Pervasive Backup Agent™

Revision 1.0
March 2004

Pervasive Software Inc

TABLE OF CONTENTS

Overview

Backing up databases used to be fairly straightforward. Wait until all users leave for the weekend, shut down the database engine, copy all data files to tape, then restart. Over time this simple procedure has become increasingly complex and unworkable. Technology and business trends have increased the demand for continuous system availability, shrinking or even eliminating downtime windows. Applications have become mission critical, increasing the frequency of backups needed to maintain acceptable data loss exposure. Applications and databases have become more complex, mandating higher file counts and backup sizes.

This paper presents the technology of Pervasive Backup Agent. Backup Agent provides value when organizations need to perform database backups, especially if the backup must be performed when the database is online.

Database Challenges

Backing up a database is more complex than simply backing up a set of files. As many people have learned the hard way, restoring a file copy of an operating database rarely succeeds. Database backups have two specific challenges, avoiding file corruption and ensuring database consistency.

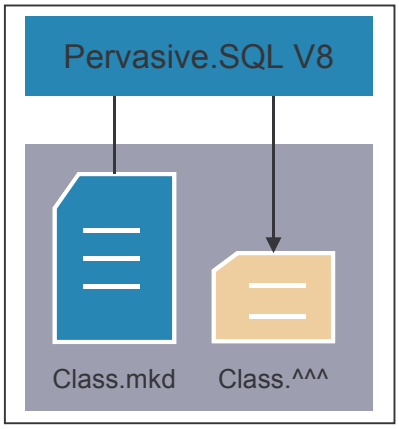
Sophisticated database engines like Pervasive.SQL assume they are the exclusive owners of data files on disk. In order to optimize performance and reliability, they may write transaction information out to any position in the file. For any individual transaction, some bits may be written to the beginning of the file while others may be written to the end. Unfortunately, the backup process for a large file may span multiple minutes. Since the database engine may be writing information to any position in the file at any time, the file image captured to tape may be invalid. The beginning of the copied file may be inconsistent with the end, resulting in a corrupt backup image.

An even more difficult challenge is ensuring database consistency. For a database like Pervasive.SQL that stores data in multiple files, a single transaction may write to multiple physical data files. Given the slow and sequential nature of a backup that copies one file at a time, it is likely that the files written to tape will be from different times and will be logically inconsistent. For example, a customer payment may include an insert to the cash table and a corresponding insert to the receivables table. It is possible that the backup process may copy the cash table before the transaction and the receivables table after the transaction, resulting in an inconsistent pair of tables in the backup set.

The easiest solution, of course, is to force all users to exit their application and shut down the database. However, as noted above, business trends are shrinking and sometimes even eliminating downtime windows. In order to meet system uptime requirements while still protecting mission critical data through frequent backups, what is needed is a way to ensure that individual database files are in a pristine state and that the collection of files in a single backup set is consistent. These needs are solved the Pervasive.SQL continuous operations feature with Pervasive Backup Agent.

Pervasive.SQL Continuous Operations

The Pervasive.SQL database engine supports *continuous operations*, a special mode that allows you to back up a data file while it is **open and in use**. The database engine opens a file like class.mkd in read-only mode to allow back up utilities to access the file's static image. Since the file is opened read-only, it is necessarily in an internally consistent state. The database engine stores changes to the original file in a temporary file called a delta file called class.^^^. The database engine automatically creates the delta file when the main file enters continuous operations mode. When the backup is complete, the database engine updates the original file by rolling in the changes stored in the delta file and then deletes the delta file as soon as all applications close the data file. The database engine properly handles operations on the file, such as user queries, by reading both the original and the delta file. The fact that a file is in continuous operations is invisible to applications and users.

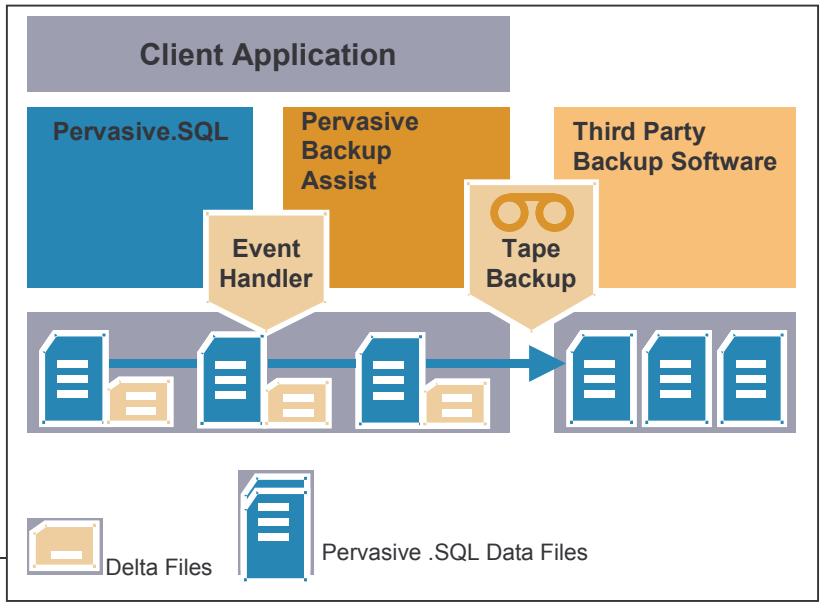


Intelligent Continuous Operations Management

While continuous operations can be used on its own, it's not easy. The challenge with continuous operations is determining what files to turn the mode on for. The typical approach for Pervasive.SQL users it to create a file with the file names of all database files, and calling the utility `butil -startbu` with the file. This turns continuous operations mode on for the listed files. This approach has two problems.

- Files are placed into continuous operations mode whether they are open or not. Since continuous operations mode physically alters the file to indicate the mode is on, the modified timestamp of the file is set. Database files that have not been modified by user activities still get their modified timestamp set, so incremental or differential backup jobs needlessly backup these files, wasting backup space and time.
- It is difficult to populate and maintain the list. This is especially true for applications that create database files during normal operations, like most accounting applications do during their periodic closing process.

Backup Agent solves these problems by implementing intelligent continuous operations management. It automatically places files that are opened during a backup session into continuous operations. Therefore, users or administrators do not have to maintain lists of files that are to be backed up (no more `butil -startbu` scripts to get out of date). Further, files that are not opened by user actions during a backup session will not be put into continuous operations, so they



will not have their modified timestamp changed. This reduces the amount of data that must be backed up when performing incremental or differential backups.

The key to Backup Agent is that it manages continuous operations for all database files. When backup mode is turned on, it turns continuous operations mode for all currently open files. Further, it manages continuous operations for subsequently opened files.

The key component of Backup Agent is its *event handler*. This component plugs into a private API of the Pervasive.SQL database engine. It monitors the database for open events (Btrieve operation 0). When an open event occurs, the event handler checks to see whether the file is already in continuous operations mode. If not, it turns continuous operations mode on for that file. This ensures that all files will be consistent with the state when backup mode was turned on.

In the case of server failure during a backup process, Backup Agent will force a roll in from the delta files to the data files upon system restart. This relieves the administrator from having to manually open individual data files.

Activation

Pervasive Backup Agent can be activated in three user-friendly ways, depending on your mode of use.

- **Graphically through the Backup Agent GUI**. This is ideal for ad-hoc backups. Starting Pervasive Backup Agent from its GUI before your backup will place the Pervasive.SQL database in backup mode. When you are done, stopping Backup Agent will return the database to normal operations.

- **Scripted from a command line**. This is ideal for use with an existing backup software solution. Add Pervasive Backup Agent start and stop commands to your backup software's pre and post commands. Your backup jobs will then automatically call Backup Agent at the appropriate times, guaranteeing consistent and reliable Pervasive.SQL database backups for your regularly scheduled backup jobs.

- **Embedded in an application**. Pervasive Backup Agent supports an API. For application developers, this can Agent your users in protecting their data. By integrating Backup Agent directly into applications, you can provide the highest level of backup data assurance.

Deployment

Pervasive Backup Agent will normally be deployed in conjunction with 3rd party backup products. The diagram below illustrates how Backup Agent integrates into this environment.

Client applications access the Pervasive.SQL database engine as usual. Applications will normally have no idea that Backup Agent is installed and operational.

Backup Agent communicates with both the Pervasive.SQL database engine and the 3rd party backup software. The backup software sends start and stop backup mode commands to Backup Agent. Backup Agent controls continuous operations for Pervasive.SQL data files. This results in the tape backup copying only clean and consistent Pervasive.SQL data files. The backup software backs up the Pervasive.SQL data files, ignoring the delta files.

Conclusion

Pervasive Backup Agent is fast, simple and easy to use. It automates the Pervasive Database Engines continuous operations function and intelligently manages database operations. Pervasive Backup Agent provides constant data availability and minimizes the impact on performance and backup size.

For More Information

For more information about Pervasive Backup Agent please visit our Web site at www.pervasive.com, email us at info@pervasive.com, or locate one of our sales offices worldwide at www.pervasive.com/company/contact.

Disclaimer: The performance demonstrated in this document is for reference purposes only and does not constitute a performance guarantee even if under similar configurations.