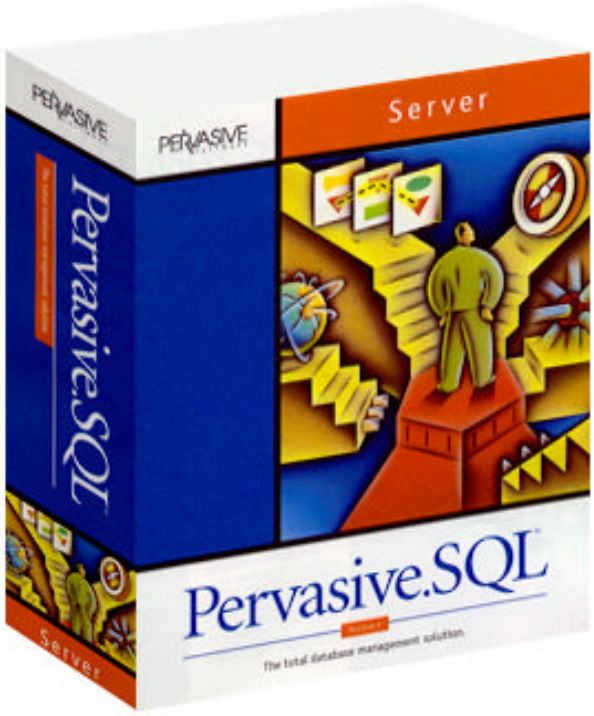
Pervasive.SQL: The Perfect Database for Businesses and Departments with Low Information Technology Resources



This document was compiled by

Scott Smith
Vice President of Product Strategy
Pervasive Software, Inc.
scott.smith@pervasive.com

Contents

Executive Summary

Pervasive.SQL is the perfect database for businesses and departments with low Information Technology (IT) resources. Several considerations go into this analysis which include:

1. **Performance: What the Application User Demands and What Pervasive.SQL Delivers**

   Pervasive.SQL allows developers to implement superior high-performance transactional applications when performance is important, while still allowing relational access to the very same data for information analysis (reporting, graphing, etc.) This is the definition of the Pervasive message, "The Best of Both Worlds" with which developers can have "Transactional" or direct access to underlying data to produce extremely fast applications and at the same time deliver "Relational" access to the same data for analysis and reporting.

2. **Zero Administration Architecture (ZAA): The Foundation of Pervasive Software**

   All Pervasive Software products are designed from the fundamental framework of Zero-Administration Architecture (ZAA). The goal of ZAA is to reduce the amount of administration that is required to use applications built on Pervasive Software's products. This goal is achieved by:

   A. Shifting administration and maintenance tasks away from the end user to inside the application.

   B. Providing all of the tools and infrastructure necessary through both GUI utilities and programmatic APIs to achieve A.

   C. Enabling distributed resource management for better centralized control of database and application servers.

   D. Providing "Scouts" that automate the process of setting up and maintaining connection-based software components such that after installation, all communication and network components are tested and verified.

   E. Keeping tuning options at a minimum, and automating tuning adjustments where appropriate.

3. **Low System Requirements and Broad Cross-Platforms Support**

   Pervasive.SQL memory and disk space are among the lowest in the industry, which is very appropriate for databases designed to be embedded and used in environments where IT resources are at a minimum. The system requirements for

Pervasive.SQL are quite minimal. Pervasive.SQL will install correctly on Windows NT 3.51, Citrix WinFrame, and multiple versions of Novell NetWare.

Pervasive.SQL also operates effectively on a wide range of hardware. Many databases, like SQL Server 7, generally require extensive hardware upgrades because the majority of businesses do not currently have hardware meeting it's requirements.

4. Total Cost of Ownership: Pervasive.SQL Has a Significant Price Advantage over Other Databases

The Aberdeen Group http://www.aberdeen.com, a Boston-based computer industry market research, analysis and consulting organization, recently published a white paper entitled "Low IT Database Cost of Ownership Study."

Aberdeen's overall findings were:

A. There is a significant difference in VCO and post-mortem TCO between Pervasive Software's Pervasive.SQL 7 and Microsoft SQL Server 7;

B. Pervasive.SQL 7 also differentiates itself with regard to other criteria key to many Low IT implementations (e.g., ease of use and maintenance resources required); and

C. These findings are likely to remain true over the next year to year and a half, as past trends in database pricing continue.

Aberdeen also found that:

A. Administrative and maintenance costs continue to increase in importance as a buying-decision factor, and proactive, "designed-in" database maintenance is key to many successful Low IT implementations; and

B. Low IT users have unique database requirements, such as "near-lights-out" administration and minimal training costs, that in many cases enterprise databases cannot meet adequately.

5. Developer Solutions: Active Resources for Application Developers

Pervasive has long valued the thousands of developers world-wide who use Pervasive technology as the data management foundation for their applications. The Pervasive Developer Solutions (PDS) organization was founded to expand the Pervasive developer base and to encourage the development of web, client/server, and mobile packaged applications that embed Pervasive technology. The PDS group strives to deliver developer tools, interfaces, and programs that keep Pervasive Software on the cutting edge of distributed data management.

Pervasive.SQL also enjoys a broad range of third-party tool and utility support. PDS manages Pervasive Software's relationship with third-party vendors and encourages broad support for Pervasive.SQL in all database applications.

Performance: What the Application User Demands and What Pervasive.SQL Delivers

Pervasive.SQL allows developers to implement superior high-performance transactional applications with simple data models when performance is important, while still allowing relational access to the very same data when that model is appropriate. This is the definition of the Pervasive marketing message, "The Best of Both Worlds" with which developers can have "Transactional" or direct access to underlying data to produce extremely fast applications and at the same time deliver "Relational" access to the same data for analysis and reporting.

This is important, because all databases at their core are Transactional. And Pervasive.SQL is the only advanced database management system that provides true Transactional access as an access option coupled with complete relational access. This is precisely why Pervasive.SQL continues to win benchmarks when compared to more cumbersome relational-only systems. As numerous benchmarks have demonstrated, record-oriented access (Transactional) is more efficient that set-oriented access (Relational).

The value of the Transactional data access mechanism in Pervasive.SQL is high-performance and simplicity of the data model. For applications where speed is of utmost importance, no well-designed application should use heavy weight relational constructs regardless of the technology upon which the database is built. With the Pervasive.SQL integrated Btrieve API, and through a number of developer interfaces including ActiveX and Java APIs, developers can build extremely high performing applications that generate data which can later be accessed by standard relational query systems like Crystal Reports.

It is interesting that two of Microsoft's own flagship technologies, Exchange and Active Directory, use a transactional ISAM database (an engine much like the Transactional engine in Pervasive.SQL) as their underlying data store. When asked why Microsoft did not use SQL Server for these products, Margaret Johnson, NT 5 Product Manager, stated, "Hierarchical and sparse data in Exchange 5 and the coming NT 5 Active Directory don't use SQL Server since they required ISAM for transaction performance."

While Pervasive.SQL's transactional access engine allows applications to achieve very high performance, at the same time, Pervasive.SQL provides high performance SQL and ODBC or "relational" access to the very same data. Pervasive sees much value in being able to offer both access methods to application developers, because the requirements of today's applications often require both.

In an application that does not require additional business rules and data integrity definitions, the abstraction or the relational data access model can slow the application down significantly. In fact, Pervasive.SQL's transactional interface can be utilized to tell the underlying engine how to change the data, or it can query the data dictionary

just like the relational engine and use that information to dictate how the data should be changed. So in this case the transactional API provides the best of both worlds by itself.

It has been stipulated by competing databases that the Transactional access methodology should be considered as traditional, incorrect, old-fashioned, or obsolete. This is simply not the case. Many interfaces to Pervasive.SQL, including ActiveX and the Java API, provide developers with the benefit of transactional speed without even knowing it – the interface does the dirty work through the transactional interface. By providing "best of both worlds" data access, you can leverage the speed of the underlying transactional layer without having to learn the mechanics of implementing the transactional (Btrieve) API.

Performance is the primary attribute that an application user required for On-Line Transaction Processing (OLTP) applications in small- and medium-size businesses and in departments of large corporations where IT resources are at a minimum. And performance is precisely what Pervasive.SQL delivers.

Zero Administration Architecture (ZAA): The Foundation of Pervasive Software

All Pervasive Software products are designed from the fundamental framework of Zero-Administration Architecture (ZAA). The goal of ZAA is to reduce the amount of administration that is required to use applications built on Pervasive Software's products. This goal is achieved in five ways:

1. Shift administration and maintenance tasks away from the end user to inside the application.

2. Provide all of the tools and infrastructure necessary through both GUI utilities and programmatic APIs to achieve #1.

3. Enable distributed resource management for better centralized control of database and application servers.

4. Provide "Scouts" that automate the process of setting up and maintaining connection-based software components such that after installation, all communication and network components are tested and verified.

5. Keep tuning options at a minimum, and automate tuning adjustments where appropriate.

Pervasive.SQL is deployed as an administration-free embedded engine built on this foundation. With ZAA, Pervasive.SQL largely eliminates the need for a DataBase Administrator (DBA) by keeping the number of necessary tuning options at a minimum and by pushing administration away from the user and into the database application itself.

The next release of Pervasive.SQL, known as Pervasive.SQL 2000, provides the unified Pervasive Control Center as well as embedded COM based controls based on the new Distributed Tuning Interface (DTI) for remote administration of multiple servers. It is the DTI that enables the application developer to shift administration tasks within the application, away from the end-user. It also allows for remote administration and distributed resource management.

Pervasive.SQL also has a process during installation called InstallScout. InstallScout verifies that everything was installed correctly on both the client and sever side of the install. This verifies the integrity of the entire installation, including network communication, and helps to debug any problems that may exist. With other databases, there is no concept or equivalent of InstallScout protection. If your network drivers or hardware are not correctly configured, operational and in sync with your database setup, you will not know during the installation process.

Low System Requirements and Broad Cross-Platforms Support

Pervasive.SQL required memory and disk space are among the lowest in the industry, which is very appropriate for databases designed to be embedded and used in environments where IT resources are at a minimum. Pervasive.SQL v7.0 requires 16MB of RAM and 36 MB of hard disk space. Microsoft SQL Server 7, in contrast, requires 32 MB of RAM and 256 MB of hard disk space.

Indeed, with Pervasive.SQL, the system requirements are quite minimal. In fact, Pervasive.SQL will install correctly on Windows NT 3.51 and Citrix WinFrame. Pervasive.SQL does not require that you have the latest Operating System and tools to install and run. It even installs on multiple Novell NetWare versions: 3.2, 4.x, or 5.x, and is NetWare 5 certified.

Not only does Pervasive.SQL run on the above platforms, it has a single source code base that runs on WinCE, Win95, Win98, NT Workstation, NT Server, and Novell NetWare. Pervasive.SQL also provides client application support for DOS, Win16, Win32, and OS/2 applications.

Pervasive.SQL also operates effectively on a wide range of hardware. Many databases, like SQL Server 7, generally require extensive hardware upgrades because the majority of businesses do not currently have hardware meeting it's vast requirements.

Pervasive Software is committed to providing broad cross-platform and cross-Operating System support, including all mobile platforms such as WinCE, the Palm OS, and devices based on other embedded operating systems.

Total Cost of Ownership: Pervasive.SQL Has a Significant Price Advantage over Other Databases

The Aberdeen Group http://www.aberdeen.com, a Boston-based computer industry market research, analysis and consulting organization, recently published a white paper entitled "Low IT Database Cost of Ownership Study."

The aim of this White Paper was to update Aberdeen's examination of the cost of ownership of the Pervasive Software Pervasive.SQL 7 and Microsoft SQL Server databases for small- to medium-sized businesses (SMBs) and workgroups and departments within larger businesses. The paper contains the following items:

1. A "Visible Cost of Ownership" (VCO) study estimating cost-of-ownership for both Pervasive.SQL 7 and SQL Server, that uses the same cost-of-ownership criteria as the previous study;

2. An examination of real users' actual Total Cost of Ownership (TCO) experience in the last 2 years, based on user data from two TCO case studies; and

3. A set of criteria that either users or Aberdeen find most useful in assessing databases for the Low IT market.

Aberdeen's overall findings were:

1. There is a significant difference in VCO and post-mortem TCO between Pervasive Software's Pervasive.SQL 7 and Microsoft SQL Server 7;

2. Pervasive.SQL 7 also differentiates itself with regard to other criteria key to many Low IT implementations (e.g., ease of use and maintenance resources required); and

3. These findings are likely to remain true over the next year to year and a half, as past trends in database pricing continue.

Aberdeen also found that:

1. Administrative and maintenance costs continue to increase in importance as a buying-decision factor, and proactive, "designed-in" database maintenance is key to many successful Low IT implementations; and

2. Low IT users have unique database requirements, such as "near-lights-out" administration and minimal training costs, that in many cases enterprise databases cannot meet adequately.

Total cost is an important consideration when choosing which database to use for an application. For environments where IT resources are already at a premium, it is

significant that Pervasive.SQL's overall long term costs are much less than many other commercial database development and deployment systems.

Developer Solutions: Active Resources for Application Developers

Pervasive has long valued the thousands of developers world-wide who use Pervasive technology as the data management foundation for their applications. In February of 1998, Pervasive Software acquired Smithware, Inc., the leading third-party developer tool solution company for Pervasive-based products. Smithware formed the foundation of a division inside Pervasive Software called "Developer Solutions."

Simply put, the mission of Pervasive Developer Solutions (PDS) is to expand the Pervasive developer base and to encourage the development of web, client/server, and mobile packaged applications that embed Pervasive technology. The PDS group strives to deliver developer tools, interfaces, and programs that keep Pervasive Software on the cutting edge of distributed data management. PDS achieves these goals by:

1. Defining and delivering SDK products containing compelling, productivity enhancing interfaces and utilities to developers.

2. Working with third-party tool vendors to build cooperative technology that compels them to embed Pervasive's SDK's, Database engines, Application Servers, or data management solutions in their development platforms.

3. Providing technical expertise in developer products and tools to drive the right requirements into our product line.

4. Driving the Pervasive Software Developer's Journal, PervasiveDevWire, and DeveloperZone as communication vehicles for Pervasive developer news.

5. Acting as technical evangelists for Pervasive at user groups and conferences, and internally within Pervasive on the behalf of developers.

Pervasive offers many solutions for developers that are available at the Pervasive Developer Zone http://www.pervasive.com/developerzone. The Developer Zone is where you can find what you need to make developing applications with Pervasive database technology faster, easier, and more powerful. Sample code, tutorials, access to an expansive technical library, and links to third-party tools and utilities are just a few of the practical resources you'll find there.

The new Pervasive.SQL SDK is a comprehensive set of developer tools designed to enable rapid development of high-performance business-critical applications. With the Pervasive.SQL SDK, you can embed a low-cost database that integrates with existing development tools and enables seamless migration from single-user to client/server and Web-based applications. The tool set includes support for today's widely used development environments, such as Visual Basic, Visual C++, Delphi, and Java.

Pervasive.SQL also offers native support for many common database API's as well, including ODBC. We also support applications written to the Remote Data Objects (RDO) and Data Access Objects (DAO) APIs. Pervasive.SQL 2000 (scheduled for release in the Summer of 1999) fully supports ADO and will add OLE-DB support for developers who adopt these new interfaces. In addition, we will continue our commitment to Java developers by offering a JDBC interface in Pervasive.SQL 2000.

One of the highlights of the Pervasive.SQL 2000 release is the greatly improved ODBC interface for the database engine. The SQL engine in Pervasive.SQL 2000 has been re-engineered to provide increased ODBC performance and SQL syntax standardization.

Fortunately, for the developer and the application end user, Pervasive.SQL also provides interfaces that enable applications to execute blindingly fast. By exploiting our core MKDE interface, these applications always out perform applications which are written to higher level interfaces. For example, our standard ActiveX Interface, written to Microsoft's proprietary VB data binding specification, offers industry standard development in Visual Basic while providing the application with unparalleled performance. This interface shields the developer from all complexity, giving all the benefits of Rapid Application Development but with blazing transactional database performance.

Pervasive.SQL targets both Microsoft and non-Microsoft development platforms for universal development. Support for a vast array of development environments has helped Pervasive.SQL win acclaim from a large group of core developers. The Pervasive Developer Zone has an up-to-date list of supported environments at http://www.pervasive.com/developerzone/thirdparty/index.html#ENVIRONMENTS which include Visual Studio, JBuilder, Delphi, Visual Café, COBOL, and more.

Pervasive Software will soon provide common development guidelines via ODBC with a common SQL syntax between Oracle PL/SQL and Pervasive.SQL applications. This means that developers can deploy the same application against either a Pervasive.SQL database backend for SMB or corporate departmental applications, or against an Oracle backend for enterprise level applications.

Pervasive.SQL continues to build strong relationships with third-party vendors who support developer and end-user tools that work with Pervasive.SQL. The Pervasive Developer Zone has a comprehensive list of all third-party tools (http://www.pervasive.com/developerzone/thirdparty/index.html) that are listed in four product categories:

1. Development Environments

2. Development Tools

3. Development Utilities

4. Other Tools and Utilities