

Changed MicroKernel Options

The following MicroKernel options have changed from version 6.x to version 7:

Option	Change
Open Files	Memory required is now 1,024 bytes per file.
Create Files in Pre-6x Format	Name changed to Create File Version. Range is now 5x, 6x, or 7x. Default is 7x.
Logging of Selected Files	Name changed to Archival Logging of Selected Files.
Handles	Support for more than 64,000 file handles.
Cache Allocation	Default is now 1,024 KB.
Perform Transaction Durability	Name changed to Transaction Durability. Default is now On.
Operation Bundle Limit	Default is now 1,000.
Initiation Time Limit	Default is now 1,000.
Worker Threads	Range is now 1 through 128.
I/O Threads	Range is now 1 through 128.
Allocate Resources at Startup	Default is now On.
Back to Minimal State if Inactive	Default is now On.

Compatibility

Pervasive.SQL 7 is compatible with all Btrieve 6.15 features and functions, including the Btrieve API and data file formats (4.x, 5.x, and 6.x). Applications developed for the Btrieve 6.15 server or workstation engines will run on Pervasive.SQL 7. Moreover, 6.15 applications running in mixed environments with 6.15 and 7 engines on multiple platforms are also fully supported as long as the data files are in 6.x format.

Note

- Applications developed for Pervasive.SQL 7 are not guaranteed to run on 6.15 engines, particularly if they make use of Pervasive.SQL 7 features. Also, Pervasive.SQL 7 utilities are not backward compatible; you must run them with a version 7 engine. More importantly, 6.15 utilities will not run against the version 7 engine, because they make assumptions about the underlying Pervasive.SQL implementation.
- Pervasive.SQL's Btrieve Interface is also compatible with Scalable SQL 3.01, Scalable SQL 4.x, and all versions of the ODBC Interface.

DDF Ease

With this utility, you can create data dictionary files (DDFs) for your existing Btrieve data files. You can also add relational capability to your Btrieve database. For explicit instructions on using DDF Ease, run its online help system or refer to the Pervasive.SQL User's Guide.

DOS Support

The Pervasive.SQL 7 Workstation release supports execution of Btrieve DOS applications running on a Win32 computer that is correctly configured with client components and is running the Workstation engine. This capability supports executing multiple DOS applications each within their own DOS box on a single machine running a single instance of the workstation engine. On Windows 9X only, you must load one copy of BTRBOX95 located in the Pervasive.SQL 7 program group.

- This configuration supports access to local workstation engines.
- The installation program automatically installs and configures VxD and VDD components on Windows 9X and Windows NT, respectively. Installation may include addition of registry settings to register VxD (on Windows 95/98) or modification of config.nt (on Windows NT).
- Btrieve DOS applications running on a DOS only computer (e.g., DOS 6.22) are not supported.
- Scalable SQL DOS applications are not supported.

Database Engine Autoload Support

The workstation engine is loaded when a user starts a Pervasive.SQL application and makes a Btrieve or Scalable SQL API call. The engine will remain loaded in memory until all Btrieve or Scalable SQL applications have correctly released all Btrieve and Scalable SQL resources (i.e. logged out, closed files, issued correct number and type of stops).

Distributed Database Support

The transactional (Btrieve) and relational (Scalable SQL) interfaces in the Pervasive.SQL workstation engine support writing distributed database applications by providing interfaces that hide the details of connecting to a local (via Pervasive.SQL Workstation) or remote database engine (via Pervasive.SQL Server) from the application. Using this architecture an application can access data that is co-located with the application (i.e. is running on the same computer as the application) while also accessing data on a remote computer.

Scalable SQL and Distributed Engines

Because the Scalable SQL engine uses Btrieve APIs, requests that the Scalable SQL engine issues to the Btrieve interface can potentially be serviced by either a local or remote engine. This type of environment allows database operations to be split across local and remote engines. Additionally, a file can be opened by either type of engine. For example, data dictionary files (DDFs) are serviced using a local engine while some data files are serviced by a remote engine.

This section describes the feature constraints or behavioral changes when a Pervasive.SQL database is not serviced exclusively by a local engine. This type of database is referred to as a mixed access database. When necessary, reference to a database that is serviced exclusively by a local engine will be referred to as local access database. Any reference to a database whose characteristics are not altered by its designation as a mixed or local access database be referred as the generic term database.

Features not supported when accessing a mixed use database:

- 1 Referential Integrity (RI)
- 2 Bound Databases
- 3 Triggers
- 4 Distributed Transaction Atomicity (requires two-phase commit)

Behavioral changes when accessing a mixed use database:

- 1 The Pervasive.SQL transactional and relational engines must be running on the same workstation to access DDFs. Access to DDFs where the relational engine is running on a separate workstation than the transactional engine is not supported.
- 2 Data files for tables that are involved in a RI relationship, or those that have any triggers defined for them or are in a Bound Named Database, cannot be opened by a remote engine.
- 3 When opening a file, the Pervasive.SQL workstation's relational engine does not verify the version of the transactional engine servicing the request. If an operation that requires version 6.30 or higher Btrieve API support (e.g. nested transactions or shared locking operations) is issued to an engine less than version 6.30 then a status code occurs. When opening DDF files or when attempting to bind a DDF or data file, the Pervasive.SQL workstation's relational engine will verify that the local workstation engine is servicing the request.

Enhanced Maintenance Utility

Both the command-line based (BUTIL) and the interactive utilities provide roll forward capability. (There is no separate utility in Pervasive.SQL for roll forward functions.) In addition, the command-line based utility provides support for large (extended) files and the ability to read a file forward or backwards in SAVE and RECOVER operations. For more information, see the Pervasive.SQL User's Guide.

Enhanced Monitor Utility

This utility now provides monitoring capabilities for remote servers. In addition, the Communication Statistics screen has been split into two screens, one for SPX and one for TCP/IP. For more information, see the Pervasive.SQL User's Guide . Previous versions of the Monitor utility will not work with the 7.0 MicroKernel.

Enhanced ODBC Interface

ODBC version 2.53 is based on ODBC 2.50 that shipped in the Pervasive.SQL 7 Server product. Both 16- and 32-bit drivers are ODBC Specification 2.5 Level 2 compliant.

ODBC configuration and reliability issues using the ODBC 2.50 code base will be addressed as follows:

ODBC 2.53 uses the Pervasive.SQL workstation's relational engine. If you are a 16-bit ODBC customer, you must thunk to a Pervasive.SQL workstation engine if you are running on a Win32 platform. ODBC 2.53 does not support Win16 only environments, meaning environments where no 32-bit Pervasive.SQL workstation engine exists.

With ODBC 2.53, access to Pervasive.SQL data stored on computers running a Btrieve v6.15 server will **not** be supported using a Pervasive.SQL 7 workstation engine.

The 32-bit ODBC installation program installs ODBC and Pervasive.SQL workstation client components. It does not install or ship the Pervasive.SQL workstation's relational engine or other Pervasive.SQL workstation components. It does use the Pervasive.SQL workstation's relational engine and other Pervasive.SQL workstation components installed as part of the Pervasive.SQL Workstation or Pervasive.SQL Server installation program.

Enhanced Setup Utility

This utility now provides configuration capabilities for remote servers. For more information, see the [Pervasive.SQL User's Guide](#) . Previous versions of the Setup utility will not work with the version 7 MicroKernel.

Enhanced Roll Forward Capability

Rolling forward is the process by which you recover changes made to a data file between the time of the last backup and a system failure. Previously, roll forward functions were performed by a separate Roll Forward utility. In Pervasive.SQL 7, the MicroKernel performs roll forward tasks when instructed by the Maintenance utility. This enhances roll forward performance and reliability.

File Conversion

Pervasive.SQL 7 uses a new file format, which supports the new large files and transaction durability features. If you want to use these features, convert your files to 7.x format. For more information, see the chapter "Converting MicroKernel Data Files" in the Pervasive.SQL User's Guide.

Engines earlier than version 7 cannot open 7.x files; however, version 7 of the MicroKernel can open pre-7.x files. When the MicroKernel opens a pre-7.x file, it does not convert the file to 7.x format. (You can configure the MicroKernel to create pre-7.x files. This may be useful if you want to use newly created files with earlier engines.)

Improved Installation Program

The installation program installs Pervasive.SQL 7 Workstation components in an InstallShield 3 based installation program. The following components are installed:

- Workstation Engines
- Workstation Client Components, including the DOS box installation and configuration
- Pervasive.SQL 7 Requesters
- ODBC (16-bit, 32-bit)
- Programming Interfaces
- Documentation
- Utilities

Use the Setup Utility, documented in the Pervasive.SQL User's Guide, to configure all Pervasive.SQL 7 Workstation components.

Installation Directories

Workstation components are installed as follows:

- Shared Windows files, DLLs and their associated resource files, help files, etc., are installed into the Windows System Directory. Under Windows NT, the 32-bit shared components are installed into the standard Windows System Directory, but the 16-bit shared components are installed into %windir%\system.
- Non-shared Windows files are installed into a directory or subdirectory under a target root directory of your choosing.

ODBC Install

The Win32 ODBC installation program can detect if a Workstation engine is not installed on the workstation and issues a warning.

Related Topics

{button „JI(\produpd.HLP>wn-small`,`Enhanced_Setup_UTILITY`)} Enhanced Setup Utility

{button „JI(\produpd.HLP>wn-small`,`Enhanced_ODBC_Interface`)} Enhanced ODBC 2.53

Internationalized Btrieve Interface

This feature extends the use of alternate collating sequences by providing a set of multi-weight collation tables you can use to sort international string values with an ISO-defined, language-specific collating sequence. The tables provide for collation in English, French, German, Spanish, and Japanese (Shift-JIS character set). For more information, see "International Sort Rules" defined in the Pervasive.SQL User's Guide.

Large Files

Data files that use the version 7 format can expand up to 64 GB. Pervasive.SQL 7 applications treat the large file as a single, logical data file, but at the operating system level the large file consists of up to 32 two-gigabyte files. This 2 GB limit allows the large file to be portable across all Btrieve-supported platforms. For more information, read Large Files in the Pervasive.SQL User's Guide.

Developers : The MicroKernel's large file support feature requires a maximum of 256 records per page. This limit can affect the optimum page size in files that have very small records and very large page sizes. For more information, read "Designing the Database" in the Pervasive.SQL Programmer's Guide.

New Configuration Options

The following options are new in Pervasive.SQL 7. See the Pervasive.SQL User's Guide for descriptions:

- Database Names Directory
- Log Buffer Size
- Maximum Databases
- Minimal State Delay
- System Data
- Thread Priority Delta
- Transaction Log Size
- Wait Lock Timeout
- Working Directory

New Diagnostic Utilities

SmartScout and InstallScout. These utilities help prevent configuration and installation problems, and are installed and run as part of the Pervasive.SQL installation process. All of these utilities can be run later to evaluate configuration and registry settings and to troubleshoot problems. See "Troubleshooting Installation Problems" in Getting Started with Pervasive.SQL.

New Interactive Rebuild Utility

This utility complements the existing command-line based utility (BREBUILD) and provides equivalent capabilities for file format conversion. For more information, see Chapter 10, "Converting MicroKernel Data Files," in the Pervasive.SQL User's Guide.

Overview of MicroKernel Database Engines

Pervasive.SQL MicroKernel Database Engine version 7 provides the following features and performance enhancements:

```
{button ,JI(`produpd.HLP>wn-  
    small`,`Changed_MicroKernel_options`)}  
    Changed MicroKernel Options  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Compatibility`)}__Compatibility  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Enhanced_roll_forward_capability`)}__  
    Enhanced Roll Forward Capability  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`File_Conversion`)}__File Conversion  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`International_Btrieve_Interface`)}__  
    Internationalized Btrieve Interface  
  
{button ,JI(`produpd.HLP>wn-small`,`Large_Files`)}  
    Large Files  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`New_configuration_options`)}  New  
    Configuration Options  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Pervasive.SQL_Event_Logging`)}__  
    Pervasive.SQL Event Logging  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Smart_Components`)}__Smart  
    Components  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Support_for_more_than_64_000_file_han  
    dles`)}__Support for more than 64,000 file  
    handles  
  
{button ,JI(`produpd.HLP>wn-  
    small`,`Transaction_Durability`)}__Transaction  
    Durability
```

Pervasive.SQL 7 Workstation Features

The Pervasive.SQL 7 Workstation supports many of the features of the Pervasive.SQL 7 Server product, as well as the following features:

```
{button ,JI(^produpd.HLP>wn-  
    small,'Database_Engine_Autoload')}} Database  
    Engine Autoload Support
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Distributed_Database_Support')}}  
    Distributed Database Support
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'DOS_Support')}} DOS Support
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Enhanced_ODBC_Interface')}}  
    Enhanced ODBC Interface
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Workstation_Installation_Program')}}  
    Improved Installation Program
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'International_Btrieve_Interface')}}  
    Internationalized Btrieve Interface
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Workstation_Documentation')}} Product  
    Documentation
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Single_User_Engine_Support')}} Single-  
    User Engine Support
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Workstation_Platform_Support')}}  
    Supported Platforms
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Supported_Uilities')}} Supported  
    Utilities
```

```
{button ,JI(^produpd.HLP>wn-small,'Tray_Icon')}}  
    Tray Icon
```

```
{button ,JI(^produpd.HLP>wn-  
    small,'Win16_Application_Support')}} Win16  
    Application Support
```

Pervasive.SQL Event Logging

The MicroKernel log file has been replaced with a new, centralized event log that Pervasive.SQL components share. This log contains informational, warning, and error messages. When receiving some types of status codes now, more detailed information can be found in this log pertaining to that error. Event logs are generated at both the client and the server machines. For more information, read Event Logging in the Pervasive.SQL User's Guide.

Product Documentation

The following documentation is included with the Pervasive.SQL 7 Workstation release in both hard copy and Windows Help formats:

- Getting Started with Pervasive.SQL (Workstation edition)
- Pervasive.SQL User's Guide
- Status Codes and Messages
- Status Codes and Message Quick Reference Card (print only)
- Pervasive Products and Services
- SQL Language Reference
- ODBC Interface Reference

Single-User Engine Support

The Pervasive.SQL 7 Workstation product incorporates the single-user database mode (SUDM). This architecture design supports single-user access to files that reside on the computer on which the engines are running (i.e. files are co-located with the engines) or that can be access via a mapped drive..

Smart Components

A new architecture governs how Pervasive's transactional interface (Btrieve) handles communication among its client components. By ensuring that outdated or incompatible components are not used by mistake, this architecture improves installation and run-time reliability and makes application troubleshooting easier. For more information about troubleshooting your installation, read "Troubleshooting Installation Problems" in *Getting Started with Pervasive.SQL*. For more information about developing Pervasive.SQL applications in the Smart Components environment, read "Component Architecture" in the *Pervasive.SQL User's Guide*.

Support for more than 64,000 file handles

Version 7 of the MicroKernel allows you to assign up to 4 billion file handles. Previously, the MicroKernel limited file handles to 64,000.

Supported Platforms

Pervasive.SQL 7 Workstation

- The Pervasive.SQL version 7 Workstation installation is a 32-bit only installation application that only operates under Windows 9X or Windows NT 4.0.
- At least 16 MB of free memory (32 MB is recommended).
- At least 28 MB of free disk space for the engine components in C:\WINDOWS\SYSTEM.
- Up to an additional 60 MB for the following client components, assuming you choose to install all components:
- Pervasive.SQL documentation in Windows Help format.
- Pervasive configuration tools and utilities, such as Setup, Rebuild, Maintenance, SmartScout, DDF Ease, and Function Executor.

Supported Utilities

The following utilities are supported in the Pervasive.SQL 7 Workstation product:

Utility	DOS, Win16, Win32
■ <u>Btrieve Maintenance Utility (GUI)</u>	Win32
■ BUTIL (command line)	DOS, Win32
■ SQLUTIL (command line)	Win32
■ <u>DDF Ease (GUI)</u>	Win32
■ <u>InstallScout (GUI)</u>	Win16, Win32
■ Function Executor (GUI)	Win16
■ <u>Rebuild (GUI)</u>	Win32
■ <u>Setup (GUI)</u>	Win16, Win32
■ <u>SmartScout (GUI)</u>	Win32
■ SQLScope (GUI)	Win16
■ View Conversion (command line)	Win32

Transaction Durability

The MicroKernel version 7 employs a new transaction logging scheme that tracks all operations on all files in a single log. This new scheme allows the MicroKernel to guarantee transaction persistence and atomicity, improve performance, and use a larger system transaction size.

Transaction durability is accomplished by uniquely identifying each record, either by a user-defined unique key or by a system-defined log key. System-defined log keys are called system data. Consequently, if a file does not have system data or a unique user-defined key, the file is not transaction durable.

To determine if a file is transaction durable, retrieve the file's statistics. You can do this using the BUTIL -STAT command, using the Maintenance utility's Create Stat Report command or calling the Btrieve Stat operation. If you have existing 6.x files that are not transaction durable because they do not have unique user-defined keys, you should add system data. Because the MicroKernel adds system data only to version 7 files, you must update the file format when you add system data.

To convert a 6.15 file to 7 format and add system data, see the chapter "Converting MicroKernel Data Files" in the Pervasive.SQL User's Guide which describes the Rebuild utility.

In Btrieve 6.x, the Transaction Durability configuration option was turned off by default. In 7.x the option is turned on by default for two reasons. First, guaranteeing transaction persistence and atomicity is beneficial. Second, the default system transaction size is larger than in Btrieve 6.x, which improves performance at the risk of losing more work in the event of a crash if durability were not turned on.

For more information about these topics, read about Transaction Durability and System Data in the chapter "Configuring Components Using the Setup Utility" in the Pervasive.SQL User's Guide.

Tray Icon

A Tray icon is displayed on the Windows 9X/NT task bars to provide a graphical indication that the workstation engine is running. Examples of tray icons are Windows NT Task Manager CPU graph, 3Com Network Adapter, Real Audio Real Player launcher, Volume control for PCs equipped with audio, etc. The tray icon does not appear when the engines are not running.

Pervasive.SQL 7 - The Best of Both Worlds

Pervasive.SQL version 7 delivers the best of both worlds in one total database management solution: the best features of a navigational architecture paired with the power of relational capabilities. With this unique structure, Pervasive.SQL provides unbeatable transaction processing and maximum programming control in addition to exceptionally fast querying and data manipulation. Pervasive.SQL brings you the ultimate in reliable, maintenance-free, resource-efficient operations, with world-class performance and usability.

Pervasive.SQL 7 Workstation

The Pervasive.SQL 7 Workstation product is targeted to single-use customers and developers. This means that a customer cannot use the Workstation engine to share files with other Workstation engines.

The Workstation product is an inexpensive means to demonstrate Pervasive.SQL applications in environments that cannot easily support a Pervasive.SQL 7 Server installation (e.g. by a Field Sales person using a laptop to demonstrate a Pervasive.SQL application).

The Mustang product can be described in terms of the following components:

- **Pervasive Database Engines.** The Pervasive Database Engines (PDE) provide Btrieve (navigational) and Scalable SQL (relational) API support. The engines support requests from applications running on the same computer.
- **Interface DLLs/DOS Requesters.** Interface DLLs for Win16 and Win32 perform the operations necessary to route requests to and replies from a PDE on the local computer. Interface DLLs are named using Smart Component Naming. DOS Requesters perform the operations necessary to route requests to and replies from a PDE on the local computer. DOS Requesters are **not** named using Smart Component Naming.
- **Glue DLLs.** Glue DLLs for Win16 and Win32 provide a well known name for applications to link to that locate, load, and bind to the correct Interface DLL. Glue DLLs are named using Glue Component Naming.
- **ODBC.** ODBC is an API interface to the ODBC Driver Manager and the selected underlying ODBC data source driver.
- **Utilities.** Utilities help troubleshoot configuration problems (SmartScout, InstallScout) and maintain databases (Maintenance, Rebuild, and DDF Ease).

Related Topics

{button „JI(^produpd.HLP>wn-
mid',`1btrintr_rtf_13166')}} Overview of
Pervasive.SQL Engines

{button „JI(^produpd.HLP>wn-
mid',`Pervasive.SQL_Workstation_Features')}}
Pervasive.SQL Workstation Features

Win16 Application Support

Win16 applications running on a Win32 computer that is correctly configured with Pervasive.SQL Workstation client components is supported using the existing Btrieve and Scalable SQL thunk mechanism. This configuration supports access to local workstation engines and remote Pervasive.SQL server engines.

Win16 applications running on a Win16 only computer (e.g. Windows 3.1, Windows for Workgroups) are not supported by the Pervasive.SQL 7 Workstation.

In the context of the Pervasive.SQL 7 MicroKernel, files are co-located with the MicroKernel Database Engine if they reside on the computer on which the engine is running.

Single-user access refers to one user on one computer accessing Pervasive.SQL data files. Multiple applications on one computer accessing the same Pervasive.SQL data file concurrently is an example of single-user access.

Thunking is the transition from executing 32-bit code to executing 16-bit code, or from executing 16-bit code to executing 32-bit code. The piece of code that makes the transition is called a thunk. This thunk is a mapping layer that converts data to allow 32-bit code and 16-bit code to communicate with each other. A thunk allows applications to use controls of another memory model without the application knowing the memory models are different.

