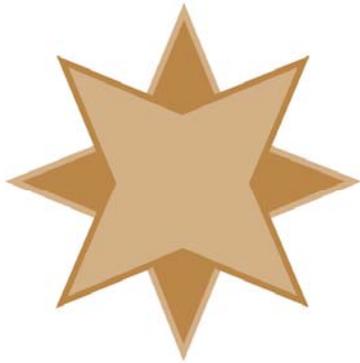


# Resolving Status 46 Error Messages

A White Paper From



**GOLDSTAR  
SOFTWARE**

*www.GoldstarSoftware.com*

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

# Resolving Status 46 Error Messages

Last Updated: 01/11/2007

Status 46's are returned by the database engine when it is trying to write back to a data file, and the file is not opened with write access. There are many different reasons for this status code, ranging from configuration issues, access rights, and other similar issues. Let's look at each cause and solution in turn.

## **1. The file is flagged ReadOnly.**

If a database file were to get flagged ReadOnly, then any subsequent access to that file would be prevented from writing to it. Files can be flagged ReadOnly by the operating system if they are copied from a ReadOnly device, such as a CDROM drive.

**Confirmation:** Check the OS rights to the file in question.

**Resolution:** Remove the ReadOnly flag from the file to repair this problem.

## **2. The file is stored in the 5.x data file format.**

Starting with Pervasive PSQL v9, the database engine is able to READ a Btrieve 5.x data file format, but cannot write to it. If you are running Pervasive PSQL v9 or newer and you are getting Status 46 messages, you should verify that all files are in 6.x or newer format.

**Confirmation:** Use either the **BUTIL -STAT** command line tool, or use the Maintenance tool's *Create Stat Report* option to view the file version information.

**Resolution:** If the database files are found to be in the 5.x version, rebuild them to 6.x or newer to allow writing back to the files.

## **3. The user has insufficient rights to the data directory.**

Each user's account must have WRITE rights at the OS layer to the database directory. Pervasive security is simplified by echoing the OS security -- if the user has rights to the file at the OS level, the user is granted database access rights as well. This simple security mode has worked for many years.

**Confirmation:** You can validate user rights by logging onto the operating system as the Administrator user and then trying the database write again. If it works, the problem is a user-specific rights issue. You can also test this by logging in as the restricted-access user, then creating or editing a text file in the same data directory. If you get an error, then the access rights need to be changed.

**Resolution:** Use your network rights tool to apply sufficient rights to the directory for the users who need such access. Required minimum rights may vary by application, so check with your vendor to be sure, or try granting ALL rights as a test.

#### **4. The engine has insufficient rights to the data directory.**

NetWare server engines will NEVER experience a problem with this, as the NLM on the server has full rights to all volumes by default. On Windows servers, however, the engine is logged onto the server (by default) with the SYSTEM account. (This can be changed in the Services applet.) On Win2003 servers, the SYSTEM account has full rights to the entire hard disk -- no problems. On WinNT and Win2000, however, the SYSTEM user is a member of the Everyone Group, and the Everyone Group is assigned full rights to the hard disk. When the SysAdmin limits security rights by removing the Everyone Group, the SYSTEM user is now denied access as well, and the database starts reporting Status 46's or 94's.

**Confirmation:** Right-Click on the database volume or directory, and select the Security tab. Look for the user SYSTEM on the top section, and see what rights have been granted on the bottom section.

**Resolution:** If necessary, add the user SYSTEM (this is an internal user defined by the OS) to the access control list and grant Full Control access rights to the directory. You can also change the "Log On As" account in the Services applet, although this may cause problems if the password or rights for this account were to ever change.

#### **5. Multiple engines may be active.**

Older database engines, including Btrieve 5.x and 6.x for DOS, Btrieve 6.15 for Windows 16-bit, and Btrieve 6.15 for Windows 32-bit, will implement a "Multi-Engine File Sharing" mode when accessing files, which allows multiple engines to access the data files simultaneously. This was great when all users had these older engines, but it does NOT work with modern engines. Every newer database engine (Btrieve 6.x server engines and all engines for PSQL7 and above) wants exclusive access to the database files that they will be accessing.

You can easily see the MEFS access on 6.15 engines by looking for files in the database directory with the .LCK extension with 0 bytes. These are Btrieve 6.15 Workstation Engine lock files. If you have them (and they are current, or if they come back after deleting them), then you may have multiple engines accessing the files. It is possible to configure Btrieve 6.x engines with Single-Engine File Sharing, but this is NOT the default configuration. In short, if you have one older Btrieve 6.x workstation engine (which could be residing on ANY workstation on the network) and a newer or server engine tries to access the file at the same time, then the newer/server database engine will be granted read rights, but no write rights. This works fine until a user tries to write to the file -- Status 46. Of course, Btrieve was a VERY common database engine, and oftentimes the engine gets left on a workstation and used without warning.

**Confirmation:** Use the Pervasive Monitor to view the Active Files list and look at the file flags on the left of the screen. If the flag is set to READONLY, then the file has likely been opened by someone at the OS level, using an older workstation engine. Use your OS tools (Novell Monitor, Windows Server Manager) to see what other users or process has the file open. If you see a USER with the file open, then that user has the Btrieve workstation engine running with access to the file, and this engine must be

eliminated. If a SERVER process has the file open, then see what process it is, and clear it out.

**Resolution:** Eliminate 6.15 Workstation Engine users first. Then, get all server-engine users out of the file from the application side. (The Pervasive Monitor will show you who has it open if you highlight the file in question in the Active Files screen.) Once all users are out and the file closes, have one user get back in, and check the flag in the Monitor again. It should now be opened as ReadWrite (or Normal). If you have a hard time getting users out, you can always kick them out with Monitor, restart the engine (a bit more drastic) or restart the server (really drastic).

**Prevention:** Check through all workstations and all server drives for the files BTRIEVE.EXE, W32MKDE.EXE, and WBTR32.EXE, and remove them if they are no longer needed. You can also lock out older database engines permanently by rebuilding the files from the older 6.x format to the 7.x or newer format. When you do this, the older engines will be unable to read the files (generating Status 2's or Status 30's), and your problems are gone.

## ***6. Other applications may be conflicting with the file.***

It is also possible that other software may be active -- usually a backup package or an anti-virus application. This is the trickiest one, since it can be extremely variable as to when it occurs and which file it affects.

When modern backup applications open a file for backup, they open the file in "DENY\_WRITE" mode. This prevents anyone from changing the file's data during the backup process. For most environments, the backup runs at off-hours, and the systems do not conflict. However, it is possible that the backup is active on a file at the SAME TIME that a user starts the application.

If this occurs, the database attempts to get write access to the file, and is denied by the OS. This was never anticipated in 1982 when the Btrieve API was designed, and the engine then decides to try opening the file in ReadOnly mode instead, since ANY access is better than no access. (This also provides compatibility with moving static data to CDROM.) This access is granted, even with the backup app running, and the engine continues normally. However, when ANY user attempts to write back to this file, the Status 46 results, since the file is open in ReadOnly mode now. Unfortunately, this error can long-survive the backup package, since the FILE is only opened ONCE by the database, and all other users get a HANDLE to the already-opened file, inheriting the ReadOnly attribute. This continues until the file is closed by the backup app **\*AND\*** is also closed by ALL users from the database app. Most sites get frustrated and simply reboot the server, which also closes the file, of course. The problem can be seen by antivirus applications that scan ALL files as well, because when the last user closes the database file, the AV solution will pick the file to scan. If another user opens the file while the scan is progressing, the engine can be denied full access rights as well.

**Confirmation:** As with the item above, use the Pervasive Monitor to view the Active Files list and look at the file flags on the left of the screen. If the flag is set to

READONLY, then the file was likely opened by another process at the OS level when the database engine went to open it. Of course, since the other process (Backup or AV) is long gone, when you use your OS tools (Novell Monitor, Windows Server Manager) to see what other users or process has the file open, you'll see no users with the file open, so there is no *obvious* conflict. This makes it impossible to confirm, but you now have suspicions.

Note: We often find the cause based on the time of the problems. Early morning problems are usually from caused by the backup system running late and users getting in while it is still running, and afternoon problems are more commonly from an AV system. (Check with your backup schedule to verify for sure.)

**Resolution:** Since no other processes are in the file any more, you need only to get all users out of the file from the application side. (The Pervasive Monitor will show you who has it open if you highlight the file in question in the Active Files screen.) Once all users are out and the file closes, have one user get back in, and check the flag in the Monitor again. It should now be opened as ReadWrite (or Normal). If you have a hard time getting users out, you can always kick them out with Monitor, restart the engine (a bit more drastic) or restart the server (really drastic).

**Prevention:** I bet you already know this one: Disable your anti-virus application or your backup application and see if the problem continues. You do NOT need to scan Btrieve data files for viruses, as they are not susceptible to virus code, and this only hurts performance and causes problems. If you can't exclude the entire database directory structure from the AV scan, try to disable the Btrieve file extensions from the scan.

If you suspect the backup process, you should use Pervasive's Continuous Operations mode to get ANY backup of a file while the database files are active. Failure to handle the backups properly can result in corrupt backups and continuing Status 46 problems. You can get more information on using ContOps mode in the Goldstar Software white paper on proper backups, available on our web site at [www.goldstarsoftware.com/press.asp](http://www.goldstarsoftware.com/press.asp).

If you still can't get it to work, contact Goldstar Software and let us work with you to help!