PERVASIVE®

DATABASE

Upgrading Btrieve™ 6.15 to Pervasive® PSQL Summit™ v10

A Pervasive Software White Paper

June 2008

Table of Contents

Overview

This paper describes two steps to upgrading applications from Btrieve 6.15 to Pervasive PSQL™ Summit v10 – simple migration to be up and running as quickly as possible, and rebuilding files to take advantage of 10 years of product improvements. These steps are typically uncomplicated, but may take a little time to execute properly. This paper will help you prepare and plan proper upgrade procedures in order to minimize the time necessary to complete the process.

The content is organized in three general sections:

Simple Migration: This section has some quick steps to get your 6.15 application up and running on Pervasive PSQL v10. It also covers installation of workgroup, servers and clients, application testing, and troubleshooting.

What's New: There have been a great many feature additions and improvements in the releases leading up to Pervasive PSQL v10. We'll provide a short review of the key features to help you evaluate the changes that will really enhance your application.

Rebuilding Files: Pervasive PSQL v10 is backward compatible with 6.15 applications. However, taking advantage of some of the new features in v10 requires rebuilding your Btrieve database files. We'll show you how.

Simple Migration

Because Pervasive has maintained backward compatibility from release to release, migrating from Btrieve 6.15 to Pervasive PSQL v10 should be as simple as installing the database and running the application.

The most important caveat to the "simple migration" statement is that the application must be using file format 6.x or later. If this is not the case, it will be necessary to rebuild the files. This additional step will provide the benefit of being able to take advantage of the feature improvements from Btrieve 6.15 through to Pervasive PSQL v10. Before going into the installation details for Pervasive PSQL v10 Workgroup or Server, it will be useful to cover some important background information.

Before You Start – Background Information

5.x File Support in Pervasive PSQL v10

As mentioned above, applications running on Pervasive PSQL v10 require file format 6.x or later. This is because Pervasive PSQL v10 databases will read 5.x files but will not write to them. Files in 5.x format, will need to be rebuilt to a more recent file format for the database to be able to update them.

The following should also be noted:

- 5.x format files will be opened read-only. Any attempt to write to a 5.x format file will fail with status 46.
- Create operation will fail if the format specified is 0500 or less. The error code returned is 41 (Operation Not Allowed).
- The File Version engine setting can no longer be set to 0500.

For a quick review of the additional benefits (other than being able to write to the files) of rebuilding your 5.x files, take a look at the "What's New" section of this paper. In the "Rebuilding Files" section of this paper, we provide instructions for rebuilding your Btrieve files to a format supported by Pervasive PSQL v10.

If you have no rebuild requirements, and are ready to test your application on the newest release of Pervasive PSQL, it's a simple process to install the database and test your application.

Migration of Existing Configuration Settings

During the installation/upgrade process, all of your existing engine configuration values are migrated to the new engine, with the following exceptions:

Configuration Setting	New Value
Server -- Compatibility -- Create File Version	**9.5 –** By default, new data files are always created in the most recent format.
Server -- Memory Usage -- System Cache	**Off –** With the addition of level 2-database cache in V8, the system cache should not be used.
Server -- Data Integrity -- Transaction Durability	**Off –** With the addition of a new setting, Transaction Logging (default: On), Transaction Durability is not needed to ensure multi-file transaction atomicity and data integrity. If your application requires true Transaction Durability, you will need to use Configuration to turn this setting On.

Installing Over Existing Pervasive Products

If you install Pervasive PSQL v10 over a Btrieve 6.15 database engine, the 6.15 components will be removed during the installation process so that there are no component conflicts. Your licenses from Btrieve 6.15 will not be migrated to Pervasive PSQL v10 (you'll need to purchase new v10 licenses).

Workgroup or Server?

The first question to ask in this process is: "Am I testing on the workgroup or server version of Pervasive PSQL?" If your application uses the Btrieve 6.15 workstation, the product you need to install is the Pervasive PSQL v10 Workgroup. The Single User Workstation Engine was replaced in the Pervasive PSQL 2000i release with the Workgroup Engine. It's faster, more reliable, and can support up to 5 concurrent users. We'll discuss the Workgroup Installation first, then the Server installation.

What's a Workgroup?

Pervasive PSQL Workgroup replaces the Workstation version of Btrieve. Workgroup is a scaled down version of the Pervasive PSQL client/server database engine designed specifically for single user and small group installations. It will support up to 5 concurrent users. The Pervasive PSQL v10 Workgroup assigns an "owner" or Gateway to the database through which all other computers in the workgroup access the data store. The Gateway can be defined statically or the system can dynamically establish the Gateway engine each time the database is opened.

For more information on the Pervasive PSQL Workgroup please read our *Understanding the Workgroup Release* white paper or review the *Getting Started with Pervasive PSQL*.

Workgroup Installation

Pervasive PSQL Workgroup should be installed on the same computer where the database files are located. The Workgroup engine is also able to access files that are stored on remote workstations, but this can and will affect performance. The Workgroup engine also includes all client components required to connect to remote Workgroup engines or PSQL v10 Server databases.

When installing Pervasive PSQL v10 for the first time on a system, Setup will check to ensure the system files meet the minimum requirements and will automatically update the system files when possible. In some cases, if these files are locked by the operating system, a reboot will be required before the Setup can continue.

To install Workgroup, follow the instructions on the CD or in the download file. Step-by-step instructions for installing Pervasive PSQL v10 Workgroup are included as part of Appendix A of this paper.

If you haven't already purchased a copy of the Pervasive PSQL Workgroup, and would like to test your application, download a trial copy from the Pervasive Web site.

Troubleshooting Workgroup Installations

On the rare occasion when you run into problems after installing Pervasive PSQL Workgroup or client requestors and your 32-bit applications do not automatically function (or seem to still be using the Btrieve 6.15 database engine), you may have a conflict with the DLLs on your system. Pervasive recommends running the Pervasive System Analyzer (PSA) to "View Loaded Components". This report details all components being loaded into memory on the client machine, and can be either printed or saved to a file.

Compare the Loaded Components report with the manifest.xls file that details components loaded by the Pervasive PSQL client requester. The components loaded into memory that are not on the manifest should be archived or deleted.

First, run the Pervasive System Analyzer Test Active Installation option. Then run the View Loaded Pervasive Modules to search for the components loaded into memory by specifying the directory where the non-PSQL (i.e. Btrieve) components were found. When these components are located, they may be safely be archived. It is recommended that the PC be rebooted once the analysis test and archiving are complete to remove the possibility of components lingering in memory.

Server Installation

To install the server engine, simply run the installation program on the CD and follow the prompts on the screen. The Pervasive PSQL install process is designed to be as simple as possible. Complete documentation regarding hardware, deployment configurations, and other installation issues is available in the Getting Started manual. A copy of this manual, and all of the documentation for Pervasive PSQL, is online at the Pervasive Library.

Client Requesters or Server First?

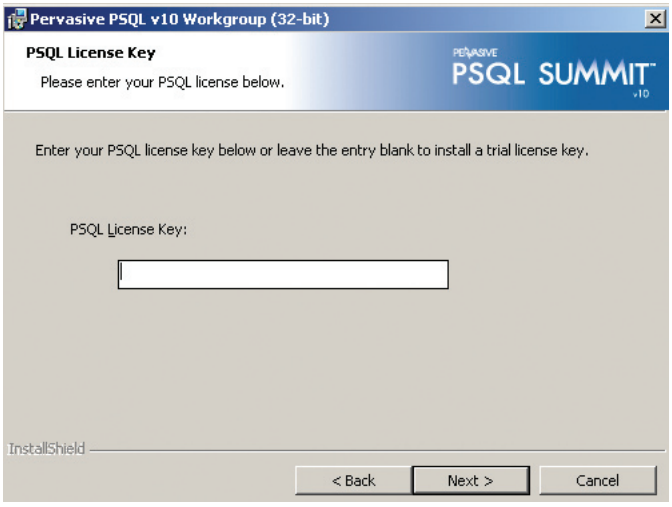
Pervasive recommends that all servers and clients have the same release number. However, Pervasive will support older requesters to a newer Server engine within a major release. **If you are migrating from Btrieve 6.15, you must use the latest versions of both the Pervasive PSQL client and server because Btrieve 6.15 components are no longer supported.**

Licensing Changes

A major change in Pervasive PSQL is the management of user counts. In Btrieve 6.15, the user count for the database engine was encoded in the server database engine. Instead of embedding the user count in the database engine, an alphanumeric user count key is provided with Pervasive PSQL. User count keys include information regarding product version, platform, user count and more.

During the server installation process, a dialog box will appear requesting a license key.

Figure 1. PSQL License Key Request



Your Pervasive PSQL Summit v10 Server is set to the number of users specified in the license key.

If you do not yet have a license key or it is not with you at the moment, you can still continue with the installation by using an evaluation license.

You may run the License Administrator utility during or after the installation to view or install a user count license key. Please note that you must restart the database engine for your license to take effect. The License Administrator utility is documented in Pervasive PSQL User's Guide in the section License Administrator.

Windows Server

The Pervasive PSQL Windows Server should be installed directly from the CD-ROM drive on the target server or from a trial download file; you cannot install it remotely from a client machine. In order to make the client installation directories available to your workstations, full administrator-level rights are required on the machine where Pervasive PSQL will be installed.

If you want individual client machines to install the requester portion of the installation from the server, you must copy the client installation to a server directory and give the clients permission to access the client installation folders, which are part of the Pervasive server install.

Note: Client installation images for Windows are not part of the server installation. You must run a separate installation program for Pervasive PSQL client requesters. Instructions for Installing Pervasive PSQL Clients are provided below.

Installing Pervasive PSQL Clients

Pervasive PSQL includes two types of requesters for Windows: Trace, and WIN32.Trace requesters are for troubleshooting (tracing) client problems at a low level. The simplest approach is to run the install program that will install both types of requesters. This install will also include a set of Pervasive PSQL utilities and a set of Pervasive PSQL documentation. Please see the readdos.txt file for more information on installing DOS requesters.

Pervasive System Analyzer

When you install the Pervasive PSQL client requesters on workstations accessing the Server engine, the Install will automatically remove any legacy components that it finds in the system path and default Btrieve/Pervasive PSQL directories. This greatly lowers the possibility of any conflicting components remaining on the system that would cause problems with your application working properly.

If you run into dll or network problems after the install, run Pervasive System Analyzer. The PSA allows you to scan the disk for any loaded components that may have remained in memory. Pervasive System Analyzer can also assist you in some basic network troubleshooting.

If you do encounter an application that will not run with Pervasive PSQL clients, please contact your application vendor or Pervasive Software Support at http://www.pervasive.com/support/.

Installing Clients Over the Network

There is no need to share the CD-ROM over the network or to physically carry the CD-ROM to all client machines in order to install the client requesters. Copying the client installation packages to the server is called “staging” the client installations on the server. Detailed instructions on how to stage the client installations on the server can be found in the book, *Getting Started with Pervasive PSQL*. Please note that this manual, along with the entire Pervasive PSQL documentation set for your platform are included on the CD-ROM as part of the help files.

You simply copy the “clients” folder from the CD-ROM to the server. You can then connect to the server from your client machines and run the appropriate installation program from one of the subfolders in the “clients” folder.

To review detailed instructions for installing the client requesters, please refer to Chapter 5: Installing the Pervasive PSQL Client for Windows.

Troubleshooting Server Installations

Pervasive provides features and tools that help prevent configuration and installation problems. Some of these utilities are installed and run as part of the installation process and all can be run later to evaluate configuration and registry settings and to troubleshoot problems.

Table 1: Pervasive PSQL Tools, Utilities, and Support

Feature/Component	Function	For More Information
Pervasive System Analyzer	Analyzes system components and runs communication tests.	See Diagnosing Problems with Pervasive System Analyzer (PSA).
Monitor Function	The Monitor utility monitors PSQL activities on a server including users, client information, data source name, and connection information.	Refer to the Advanced Operations Guide.
Gateway Locator	A utility that can be used to determine or change the Gateway being used for a particular data dictionary.	See Configuring the Workgroup Engine.
Knowledge Base	Provides information about many Pervasive software configurations and common environments.	Search the Pervasive Knowledge Base at: http://ww1.pervasive.com/kb/

You can find the Chapters referred to in the More Information column above in the Pervasive PSQL v10 documentation. Documentation is installed as part of the Workgroup and Server installs. The latest versions of all Pervasive PSQL documentation can also be found online at http://www.pervasive.com/library.

Your installation process should complete without problems. However in case of issues, please refer to the following checklist.

Checklist for Problems

- Did you see any error messages displayed during installation?
- Does the Network function correctly?
- Is the Engine running?
- Is the Client software correctly functioning?
- Are there errors in your PVSW.LOG file?

If your problems persist, you can access Pervasive Support online at http://www.pervasive.com/support. From this portal you can find both online resources (Knowledge Base, FAQs, etc.) as well as a full set of information regarding how to work with Pervasive Support.

Pervasive also offers a comprehensive set of courses for Pervasive PSQL Service and Support to get you up to speed with the latest product offerings.

Upgrade Complete

Unless you have version 5.x or earlier files, or you want to take advantage of the new Pervasive PSQL features and functionality that a file rebuild can offer you, your upgrade is complete.

To help you make an informed decision about whether or not to rebuild your files, let's take a look at some of the feature improvements from Btrieve 6.15 to Pervasive PSQL v10.

What's New?

Pervasive PSQL v10 offers you new and powerful features that take advantage of the Btrieve API and offer you significant performance improvements over Btrieve 6.15. Over 10 years of development effort have delivered tremendous improvements in the Pervasive database engine from Btrieve 6.15 to Pervasive PSQL v10. The remainder of this section includes a summary of the key areas of improvement. To learn all about the latest release, Pervasive PSQL v10, go to http://www.pervasive.com/psql/index.asp.

1) Performance – The key to gains in performance, over an already very fast Btrieve database engine, comes from supporting newer processor architectures and focusing on minimizing disk and network I/O.

    a. Pervasive PSQL Summit v10 provides support for 64-bit Intel and AMD processors. PSQL v10 applications can take advantage of new faster hardware, more memory, and larger address space available in the latest 64-bit systems.

    b. Xtreme I/O for Pervasive PSQL Summit v10 accelerates database file I/O, minimizes processor contention and maximizes memory capacity. Xtreme I/O is designed for applications with large data sets, a high percentage of random writes, and data that can be significantly compressed.

    c. Page and record compression reduce the size of data files to improve performance. Records with contiguous repeating characters are compressed to reduce record size. Pages are compressed before writing to improve write times and decompressed into cache to improve read times.

    d. Turbo Write Accelerator enables a background writer thread to find (or create) holes in the physical file so that multiple pages can be written as a single system write. Page

write coalescing improves performance because fewer writes are required and there are larger contiguous blocks.

e. Self-tuning dynamic cache manages both static and dynamic cache to increase memory utilization on the server, dramatically improving read performance.

f. Client cache architecture greatly improves disk I/O performance, using local caching of pages to eliminate network requests.

2) Security – Recent releases of Pervasive PSQL have greatly improved the security features relative to Btrieve 6.15.

a. Pervasive PSQL v10 provides improved security functionality for controlling permissions for views and stored procedures. Also new for PSQL Summit v10 are signed Pervasive executables, for both the executables and the new installer.

b. Pervasive Control Center can now be used to set up OS independent database authentication and authorization for the Btrieve database engine. Pervasive PSQL security for Btrieve now offers three modes of authentication for the database: 1) Classic (OS authority); 2) Database (users must be defined for the database); and 3) Mixed (OS authentication Database authorization). Relational access security can be used to set access rights, including database and table permissions, for groups and users. The PSQL client can be configured to automatically pop-up a login prompt to facilitate the entry of user Ids and passwords, allowing you to add security without application changes.

c. Pervasive PSQL also supports encrypting the database related network traffic that occurs when using the database. This type of encryption is often called wire encryption because it protects the data when it is traveling on the network, or on any network infrastructure, including wireless.

3) SQL Interface – Pervasive PSQL now fully integrates a greatly improved SQL interface. Query optimization and a greatly expanded SQL syntax really do deliver the “best of both worlds”.

4) Expanded Suite of Developer Tools – With Pervasive PSQL, customers get much more than a very fast database engine. Developer tools now include:

a. A brand new GUI Pervasive Control Center

b. Updated version of DDF Builder (using the same interface architecture as PCC)

c. .NET 2.0 and Visual Studio 2008 support

5) Add-on Security Solutions – Customers can also enhance the value of their application with tools for backup, audit, and multi-database data synchronization.

a. AuditMaster™ monitors and reports all activity occurring in a Pervasive PSQL database. With the ability to be plugged directly into the Pervasive PSQL database engine, it has audit functionality that can be added without coding. Customers use AuditMaster to ensure regulatory compliance, to avoid fraud, and to diagnose data-related problems.

b. Backup Agent™ helps backup operating databases by allowing applications to stay in production during backups. The agent intelligently ensures data files are clean and consistent for safe backup to tape, disk, or other media. A complement to 3rd-party backup packages, it can be integrated into existing backup processes.

c. DataExchange™ synchronizes data in real-time between Pervasive PSQL databases and supports multiple network topologies like point-to-point and peer-to-peer. DataExchange can be used with applications maintaining a warm backup, loading a reporting server, sharing data among multiple offices, and connecting remote users.

For a detailed table comparing the most recent versions, refer to Appendix C: *Pervasive PSQL Version Comparison Guide*.

Rebuilding Files

To take advantage of the new version features, such as larger file sizes, large page sizes, or system key generation, data files must be rebuilt to use the version 10 file format. If your database uses dictionary files (DDFs), these must be rebuilt as well as the data files. The Pervasive PSQL Rebuild utility is what you will use to convert the data files and DDFs. Rebuild comes in two forms: a GUI version for 32-bit versions of Windows, and a command-line version for Linux and Windows. See the *Advanced Operations Guide*, "Converting Data Files" for detailed information on how to use the Rebuild utilities to convert your data files.

The following information will help you make better use of the Rebuild tools:

- Make sure no one is working in the file while it is being built.
- Use the GUI Rebuild Utility if you have a large number of files to rebuild. The GUI version of the utility allows you to select many files at once, making the selection process easier.
- Convert DDFs as well as data files.
- For converting multiple files, the best approach when using a Windows server is to use the GUI Rebuild utility to build a list of all the files you want to convert, and then convert them all at once.
- Large files, such as 3 or 4 GB, may take several hours to convert. If you have more than one database engine available, you can reduce rebuild time by sharing the process among a number of CPU's. During scheduled down time for the upgrade, you may find it helpful to copy large files out to the other computers with server engines, and then copy the files back when you are finished converting.
- Use the "Continue on Error" option in combination with a *.* file list to convert all the data files in a directory, even if other types of files exist in the directory. The other types of files generate an error when the Rebuild utility attempts to open them. If the "Continue on Error" option is set, the utility ignores each of these errors and skips to the next file.
- There is no facility to perform a recursive or multi-directory conversion without building a list of all the individual directories or individual files that you wish to convert. Build a list of data files, and then select the specific files to convert in that run.

Optimizing the Rebuild Process

The Rebuild utility creates copies of files by making Btrieve calls to the database engine. The database configuration settings, as well as the amount of physical RAM present can have a great effect on the amount of time required to rebuild data files, especially large files.

Building indexes generally requires much more time than building data pages. If you have a data file with many indexes, it will require more time to rebuild than would the same file with fewer indexes.

The Rebuild utility is capable of rebuilding a file using two different methods.

Default Rebuild

In the default method, the Rebuild utility uses Btrieve operations to perform the following steps:

1) Create a new, empty data file with the same record structure and indexes as defined in the source file.
2) Drop all the indexes from the new file.
3) Copy all the data into the new file, without indexes.
4) Add the indexes, using the steps below.
    a) For a particular key in the source file, the database engine reads as many key values as it can into the memory buffer using the Extended Step operation with the appropriate filter.
    b) The database engine sorts the values in the memory buffer and writes out the sorted values to a temporary file.
    c) The engine repeats steps a) and b) until it has processed the key value from every record. The temporary file now contains several key value sets, each of which has been individually sorted.
5) The engine merges these key value sets into index pages, filling up each page to capacity. Each index page is added to the Btrieve file at the end, extending the file length.
6) Steps 4 and 5 are repeated for each remaining key.

If any failure occurs during this process, such as failure to open or write the temporary file, Rebuild starts over and uses an alternative method to build the file.

Alternative Rebuild

The alternative method is used by Rebuild when there is not enough physical memory to use the default method, or if the default method encounters processing errors. In this situation, Rebuild:

1) Creates a new, empty data file with the same record structure and indexes as defined in the source file.
2) Drops all the indexes from the new file.
3) Copies all the data into the new file, without indexes.
4) Adds the indexes, one by one, using the following process:
    a) For a particular key in the source file, reads one record at a time using the Step Next operation.
    b) Extracts the key value from the record and inserts it into the appropriate place in the index. This necessitates splitting key pages when they get full.
    c) Repeats steps a) and b) until it has processed the key value from every record.
5) Repeats step 4 is repeated for each remaining key.

The alternative method is typically much slower than the default method. If you have large data files with many indexes, the difference between the two methods can amount to many hours or even days. The only way to ensure that the default (fast) method is used is to be sure that you have enough physical memory on your server. The following sections explain how to calculate memory requirements for rebuilding your files.

Configuration Parameters and Rebuilding Files

The amount of free memory on the computer, along with certain configuration parameters, has a direct effect on how much time the index rebuild process takes. This section helps you determine whether the default or alternative method of rebuilding indexes will be used, and what actions you can take to help speed the process.

Sort Buffer Size

Optimal memory is enough memory to store all merge blocks in RAM. The minimum amount of memory is enough to store one merge block in RAM. The section below titled “How Much Memory Do You Need?” explains the method for calculating optimal and minimum memory requirements.

If the Sort Buffer Size is set to zero (the default), Rebuild calculates the value of optimal memory bytes and allocates memory based on that value. If the memory allocation succeeds, the size of the block must be at least as large as the value defined for *Minimum Memory Bytes Required*.

If the Sort Buffer Size is set to a non-zero value, and it is smaller than the calculated *Minimum Memory Bytes Required* value, Rebuild uses the value to allocate memory.

Rebuild compares the amount of memory that it *should* allocate with 60% of the amount that is actually available. It then attempts to allocate the smaller of the two. Rebuild uses the 60% rule to avoid taking all the memory in the computer.

If the memory allocation fails, the database attempts to allocate 80% of the last attempted amount, and so on. If, in the end, the memory allocation fails completely, Rebuild builds the index using the alternative method.

If the memory allocation succeeds, the size of the block that has been allocated, the *Allocated Block* must be at least as large as defined in the formula for *Minimum Memory Bytes Required* in the next section. In addition, the number of blocks of memory required must meet certain criteria. See the formulas in the following section for the details.

Index Page Size

The page size in your file also affects the speed of index creation. If Rebuild uses the alternative method, small key pages dramatically increase the time required to build indexes. Key page size has a lesser effect on building indexes if Rebuild uses the default method.

Rebuild can optimize page size for performance or disk storage. To optimize for performance (your application accessing its data) Rebuild uses a page size of 4096 bytes. This results in larger page size on physical storage and slower rebuild times.

Assume that your application has 8 million records, a 20-byte key, and uses a page size of 512 bytes. The database engine places between 8 and 18 key values in each index page (index pages are automatically kept at least half full). This lessens the amount of physical storage required for each page. However, indexing 8 million records creates a B-tree about seven levels deep, and most of the key pages will be at the seventh level. Performance will suffer.

If a page size of 4096 bytes is used, the database engine places between 72 and 145 key values in each index page. This B-tree is only about four levels deep, requiring fewer pages to be examined when inserting each new key value. Performance is increased, but so is the requirement for physical storage.

The time required to build the indexes increases exponentially with increasing depth of the B-tree.

The approximate maximum number of key values per index page can be calculated using the following formula:

Max Keys = (Page Size – 12) / (Key Length + Key Overhead)

How Much Memory Do You Need?

You can use the formulas below to estimate the optimal and minimum amount of contiguous free memory required to rebuild your indexes on a particular file using the default method. The optimal memory amount is enough memory to store all merge blocks in RAM. The minimum amount of memory is enough to store one merge block in RAM.

Key Length = total size of all segments of largest key in the file.

Key Overhead = 8 unless the key type is Linked Duplicate, in which case the value is 12.

Record Count = number of records in the file.

Optimal Memory Bytes = (((Key Length + Key Overhead) * Record Count) + 65536) / 0.6

Minimum Memory Bytes = Optimal Memory Bytes / 30

For example, if your file has 8 million records, and the longest key is 20 bytes (not Linked Duplicate), the preferred amount of memory is (((20 + 8) * 8,000,000) + 65536) / 0.6 = 373,442,560 bytes of memory, or 373.5 MB. The minimum amount of memory is 1/30th of that amount, or 12,448,086 bytes of memory, or 12.45 MB.

The divisor 30 is used because the engine keeps track of no more than 30 merge blocks at once, but only one merge block is required to be in memory at any time. The divisor 0.6 is used because the engine allocates no more than 60% of available physical memory for this process.

If you do not have this minimum amount of physical memory available, Rebuild will use the alternative sorting method to rebuild your data file.

In this example, the optimal number is 373,442,560, or 373.5 MB of contiguous free memory. If you have this much contiguous free memory available, the Rebuild process takes place entirely in RAM. Because of the 60% allocation limit, this number is actually the amount of contiguous physical memory you need to have free when the rebuild process starts, not the amount that the rebuild process would actually use. Multiply this value by 0.6 if you want to figure the maximum amount that would actually be used by the rebuild process.

Finally, the block that is actually allocated must meet two additional criteria:

Blocks Required must be less than or equal to 30, where:

Blocks Required = Round Up (*Optimal Memory Bytes* / *Allocated Block*)

And, *Allocated Block* size must be greater than or equal to:

((2 * *Max Keys* + 1) * (*Key Length* + *Key Overhead*)) * *Blocks Required*

Continuing with the same example, assuming a 512-byte page size, and a block of 12.45 MB successfully allocated, let's calculate the test values:

Blocks Required = 373,500,000 / 12,450,000 = 30

This part of the test passes. On to the next part:

Max Keys = (512-12) / 28 = 18

(((2 * 18) + 1) * (20 + 8)) * 9 = 9324

Is *Allocated Block* (12.5 million bytes) larger than 9324 bytes? Yes, so part two of the test passes. The index keys will be written out to a temporary file in 12.45 MB pieces, sorted in memory, and then written to the index.

Cache Allocation Setting

The Cache Allocation setting determines how much memory is available to the database engine for accessing data files, not for use when indexes are built. Increasing your Cache Allocation to a high value does not help the indexes build faster and may even slow the process by taking up crucial memory that is now unavailable to the rebuild process going on outside the cache. When rebuilding large files, decrease the cache value to a low value, such as 20% of your current value but not less than 5MB. This leaves as much memory as possible available to the index rebuild process.

Summary

Upgrading from Btrieve 6.15 to Pervasive PSQL is a straightforward process. The information presented in this document should help you migrate your application successfully. If you do run into questions or problems during your migration, give us a call or send an email and we'll do our best to help you make the migration simple and quick, and enable your application with all of the great new features of Pervasive PSQL.

If you're committed to upgrading, the only real decision to make is whether to rebuild your application files in order to take advantage of twelve years of product improvement. If your application is using file formats earlier than 6.15, they must be rebuilt in order to run on the latest Pervasive PSQL release.

For single user or small workgroup (up to 5 users), the Pervasive PSQL Workgroup edition is ideal. For anything more than 5 users, the Pervasive PSQL Server edition will be the best approach. The installation process for both Workgroup and Server have been greatly simplified with tools and utilities including Pervasive System Analyzer, which identifies installed components and tests both Btrieve and relational access, and Pervasive Control Center, a GUI based tool to create and manage the database.

Other notable improvements from the 6.15 release include:

- Performance – minimized disk and network I/O
- Security – OS independent database access authentication and encryption
- Relational Interface – Fully integrated and optimized SQL access
- Developer Tools – Improved DDF Builder for creating and managing data definitions, and a long list of database access methods
- Value Added Security Solutions – AuditMaster™, Backup Agent™, and DataExchange™ all offer great ways to add value to your application

Whether you choose a simple migration or rebuilding your files, Pervasive has a tremendous set of resources to make the job as simple as possible. The following links are just a start. If you find you need more information, please give us a call or send an email. We're ready to help.

Pervasive Online Library – http://www.pervasive.com/library

Pervasive Developer Center - http://www.pervasive.com/developerzone/

Pervasive PSQL Updates and Service Packs - http://www.pervasive.com/support/updates/?product=psql

Pervasive Support Knowledge Base - http://www.pervasive.com/support/

Pervasive Support Forum - http://www.pervasive.com/devtalk/

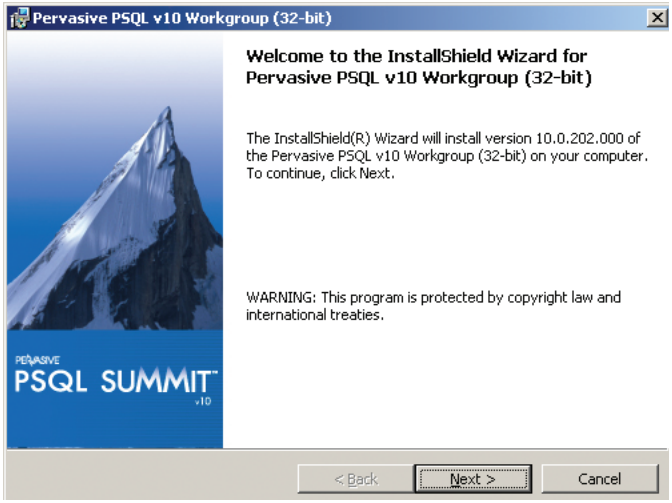
Appendix A: Installing Pervasive PSQL v10 Workgroup on Windows

1. Stop the Engine currently running.

   If your Pervasive PSQL Workgroup engine is currently running, you must stop the engine and exit before you begin installation. To do this, right-click the engine icon in the task bar at the bottom of your screen and select the “Stop Engines” and “Exit” option.

2. Launch the installation program from your Windows workstation:

   a. Insert the Pervasive PSQL Summit v10 Workgroup CD in the CD-ROM drive of your Windows workstation.



   b. If the installation does not start automatically, click Start, select Run, and type *drive:\autorun\autorun*, where drive is the drive letter of your CD-ROM device.

The installation selection dialog displays.

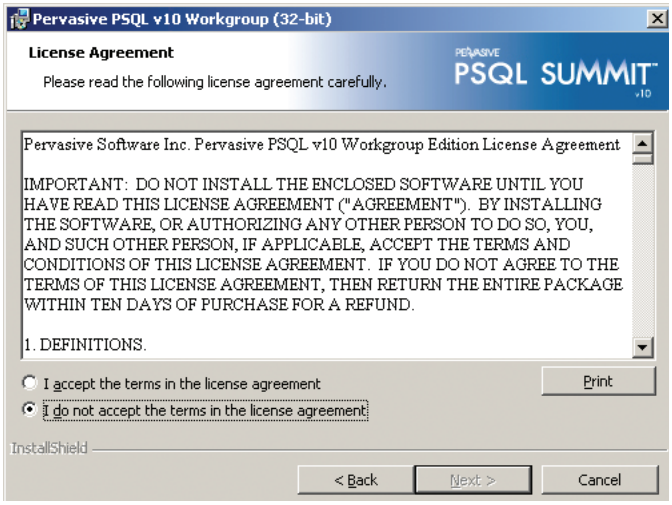
3. Click the button for the “Workgroup” installation.

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

4. Click Next to proceed with the installation.

5. Read and accept the Software License Agreement

Pervasive Software's Software License Agreement

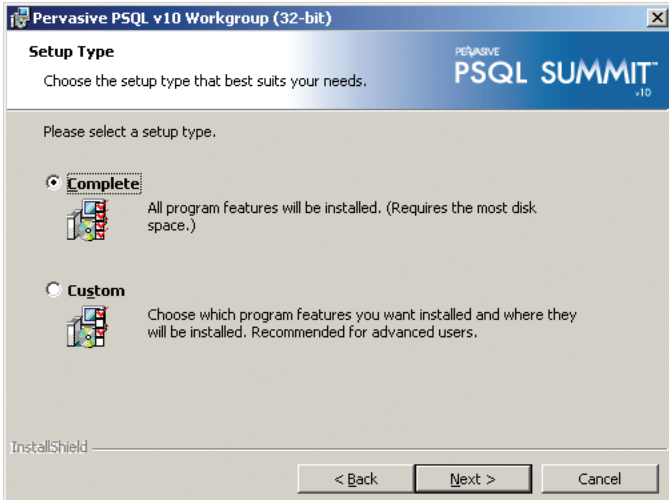


Read the Pervasive Software License Agreement. Select the choice that indicates you accept the terms in the license agreement.

6. Click "Next" to continue the installation.
   See "User Licenses" for more information about Workgroup licenses.

7. Select the setup type: Complete or Custom.

Setup Type Dialog Box



The Complete installation, recommended for most users, takes default actions for most operations performed during the installation. The Complete Workgroup installation installs the following components to drive C:

- PSQL database and all requested components for Transactional and Relational access
- Pervasive PSQL Control Center
- Utilities
- Online documentation

The Custom installation is typically for advanced users. It allows you to specify the installation location, select the components to install, and determine the space requirements for the components.

8. Click “Next”, then continue based on your choice of Complete or Custom:

   • If you choose a Complete install, continue with the next step.

   • If you choose a custom install, skip now to Custom Installation Path.

   At this point, if you want, you may click “Back” to change or review any of the installation settings, or click “Cancel” to exit the installation program. After you click “Install”, you may still exit the installation, but you will be unable to change or review settings.

9. Click Install.
   If required, close any running applications that may interfere with the Pervasive PSQL installation.

   Decide how to continue with the installation. Your choices are:

   a. To exit all of the programs that may interfere, then click “Next”.

   b. Not to exit all of the programs that may interfere, then click “Ignore”.

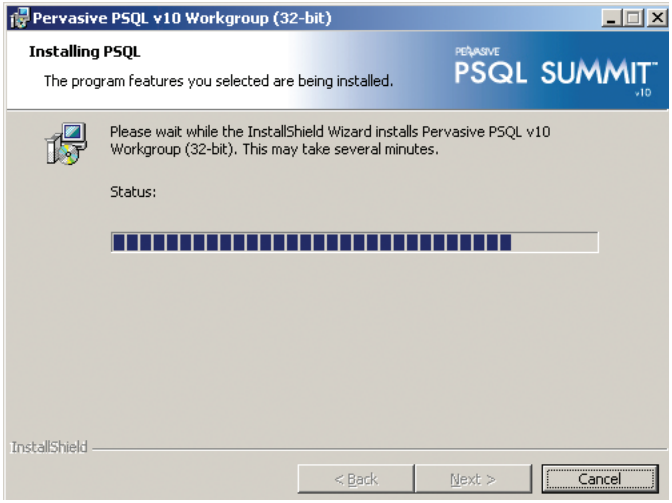
NOTE:

Next does not proceed with the installation unless you exit all programs that may interfere.

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

If no running applications interfere with the Pervasive PSQL installation, the installation process continues. A dialog box appears that gives you a status.

Installing PSQL in Progress



Installation actions include, but are not limited to, the following:

Script operations

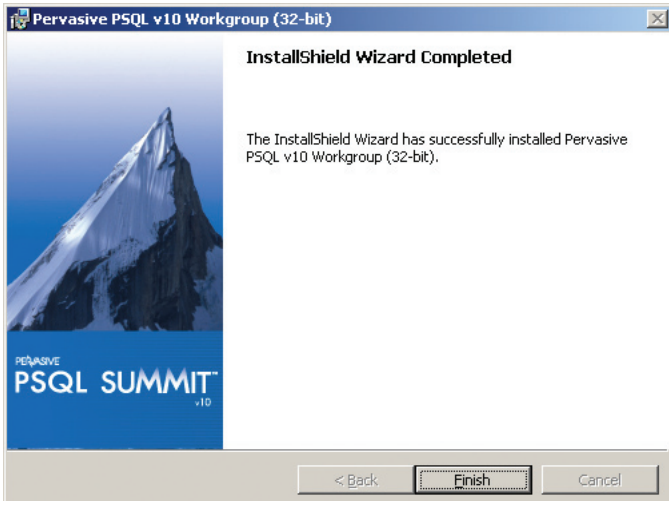
a. Component registration

b. File copying

c. Shortcut creation

d. System registry updates

e. Sample database creation



NOTE:

To make these instructions brief, we're reviewing only the Complete installation. For information on Custom installations, please review Chapters three and six of the Getting Started with Pervasive PSQL documentation.

Appendix B: Pervasive PSQL Version Comparison Guide

	Btrieve	v10
Performance: Turbo Write Accelerator, Dynamic Cache, Client Cache		✔
Security: OS independent database authentication, encryption		✔
Workgroup: Single point control for data access for single user or small groups		✔
Support for XP/2000/2003/Server 2008 and Linux		✔
Fully Integrated Relational Interface		✔
Transaction logging durability with and without unique keys		✔
SQL Outline View: Simplify complex query analysis		✔
Utilities – Pervasive Control Center and Pervasive System Analyzer		✔
Distributed Tuning Interface – API for cataloging, configuring, monitoring and diagnostics		✔
256GB Table Size		✔
16K Page Size		✔
AuditMaster – Auditing and real-time monitoring		✔
Backup Agent – Database backup utility		✔
DataExchange – Database replication and synchronization		✔
DDF Builder – Java based tool to manage table definitions		✔
User Defined Functions		✔
System Stored Procedures		✔
Bulk Data Utility – High Performance data import		✔
Configuration Properties Utility – Simplified configuration changes		✔
Monitor Utility – Monitor database resources		✔
DDL Utility – Execute SQL from a command line interface		✔
SQL Support for Variant Records – Simplified SQL access to Btrieve files		✔

Contact Information

Pervasive Software Inc.
12365 Riata Trace Parkway, Building II
Austin, Texas 78727

United States
800.287.4383
512.231.6000
Fax: 512.231.6010
info@pervasive.com

EMEAA
+800.1212.3434
cic@pervasive.com

http://www.pervasive.com