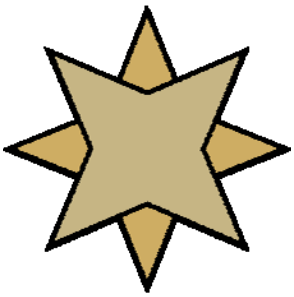
Btrieve Data File Rebuilding

A White Paper From

Goldstar Software Inc.



For more information, see our web site at
http://www.goldstarsoftware.com

Btrieve File Rebuilding

Last Updated 01/05/2006

The Pervasive.SQL database engines can read files in ALL file formats, going back as far as Btrieve v3! As such, you are not required to upgrade your files when you upgrade your database engine. In fact, you can continue to run the old file format for whatever period of time you wish, until you are certain that there will be no need to ever go back to the previous database version. This makes it far superior to MS SQL Server and some other databases that require a full rebuild and have an extensive recovery if you ever need to roll back.

However, the newer database file formats do provide some benefits:

- Newer formats provide additional features that may be required for some applications, like VATs.
- Newer formats are typically more stable on the newer engines, since they are better tested.
- Newer formats can provide additional performance benefits over older versions. This is especially true of the upgrade from the older formats to 6.x format, and again in the upgrade to the 8.x format.
- The 6.x format supports at most a 4GB file. The 7.x and 8.x formats allow files to grow to a logical 64GB in size, depending on the page size of the file. It does this through the use of 2GB extents, that have names like *filename.^01*. The 9.x format supports a 128GB file when used with 8K pages.
- The Pervasive PSQL v9 database engine supports reading 5.x files only – it does NOT support writing to these files. As such, you may need to rebuild them to 6.x or newer.

To take advantage of these benefits, you may need to rebuild your files. Of course, before you start, you should contact your application vendor to verify that they will support you if you have the newest files! If they are not using the same database engine, they may have problems reading the data if you ever needed to send them the files for help.

There are three primary tools for rebuilding files, the Rebuild Utility (a GUI utility shipping with all products above Pervasive.SQL V7, BREBUILD (shipping with NetWare and older DOS engines), and RBLDCLI (a utility shipping new with Pervasive.SQL V8.5). The instructions for running all three can be found in the product manuals for your database engine. Users of Pervasive.SQL 2000i and above can look in the Advanced Operations Guide under the chapter entitled *Converting Pervasive.SQL Data* or *Rebuilding Data Files*.

Using the Rebuild Utility

The GUI-based Rebuild Utility (also known as the Rebuild Wizard in Pervasive.SQL V8.5 and newer) is the easiest way to rebuild files if you are not familiar with command-line applications, because it uses the standard Windows graphical user interface to select files and options. Using this program is pretty simple:

1. First, select Options/Select Files to open the dialog box to select files. Click the *Add* button to add files to the list to be processed.
2. Then, select Options/Settings to open the Settings dialog box. Here, you can set the following options:
    - **Output Directory**: Leave this blank to update the files in place. Set this directory if you want to rebuild them into a new location.
    - **Status File Path**: This is the log file for the rebuild. Leave the default for best results.
    - **Continue on Error**: If doing one file, set to *No*. Otherwise, set this to *Yes* to have the system work on the next file if an error occurs. Be sure to check the log file after a run to make sure that all files were converted properly.
    - **Save Settings On Exit**: Saves these settings for the next time you run the Rebuild program.
    - **Key Number**: Leave to *NONE* for the best performance. *Some* applications' data files MAY require using a specific key. Check with your vendor for details.
    - **Page Size**: Leave at *Existing* if you trust the original application developers, or change to 4096 for the best performance in most cases.
    - **System Data**: Check the *Data and Key* box if you are planning to use DataExchange, Pervasive's replication solution.
3. Make sure that all users are out of the database files and that all files are closed in the Pervasive Monitor. If in doubt, reboot the database server!
4. Select Run/Start Rebuild to start the process. Each selected database file will be rebuilt with the settings you specified.
5. Finally, check the rebuild log file for any errors. If there were no problems, allow users back into the application and enjoy the performance gains!

Using BREBUILD

The NetWare program known as BREBUILD.NLM is a command-line-based application that runs on NetWare database servers. The base command is LOAD BREBUILD <filespec>, and you will add the following options as needed. When the options are the same as those listed above, little documentation is available in the program. Database versions 8.5 and above have a new file, called RBLDCLI, that supports many of the same options.

- -b<directory>: **Output Directory**
- -c: **Continue on Error**
- -d: Convert repeating-duplicatable keys to linked-duplicatable keys
- -m0: Clone and copy the files without dropping indices. This may be required if you have a very old data file with some internal corruption, or it may be required by some data files for some applications. You should ALWAYS use the -M0 switch for rebuilding VIEW.DDF.

- -m2: The default value, this option drops the indices before importing the data. This provides a substantial performance gain, and should always be used whenever possible.
- -p####: **Page Size**: If ommitted, the existing page size is used in the new file. If the -P is specified by itself, the optimal page size is selected. If the page size is present, then the specified page size will be used.
- -k##: **Key Number**
- -sD: Rebuilds the file and creates the System Data at the same time.
- -sK: Rebuilds the file to include the system data AND key.
- -l<logfilepath>: **Status File Path**
- -f#: Specifies the file version to rebuild to in RBLDCLI. Specify –f7 to force the files into the 7.x format.

Note that the command line applications also take a "filespec" – an indication of the files to rebuild that can include wildcards. Remember that NetWare servers have no concept of a current directory, so you should specify the entire path from the volume, as in the following example:

```
LOAD BREBUILD -p -c VOL1:APPS\DATA\*.BTR
```

Be sure to check the SYS:SYSTEM\BREBUILD.LOG file when finished to verify that all files rebuilt successfully.

Optimizing the Rebuild Time

The file rebuilds **must** be done during system downtime, since all users must be out of the files for the rebuild to be successful. Large files on slower servers will take a long time to rebuild, as will huge files on even the fastest server. There is no way to tell how long a process is going to take in advance, as it varies based on MANY factors. Luckily, you do NOT need to rebuild all files at the same time, since the engine supports accessing mixed file formats freely. You can run some tests on backup copies of your files, but remember that the larger a file is, the longer the time will be, and it can be expoentially longer on some systems.

To optimize the system for rebuilding, you want to split your system memory about 50/50 between the database cache and free memory. This means that on Pervasive.SQL 2000i or lower, your database cache should be set to a reasonable limit, such that there is sufficient free memory left over afterwards. On Pervasive.SQL V8 and above, you should *also* disable the Level 2 (L2) cache by setting the Maximum Memory Size down to 0%. Remember to change any settings back when you are done, and that you must restart the engine to allow the change to take effect.