encoding
PCC and the database automatically establish compatible encoding. The database metadata and data are translated from the encoding specified for “Database Code Page” to the encoding used on the system where PCC is running.	PCC ignores “Database Code Page” and uses the encoding specified to read and insert data and metadata. (This is the legacy behavior of PCC.)

Note “PCC connection encoding” applies only to PCC. It has no affect on other client applications.

Encoding Setting in Previous Versions

Prior versions of PCC (Pervasive PSQL v9 through Pervasive PSQL v10.01) contained an “Encoding” option on the “New Server” dialog and an “Encoding” property on the “General” property sheet for both server and database.

The “Encoding” option and the “Encoding” properties have been removed. In PCC, encoding can no longer be applied to a server. It is now a property associated with a database.

Note The “Encoding” property is not retained if you upgrade from Pervasive PSQL v9 or any later release, including Pervasive PSQL v10.10 Beta.

If you set “Encoding” in a previous release and want to keep the same value for Pervasive PSQL v10.10, you need to reset it using the new property.

“Encoding” as a database property is now “PCC Connection Encoding.” See “PCC Connection Encoding” on page 1-13.

“Encoding” as a server property has no equivalent setting in Pervasive PSQL v10.10. The “Encoding” server property no longer applies.

If you used the default encoding values for server and database, no changes are required.

SQL Statements

The CREATE DATABASE statement contains new syntax to set a database code page. The syntax supports a code page name, code page number, or the key word “Default.” “Default” specifies the operating system encoding on the server.

See “CREATE DATABASE” on page 3-46 in *SQL Engine Reference*.

Dbmaint

Dbmaint is a utility for managing named databases on Linux. This utility now accepts a parameter, -c, to set database code page.

See “dbmaint” on page 8-20 in *Pervasive PSQL User's Guide*.

Dsnadd

Dsnadd is a utility for creating and managing DSNs on Linux. This utility now accepts a parameter, -translate, to specify the desired encoding option (“none” or “auto”).

See “dsnadd” on page 8-23 in *Pervasive PSQL User's Guide*.

Access Methods

The Pervasive PSQL software development kit (SDK) provides various access methods by which an application can use a Pervasive PSQL database. The encoding enhancements affect the access methods ODBC, DTI, and DTO.

ODBC

ODBC uses the connection string attribute PvTranslate to specify the data encoding when the client connects to the database engine. The attribute can either be absent or empty, or have a value set to “auto” (for “automatic”).

If the attribute is absent or empty, ODBC does not translate any character data. This is the legacy behavior.

If the attribute is set to “auto,” the client and server automatically establish compatible encoding. Data translation, if required, occurs on the client. Note that “auto” overrides the “OEM/ANSI” setting in a DSN. PvTranslate defaults to “auto” if the local DSN is configured for Automatic. See “DSN Configuration” on page 1-17.

The PvTranslate attribute can be included with the ODBC function SQLDriverConnect (see following example) or with SQLConnect (through the DSN setting for automatic encoding; see “DSN Configuration” on page 1-17).

Example

An ODBC client application uses SQLDriverConnect to connect to a server DSN named “mydata” on a remote server named “MyServer” and establishes automatic encoding support:

```
Driver={Pervasive ODBC Client Interface};
ServerName=MyServer;ServerDSN=mydata;PvTranslate=auto;
```

Note that the PvTranslate attribute is explicitly stated because a server (an engine) DSN is being used.

DTI

The DTI application programming interface (API) has new functions to support database code page and client application encoding. Refer to the following in *Distributed Tuning Interface Guide*:

- “PvCreateDatabase2()” on page 2-47
- “PvGetDbCodePage()” on page 2-99
- “PvModifyDatabase2()” on page 2-204
- “PvCreateDSN2()” on page 2-55
- “PvGetDSNEx2()” on page 2-118
- “PvModifyDSN2()” on page 2-210

For string argument encoding, a user application still uses the client’s OS encoding at the API level. DTI handles the differences between OS encoding on the server and client. See “String Arguments Encoding” on page 1-2 in *Distributed Tuning Interface Guide*.

New DTI Status Codes

See “Status Codes” on page 1-39.

DTO

The DTO application programming interface (API) has new properties and enumerations to support database code page and client application encoding. Refer to the following in *Distributed Tuning Objects Guide*:

- "DtoDatabase Object" on page 5-5 (DBCodePage property)
- "DtoDSN Object" on page 5-15 (Translate property)
- "DSN Translate Option" on page 6-10 (enumeration)
- "Database Code Page" on page 6-11 (enumeration)

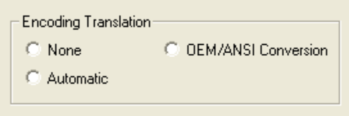
ODBC Administrator and DSNs

A new option, Code Page, is available when you create a database through ODBC Administrator. The option is the same as if you specified it on the New Database dialog in PCC. See "New Database Dialog" on page 1-12.

See also "Create Database From DSN Setup Dialog" on page F-8 in *SQL Engine Reference*.

DSN Configuration

A new option, "Automatic," is available when creating or configuring data source names (DSNs).



This option instructs the Pervasive PSQL ODBC client to automatically translate data when the database code page on the engine machine differs from the OS encoding on the client machine.

The Pervasive PSQL ODBC client ensures that the client and server use the same encoding. Data translation, if required, occurs on the client. (No data translation is required if the database code page on the engine machine is the same as the OS encoding on the client machine.)

"Automatic" requires that the client and the server be Pervasive PSQL v10.10 or greater.

See also "Encoding Interaction" on page 1-11 in this book and "Encoding Translation" on page F-5 in *SQL Engine Reference*.

None and OEM/ANSI

The previous option, “OEM/ANSI Conversion,” is also still available, but has been changed from a single choice to two separate choices. The single choice had two states: *not selected* or *selected*.

The *not selected* state is now labelled “None,” and is the default. (By default in previous releases, “OEM/ANSI Conversion” was *not selected*, which meant that no character data was translated between the client and server.) Therefore, “None” means that no character data is translated between the client and server. (The assumption is that the client and server use the same OS encoding.) The *selected* state is now labelled “OEM/ANSI Conversion.”

The functionality remains the same as in the previous release. See “OEM/ANSI Conversion” on page F-5 and “None” on page F-5, both in *SQL Engine Reference*.

Interaction Between Settings

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

Table 1-4 Interaction Between Database Code Page and DSN Settings

If Database Code Page Is	And the DSN Encoding Translation Is	Pervasive PSQL ODBC Driver
Server Default	None (The equivalent default behavior in versions prior to Pervasive PSQL v10.10.)	Performs no translation of data or metadata. The assumption is that the OS encoding on the server matches the OS encoding on the client. For compatible data interpretation, the encoding used by the client machine must match the encoding of the data and metadata in the database.
A specific code page	None (The equivalent default behavior in versions prior to Pervasive PSQL v10.10.)	Performs no translation of data or metadata. The assumption is that the OS encoding on the server matches the OS encoding on the client. For compatible data interpretation, the encoding used by the client machine must match the encoding of the data and metadata in the database.
Server Default *or* A specific code page	OEM/ANSI	Ignores database code page and translates data and metadata from the OEM encoding of the database to ANSI Windows encoding for the client application.
Server Default	Automatic	Translates data and metadata from the default OS encoding on the server to the OS encoding on the client.
A specific code page	Automatic	Translates data and metadata from the database code page to the OS encoding on the client.

Security

The database engine allows you to restrict access to a data file by specifying an owner name for the file. An owner name is a password required to access the data file.

The length of owner names has been increased from 8 bytes to 24 bytes. For this reason, the new owner name format is referred as “long.” The legacy format is referred to as “short.” In addition, stronger encryption is now used for owner names. The increased length and stronger encryption provide enhanced security for owner names.

Long owner names require a file format of 9.5 or greater. An owner name, long or short, with less than the maximum allowed bytes is padded with spaces to the maximum length (8 or 24 bytes).

Restrictions

The following restriction apply to the use of long owner names.

- Once a long owner name is specified, the data file cannot be read by a database engine prior to Pervasive PSQL v10.10.
- A data file with a long owner name cannot be rebuilt to a file format prior to 9.5 unless the owner name is first removed. (See “Clear Owner (30)” on page 2-7 in *Btrieve API Guide*.)
- Long owner names are supported by the transactional interface (Btrieve API), the relational interface, JCL, and JDBC. None of the other access methods support it. (For the relational interface and JDBC, see “GRANT” on page 3-154 and “SET OWNER” on page 3-248, both in *SQL Engine Reference*.)

Btrieve API

The Btrieve API operation Set Owner (29) now supports a bias, +17000, to specify a long owner name. This bias is also defined in btrconst.h as B_LONG_OWNER_NAME_BIAS. Btrconst.h is provided with the Btrieve software development kit (SDK) that can be downloaded from the Pervasive Software Web site.

The format of the data buffer with bias +17000 is as follows:

Length	Description
Up to 24 bytes	Owner name
1 byte	Null terminator (binary zero)

See “Set Owner (29)” on page 2-116 in *Btrieve API Guide*.

JCL

The JCL access method contains a new API, setDataFileLongOwner, to specify a long owner name. See the Javadoc HTML files provided with the JCL SDK that can be downloaded from the Pervasive Software Web site.

Utilities

Long owner names affect the following utilities:

- Maintenance
- Butil
- Function Executor

Maintenance

The Maintenance utility lets you perform a variety of actions on data files. One action is to set and clear owner names. The Set/Clear Owner Name dialog contains a new option, “Long Owner Name,” to set a long owner name.

See “To set or clear an owner name” on page 13-26 in *Advanced Operations Guide*.

Butil

Butil is the command-line version of the Maintenance utility. Butil contains a new parameter, /L, for the setowner command. The /L parameter designates a long owner name.

See “Setowner” on page 13-61 in *Advanced Operations Guide*.

Function Executor

Function Executor allows individual execution of Btrieve operations to simulate database calls from an application. The “Set Owner Name” dialog in Function Executor now contains an additional option: Use Long Owner Name.

See “Performing Operations Tasks” on page 12-23 in *Advanced Operations Guide*.

Performance

By default, the Pervasive PSQL Server installation for Windows installs a database accelerator called Xtreme I/O (XIO) if the system meets the minimum requirements for XIO. See “Database Accelerator” on page A-9. A message is now written to the pvsw.log file if XIO is installed and the database engine can cache files in the XIO cache. The log file contains the following message:

```
The Microkernel has acquired an active linkage to the XIO
Cache driver.
```

Note If XIO is installed but the pvsw.log does not contain the message, then the database engine is unable to communicate with XIO and no files will be cached. One possible corrective action is to stop and then start the transactional service. (See “Starting and Stopping the Server Engine on a Windows Server” on page 2-2 in *Pervasive PSQL User's Guide*.) Upon restart, if the database engine can communicate with XIO, the message is written to pvsw.log.

See also “Xtreme I/O Driver” on page 5-19 in *Advanced Operations Guide*.

Operating System Support

The Pervasive PSQL v10.10 release is now supported on the following operating systems:

- Windows Server 2008 (see “Windows Server 2008 Certification” on page 1-3)
- Windows Vista SP1 (SP1 is a requirement for Pervasive PSQL v10.10 to be installed)
- Windows XP SP3
- Linux distributions running Kernel 2.6.0 or higher

Refer to the Pervasive Software Web site for the complete list of the supported platforms for Pervasive PSQL v10.10.

Relational Interface

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

CREATE DATABASE

The CREATE DATABASE statement now contains syntax to specify a code page for the database. See “Encoding” on page 1-9 in this book.

See also “CREATE DATABASE” on page 3-46 in *SQL Engine Reference*.

INSERT

The INSERT statement now supports the optional clauses UNION, UNION ALL, and ORDER BY.

See “INSERT” on page 3-172 in *SQL Engine Reference*.

SELECT (with INTO)

Previous versions of Pervasive PSQL supported SELECT INTO only with temporary tables. SELECT INTO now supports creating regular tables. In addition, SELECT INTO supports the optional clause UNION and UNION ALL, and now can be used inside a stored procedure.

See “SELECT (with INTO)” on page 3-207 in *SQL Engine Reference*.

SOUNDEX

Pervasive PSQL v10.10 now supports the SOUNDEX string scalar function. SOUNDEX converts an alpha string to a four-character code to find similar sounding words or names. This new function conforms to the current rule set for the official implementation of SOUNDEX used by the United States Government.

See “SOUNDEX (string)” on page 5-3 in *SQL Engine Reference*.

Utilities

This section discusses the additions and changes to Pervasive PSQL utilities.

GUI Utilities

The GUI utilities for Windows include changes to License Administrator, Query Plan Viewer, and DDF Builder. DDF Builder is also new for Linux. Pervasive PSQL Control Center (PCC), which applies to both Windows and Linux, includes revised and new functionality.

License Administrator

In Pervasive PSQL v10.10, a permanent license has a maximum user count. You now increase the user count by applying a user count increase license key. License Administrator allows you to increase user count only up to the maximum allowed by the permanent license.

See “Licensing” on page 1-8 for complete details.

Query Plan Viewer

When opened, query plans are now sized by default in the Plan Pane so that the entire plan is viewable. The Plan Pane also has additional sizing commands such as zoom in, zoom out, and view at a desired percent.

See “To adjust the display size of a query plan in the Plan Pane” on page E-8 in *SQL Engine Reference*.

DDF Builder

DDF Builder is now available on Linux. This utility is used to view, create, and change Pervasive PSQL data dictionary files (DDFs) without modifying the underlying data file. The utility’s primary purpose supports the following:

- Creating the table definitions required to enable relational access to data files
- Modifying existing table definitions to ensure that relational access is enabled correctly for data files.

Note that DDF Builder is a specialized utility intended for advanced users and is not a utility that you would typically use daily.

For details, including tutorials, refer to the DDF Builder documentation that is accessed from the utility.

Check Table Consistency

DDF Builder provides functionality with which you check the consistency of a table. A consistency check uses a set of validation rules to compare the physical data file against its metadata (the table against the data dictionary files). See “Check Table Consistency” on page 2-24 in *DDF Builder User’s Guide*.

Use In A Client/Server Environment

DDF Builder can now be used in a client/server environment to connect to remote databases. You no longer are required to install DDF Builder on a server to create or repair DDFs.

Pervasive PSQL Control Center

PCC includes enhancements to the data grid, the text window view, and new functionality to export the schemas for all, or some, of the tables in a database.

Grid Enhancements

The grid view shows in a matrix format, like a spreadsheet, the result of running SQL statements. Each field is represented as a column and the data appears in cells within the columns. The grid has been enhanced in the following ways:

- Faster display of data result sets (no delay when you scroll to the end of a result set, for instance).
- Ability to display much larger data result sets.
- Support for Ctrl+Home and Ctrl+End to navigate to the top and bottom of the grid, respectively.
- A tool tip displaying the row number when you scroll the grid vertically using the scroll bar.

In addition, two new properties are available for the grid:

- Maximum number of rows to retrieve
- Hide maximum row limit information dialog

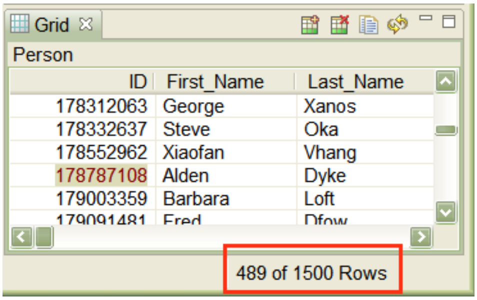
You access the grid properties in PCC by clicking Window then Preferences. Expand the Pervasive node then click Data Grid.

Property	Description
Maximum number of rows to retrieve	This property lists the maximum number of rows to display in the grid (in thousands). The default is 2,000 (2,000,000 rows). Note that this setting is affected by the amount of memory available for the machine. If you set the value too high, you may receive an "out of memory" error when PCC attempts to display the result set in the grid. You can increase the amount of memory available to PCC with the -vmargs parameter. See "PCC runs slowly or hangs when retrieving large record sets" on page 9-7 in *Pervasive PSQL User's Guide*.
Hide maximum row limit information dialog	This property suppresses the information dialog that displays if a result set exceeds the maximum rows to retrieve. For example, if the result set contains 3,000 rows and you have the maximum rows to retrieve set to 2 (2,000), a message similar to the following displays: `The recordset contains 3,000 rows. Only the first 2,000 will be retrieved. You may change the maximum number of rows to retrieve in the grid in the grid preferences.` The information dialog also contains an option "Do not show this dialog in the future." That option, and the **Hide maximum row limit information dialog** property, provide the same result.

Display of Row Positioned On and Total Rows

If you position on a record in the grid, the lower right corner of PCC lists the row positioned on and the total number of rows in the result set.

For example, the following image shows being positioned on the record with ID 178787108 is row 489 in a result set with 1,500 rows.



Person

ID	First_Name	Last_Name
178312063	George	Xanos
178332637	Steve	Oka
178552962	Xiaofan	Vhang
178787108	Alden	Dyke
179003359	Barbara	Loft
179091481	Fred	Dfow

489 of 1500 Rows

If you are not positioned on a row but the grid has focus, the display lists the total rows in the result set and the number of rows displayed based on the vertical size of the grid. For example, if the grid is vertically sized to view 20 records and the result set has 1,200 rows, the lower right corner of PCC displays “20 of 1,200 rows.”

Text Window View

The Text window view shows in a text format the result of running SQL statements. The Text window view has a new property to specify the number of rows displayed from a result set: Maximum number of rows to display. The default is 5,000.

See “Preferences for PCC Window Views” on page 3-10 and “Text Window View” on page 6-5, both in *Pervasive PSQL User's Guide*.

Export Table Schema

The Control Center now contains functionality to export to a text file the schemas for all, or some, of the tables in a database. The exported schemas contain the CREATE TABLE SQL statements (and CREATE INDEX statements if applicable) to create the tables and their indexes. The exported file is called an SQL script, which can be run (executed) in SQL Editor.

The export feature allows the inclusion of an IN DICTIONARY clause or a USING clause. You can export a schema for a particular table (or tables) or, at the database level, for all tables at once.

See “Metadata” on page 3-33 in *Pervasive PSQL User's Guide*.

New Database Dialog

The New Database dialog has been revised as the following table explains.

GUI Component	Discussion
“DBName Options” group box	The “DBName Options” group box has been renamed to “Database Options.”
“Metadata” group box	The “Metadata” group box, which had options “Version 1” and “Version 2,” has been removed. A new option, “Long Metadata (V2 metadata),” lets you set metadata version: *not* selected specfies V1 metadata and selected specifies V2 metadata.
“Create dictionary files” option	This option has been clarified to read “Create dictionary files if they do not exist.”
“Code Page” option	This option has been added for encoding support. See “Database Code Page” on page 1-9.
“Open mode” as a DSN option	This option has been removed. By default, DSNs are created with an open mode of “normal.” The open mode can still be configured through ODBC Administrator. See “DSN Open Mode Options” on page F-3 in *SQL Engine Reference*.

Command Line Interface (CLI) Utilities

The CLI utility pvmdconv is new on Linux. The Linux utilities that have changed include dbmaint and dsnadd. The PSC utility (Windows only) has changed.

Pvmdconv

Pvmdconv converts V1 metadata to V2 metadata. See “pvmdconv” on page 8-33 in *Pervasive PSQL User's Guide*.

Dbmaint

Dbmaint is a utility for managing named databases on Linux. This utility now accepts a parameter, -c, to set database code page.

See “Encoding” on page 1-9 in this book and “dbmaint” on page 8-20 in *Pervasive PSQL User's Guide*.

Dsnadd

Dsnadd is a utility for creating and managing DSNs on Linux. This utility now accepts a parameter, translate, to specify the desired encoding option (“none” or “auto”).

See “Encoding” on page 1-9 in this book and“dsnadd” on page 8-23 in *Pervasive PSQL User's Guide*.

PSC

PSC is the Pervasive service controller utility that retrieves and sets control information about the Pervasive PSQL services on Windows. Pervasive PSQL no longer supports the PSC create and delete functions. The PSC utility is provided only with the Pervasive PSQL Control Center.

Access Methods (Software Development Kit)

This release of Pervasive PSQL includes changes to the following programming access methods:

- "ADO.NET" on page 1-30
- "Btrieve API" on page 1-30
- "Distributed Tuning Interface (DTI)" on page 1-30
- "Distributed Tuning Objects (DTO)" on page 1-31
- "JCL" on page 1-31
- "JDBC" on page 1-31
- "ODBC" on page 1-31

ADO.NET

The Pervasive PSQL ADO.NET data provider supports the following new features.

- Windows Vista running Pervasive PSQL Server, Workgroup or Client
- Pervasive PSQL Server 64-bit
- V2 metadata (see "Versions of Metadata" on page 2-2 in *SQL Engine Reference*)
- Terminal Server licensing (see "Terminal Server Licensing" on page 4-4 in *Pervasive PSQL User's Guide*)

See also *Data Provider for .NET Guide*.

Btrieve API

The Btrieve API includes support for 64-bit applications on Linux. See "Linux" on page 1-5.

The Btrieve API operation Set Owner (29) now supports a bias, +17000, to specify a long owner name. See "Security" on page 1-20.

Distributed Tuning Interface (DTI)

DTI includes support for 64-bit applications on Linux. See "Linux" on page 1-5.

New Functions

DTI contains new functions to support data translation encoding and database code page. See "DTI" on page 1-16.

Changes to Existing Functions

The functions PvGetProductsInfo and PvGetTable have changed.

PvGetProductsInfo

The “feature” and “edition” tags returned in the XML string have been deprecated. They always return zero. A new tag, “maxUserCount,” has been added to the *productinfo* string to support the Pervasive PSQL v10.10 licensing changes (see “Licensing” on page 1-8). See also “PvGetProductsInfo()” on page 2-154 in *Distributed Tuning Interface Guide*.

PvGetTable

The offset of a field within its row is now accessible through the PvGetTable() function. The COLUMNMAP structure has been modified in ddfstrct.h to contain this additional information. To use this new feature, existing applications need only to recompile against the DTI headers provided with Pervasive PSQL v10.10. This new field is ignored when calling the PvAddTable and PvFreeTable functions. Refer to ddfstrct.h and ddf.h, and see “PvAddTable()” on page 2-23 in *Distributed Tuning Interface Guide*.

Distributed Tuning Objects (DTO)

As a COM wrapper of DTI, DTO also supports the new DTI functions for data translation encoding and database code page. See “DTI” on page 1-16.

JCL

The Java Class Library (JCL) now supports the BIT data type. See “Pervasive PSQL Supported Data Types” on page A-2 in *SQL Engine Reference*.

JCL contains a new API, setDataFileLongOwner, to specify a long owner name. Refer to the javadocs that are installed as part of the JCL samples and header files. The samples and header files are a Web download archive available from the Pervasive Developer Center. See also “Security” on page 1-20 in this book.

JDBC

JDBC supports long owner names through the GRANT and SET OWNER SQL statements. See “Security” on page 1-20.

ODBC

ODBC supports database code page and the pvtranslate attribute in a connection string. See “Encoding” on page 1-9.

Documentation

The viewer for the documentation library has been integrated into Pervasive PSQL Control Center (PCC). The documentation library no longer runs standalone. On Windows, the documentation is no longer a separate command on the Start menu.

The documentation library is now accessed through the PCC interface on the Welcome view, in the Help menu, by pressing F1 (Windows) or Shift F1 (Linux). (The PCC properties for Help include an item for accessing help from a remote infocenter. The installation of the Pervasive PSQL documentation does not currently support an infocenter.)

The behavior of the help viewer differs from the previous viewer. Here are some usage tips to help you get the most out of the documentation library.

Index and Search

The help viewer contains both an index and a search feature. The index is more exacting in the results returned for specific terms. Use the index to find specific terms, such as “create database” when looking for the CREATE DATABASE statement. (Also use the index to find status codes. See “Finding a Status Code” on page 1-35.)

The search feature does pattern matching based on the search scope (see “Limiting the Scope of Search” on page 1-34). The default search scope is to search all topics in all books. Typically, the search feature is most handy for finding generic or broad categories of information. For example, if you are interesting in access methods, searching for “access methods” returns the numerous locations where they are discussed.

We recommend that you try the index first. If the search term is not in the index, then try the search feature. The next topic explains the syntax available for the search feature.

Search Query Syntax

Use the following search expression rules for searching the content of the Pervasive PSQL documentation.

- The search engine does “fuzzy” searches and word stemming. If you enter create, the search engine returns topics that contain “creates,” “creating,” “created,” and so on. To prevent the search engine from stemming terms, enclose the search term in double quotes.
- Unless otherwise stated, an implied AND is between all search terms. In other words, topics that contain all of the search terms are returned.

  For example: `database security` returns topics that contain the word “database” and the word “security,” but does not return topics that contain only one of these words.
- Use OR before optional terms.

  For example: `client OR connection` returns topics that contain the word “client” or the word “connection” (or both).
- Use NOT before terms you want to exclude from search results.

  For example: `PCC NOT Maintenance` returns topics that contain the word “PCC” and do not contain the word “Maintenance.”

Note NOT works only as a binary operator (that is, “NOT PCC” is an invalid expression).

- Use ? for a single-character wildcard and * for a multi-character wildcard.

  For example: `rebuil?` returns topics that contain “rebuild” or “rebuilt,” but not “rebuilding.” On the other hand, `rebuil*` returns topics that contain “rebuild,” “rebuilt,” “rebuilding” and so on.

Note The search engine does not accept terms with a wild card at first character position.

- Use double quotation marks around terms you want treated as a phrase.

  For example: `"transaction durability"` returns topics that contain the entire phrase "transaction durability," and not "transaction" or "durability" on its own.

- The search engine ignores character case.

  For example: Server returns topics that contain "Server," "server," and "SERVER."

- The following stop words are common words which are ignored (not searched for) if they appear in the search expression:

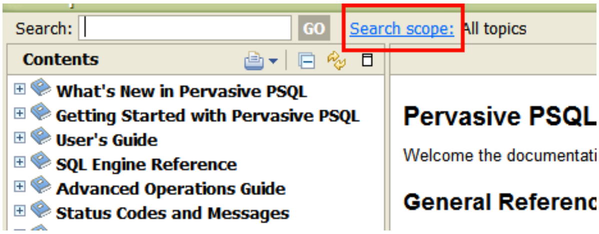
  a, and, are, as, at, be, but, by, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, to, was, will, with

Note You can search for these words if you enclose them in double quotes.

Limiting the Scope of Search

This feature is handy if you want to limit documentation searches to a single book or chapter, or to a desired group of books and chapters. The default is to search all topics in all books.

1. In PCC, click Help then PSQL Documentation Library.
2. Click Search scope.



3. Click Search only the following topics.
4. Click New.
5. In the List Name field, type a name for your search scope.

6 Click the topics (entire books or individual chapters) that you want to include in your search scope.

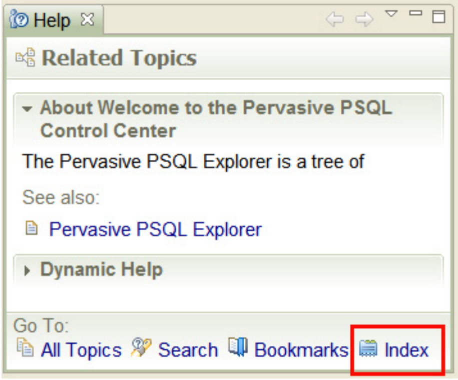
7 Click OK, then OK.

Finding a Status Code

Looking up a status code is quick and easy.

1 In PCC, press F1 (or Shift F1).

2 In the dynamic help view, click Index.



About Welcome to the Pervasive PSQL Control Center

The Pervasive PSQL Explorer is a tree of

See also:

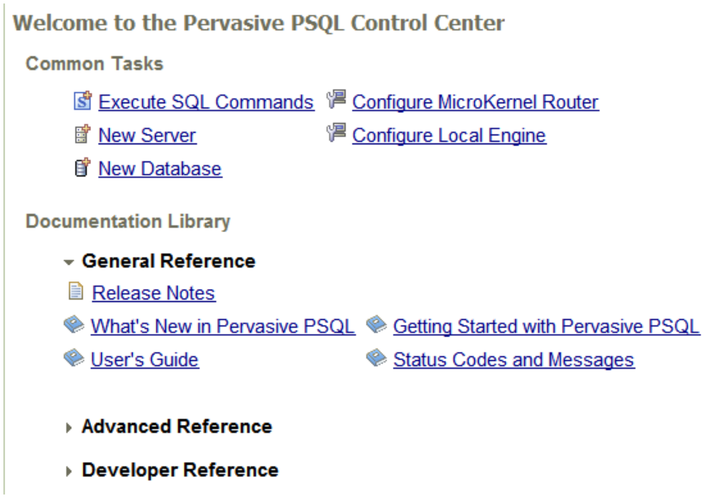
Pervasive PSQL Explorer

3 Type the desired status code number and press Enter.

Note The first time you access the index, the documentation system must build some local index files. A short delay occurs while the index files are being built. The delay occurs only the *first* time you access the index functionality.

Viewing Documentation

The PCC Welcome view lists the categories of documentation. Each category can be expanded or collapsed by clicking on the category name. For example, the following image shows the “General Reference” category expanded, and the “Advanced Reference” and “Developer Reference” categories collapsed.



Welcome to the Pervasive PSQL Control Center

Common Tasks

Execute SQL Commands
Configure MicroKernel Router
New Server
Configure Local Engine
New Database

Documentation Library

General Reference

Release Notes
What's New in Pervasive PSQL
Getting Started with Pervasive PSQL
User's Guide
Status Codes and Messages

Advanced Reference

Developer Reference

Viewing Documentation in a Separate Window

You may prefer to view the documentation in a separate window rather than as an additional page within PCC. A separate window offers the advantage of sizing and positioning the documentation without crowding other pages within PCC.

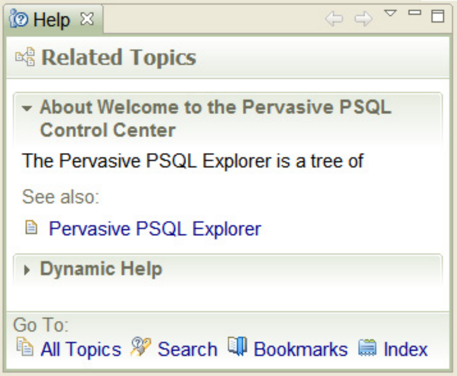
1. In PCC, click Help then PSQL Documentation Library.
2. Size and position the window as desired.

Changing How Context Help Displays

You can change how context help displays in the viewer. The default is *dynamic help*, which displays as a view within PCC.

1. In PCC, click Window then Preferences.
2. Click the Help node in the tree on the left.
3. Set the properties as desired.

   Dynamic help displays as a view within PCC:



Related Topics

About Welcome to the Pervasive PSQL Control Center

The Pervasive PSQL Explorer is a tree of

See also:

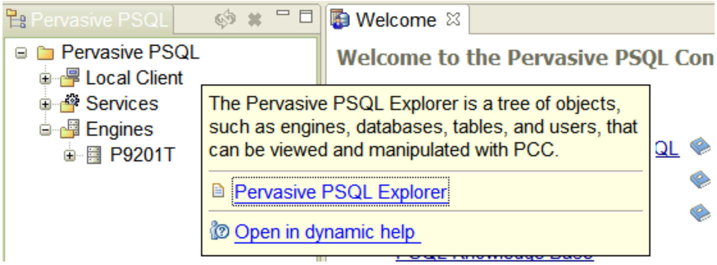
Pervasive PSQL Explorer

Dynamic Help

Go To:

All Topics Search Bookmarks Index

If desired, you can display the content as a pop-up dialog called an *infopop*:



Welcome to the Pervasive PSQL Con

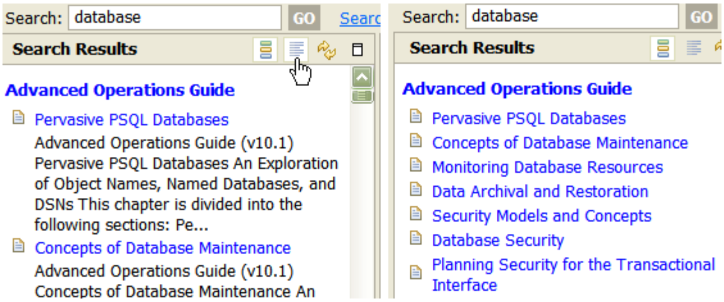
The Pervasive PSQL Explorer is a tree of objects, such as engines, databases, tables, and users, that can be viewed and manipulated with PCC.

Pervasive PSQL Explorer

Open in dynamic help

Viewing Search Results

You can show or hide descriptions for search results by clicking the Show result descriptions icon:



Search: database GO Searc

Search Results

Advanced Operations Guide

Pervasive PSQL Databases
Advanced Operations Guide (v10.1) Pervasive PSQL Databases An Exploration of Object Names, Named Databases, and DSNs This chapter is divided into the following sections: Pe...

Concepts of Database Maintenance
Advanced Operations Guide (v10.1) Concepts of Database Maintenance An

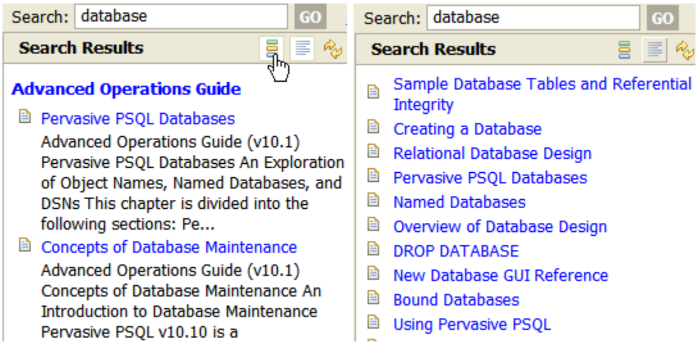
Search: database GO

Search Results

Advanced Operations Guide

Pervasive PSQL Databases
Concepts of Database Maintenance
Monitoring Database Resources
Data Archival and Restoration
Security Models and Concepts
Database Security
Planning Security for the Transactional Interface

You can change the categories by which the results are sorted by clicking the Show result categories icon:



Search: database GO

Search Results

Advanced Operations Guide

Pervasive PSQL Databases
Advanced Operations Guide (v10.1) Pervasive PSQL Databases An Exploration of Object Names, Named Databases, and DSNs This chapter is divided into the following sections: Pe...

Concepts of Database Maintenance
Advanced Operations Guide (v10.1) Concepts of Database Maintenance An Introduction to Database Maintenance Pervasive PSQL v10.10 is a

Search: database GO

Search Results

Sample Database Tables and Referential Integrity
Creating a Database
Relational Database Design
Pervasive PSQL Databases
Named Databases
Overview of Database Design
DROP DATABASE
New Database GUI Reference
Bound Databases
Using Pervasive PSQL

Finding Text Within a Displayed Topic

You can find text within a documentation topic, such as a chapter within a book, using the standard “find” command. (This differs from searching for text with the Search command.)

1 Within a topic, press Ctrl+f to display the Find dialog.

2 Type the text you wish to find and click Find Next.

Status Codes

This section lists the new and revised status codes.

New

The following categories contain new status codes:

- Microkernel Database Engine on page 1-39
- Distributed Tuning Interface on page 1-39
- License Administrator on page 1-40

Microkernel Database Engine

120: Maximum number of B-Tree index levels reached

This status code may result if you use a large key size for an index and small page sizes. Index keys can fill the B-Tree to the allowable depth even though the B-Tree is not completely full.

To prevent this status code, try one or both of the following:

- Rebuild the data file with a larger page size to increase the number of keys stored per page.
- Turn on index balancing to maintain a better distribution of index keys (performance decreases somewhat with index balancing on). See “Index Balancing” on page 4-45 in *Advanced Operations Guide*.

Distributed Tuning Interface

7043 Invalid character sent by client

One or more characters sent by the client in a DTI function argument string are invalid and cannot be translated to the server’s encoding. Verify the function arguments.

7044 Invalid character received by client

One or more characters received from the server in a DTI function argument string are invalid and cannot be translated to the client’s operating system encoding. It may be necessary to use a different client machine.

7045 Internal buffer too small

An internal buffer is too small to process a DTI argument. One possible workaround is to use shorter arguments.

7046 Invalid encoding translation option

The translation option in the DTI function “PvCreateDSN2()” or “PvModifyDSN2()” specifies an invalid encoding. For the valid options, see the argument “translate” on page 2-55 in *Distributed Tuning Interface Guide*.

7047 Invalid code page for database

The code page specified for the database is invalid (not supported by the database engine). Verify the code page and use one supported by the database engine. The code pages supported can be viewed from the Create Database dialog within Pervasive PSQL Control Center. See “New Database GUI Reference” on page 3-22 in *Pervasive PSQL User's Guide*.

License Administrator

7124: A User Count Increase cannot be applied without a permanent license.

The machine on which you are attempting to apply a user count increase does not contain a permanent license key. You may not apply a user count increase without a permanent license.

Apply a permanent license key prior to applying a user count increase.

7125: A permanent license already exists.

The machine on which you are attempting to apply a permanent license key already contains a permanent license key. You may not apply more than one permanent license key.

If a permanent license key has already been applied, you may only apply a user count increase. Remove the existing license key and then apply the new permanent license key.

7126: Cannot increase the user count beyond the maximum allowed.

The user license increase you are attempting to add would increase the user count beyond the maximum number of users allowed. You cannot increase the user count beyond the maximum count allowed.

Verify the user license increase count.

Revised

The following status codes have been revised for the MicroKernel Database Engine.

86: The file table is full

Explanation about this status code has been added. The documentation erroneously stated that this status code was obsolete.

113: The MicroKernel is unable to open the archival log for the specified file

The wording of this status coded changed from “The Microkernel cannot find the archival log for the specified file” to more accurately reflect the error conditions.

Appendix

A

What Was New in Pervasive PSQL v10

An Overview of Features in Pervasive PSQL v10

This chapter summarizes and explains the major new features and differences in behavior for Pervasive PSQL v10. Where applicable, links are provided to additional information for a given feature.

List of New Features and Improvements

This release includes the following new features and changes:

- “Support for Windows Vista” on page A-3
- “Support for 64-bit Operating Systems and Applications” on page A-6
- “Database Accelerator” on page A-9
- “Relational Interface Support” on page A-17
- “Utilities” on page A-22
- “Installation” on page A-10
- “Configuration Settings” on page A-25
- “Page Compression” on page A-27
- “Access Methods (Software Development Kit)” on page A-29
- “File Format and Page Size” on page A-32
- “Network Communications” on page A-33
- “Documentation” on page A-35
- “Operating System Support” on page A-36
- “Status Codes” on page A-37
- “Complimentary Data Management Products” on page A-39
- “Deprecated Features” on page A-40

These features are described in the sections that follow. Also see the Readme file for additional information about this release that *What’s New* may not contain.

Support for Windows Vista

In many aspects, Windows Vista is a departure from previous Windows operating systems. Among other features, Vista includes many changes to the Graphical User Interface (GUI), a new security model using User Account Controls (UAC), changes to the Windows firewall and additional Internet protection.

You may find that your software products—not just Pervasive PSQL products—behave differently under Vista because of the operating system itself.

Tips When Using Vista

Here are some tips for working with Windows Vista to help you use applications, including Pervasive PSQL applications.

- Initially, client/server applications may not function correctly because of Vista's increased security. You may need to adjust some of Vista's security settings to enable client/server applications. See "Security and File Permissions" on page A-5.
- Windows Vista uses two main types of user accounts: standard users and administrators. The permissions differ for these two types of users.

  If a standard user needs to perform a task that requires additional privileges, Vista prompts (run as administrator) for an administrator password.

  Vista includes a built-in administrator, admin, whose account is disabled by default. The Vista installation requires that you create a secondary administrator account. You may also create additional administrator accounts after Vista is installed.
- You can create system data source names (DSNs) only if logged on as an administrator. A standard user cannot create system DSNs.
- Floating point precision has changed with Windows Vista. You may notice a change in the results from your application. Refer to the documentation for Windows Vista.

Pervasive PSQL and Vista

The Pervasive PSQL Server Engine, Workgroup Engine, and Client are Vista enabled for installation and execution. (See "Operating System Support" on page A-36 for a list of all supported platforms.)

This section discusses topics of which you should be aware when you install and use Pervasive PSQL on Windows Vista.

Types of Users

In addition to the "Tips When Using Vista" mentioned previously, the following behavioral differences with Pervasive PSQL occur depending on how you are logged on:

- If you are logged on as a standard user, Vista prompts for an administrator password if you attempt to install Pervasive PSQL.
- The Pervasive PSQL utilities create only system data source names (DSNs), not user DSNs. The creation of system DSNs requires that you are logged on as an administrator.

Internet Protocol

Pervasive PSQL supports Internet Protocol (IP) v4, not IP v6.

Tip Consider changing the operating system default configuration from IP v6 to IP v4. Using IP v4 as the default prevents the delay caused by the database engine when the operating system network layer attempts to use IP v6 then must revert to IP v4. Refer to the operating system documentation for setting the IP default.

Configuration Settings

The registry configuration settings allow full access to all users. Pervasive PSQL configuration settings are persistent for all user types on Vista.

Security and File Permissions

Windows Vista enables the firewall by default. The Pervasive PSQL Server and Workgroup installation adds files to the firewall access list to enable access to the database engine. If the operating system security prompts you to unblock or allow communication with a Pervasive PSQL component, select OK (yes).

If you encounter problems with your client/server applications not working correctly after installation, check the firewall access list or the ports. You may need to adjust some of the security settings to enable client/server applications. You can add files to the access list or open ports. You do not need to do both.

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

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

If you want to opens ports, Pervasive PSQL communicates via the following ones: 3351 for the transactional interface, 1583 for the relational interface, and 139 for named pipes. Note that opening a port opens it for all access, not just for Pervasive PSQL.

Refer to the operating system documentation on security, firewall settings, and ports.

Workgroup Engine and Cache Engine

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

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

If it is desirable to have multiple local users accessing a local engine, install the Workgroup or Cache Engine as a service. That way, the engine is already running before the first user session starts.

Support for 64-bit Operating Systems and Applications

Pervasive PSQL supports the transactional interface on native 64-bit operating systems running on machines with 64-bit architecture. The support applies only to the transactional interface and to the SDK interfaces Btrieve and DTI (see “SDK Interfaces”). The relational interface is 32-bit only. The relational interface functions the same on 64-bit operating systems as it has in previous 32-bit releases.

Database Products

Pervasive Software now offers 32-bit and 64-bit versions of the Server Engine and the client. The Workgroup Engine is available only in a 32-bit version. The following table summarizes the platforms on which the products can be installed and the type of applications supported by the product.

Table A-1 Install Platforms and Application Support

Product	Install Platform		Local Applications[1]		Remote Applications[1]	
	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
Server Engine 32-bit	✓	✓[2]	✓		✓	✓
Server Engine 64-bit[3]		✓	✓	✓	✓	✓
Workgroup Engine[4]	✓	✓	✓		✓	✓
Client 32-bit	✓	✓	✓			
Client 64-bit		✓	✓	✓		

[1] “Local” and “remote” are relative to the product. For example, a client can run an application that is local to the client itself but remote to the server engine.

[2] The 32-bit Server installed on a Windows 64-bit machine runs under the Windows-on-Window (WOW) execution layer.

[3] Only the transactional interface is 64-bit. The relational interface is 32-bit.

[4] The Workgroup Engine is 32-bit only. A 64-bit version is not available.

Note that the 32-bit Pervasive PSQL products are supported on 64-bit operating systems. In addition, the 64-bit transactional engine and 64-bit DTI and Btrieve API are supported on 64-bit operating systems. Application created with a 32-bit interface can be used with the 64-bit Server engine.

SDK Interfaces

The 64-bit client supports 64-bit applications that use the Btrieve API or the distributed tuning interface (DTI). Other software development kit (SDK) interfaces are not supported for 64-bit access.

The header files for the Btrieve API and DTI have been enhanced to support 64-bit applications. For a 64-bit application using the Btrieve API, you need to define the preprocessor symbol BTI_WIN_64 (for Windows platforms) and link against w64btrv.lib.

For a 64-bit application using DTI, link against w6dba.lib.

Chunk Operations

Chunk descriptors are larger in size when used in a 64-bit application than when used in a 32-bit application. Pointers are 64 bit (8 bytes wide) in a 64-bit application.

Application	Chunk Offset (bytes)	Chunk Length (bytes)	User Data Pointer (bytes)
32-bit	4	4	4
64-bit	4	4	8

As with previous releases of Pervasive PSQL, the User Data field is only used in indirect chunk descriptors. You should initialize it to zero in direct chunk descriptors. Refer to the various chunk operations in the “Btrieve API Operations” chapter in *Btrieve API Guide*.

Registry

Registry changes apply more to the operating system and less to how your interact with Pervasive PSQL. However, because Pervasive PSQL now runs on 64-bit environments, some general information about the registry is useful background.

On 64-bit operating systems the registry is split at certain important nodes into a 32-bit section and a 64-bit section. Access to keys is redirected by default to the appropriate section depending on whether the calling application is 32-bit or 64-bit. The Windows API allows applications to request which specific section to access. Refer to your operating system documentation for specifics about registry architecture on 64-bit platforms.

The Pervasive PSQL components transparently access the 32-bit or 64-bit section of the registry as required.

Pervasive PSQL Settings

When Pervasive PSQL v10 is installed on a 64-bit Windows operating system, most components store their registry entries in the 64-bit section of the registry. In the 64-bit version of Pervasive PSQL, both 32-bit and 64-bit versions of certain components are present. Both versions read their settings from the same section of the registry. For example, if you enable debug tracing for the client components, the tracing applies to both 32-bit and 64-bit client components.

To read or modify commonly used settings, use Pervasive PSQL Control Center or the DTI API.

Database Accelerator

By default, the Server Engine installation installs a database accelerator called Xtreme I/O (XIO) if the system meets the minimum requirements for XIO. See “System Requirements” on page 5-20 in *Advanced Operations Guide*.

XIO increases database performance by accelerating disk access time for Pervasive PSQL data files. XIO and the database engine work together transparently—no intervention is required by a user or an application. The larger the data set size compared to the size of the Windows system cache, the more performance improvement XIO provides.

For a custom installation, you can omit XIO if you choose.

XIO provides two utilities for working with the driver: xiomgr and xiostats. Note that, on Windows Vista, you must have administrative privileges to run xiostats.

See “Xtreme I/O Driver” on page 5-19 in *Advanced Operations Guide*.

Installation

This section discusses the enhancements to the Pervasive PSQL installation:

- “Installer Technology” on page A-10
- “Product Installations” on page A-10
- “Application or Service for Workgroup or Client Cache” on page A-12
- “Default Location of Installed Components” on page A-13
- “PSA” on page A-13
- “Service Names” on page A-14
- “Upgrade” on page A-15
- “Uninstall” on page A-15
- “Non-interactive (Silent) Install Examples” on page A-16

See also the Readme file provided with the product release for information that may not be included in this book.

Installer Technology

The Pervasive PSQL installation on all Windows operating systems now uses Microsoft Windows Installer technology (MSI) for product installations, updates, and uninstalls. The MSI installer (msiexec.exe) uses the configuration settings specified in the ptksetup.ini file for customized installations.

The MSI technology provides better deployment and a standard format for component management. The overall flow and look-and-feel of the installation is similar to previous releases. The underlying technology, however, is new and improved.

See “MSI Technology” on page 1-2 in *Installation Toolkit Handbook*.

Product Installations

A separate installation exists for each of the following:

- Server 32-bit
- Server 64-bit
- Workgroup
- Client 32-bit
- Client 64-bit

For each component, the installation program is uniquely named in the format setup*product-type*.exe, where *product-type* indicates the type of product. For example, the program for the 32-bit Server product is SetupServer_x86.exe.

As with previous version of Pervasive PSQL, only a single database engine may be installed on the same machine at a time. Note, however, that both the 32-bit and the 64-bit Client can be concurrently installed on the same machine provided the machine is running a 64-bit operating system.

Authenticity of Executable Files

All Pervasive PSQL executable files, including the installation program, are now digitally signed. Some Windows operating systems check executable files to verify the authenticity of the executable. On such platforms, the authentication shows that the executable is from Pervasive Software Inc.

Client Components

A client installation image is not included as a component of the Pervasive PSQL Server product. You must run the separate Client setup program to install a Pervasive PSQL Client on a machine that does not include the Pervasive PSQL Server or Workgroup product. The Pervasive PSQL Server and Workgroup products include the client components so a separate client installation is not necessary.

Data Access Components

The data access methods are part of the Pervasive PSQL software development kit (SDK). A complete installation for any of the products also installs the runtime components for all of the data access methods:

- ActiveX
- ADO.NET
- Btrieve API
- Distributed Tuning Interface (DTI)
- Distributed Tuning Objects (DTO)

- Java Class Library (JCL)
- JDBC
- OLE DB
- Pervasive PSQL Direct Access Components (PDAC)

A custom installation allows you to include or exclude specific access methods.

The headers, libraries, samples and so forth for each access method remain separate downloads from the Pervasive PSQL Web site.

Java Runtime Environment

For a complete installation, the Pervasive PSQL installation installs Java Runtime Environment version 6 (JRE 6). Previous versions installed JRE 5. The JRE is an optional component for a custom installation depending on which Pervasive PSQL components you select. Some components, such as Pervasive PSQL Control Center and DDF Builder, require the JRE.

Because of limitations with Microsoft Windows Installer technology, the JRE installation cannot be performed during a silent installation of Pervasive PSQL v10. You must install the JRE separately for a silent installation.

Security and File Permissions

Certain Windows operating systems enable the firewall by default. On such platforms, the Pervasive PSQL installation adds executable files to the firewall access list. With the firewall enabled, users cannot access the database engine unless certain engine files are added to the firewall access list.

Refer to the operating system documentation about firewalls and access lists.

Application or Service for Workgroup or Client Cache

The installation for Workgroup Engine and for Client Cache Engine now offers a choice to install the engine as an application or as a service. Previous versions of Pervasive PSQL installed the engine as an application.

The default is to install the engine as an application unless Terminal Services is detected on the machine. If you install either engine on a

Terminal Services machine, the default is to install the engine as a service using the “Local System” account.

If you install the product as a service, you must provide the installation program with an account under which the service runs. This can be the “Local System” account or a specific user account.

Default Location of Installed Components

The default location of installed components differs in Pervasive PSQL v10 from previous releases. This change allows the Pervasive PSQL products to adhere to installation guidelines provided by the operating system vendor.

Installation now places files into different root locations depending on the product and platform. (See “Where are the Pervasive PSQL files installed?” on page 7-2 in *Getting Started With Pervasive PSQL*.)

Table A-2 Default Locations of Installed Components

Contents	**Default Location**[1]	**On Windows Platform**	
		Vista	**2003, 2000, XP**
On 32-bit platforms, binary and system files used by Pervasive PSQL, such as executable files, dynamic link libraries, JAR files, client components, and so forth. On 64-bit platforms, only the binary and system files required for 64-bit platforms.	C:\Program Files\Pervasive Software\PSQL\subfolders	yes	yes
On 64-bit platforms, 32-bit binary and system files used by Pervasive PSQL, such as executable files, dynamic link libraries, JAR files, and so forth.	C:\Program Files (x86)\Pervasive Software\PSQL\subfolders	yes, if 64-bit platform	yes, if 64-bit platform
Demodata sample database, DefaultDB system database, Pervasive PSQL log files, sample files, archived versions of previous installed products, and so forth.	C:\ProgramData\Pervasive Software\PSQL\subfolders[2]	yes	no
	C:\Documents and Settings\All Users\Application Data\Pervasive Software\PSQL\subfolders[2]	no	yes

[1] During install, you can specify the location where you want the Pervasive PSQL products. This location could be “C:\PVSW” if you choose. However, we recommend that you use the new default locations.

[2] By default, the operating system hides **C:\ProgramData** and **C:\Documents and Settings\All Users\Application Data**. To display these directories, change the folder options to “show hidden files and folders.” Refer to the operating system documentation for “folder options.”

PSA

The Pervasive System Analyzer (PSA) is now installed as part of the other Pervasive PSQL utilities and resides in the same location as the other utilities. Also, during installation, PSA no longer runs its tests of the network and database engines. (Many of the features formerly provided by PSA are now available through the Microsoft Windows Installer technology.) You may run those tests after installing the product if you choose.

Note that PSA no longer provides the following functionality:

- Archive the product
- Analyze an archive
- Restore components or an archive
- Delete components or an archive. The Pervasive PSQL uninstall program now handles all deletion requirements.

See "Pervasive System Analyzer (PSA)" on page 7-1 in *Pervasive PSQL User's Guide*.

Service Names

The *display* names of the Pervasive PSQL transactional and relational services have changed. The display name is what you see in Pervasive PSQL Control Center and in Windows Services, for example. The *service* names—the name that identifies the service to the operating system—remain the same.

In addition, Pervasive PSQL v10 allows you to install the Workgroup Engine or the Cache Engine as a service. The service name and the display name for both products are new.

The following table summarizes this information.

Table A-3 Names of Pervasive PSQL Services

Component	Service Name	Display Name
Transactional Engine	Pervasive.SQL (transactional)[1]	Pervasive PSQL Transactional Engine
Relational Engine	Pervasive.SQL (relational)[1]	Pervasive PSQL Relational Engine
Workgroup Engine	psqlWGE[2]	Pervasive PSQL Workgroup Engine
Cache Engine	psqlCE[2]	Pervasive PSQL Cache Engine

[1] Same as in previous versions of Pervasive PSQL

[2] If product installed as a service

Because the service names remain the same, any scripts or applications you coded that start or stop the Pervasive PSQL services should work as before provided the code refers to the services as Pervasive.SQL (transactional) and Pervasive.SQL (relational).

Upgrade

Pervasive PSQL v10 installs to its default locations, *not* to the location where the previous PSQL version was installed. See “Default Location of Installed Components” on page A-13.

Versions prior to Pervasive PSQL v10 installed to a default location of C:\PVSW. Pervasive PSQL v10 no longer uses that default location.

The Pervasive PSQL installation archives the previous version of the product and places the archive in a folder under the application data location. See Table A-2 on page A-13 in this book, and “Where are the Pervasive PSQL files installed?” on page 7-2 in *Getting Started With Pervasive PSQL*.

Uninstall

The uninstall program removes all of the Pervasive PSQL components except for the following:

- Dbnames.cfg. Note, however, that entries in dbnames.cfg that were added by the Pervasive PSQL v10 installation are removed. The file itself is retained, as are entries resulting from user-initiated actions such as the creation of a new data source name (DSN).
- License keys. Product license keys are never uninstalled.
- User-created data source names (DSNs).

User-created databases and data files are not removed.

An uninstall removes all Pervasive PSQL settings, including user-modified ones. If you reinstall the product, the default settings are applied.

Uninstalling the Workgroup or Cache Engine and the Service

If you set up the Workgroup Engine or Cache Engine as a service through the Pervasive PSQL installation, the Pervasive PSQL uninstall removes the service as well as the product.

However, it is possible to set up a service by other means, such as third-party utilities or Pervasive PSQL utilities. For those situations, the Pervasive PSQL uninstall may not detect the service. After uninstalling Pervasive PSQL, check if the Workgroup Engine or Cache Engine service still exists. If it does, delete the service through the operating system Service Manager or by some other means.

Non-interactive (Silent) Install Examples

See “Installing Using PTKSetup.ini Settings” on page 1-13 in *Installation Toolkit Handbook*.

Relational Interface Support

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

New Functionality and Features	Revised Functionality and Features
“Metadata Versions” on page A-17	“GRANT and REVOKE” on page A-21
“CREATE DATABASE Statement” and “DROP DATABASE Statement” on page A-20	“SET PASSWORD” on page A-21
“Partial Indexes” on page A-20	“String Functions” on page A-21
“ORDER BY in VIEW” on page A-20	
“SET DEFAULTCOLLATE” on page A-20	
“@@SESSIONID Variable” on page A-20	

Metadata Versions

The Pervasive ODBC Engine Interface in Pervasive PSQL v10 supports two versions of metadata, referred to as version 1 (V1) and version 2 (V2).

Metadata version is a property of the database that you specify when you create a database. V1 metadata is the default. When you create a database, you must specify V2 metadata if you want that version.

Metadata version applies to all data dictionary files (DDFs) within that database. A single database cannot use some DDFs with V1 metadata and others with V2 metadata. DDFs from the two versions cannot interact.

The database engine can, however, concurrently access multiple databases and each database can use either V1 metadata or V2 metadata. See

A conversion utility is available to convert V1 metadata to V2 metadata. See “pvmdconv” on page 8-33 in *Pervasive PSQL User's Guide*.

Comparison of Metadata Versions

The primary features of V2 metadata include the following:

- Identifier names up to 128 bytes long for many identifiers. See "Identifier Restrictions by Identifier Type" on page 1-3 in *Advanced Operations Guide*.
- Permissions on views and stored procedures. See below.
- Data dictionary files (DDFs) specific for V2 metadata. The DDFs are named differently than for V1 and in many cases contain additional fields and changes to existing fields. See "System Tables" on page C-3 in *SQL Engine Reference*.

Permissions on Views and Stored Procedures

Permissions can now be specified for views and stored procedures. In addition, views and stored procedures can be trusted or non-trusted, depending on how you want to handle the permissions for the objects referenced by the view or stored procedure.

A *trusted* view or stored procedure is one that can be executed without having to explicitly set permissions for each referenced object. A *non-trusted* view or stored procedure is one that cannot be executed without having to explicitly set permissions for each referenced object.

See "Permissions on Views and Stored Procedures" on page 3-158 in *SQL Engine Reference*.

Additional Reading

See the following areas in the documentation for details about metadata versions.

Table A-4 Metadata Versions References

Area of Discussion	Related Content
Concepts	"Versions of Metadata" on page 2-2 in *SQL Engine Reference*
Identifiers	"Identifier Restrictions by Identifier Type" on page 1-3 in *Advanced Operations Guide*
Conversion Utility	"pvmdconv" on page 8-33 in *Pervasive PSQL User's Guide*
New Database Wizard	"New Database GUI Reference" on page 3-22 in *Pervasive PSQL User's Guide*
Create Database from ODBC Setup	The online help from the ODBC Setup dialogs when you create a Pervasive PSQL engine or client DSN.
Relational Interface Conditions	"Relational Interface Limits" on page 2-2 in *SQL Engine Reference*
Object Permissions	"Permissions on Views and Stored Procedures" on page 3-158 in *SQL Engine Reference*.
System Tables	"System Tables" on page C-3 "System Tables Structure" on page C-4 Both references are in *SQL Engine Reference*
SDK Changes	"PvCreateDatabase()" on page 2-43 "PvGetDbFlags()" on page 2-105 "PvModifyDatabase()" on page 2-201 All references are in *Distributed Tuning Interface Guide*

CREATE DATABASE Statement

A new statement, CREATE DATABASE, allows you to create a database using SQL syntax. See “CREATE DATABASE” on page 3-46 in *SQL Engine Reference*.

DROP DATABASE Statement

A new statement, DROP DATABASE, allows you to delete a database using SQL syntax provided the database does not use the “database” security model. See “DROP DATABASE” on page 3-134 in *SQL Engine Reference*.

Partial Indexes

A new keyword, PARTIAL, allows you to create partial indexes on CHAR or VARCHAR columns larger than 255 bytes, as long as the CHAR or VARCHAR column is the last or only column in the Index.

The CREATE INDEX statement allows you to use PARTIAL to create partial indexes.

See “PARTIAL” on page 3-192 and “CREATE INDEX” on page 3-61 in *SQL Engine Reference* and “Index Attributes” on page 12-15 in *Pervasive PSQL Programmer's Guide*.

ORDER BY in VIEW

The CREATE VIEW syntax now supports the ORDER BY clause. See “CREATE VIEW” on page 3-113 in *SQL Engine Reference*.

SET DEFAULTCOLLATE

A new statement, SET DEFAULTCOLLATE, allows you to specify the collating sequence file to use for all columns of data type CHAR, VARCHAR, or LONGVARCHAR.

See “SET DEFAULTCOLLATE” on page 3-245 in in *SQL Engine Reference*.

@@SESSIONID Variable

A new global variable, @@SESSIONID, has been added. This variable returns an eight-byte integer value for the Pervasive PSQL connection. The integer is a combination of a time value and an incremental counter. This variable can be used to identify uniquely each Pervasive PSQL connection.

See “@@SESSIONID” on page 3-300 in *SQL Engine Reference*.

GRANT and REVOKE

The GRANT and REVOKE statements now allow you to grant or revoke permissions on views and stored procedures in addition to tables. The asterisk (*) in their syntaxes now refers to all objects: tables, views and stored procedures.

See “GRANT” on page 3-154 and “REVOKE” on page 3-200, both in *SQL Engine Reference*.

SET PASSWORD

The SET PASSWORD statement now permits a normal user (non-Master user) to change his or her logon password to the database. The user must be logged on the database to issue the statement. The changed password takes effect the next time the user logs on the database.

See “SET PASSWORD” on page 3-250 in *SQL Engine Reference*.

String Functions

The string functions now support multiple-byte character strings. See “String Functions” on page 5-2 in *SQL Engine Reference*.

Note, however, that the CASE (string) keyword does not support multiple-byte character strings. The CASE keyword instructs the database engine to ignore case when evaluating string columns (case insensitive). The keyword assumes that the string data is single-byte ASCII. See “CASE (string)” on page 3-39 in *SQL Engine Reference*.

Utilities

This section discusses the additions and changes to Pervasive PSQL utilities.

GUI Utilities

The GUI utilities include a new one, DDF Builder, and changes to PCC, License Administrator, and License Generator.

DDF Builder

DDF Builder is a Windows-based utility that allows you to view, create, and change Pervasive PSQL data dictionary files (DDFs) without modifying the underlying data file. The utility's primary purpose supports the following:

- Creating the table definitions required to enable relational access to data files
- Modifying existing table definitions to ensure that relational access is enabled correctly for data files.

Note that DDF Builder is a specialized utility intended for advanced users and is not a utility that you would typically use daily.

DDF Builder is included as part of the complete installation and is an option under the custom installation.

For details, including tutorials, refer to the DDF Builder documentation that is accessed from the utility.

Check Table Consistency

DDF Builder provides functionality with which you check the consistency of a table. A consistency check uses a set of validation rules to compare the physical data file against its metadata (the table against the data dictionary files). See "Check Table Consistency" on page 2-24 in *DDF Builder User's Guide*.

Pervasive PSQL Control Center

PCC provides enhanced functionality to its Index Editor and New Database wizard, wizards to export data and import data, and user assignments to groups.

Index Editor

The Index Editor in PCC now supports creating partial indexes. The editor also now uses the `unique` option instead of `allow duplicates`.

See "To create a unique index" on page 5-26 and "To create a partial index" on page 5-28 in *Pervasive PSQL User's Guide*, and "PARTIAL" on page 3-63 in *SQL Engine Reference*.

New Database Wizard

The New Database wizard contains options to specify V1 metadata or V2 metadata. The default is V1 metadata. See "Metadata Versions" on page A-17.

Export Data Wizard and Import Data Wizard

The Export Data wizard exports data from a table to a text file. This wizard has been streamlined from its previous version and provides a more efficient interface.

The Import Data wizard reads delimited data from a text file and adds the data to a table. This wizard is new in Pervasive PSQL v10.

Both wizard are accessed by right-clicking on a table name in Pervasive PSQL Explorer.

See "Exporting Data with Export Data Wizard" on page 3-32 and "Importing Data with Import Data Wizard" on page 3-31 and *Pervasive PSQL User's Guide*.

User Assignments to Groups

Existing users can now be added to a group. In previous releases, a user could be added to a group only when creating the new user. See "To assign a user to a group using Pervasive PSQL Explorer" on page 3-46 in *Pervasive PSQL User's Guide*.

License Administrator

License Administrator now accepts and displays license keys specific to platforms. Such keys restrict the use of the database engine to an operating system(s) and the bit architecture of the operating system(s).

For example, if your license key has a platform of "Win64," you can apply that license to a database engine running only on a Windows

64-bit platform. The license is invalid for Windows 32-bit platforms. Other types of platform keys permit product use on all operating systems supported by Pervasive PSQL.

See “Pervasive PSQL Key Platforms” on page 4-2 in *Pervasive PSQL User's Guide*.

License Generator

The License Generator utility now supports license generation for specific platforms that includes the bit architecture of the platform (the licenses are referred to as platform specific license keys). See “Pervasive PSQL Key Platforms” on page 4-2 in *Pervasive PSQL User's Guide*.

Note that License Generator is available only to Pervasive Software Original Equipment Manufacture (OEM) Partners.

Command Line Interface (CLI) Utilities

The CLI utilities include new utilities pvmdconv and psc, and changes to pvddl.

Pvmdconv Utility

This utility provides functionality to convert V1 metadata to V2 metadata.

See “Metadata Versions” on page A-17 and “pvmdconv” on page 8-33 in *Pervasive PSQL User's Guide*.

Psc Utility

The psc utility replaces the btisc utility. Psc retrieves and sets control information about Pervasive PSQL services. Note that, on Windows Vista, you must have administrative privileges to run psc.

See “psc” on page 8-26 in *Pervasive PSQL User's Guide*.

Pvddl Utility

Pvddl, which is used to execute a series of SQL statements in a command file, has two new options. One option allows you to specify the character separator to use between SQL statements in the command file. The other option allows you to log output to a file instead of to standard output (stdout).

See “pvddl” on page 8-31 in *Pervasive PSQL User's Guide*.

Configuration Settings

This section discusses the configuration settings that have changed in Pervasive PSQL v10.

Cache Allocation Size

The value required and returned for the cache allocation size varies depending on the combination of Pervasive PSQL server and clients, as show in the following table.

Table A-5 Cache Allocation Size Value By Product Version

PSQL Server		PSQL Client		Value Required and Returned		Example: 1 Gigabyte (GB) Cache
v10	Prior to v10	v10	Prior to v10	Bytes	Megabytes (MB)	
✓		✓			✓	1,024
			✓	✓		1,073,741,824
	✓	✓		✓		1,073,741,824
			✓	✓		1,073,741,824

When you configure the setting, you enter a value either in bytes or megabytes depending on the combination of server and client as shown above. The Pervasive PSQL utilities inform you of the units required and returned. For example, if you use a Pervasive PSQL v10 server with v10 clients, the Pervasive PSQL Control Center displays the setting as “Cache Allocation Size in megabytes.” For any other combination of server and clients, the Pervasive PSQL Control Center displays the setting as “Cache Allocation Size in bytes.”

Also note that the limit for cache allocation size in 64-bit Pervasive PSQL is 4 terabytes. The limit is 4 gigabytes in 32-bit Pervasive PSQL.

Using Distributed Tuning Interface (DTI), cache allocation size is configured the same as any other setting. In *Distributed Tuning Interface Guide*, see “Obtaining a Setting ID Using DTI” on page 1-8 and “PvSetSelectionValue()” on page 2-224.

Supported Protocols

The engine configuration setting “Supported Protocols” now defaults to TCP/IP. (The client setting still defaults to all three supported protocols.) See “Supported Protocols” on page 4-26 in *Advanced Operations Guide*.

Page Compression

Pervasive PSQL v10 provides two types of data compression: record and page. Record compression was formerly termed “data compression.”

Record compression and page compression may be used separately or together. The primary purpose for both compression types is to reduce the size of the data files and to provide faster performance depending on the type of data and on the type of data manipulation.

Record compression requires a file format of 6.0 or later. Page compression requires a file format of 9.5 or later.

Note that page compression is predominately a feature of the transactional interface. The relational interface supports page compression through the CREATE TABLE syntax.

Additional Reading

See the following areas in the documentation for details about page compression.

Table A-6 Page Compression References

Area of Discussion	Related Sections
Concepts	"Record and Page Compression" on page 13-4 in *Advanced Operations Guide*
Maintenance utility	• "File Information Editor" on page 13-8 • "Statistics Report Tasks" on page 13-28 • "Sample Description File for the CREATE Command" on page 13-57 All references are to *Advanced Operations Guide*
BUTIL command-line utility	• "Creating and Modifying Data Files" on page 13-53 • "Clone" on page 13-53 • "Viewing Data File Statistics" on page 13-65 • "Stat" on page 13-65 All references are to *Advanced Operations Guide*
Rebuild utility	"Rebuild Utility GUI Reference" on page 14-12 in *Advanced Operations Guide*
Function Executor utility	• "Function Executor Features" on page 12-2 • "Create a File Dialog GUI Reference (Advanced)" on page 12-18 • "Creating a Btrieve File Tasks" on page 12-26 All references are to *Advanced Operations Guide*
Pervasive PSQL Control Center	"Table Properties" on page 3-28 in *Pervasive PSQL User's Guide*
Description file elements	"Description File Elements" on page A-6 in in *Advanced Operations Guide*
CREATE TABLE syntax	"CREATE TABLE" on page 3-83 in *SQL Engine Reference*
SDK Changes	• "Create (14)" on page 2-15 in *Btrieve API Guide* • "Stat (15)" on page 2-119 in *Btrieve API Guide* • "PvGetTableStat2()" on page 2-192 in *Distributed Tuning Interface Guide*

Access Methods (Software Development Kit)

This release of Pervasive PSQL includes changes to the following programming access methods:

- "Btrieve API" on page A-29
- "Distributed Tuning Interface (DTI)" on page A-29
- "Distributed Tuning Objects" on page A-30
- "JDBC" on page A-31
- "OLE DB" on page A-31

Also note that the runtime components for all access methods can now be installed with the database engine or client. See "Data Access Components" on page A-11.

Btrieve API

The 32-bit import library for the Btrieve API has been compiled with Microsoft Visual Studio 2005. The Btrieve API includes support for 64-bit applications and page compression. See the following:

- "SDK Interfaces" on page A-7
- "SDK Changes" on page A-27

Distributed Tuning Interface (DTI)

The DTI 32-bit import library has been compiled with Microsoft Visual Studio 2005. DTI includes support for 64-bit applications, metadata versions, page compression, partial indexes, and user and group management. See the following:

- "SDK Interfaces" on page A-7
- "SDK Changes" (metadata) on page A-19
- "SDK Changes" (page compression) on page A-27
- "Partial Indexes" on page A-29
- "User and Group Management" on page A-30

Partial Indexes

New flags have been added to the INDEXMAP structure to support partial indexes. See "Partial Indexes" on page A-20 in this guide and "DDFSTRCT.H Structures" on page 2-13 in the *Distributed Tuning Interface Guide*.

INDEXMAP Flags

```
B_FLAG_DUPLICATES = 1
```

Duplicates allowed in index.

```
B_FLAG_MODIFIABLE = 2
```

Index is modifiable.

```
B_FLAG_SORT_DESCENDING = 64
```

Sort index descending.

```
B_FLAG_PARTIAL = 512
```

Index is partial. Partial Index flags on segments that are not the last segment in the index are ignored. Partial Indexes only apply to the last segment in an index.

User and Group Management

New APIs exist to manage users and group. See the following in *Distributed Tuning Interface Guide*:

- "PvAddUserToGroup()" on page 2-25
- "PvAlterUserName()" on page 2-27
- "PvAlterUserPassword()" on page 2-29
- "PvCreateGroup()" on page 2-58
- "PvCreateUser()" on page 2-60
- "PvDropGroup()" on page 2-73
- "PvDropUser()" on page 2-80
- "PvRemoveUserFromGroup()" on page 2-216

Distributed Tuning Objects

Distributed Tuning Objects includes support for metadata versions and partial indexes.

Metadata Versions

Database Flag contains a new enumeration to support V2 metadata.

Enumeration	Value
32	dtoDFlagLONGMETADATA

Partial Indexes

In the DtoIndex object, only the last column in an index segment can have a partial index flag. Index segments that are not the last segment in the index and that use the partial index flag ignore the partial flag.

The Index Flag contains a new enumeration for partial index.

Enumeration	Value
512	dtoIndexPartial

JDBC

The JDBC driver is a pure Java application. No dynamic link library is required if you connect to the database engine using sockets.

The DLL, pvjdbc2.dll, is required only if you connect using shared memory or IPX.

JDBC supports V2 metadata. See “Metadata Versions” on page A-17.

OLE DB

OLE DB supports V2 metadata. See “Metadata Versions” on page A-17.

File Format and Page Size

The default file format for Pervasive PSQL v10 is 9.5 (there is not a version 10.0 file format). The 9.5 file format was available with Pervasive PSQL v9 SP2, but it is now the default.

The maximum size of a data file is 256 GB for a 9.5 format file.

The maximum page size has been increased to 16,384 bytes for a version 9.5 file. Only five page sizes are supported for a version 9.5 file:

- 1,024 bytes
- 2,048 bytes
- 4,096 bytes
- 8,192 bytes
- 16,384 bytes

For a version 9.5 file, if you specify a different page size (such as 512; 1,536; 2,560; 3,072; or 3,584), the database engine rounds up the size to the next supported size.

Note that files using file format 6.0 or higher are still fully accessible (read and write). Files with file format 5.x or older are read-only.

Network Communications

Pervasive PSQL v10 includes an enhancement to Terminal Services.

Terminal Services

Pervasive PSQL v10 clients running within Terminal Services client sessions can now perform Pervasive PSQL administrative functions by default. For example, a user with such a client can change configuration settings for Pervasive PSQL, create DSNs, and use the Monitor utility. In prior releases, the ability to perform administrative functions was prohibited from the client.

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

1. From PCC, open the properties for the MicroKernel Router under Local Client. See "To access configuration settings in PCC for a local client" on page 4-4 in *Advanced Operations Guide*.
2. On the Properties dialog, check the option Restrict Administrative Functions from a WTS Client.
3. Click OK, then exit PCC and all applications using the Pervasive PSQL database engine.
4. Start PCC again for the setting to take effect.

Licensing

Each Terminal Server *client* session with the database engine now counts as one user. In previous versions of Pervasive PSQL, only the session with the Terminal Server *itself* counted as one user.

The Pervasive PSQL Monitor utility can now differentiate each user connected to the database engine through Windows Terminal Server or through Citrix Presentation Server. (For example, "network address" ends with ":$3" if the user is in Terminal Services session number three. A local session would show "Local:$3.")

Also note that, collectively, all applications that access the database engine and run on the same machine as the database engine also count as one user. Your user count license must be sufficient for the number of users accessing the database engine.

For example, suppose that you have a user license for 20 users and your application runs on the same machine as the database engine. The application itself counts as one user. The database engine accepts up to 19 more concurrent Terminal Server client sessions and remote database sessions (19 + 1 = 20).

See also “License Administrator” on page A-23.

Documentation

WinHelp

Pervasive PSQL no longer includes any documentation in the WinHelp format. The WinHelp format is identified by files with “hlp” as the file extension. The WinHelp content is now provided in the other documentation formats.

HTMLHelp

The documentation in HTMLHelp (CHM) format has been reduced to only the context sensitive help for the following utilities:

- Function Executor
- License Administrator
- Maintenance
- Monitor
- ODBC Setup
- Pervasive System Analyzer (PSA)
- Rebuild

Note that the same content is also provided in JavaHelp. JavaHelp is the primary repository of Pervasive PSQL documentation.

JavaHelp

The JavaHelp documentation is now in a single JAR file and requires substantially less physical storage. The JavaHelp typically displays more quickly on initial invocation.

Query Plan Viewer

Query Plan Viewer is a graphical utility with which you can view query plans selected by the database engine. A query plan can be viewed for a SELECT, INSERT, UPDATE, or DELETE statement.

Query Plan Viewer has been distributed with versions prior to Pervasive PSQL v10 but was not documented. The utility is now documented in *SQL Engine Reference*. See “Query Plan Viewer” on page E-1.

Operating System Support

The Pervasive PSQL v10 release runs only on Windows 32-bit and 64-bit platforms. A subsequent release will support Linux. NetWare support has been deprecated (see “Deprecated Features” on page A-40).

Refer to the Pervasive Web site for a list of the supported Windows platforms for Pervasive PSQL v10.

Status Codes

This section lists the new and revised status codes.

New

The following status code are new:

- "-5248: Invalid partial column"
- "3032: Failed to initialize shared memory to local engine"
- "7042: Bound database requires data dictionary files"
- "8097: General security error"

-5248: Invalid partial column

The last (or the only) column in a partial index is not of data type CHAR or VARCHAR.

3032: Failed to initialize shared memory to local engine

This status code results if a second user attempts to access another user's database engine through the transactional interface. The application is unable to connect to the local engine. Possibly a database engine has been started by another user on the machine in a non-elevated mode. If so, other users will not be able to connect to that engine.

One situation in which this status code may result is if a user starts the Workgroup Engine or Cache Engine in a Terminal Services session or in a multiple-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. To allow multiple users access to a Workgroup Engine or Cache Engine in a Terminal Services session or through fast-user switching, start that engine as a service instead of as an executable.

7042: Bound database requires data dictionary files

The creation of a bound database requires data dictionary files (DDFs). For a bound database, ensure that the CREATE DATABASE statement or the Create New Database wizard also specifies the creation of dictionary files.

8097: General security error

This status code indicates that a general error occurred when checking or setting database security. The database engine was unable to associate the error with a more specific status code. If the error persists, contact Technical Support at Pervasive Software.

Revised

The following status code has been revised to support platform specific license keys. See “License Administrator” on page A-23.

7113: Invalid license key. Verify the provided key.

The license key you attempted to apply is not a valid license. Ensure that you have a valid license key issued by Pervasive Software or by your application vendor if the Pervasive PSQL database engine is embedded in an application.

This status code also displays if your key is for a specific platform that does not match the platform on which you are attempting to apply the license. For example, if your license key is for “Win64,” you cannot install the license on a Windows 32-bit operating system or on a Linux platform.

If you are using the License Administrator GUI to apply the key, repaste or retype the key into the License Key field, then click Apply License Key.

Complimentary Data Management Products

The data management products Backup Agent and AuditMaster are supported with Pervasive PSQL v10.

AuditMaster

Pervasive AuditMaster 6.4 is an upgrade from AuditMaster 6.2 or 6.3. The upgrade enables the functionality of your 6.2 or 6.3 release to be compatible with Pervasive PSQL v10 Server, 32-bit.

Pervasive AuditMaster 6.4 does not include the following:

- Operation on Windows Vista
- 64-bit support or V2 metadata

You do not need to uninstall AuditMaster 6.2 or 6.3 to upgrade. Pervasive AuditMaster 6.4 can be installed to your existing version without loss of existing audit records, settings, or saved queries, reports, or alerts.

The upgrade uses the same directories as the AuditMaster 6.2 or 6.3 product from which you are upgrading (for example, default locations such as C:\PVSW\Audit).

Backup Agent

Pervasive Backup Agent 1.2 is an upgrade from Backup Agent 1.x. The upgrade enables the functionality of your 1.0 or 1.1 release to be compatible with the Pervasive PSQL v10 Server, 32-bit.

Pervasive Backup Agent 1.2 does not include the following:

- Operation on Windows Vista
- 64-bit support

You do not need to uninstall Backup Agent 1.0 or 1.1. to upgrade. Pervasive Backup Agent 1.2 can be installed to your existing version. The upgrade uses the same directories as the product from which you are upgrading (for example, default locations such as C:\PVSW\PBA).

Deprecated Features

The following features are no longer supported in Pervasive PSQL:

- "NetWare" on page A-40
- "Select Windows Platforms" on page A-40
- "DOS 6.22" on page A-40
- "16-bit Applications" on page A-40
- "Smart Components" on page A-40
- "DOS TSR Programs" on page A-41

NetWare

NetWare is no longer a supported platform for Pervasive PSQL. If you are running a previous version of Pervasive PSQL for NetWare, consider upgrading to Pervasive PSQL for Windows or for Linux.

Select Windows Platforms

The following Windows platforms are no longer supported for any of the Pervasive PSQL products:

- Windows NT
- Windows 98/ME

DOS 6.22

DOS 6.22 is no longer a supported platform for any of the Pervasive PSQL products.

16-bit Applications

Support is no longer available for 16-bit products or 16-bit application development with Pervasive PSQL.

The 16-bit utilities are no longer available in Pervasive PSQL, nor are the 16-bit client components. The configuration settings for 16-bit clients, such as Use Thunk and so forth, are no longer available.

Smart Components

The smart component management (SCM) technology is no longer used in Pervasive PSQL. SCM provided a method to avoid component mismatches and the occurrence of an older program overwriting newer components. Pervasive PSQL now uses different techniques to provide the same functionality.

DOS TSR Programs

The DOS Terminate and Stay Resident (TSR) programs are no longer supported. This includes the following programs:

- BREQUEST DOS requester
- BREQNT DOS requester
- BREQTCP DOS requester
- Sockets.exe

Note that the DOS requester btrbox *is* still supported on 32-bit Windows platforms. DOS applications are not supported on 64-bit Windows platforms; therefore, btrbox is not supported on 64-bit Windows platforms.

Index