1 to 199

MicroKernel Database Engine Status Codes

This section describes status codes that the MicroKernel returns.

1: The operation parameter is invalid

The specified operation does not exist or is not valid.

You may receive this error if you are running a general-release version of the V8 client software against a pre-release version of the V8 database engine. If this is the case, you must uninstall your database engine and install the general release version.

2: The application encountered an I/O error

This status code typically indicates a corrupt file, an error while reading from or writing to the disk. One of the following has occurred:

- The file is damaged, and you must recover it. Refer to the *Advanced Operations Guide* for more information on recovering files.
- For pre-v6.0 data files, there is a large pre-image file inside a transaction, and there is not enough disk space for a write to the pre-image file.
- For pre-v6.0 data files, there is one pre-image file for multiple data files. For example, if you name the data files CUSTOMER.ONE and CUSTOMER.TWO, both files have pre-image files named CUSTOMER.PRE.
- For pre-v6.0 data files that are larger than 768 MB, there is a conflict among locking mechanisms. The file has not been corrupted. Your application can retry the operation until the conflict is resolved (when the competing application releases the lock your application requires).
- A pre-v6.0 Btrieve engine attempted to open a v6.*x* or later MicroKernel file.
- With Btrieve for Windows NT Server Edition v6.15.445, 32 bit Windows application may return Status 2 or “MKDE Terminated with Service Specific Error 0” after running an application for an extended period of time.

Note: Please see the Pervasive PSQL Knowledge Base for new and updated articles on troubleshooting this status code. You can access the Knowledge Base at the Pervasive Software website.

3: The file is not open

The operation cannot execute because the file is not open. The application must perform a successful Open operation before the MicroKernel can process any other operations. The MicroKernel also returns this status code if the application passed an invalid position block for the file, or if the application passed a position block with a client ID other than the client ID used to open the file.

Note: Please see our Pervasive PSQL Knowledge Base for new and updated articles on troubleshooting this status code. You can access the Knowledge Base at the Pervasive Software website.

4: The application cannot find the key value

The MicroKernel cannot find the specified key value in the index path.

- When you receive this status code on an Update or Delete operation, it usually means that the file is damaged, and you must recreate it. Occasionally, a corrupt key can cause this status code. Drop the key, then add it again.
- The MicroKernel returns this status code when an application performs a Get Equal operation to search on field type char. It is caused by a mismatch of two fields at the char level. To resolve, fill the KeyBuffer with the same fill char type as the field.

5: The record has a key field containing a duplicate key value

The MicroKernel cannot add or update a record because the record has a key field that contains a duplicate key value for an index that does not allow duplicate values. The MicroKernel also returns this status code when it cannot create an index that does not allow duplicate key values because a duplicate key value already exists.

6: The key number parameter is invalid

The value stored in the key number parameter is not valid for the file being accessed. The key number must correspond to one of the keys defined for the file. Valid key numbers are 0 through 118.

7: The key number has changed

The key number parameter changed before a Get Next, Get Next Extended, Get Previous, or Get Previous Extended operation. The operation requires the same key number parameter as the previous operation, because the MicroKernel uses positioning information relative to the previous key number.

In a related situation, the MicroKernel returns this status code when an application performs a Delete or Update operation immediately following a Get operation. If the application changes the value of the key number in the Delete or Update operation (from the value used with the preceding Get operation), the MicroKernel deletes or updates the record as requested and does *not* return this status code, at least not at this point. However, the MicroKernel *does* return this status code on the very first Get Next, Get Next Extended, Get Previous, or Get Previous Extended operation performed after the deletion or update, even if that Get operation uses the same key value the application passed to the Delete or Update operation.

If you need to change key numbers between consecutive Get Next, Get Next Extended, Get Previous, or Get Previous Extended operations (or in Delete or Update operations as described in the preceding paragraph), use a Get Position operation followed by a Get Direct/Record operation to reestablish positioning for the new index path.

8: The current positioning is invalid

You must establish the current position in order to update or delete a record. Perform a Get or Step operation to establish the current position. The MicroKernel also returns this status code if the application passed an invalid position block for the file.

9: The operation encountered the end-of-file

The MicroKernel returns this status code in the following situations:

- The operation encountered an end-of-file boundary or tried to read past a file boundary (end-of-file or start-of-file).
- In a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, the number of records satisfying the filtering condition is less than the number of specified records to be returned, and the reject count or filter limit has not been reached.
- When reading a file in ascending order according to an index path, the MicroKernel has already returned the last record in that index path. When reading a file in descending order according to an index path, the MicroKernel has already returned the first record in the index path.
- When using the Get By Percentage operation, either the value supplied for the percentage is too high—it exceeds 10,000 decimal (0x2710)—or the file contains no records.
- When using the Get operation using ActiveX Data Control, this error will occur only after the application is compiled and deployed. This error will not occur at design time during development. The error results from a missing MSDADC.DLL on the deployment machine. Make sure to include this file (MSDADC.DLL) in your installation script. This file is a Microsoft MDAC (Microsoft Data Access Component) file.

Note: Please see our Pervasive PSQL Knowledge Base for new and updated articles on Btrieve ActiveX Controls. You can access the Knowledge Base at the Pervasive Software website.

10: The key field is not modifiable

During an Update operation, the application attempted to modify a key field that cannot be modified by definition.

11: The specified filename is invalid

The MicroKernel returns this status code in the following situations:

- The specified filename does not conform to file naming conventions or the pathname is invalid. Make sure the filename or pathname is valid for the environment.
- If operating in the client/server environment:
  - The application attempted to open a file that has .^^^ as its extension. This extension is reserved for the MicroKernel to use during continuous operation. (Only server engines can use continuous operation.)
  - The data buffer for a Begin or End continuous operation is not set up correctly.
  - You attempted to load a remote file when your client configuration settings for **Local MicroKernel Engine** and **Use Remote MicroKernel Engine** are incorrectly set to **On** and **Off**, respectively. To resolve this condition, at the client workstation, open Pervasive PSQL Control Center (see Starting PCC on Windows in *Pervasive PSQL User's Guide*). Under the Access properties category for **MicroKernel Router**, set **Use Remote MicroKernel Engine** to **On** (click the option). See To access configuration settings in PCC for a local client in *Advanced Operations Guide*.
  - You attempted to open a local file with a Workgroup engine that isn't the designated Gateway engine for the file. Go to the directory where the file you attempted to open resides. Check to see if the `~pvsw~.loc` in that directory is flagged read only. If it is, change it to read-write.
- If you are using the dynamic locator file with your Workgroup engine:
  - Make sure the name of the second-level locator file specified in your first-level locator file does not have the same name as an existing directory. Also, make sure they are all on the same drive.
  - Make sure the second-level locator file specified in your first-level locator file can be accessed by the engine.
  - Make sure all the Workgroup engines sharing the dynamic locator feature have the exact same drive mapping to the server location where the data files reside.
- If you are accessing files on a DOS client:
  - A NET START FULL for the Windows for Workgroups workstation was not used when it was booted into DOS. Use a NET START FULL to get a full redirector in the DOS client mode. BREQNT.EXE requires a full redirection. At the DOS prompt type `net ver` and press Enter. Here is the list of required components for a DOS workstation to connect to a Windows server:
    LSL.COM LAN Card Driver
    IPXODI.COM
    IFSHLP.SYS
    NET START FULL

    These can be loaded high, using emm386. NET START FULL will load in upper memory.
- You attempted to open a file with a long filename on NSS volumes. The MicroKernel queries the volumes using OS calls and then loads the appropriate drivers for the associated name spaces it find for the volumes. In this case, the MicroKernel was being loaded before mounting the volumes so it did not find the requirement for long filename support by the NSS volumes.
  - Issue the MGRstart or Bstart command after loading the volumes. An example would be:
    LOAD NSS
    MOUNT ALL
    SYS:ETC\INITSYS.NCF
    MGRSTART or BSTART
- Embedded spaces configuration setting is "on." "On" is the default starting with Pervasive PSQL v9. Change the setting to "off" if your applications do not allow embedded spaces in file names.
- If you are working in the Microsoft Terminal Server environment:
  - Approximately 5 users can work in a Windows application on 2 different Terminal Servers that are connected to a primary Windows server. If you have attempted to run on top of this limit, you will receive status codes 11 and 35. For the recovery solution for this instance, refer to Microsoft Knowledgebase Article 190162, "Terminal Server and the 2048 Open File Limitation."

12: The MicroKernel cannot find the specified file

- Check that the file exists and that you specified the correct file.
- Check the key buffer parameter to make sure the pathname is terminated with a blank or a binary zero.

- When accessing a file on a server, ensure that you have FILE SCAN rights to the directory in which the file resides. Occasionally, a corrupt key can cause this status code. Drop the key, then add it again.
- This error code may also be returned when the file DBNames.CFG has been removed (for example, by the Pervasive PSQL cleanup utility) and old Data Source Names (DSNs) which reference that file are not removed from the ODBC configuration.
- You attempted to open a local file with a Workgroup engine that isn't the designated Gateway engine for the file. Go to the directory where the file you attempted to open resides. Check to see if the `~pvsw~.loc` in that directory is flagged read only. If it is, change it to read-write.
- You may have a Status 12 returned and see X$FILE.DDF for the file name in your DDFs.
  - If the file.ddf is examined with a Btrieve utility, the location name for the dictionary files will be x$file.ddf, x$field.ddf, x$index.ddf, instead of the standard file.ddf, field.ddf, index.ddf. An unsupported DDF Creation utility called DDL Services (DDLSVCS.DLL) created the DDFs. DDL Services has a known bug that causes the system table to be populated with incorrect data. .
- You may get Status 12 when a file with a filename or path with embedded spaces is opened on certain Windows 32-bit platforms. Btrieve data files can be accessed using long names but long names cannot be used for other files.
- To Enable Embedded Spaces in Pervasive.SQL 2000i or later:

1. Start Pervasive PSQL Control Center (see Starting PCC on Windows in *Pervasive PSQL User's Guide*).
2. Expand the nodes for **Pervasive PSQL** (click the plus (+) sign).
3. Expand the **Local Client** node.
4. Right-click **MicroKernel Router** then click **Properties**. Login if prompted.
5. Click **Application Characteristics** in the Properties tree.
6. Click **Embedded Spaces (a check mark indicates that the option is enabled)**.

13: The MicroKernel could not open the extension file for an extended file

The MicroKernel could not open the extension file for an extended file that the application tried to open. An extended file can consist of a base file and up to 31 extension files. Extension files must remain in the same volume and directory as their base files. The MicroKernel returns this status code if you delete, move, or rename the extension files.

14: The MicroKernel cannot create or open the pre-image file

The MicroKernel uses pre-image files only for pre-v6.0 data files.

The MicroKernel returns this status code in one of the following situations:

- The MicroKernel cannot create a new pre-image file because the disk directory is full. The MicroKernel must be able to create a pre-image file.
- The MicroKernel cannot open the pre-image file to restore file integrity. If the pre-image file is erased or damaged, the MicroKernel cannot restore the file integrity. Refer to *Advanced Operations Guide* for more information about recovering damaged files.
- The workstation MicroKernel cannot assign a handle to the pre-image file because the MicroKernel was not started by a user with access rights to the pre-image file.
- The file structure of a pre-image file created by this MicroKernel is different from the file structure of a pre-image file created by a v5.*x* MicroKernel. If you have an extraneous .PRE file in v5.*x* format, the MicroKernel returns this status code when you try to open the data file to which the .PRE file belongs.

15: The application encountered an I/O error during pre-imaging

The MicroKernel uses pre-image files only for pre-v6.0 data files.

- The pre-image file is damaged and the integrity of the data file cannot be ensured. Refer to *Advanced Operations Guide* for more information about recovering damaged files.
- The disk is full. Erase any unnecessary files.

16: The application encountered an expansion error

This status code is obsolete in MicroKernel versions 6.0 and later.

The MicroKernel encountered an error while writing the directory structure to disk prior to creating the expanded file partition. Either the MicroKernel cannot close the file, or a new page was added to the file and the MicroKernel cannot close and reopen the file to update the directory structure. Check for a disk hardware failure.

18: The disk is full

The MicroKernel can return this status code in the following situations:

- The disk is full and the MicroKernel cannot expand the file to accommodate additional records. Erase any unnecessary files.
- There is not enough space to append a new page to the data file.
- The pre-image file is out of disk space. If your files are in pre-v6.0 format and you are in a transaction, the pre-image file size increases for the duration of the transaction. If you receive this status code, either reduce the number of operations in the transaction, or obtain more disk space.
- In some environments, you can restrict the amount of disk space available to each user. This status code indicates that the application attempted to expand a data file beyond the amount of disk space allocated to the file owner.
- You tried to read or modify a file which was not closed properly after a disk full error. Make sure that every application using the file at the time of the disk full error closed the file successfully.
- If a client connected to a Pervasive PSQL server encounters this status code, other clients performing read-only operations from the same disk may also receive a non-zero status.

19: The application encountered an unrecoverable error

To ensure file integrity, recover the file as described in *Pervasive PSQL User's Guide*.

20: The MicroKernel or Btrieve Requester is inactive

Note: For an expanded list of possible recovery solutions, consult the Pervasive PSQL Knowledge Base at the Pervasive Software website.

- If you are running an application in a client/server environment:
  - Make sure the Btrieve requester is loaded.
  - Verify that the IPX/SPX or TCP/IP protocol is properly installed at the client machine and that no two machines on the network have the same Internal Network Number.
  - Make sure at least one of the client configuration options, **Access ▸ Use Local MicroKernel Engine** or **Access ▸ Use Remote MicroKernel Engine** is enabled. If your environment includes both a server engine and Workgroup engines, you should have both settings enabled.
- If you are running an application in a workstation/workgroup environment, make sure the MicroKernel is loaded.
- If you are running an application in a client/server environment and also need to access files located on a local drive:
  - Make sure the Btrieve Requester is loaded.
  - Make sure both of the client configuration options, **Access ▸ Use Local MicroKernel Engine** or **Access ▸ Use Remote MicroKernel Engine** are enabled.
  - Make sure a local MicroKernel is available and loaded.
- If you are operating in a DOS server environment:

  Pervasive PSQL v11 SP3 installs BTRBOX95 by default. As long as this is installed no other requester can be used.

  - If you want to use BTRBOX95:

    a. and run the setup utility. This loads the appropriate file for clients running Windows 32-bit operating system.

    b. After installation, reboot the client.

  - For Windows 32-bit platform users: open a command prompt and run a DOS Btrieve application.
- If you are operating in a Windows server environment:
  - Make sure the MicroKernel is started before generating any requests.
  - Make sure the Windows DLLs are in your path.
  - Make sure the appropriate communications modules are loaded at the server.

21: The key buffer parameter is too short

The key buffer parameter is not long enough to accommodate the key field for the requested index path. Verify that the length of the key buffer equals the defined length of the key specified in the key number parameter.

22: The data buffer parameter is too short

- The data buffer parameter specified by the application was not large enough to accommodate either the minimum length of the record for an Insert or Update operation, or the entire record length for a Get or Step operation. Also, the data buffer may not be large enough to accommodate the length of data required for operations such as Create, Create Index, Stat, Get By Percentage, Find Percentage, or Version.
  - For Get or Step operations, the MicroKernel returns as much data as it can and this status code, indicating that it cannot return the entire record.
  - For an Insert operation, the MicroKernel does not insert the record if the data buffer is shorter than the fixed-length portion of the record.
  - For an Update operation, if the data buffer is too short to contain the fixed-length portion of a record, the MicroKernel does not update the record.
  - For the Create, Stat, and Create Index operations, the data buffer is not long enough to contain all the file specifications, the key specifications, and (if specified) the Alternate Collating Sequence (ACS) definition.
  - For the Get by Percentage or Find Percentage operation, the data buffer length is less than 4 bytes.
  - For the Version operation, the data buffer length is less than 5 bytes.
  - The data buffer parameter is too short when access table with more than 60 field using ActiveX. Download the latest ActiveX controls from the Pervasive Software web site (http://www.pervasivedb.com).
- A corrupt file may be indicated if the file allows variable-length records and you receive this status code on a Get or Step operation. In such a corrupt file, you can receive Status Code 54 when you use Get or Step operations to read other records. Recover the file according to the instructions in *Pervasive PSQL User's Guide*.

23: The position block parameter is not 128 bytes long

This status code is obsolete in Btrieve language interface versions 6.0 and later.

The position block parameter must be exactly 128 bytes long.

24: The page size or data buffer size is invalid

The MicroKernel returns this status code in one of the following situations:

- The page size you specified when creating a file is invalid. The page size must be a multiple of 512 bytes and cannot exceed 4096 bytes (up to 8.x file format) or 8192 bytes (9.0 file format) or 16384 (9.5 file format).
- During a Create operation, the page size is the first file specification the MicroKernel checks. If you receive this status code at this point, it can indicate an invalid data buffer parameter.

25: The application cannot create the specified file

The MicroKernel returns this status code in one of the following situations:

- If an application attempted to create a data file, the disk directory or the disk itself may be full.
- If an application tried to create a file over an existing file, the existing file is open or the operating system will not allow the operation for another reason.

- In an attempt to create a Btrieve file over existing Btrieve file, this status will be returned. The keybuffer on the Btrieve create operation API (opcode 14) is set properly to create a file over an existing file.
  - This problem may be caused by Antivirus software (such as Innoculan)

Note: This happens when the Operating system returns an unusual status code to the engine. Normally, the engine expects either a success or the file already exists. In one situation, the error code was being returned because the file handle that the engine was using was not functional; however, the OS call that the engine makes is supposed to return a file handle.

One solution is to disable the Antivirus software. Contact the third party vendor for additional information on configuring the Antivirus software to eliminate scanning specific data files.

26: The number of keys specified is invalid

The number of keys specified for the page size is invalid. Note that the maximum number of *keys* is 119 for all file versions.

The number of *key segments* can vary but must be within the limits shown by the following table.

Page Size (bytes)	Maximum Key Segments by File Version		
	8.x and prior	**9.0**	**9.5**
512	8	8	rounded up[2]
1,024	23	23	97
1,536	24	24	rounded up[2]
2,048	54	54	97
2,560	54	54	rounded up[2]
3,072	54	54	rounded up[2]
3,584	54	54	rounded up[2]
4,096	119	119	119 or 204[3]
8,192	n/a[1]	119	119 or 420[3]
16,384	n/a[1]	n/a[1]	119 or 420[3]

[1] ”n/a” stands for “not applicable”

[2] ”rounded up” means that the page size is rounded up to the next size supported by the file version. For example, 512 is rounded up to 1,024, 2,560 is rounded up to 4,096, and so forth.

[3] The maximum number of index segments that can be used with the relational interface is 119. For the transactional

interface, the maximum number is 204 for a page size of 4,096, and 420 for page sizes 8,192 and 16,384.

Conditions For Which Status Code 26 Is Returned

The following conditions apply to the Btrieve Create API operation. See Create (14) in *Btrieve API Guide*, which is part of the Pervasive PSQL Software Development Kit (SDK).

- Number of keys or key segments exceeds the permissible limit for the given file format and page size. The maximum number of keys is 119 for all file formats and page sizes. The number of segments is listed in the table above.
- A key-only file is being created and more than one key is supplied in the data buffer.
- A key-only file is being created, the “Reserve Duplication Pointers” flag is “on,” and the “Number of Duplicate Pointers to Reserve” field is equal to or greater than 1.
- The number of segments specified in the data buffer exceeds the limit for maximum segments.
- The “Number of Duplicate Pointers to Reserve” field exceeds the limit for maximum number of keys.

The following condition applies to the Btrieve Create Index API operation. See Create Index (31) in *Btrieve API Guide*, which is part of the Pervasive PSQL Software Development Kit (SDK).

- The number of segments specified in the data buffer exceeds the limit for maximum segments.

Nullable Columns

Note that nullable columns must also be considered. For example, in a data files with 4096 byte page size you are limited to 119 index segments per file. Because each indexed nullable column with true null support requires an index consisting of 2 segments, you cannot have more than 59 indexed nullable columns in a table (or indexed nullable true null fields in a Btrieve file). This limit is smaller for smaller page sizes.

Any file created with file version set to 7.x or later, and TRUENULLCREATE set to the default value of On, has true null support. Files created using an earlier file format, or with Pervasive.SQL 7, or with TRUENULLCREATE set to Off, do not have true null support and do not have this limitation.

27: The key position is invalid

The specified key field position is less than 1 or exceeds the defined record length for the file. Either the key position is greater than the record length or the key position plus the key length exceeds the record length.

28: The record length is invalid

The physical record length, which is the logical record length specified when creating the file plus any additional overhead for duplicate keys, reserved duplicate pointers, variable record pointers, and blank truncation information must be less than or equal to the page size minus 10 bytes (8 bytes if creating files in version 5 file format), and must be greater than or equal to 4 bytes.

For key-only files, the maximum record length is 253 bytes (255 bytes if creating files in version 5 file format).

For more information about calculating the physical record length, see *Pervasive PSQL Programmer's Guide*.

29: The key length is invalid

The MicroKernel returns this status code for the following situations pertaining to the Btrieve Create or Create Index API operations. See Create (14) and Create Index (31), both in *Btrieve API Guide*, which is part of the Pervasive PSQL Software Development Kit (SDK).

- The length of the entire key (all segments) exceeds 255.
- The length of the entire key (all segments) is so large that less than 4 keys fit on a key page. For example, a 200 byte key on a 512 byte page.
- A key segment length is 0 or greater than 255 bytes. The length of a key segment must agree with its key type if the key type implies a length (for example, an integer key must have a length evenly divisible by two). Each key page in the file must be large enough to hold at least four keys. If the page size is too small to accommodate four occurrences of the specified key length (plus overhead), you must increase the file page size or decrease the key length.

An additional byte of storage space is needed for the null indicator for the column. This error occurs through a SQL CREATE INDEX statement, or through the creation of a SQL PRIMARY KEY or FOREIGN KEY, if the index, or key, references a null CHAR column of 255 characters (or VARCHAR of 254). This additional byte causes the actual length of the index to be one byte longer, or 256 bytes. To resolve the error, reduce the size of the column or create the column as NOT NULL and try again. For a foreign key, if you decrease the size of the column, you must decrease both the referencing column and the referenced column.

- A key segment length is greater than 4 and the key segment data type is AUTOINCREMENT.
- A key segment data type is DATE, TIME, BFLOAT, or AUTOINCREMENT and the segment length is an odd number.
- A key segment data type is NUMERICSTS and the segment length is less than 2.
- A key segment data type is CURRENCY or TIMESTAMP and the segment length is not 8.
- A key segment data type is DATE, TIME, BFLOAT, or AUTOINCREMENT and the segment length is an odd number.
- A key segment data type is NULL INDICATOR SEGMENT and the segment length is not 1.
- A key segment data type is GUID and the segment length is not 16.
- A key segment data type is BINARY (“Use Old Style Binary Data Type” flag is “on”) and the segment length is odd.

30: The file specified is not a MicroKernel file

This status code is returned in one of the following situations:

- The MicroKernel did not create the file, or a pre-v3.*x* MicroKernel created it.
- While using an earlier version of Btrieve, you opened a file created by a later version that has a format incompatible with the earlier version.
- The first page of the file may be damaged. Use a backup copy of your data file. If you receive this status code and you suspect that the header page of the source file is damaged, recover the file as described in *Advanced Operations Guide*.
- You have attempted to access a valid Btrieve file. This status code is returned when old engines access newer file formats. A likely scenario is that data created by a new server engine is later used by an earlier Workgroup engine. Status 30 can be reported if the file format is newer than the MicroKernel engine attempting to open it. Particularly, accessing a 7.x file with a 6.x engine causes this error.

  NOTE: Previously, accessing a 6.x file with a 5.x engine returned Status 2: "the application encountered an I/O error".

31: The file is already extended

This status code is obsolete in MicroKernel versions 6.0 and later.

An application tried to extend a file that had already been extended; you can only extend a file once.

32: The file cannot be extended

The MicroKernel must create an extension file to accommodate a file which is growing larger than the operating system file size limit. However the MicroKernel encounters an error from the operating system when it tries to create and open the new extension file. Possible causes for receiving this status code include the following: the directory is full, the disk is full, or the MicroKernel has not been granted sufficient rights by the operating system.

33: The MicroKernel cannot unload

In the DOS environment, The MicroKernel returns this status code for the following reasons:

- You attempted to unload the MicroKernel when you have loaded another terminate and stay resident (TSR) program after you loaded the MicroKernel. Unload the other TSR before unloading the MicroKernel.
- You attempted to unload the MicroKernel from a 32-bit application that uses the BSTUB interface with the DOS/4G extender.

34: The specified extension name is invalid

This status code is obsolete in MicroKernel versions 6.0 and later.

An application specified an invalid filename for the extended partition. Check the validity of the filename.

35: The application encountered a directory error

Either a Get Directory operation specified a drive that does not exist, or a Set Directory operation specified an invalid pathname. Check the validity of both the drive and the pathname.

37: Another transaction is active

The application issued a Begin Transaction (19 or 1019) operation while another transaction was active by the same user or task; the active transaction can be nested or non-nested. This status code often indicates a problem in nested transactions within your application.

38: The MicroKernel encountered a transaction control file I/O error

This status code is obsolete in MicroKernel versions 7.0 and later.

The MicroKernel tried to write to the transaction control file. Possible causes for receiving this status code are that the disk is full, the disk is write protected, the transaction control file (BTRIEVE.TRN) that is created when you load the MicroKernel has been deleted, or the transaction control file is flagged read-only or is corrupt.

39: A Begin Transaction operation must precede an End/Abort Transaction operation

The application issued an End Transaction (20),or Abort Transaction (21) operation without a corresponding Begin Transaction (19 or 1019) operation. Make sure that each End or Abort Transaction operation in your program is executed only after a successful Begin Transaction operation.

40: The file access request exceeds the maximum number of files allowed

This status code is obsolete in MicroKernel versions 6.0 and later.

The application tried to access more than the maximum number of files allowed within a transaction. You set the maximum number of different files that you can access during a logical transaction when you configure the MicroKernel.

41: The MicroKernel does not allow the attempted operation

The application tried to perform an operation that is not allowed under these operating conditions. For example:

- The application attempts to perform a Step operation on a key-only file.
- If using a server engine, the key number parameter of a continuous operation MicroKernel call is not valid.
- The MicroKernel prohibits certain operations during transactions because they have too great an effect on the file or on performance. These operations include Set Owner, Clear Owner, Create Index, and Drop Index.
- An application running on a 9.x or higher engine attempts to create a format file prior to 6.x (0600).

42: A file previously opened in Accelerated mode was not closed

This status code is obsolete in MicroKernel versions 6.0 and later.

The MicroKernel returns this status code for the following reasons:

- Either the application tried to open a v5.*x* data file that was previously accessed in Accelerated mode by a v5.*x* MicroKernel and never successfully closed, or the application tried to open a file for which a v6.0 or later MicroKernel encountered an unrecoverable error during a Set or Clear Owner operation. The file integrity cannot be ensured. Refer to *Advanced Operations Guide* for information about recovering damaged files.
- Your application tried to open a file in MicroKernel v5.*x* format using a v5.*x* MicroKernel; however, that same file was previously accessed by a v6.0 or later MicroKernel, which failed to close the file successfully and left a pre-image file on the disk. Version 5.*x* MicroKernels cannot read pre-image files created in v6.0 or later format.

43: The specified record address is invalid

The MicroKernel returns this status code for the following reasons:

- The record address specified for a Get Direct operation is invalid. Either the address is outside the file boundaries, or it is not on a record boundary within or on a data page, or the record at the specified address has been deleted. For a Get Direct operation, specify the 4-byte address obtained by a Get Position operation.
- If the records' file is in v5.*x* format, this status code can indicate a file access conflict. For example, task 1 has a file locked in an exclusive transaction. Task 2 is reading records from the same file and tries to update a record that the transaction inserted. If task 2 reads the record and then task 1 aborts the transaction, task 2 receives this status code when issuing the Update operation.
- For a Find Percentage operation that is seeking a percentage based on a record's physical location within the file, the specified record address is invalid.
- The file may be corrupt, and you must recover it. Refer to *Advanced Operations Guide* for information about recovering damaged files.

44: The specified key path is invalid

The application tried to use the Get Direct/Record operation to establish positioning on an index path for a key whose value is null in the corresponding record. The MicroKernel cannot establish positioning based on a null key value.

45: The specified key flags are invalid

The key flags specification on a Create operation is inconsistent. If a key has multiple segments, the duplicate, modifiable, and null attributes should be the same for each segment in the key. Also, you cannot use the null or manual key attributes in a key-only file. The MicroKernel also returns this status code if the application attempted to specify a different Alternate Collating Sequence (ACS) for two or more segments of a segmented key.

46: Access to the requested file is denied

The MicroKernel returns this status code for the following reasons:

- The application opened a file in Read-only mode and tried to perform a Write operation on that file.
- The application attempted to perform a Write operation on a file that is flagged read-only by the operating system.
- When the application opened the file, it did not correctly specify the owner name required for updates.
- (Workgroup engine only) If a Workgroup engine user or task opens a file that a client machine has opened using a server MicroKernel, the MicroKernel returns this status code if the Workgroup engine user attempts to write to the file.
- (9.x and higher engines only) The application attempted to perform a Write operation on a 5.x format file. When using a 9.x or higher engine, you cannot perform a write operation such as insert or delete on a 5.x format file.

47: The number of files opened exceeds the maximum allowed

This status code is obsolete in MicroKernel versions 6.0 and later.

Pre-v6.0 workstation MicroKernels return this status code when the number of files opened in Accelerated mode exceeded the number of buffers available in the MicroKernel cache. When a file is opened in Accelerated mode, the MicroKernel reserves one of its cache buffers for the file. It always reserves five empty buffers for index manipulation. Reconfigure Btrieve with both a smaller **/P** configuration option (to allocate more buffers) and a larger **/M** option (to increase the cache allocation).

48: The alternate collating sequence definition is invalid

The MicroKernel returns this status code for the following reasons:

- The first byte of an Alternate Collating Sequence (ACS) definition (the identification byte) does not contain the hexadecimal value *AC* (for user-defined ACSs), *AD* (for locale-specific ACSs), or *AE* (for international sorting rules support). Make sure that the first byte contains the appropriate value.

- You set the **Create File Version** option to v5.*x*, and you attempted to create a file that contains a key with a locale-specific ACS. Pre-v6.0 files do not support locale-specific ACSs.

49: The extended key type is invalid

The MicroKernel returns this status code for the following reasons:

- You tried to create a file or an index with an invalid extended key type.
- You tried to assign an Alternate Collating Sequence (ACS) to a BINARY key or key segment. You can assign an ACS only to a STRING, LSTRING, WSTRING, WZSTRING, or ZSTRING key type.
- You defined an index requiring an ACS, but no ACS definition exists either in the file or in the key definition passed in the data buffer.
- You attempted to create a key segment with both the Case Insensitivity and the ACS flags set, and the MicroKernel is configured to create files in v5.*x* format. This combination is invalid for v5.*x* files.
- You set the **Create File Version** value to v5.*x*, and you attempted to create a file with a NUMERICSA or NUMERICSTS key. Pre-v6.*x* files do not support these key types.

  OR

- You set the **Create File Version** value to v6.x, and you attempted to use one of the new Pervasive.SQL V7 data types, such as CURRENCY or TIMESTAMP. Pre-v7.*x* files do not support these key types. Increase the setting for this component.
- You set the Create File Version value to v9.0, and you attempted to use one of the new Pervasive PSQL v11 SP3 data types, such as GUID. File versions prior to Pervasive PSQL v11 SP3 do not support the GUID data type.

► To change the Create File Version setting:

1. Start Pervasive PSQL Control Center (see Starting PCC on Windows in *Pervasive PSQL User's Guide*).
2. Expand **Engines** and find the desired engine name.
3. Right-click on the engine name and click **Properties**.
4. Click **Compatibility**.
5. In the right hand frame, adjust the **Create File Version**.

50: The file owner is already set

The application tried to perform a Set Owner operation on a file that already has an owner. Use the Clear Owner operation to remove the previous owner before specifying a new one.

51: The owner name is invalid

The MicroKernel returns this status code for the following reasons:

- If the application received this status code from a Set Owner operation, the owner names specified in the key buffer and data buffer do not match.
- If this status code occurred during an Open operation or a DROP TABLE statement, the application attempted to open a file that has an owner name assigned to it. The application must specify the correct owner name in the data buffer. Ensure that the owner name is null-terminated in the data buffer and that the data buffer length is set long enough to include the owner name plus the null terminator.

52: An error occurred while writing to the cache

This status code is obsolete in MicroKernel versions 6.0 and later.

While trying to make a cache buffer available, the MicroKernel attempted to write data to a disk from a file that was previously opened in Accelerated mode. The operating system returned an I/O error during the write. This generally indicates a hardware memory problem. Unload and reload Btrieve before you continue.

53: The language interface version is invalid

An application tried to access a file containing variable-length records with a language interface from Btrieve v3.15 or earlier.

54: The variable-length portion of the record is corrupt

During a Get or Step operation, the MicroKernel could not read all or part of the variable-length portion of a record. The MicroKernel returns as much data as possible to the application. This status code usually indicates that one or more pages used to store variable-length records are corrupt. Check the data buffer length the MicroKernel returns to see how much of the record was returned. Recover the damaged file as described in *Pervasive PSQL User's Guide*.

55: The application specified an invalid attribute for an AUTOINCREMENT key

The data field indexed by an AUTOINCREMENT key can be part of a different segmented key only if the key number of the AUTOINCREMENT key is less than the key number of the new segmented key and the new data type flag referencing the field is not AUTOINCREMENT

56: An index is incomplete

An index can be damaged if a Create Index operation (31) or a Drop Index operation (32) is interrupted before it runs to completion. Perform a Drop Index operation to completely remove the damaged index from the file, then rebuild the index with the Create Index operation, if desired.

57: An expanded memory error occurred

This status code is obsolete in MicroKernel versions 6.0 and later.

Btrieve for DOS returns this status code if it receives an error from the Expanded Memory Manager. This status code usually means that the MicroKernel was unable to save or restore the memory mapping register context, indicating an incompatibility with another application that uses expanded memory.

58: The compression buffer length is too short

This status code is obsolete in Pervasive.SQL 2000i and later versions.

59: The specified file already exists

During a Create operation, the application specified -1 in the key number parameter and the name of an existing file in the key buffer parameter. To overwrite the existing file, remove the -1 from the key number parameter. To preserve the existing file, alter the filename specified in the key buffer parameter.

60: The specified reject count has been reached

The MicroKernel rejected the number of records specified by the reject count before a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation found the requested number of records that satisfy the filtering condition. Check the first two bytes returned in the data buffer for the number of records that were retrieved.

61: The work space is too small

The Get Next Extended, Get Previous Extended, Step Next Extended, and Step Previous Extended operations use a buffer as work space. This status code indicates that the work space (set by default to 16 KB) is not large enough to hold the filtering data buffer structure and the largest record to be received. You will receive Status Code 0 if the work space is large enough to hold the filter/extraction expression and enough of the record to include all of the fields to be filtered or extracted.

62: The descriptor is incorrect

This status code is returned in the following situations:

- The descriptor (data buffer structure), which is passed for a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, is incorrect. The descriptor length (the first two bytes of the data buffer) on the extended operation call must be the exact length of the descriptor. This requirement does not apply to the data buffer length option, which can still be declared longer than necessary.

- On a Stat Extended operation, the signature field in the data buffer is not set to 0x74537845, the subfunction field is not set to 0x00000001, or the Pervasive PSQL Explorer field is not set to 0x00000000.
- On a Get Direct/Chunk or Update Chunk operation, the descriptor structure in the data buffer is incorrect, or is inconsistent either internally or with respect to the data buffer length.
- ActiveX control's buffers are not cleared and reallocated. Use the Init method to clear and reallocate the control's buffers before the use of any extended operations in the code. In addition, if you are using AutoMode, it is necessary to establish logical position (GetLast, GetFirst, GetEqual, and so forth) before making the call to Init.

63: The data buffer parameter specified on an Insert Extended operation is invalid

An Insert Extended operation provided an invalid buffer. Either the buffer length is less than 5 bytes, or the number of records specified is 0. Correct the buffer length or the number of records.

64: The filter limit has been reached

The MicroKernel returns this status code for the following reasons:

- During a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, a rejected record was reached; no other record can satisfy the given filtering condition, going in the direction that the operation specified. This is applicable only if the first segment of the key that the key number specified is also used as the first term of the filtering field.
- The number of records to be retrieved is greater than the number of records present in the file that satisfy the filter condition. This option is specified in the data buffer of the extended operation.

65: The field offset is incorrect

The field offset in the extractor of a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation is invalid based on the length of the retrieved record. Make sure that the field offset is a valid value (from 0 through the record length minus 1).

66: The maximum number of open databases has been exceeded

This status code is obsolete in Pervasive.SQL 2000i and later versions.

The MicroKernel tried to open files bound to too many MicroKernel databases. To avoid receiving this status code, you must set a higher value for the number of databases that the MicroKernel can open.

Refer to the *Advanced Operations Guide* for more information about bound files.

67: The MicroKernel cannot open the SQL data dictionary files

The MicroKernel returns this status code for the following reasons:

- An application attempted to use a data file that is bound to a the MicroKernel database, but the MicroKernel could not open one of the MicroKernel data dictionary files (FILE.DDF or, if the file has RI definitions, RELATE.DDF) or the configuration file (DBNAMES.CFG).
- You attempted to create a file with the **Replace** option, and a bound MicroKernel data file with the same name and location already exists. However, the MicroKernel could not open the MicroKernel data dictionary file FILE.DDF, or the configuration file (DBNAMES.CFG).

If the data file has RI definitions, the DBNAMES.CFG file must be in the location specified in the DBNames Configuration Location option in the server configuration settings. Also, ensure that FILE.DDF and RELATE.DDF (if the file has RI definitions) are in the locations specified by the Working Directory option in the server configuration settings.

68: The MicroKernel cannot perform the RI Delete Cascade operation

The MicroKernel cannot enforce the Delete Cascade rule on a file under RI control because the record that the application attempted to delete has more than 16 levels of descendants. Delete records from the lower levels, and then try again to delete the record that the application was attempting to delete initially. Refer to *Advanced Operations Guide* for more information about RI.

69: The Delete operation specified a file that is damaged

The application encountered an error while the MicroKernel was attempting to enforce the Delete Cascade rule in response to a Delete operation. This status code indicates that the related file has been damaged and must be recreated. Refer to *Advanced Operations Guide* for more information about RI and the Delete Cascade rule.

71: There is a violation of the RI definitions

- If you attempted an Insert operation on a file under RI control, a foreign key value in the record to be inserted does not have a corresponding primary key in the referenced file.
- If you are performing an Update operation, there are two possible causes for this status code:
    - You attempted to change the value of a primary key.
    - You attempted to change the value of a foreign key to a value that does not exist for the defined primary key.
- If you attempted a Delete operation, the restrict rule is enforced, and a primary key value in the record you are trying to delete references a foreign key in the referenced file. Refer to *Advanced Operations Guide* for more information about RI.

72: The MicroKernel cannot open the RI referenced file

The referenced file cannot be found at the location specified by FILE.DDF and DBNAMES.CFG. Be sure that the referenced file is in one of the data file locations that the DBNAMES.CFG file specifies for the named database.

- If the DBNAMES.CFG file is defined on a server, verify that the file location does not contain a drive letter.
- If the DBNAMES.CFG file is defined for a Workgroup engine, make sure that the drive letters are the same (and map to the same locations) as specified in DBNAMES.CFG.

Refer to *Advanced Operations Guide* for more information about RI.

73: The RI definition is out of sync

The MicroKernel returns this status code for the following reasons:

- You tried to open a data file that is bound to a MicroKernel database, and the database to which the file is bound was not found in the DBNAMES.CFG file.
- You tried to open a data file with RI (Referential Integrity) definitions that are bound to a MicroKernel database, and the table to which the file is bound was not found in the database FILE.DDF file, or the table location and filename do not match the file location and filename as configured in the DBNAMES.CFG or FILE.DDF file.
- You attempted to modify a bound file, and the RI definition for that file disagrees with the definition in the RELATE.DDF file.
- You attempted an Insert, Delete, or Update operation that would change a foreign key, if the file related to this file is out of sync (an attempt to open or modify the related file would have returned this same status code).
- You attempted to create a file with the **Replace** option, and a bound MicroKernel data file with the same name and location already exists. However, the MicroKernel detected that the existing bound file was out of sync (that is, an attempt to open the existing file would have returned this same status code).

Note: The same named database cannot exist on two servers on the same network. So, if the intent is to move the dictionaries to another server on the same network, one way would be to delete the named database on the old server before creating the same named database on the new server.

Check the RI constraints on your database. For information about how to do this, refer to *Pervasive PSQL User's Guide*.

76: There is a conflict on the referenced file

An application attempted to perform an Update, Insert, or Delete operation on an RI-controlled file that references another file. The application cannot open the referenced file for RI checking because it is already open in Exclusive mode. Wait until the referenced file is closed or is opened in a mode other than Exclusive, and then retry the operation. Refer to *Advanced Operations Guide* for more information about RI.

77: The application encountered a wait error

This status code is obsolete in MicroKernel versions 7.0 and later.

This is an informative status code. You must retry the operation yourself; the MicroKernel does not automatically retry the operation. A client/server MicroKernel returns this status code in one of the following situations:

- The application specified a wait lock bias for an operation, but another user has locked the requested resource.
- The application is currently processing a wait transaction and tried to access a file that another user has locked.

When you are using the Btrieve Requester to access the MicroKernel, the Requester waits and retries if a requested resource is locked. When a server-based application is accessing the MicroKernel and the requested resource is locked, a wait is also required. In this case, the MicroKernel is expected to perform the wait. Because this would occupy the MicroKernel and lock out other users who might be trying to release the requested resource, the MicroKernel does not perform the wait. Instead, it returns this status code, and the server-based application must retry later.

78: The MicroKernel detected a deadlock condition

The application should clear all resources by aborting, ending the transaction, or releasing all record locks before proceeding. This breaks the deadlock, allowing other applications to access the resources for which they are waiting.

79: A programming error occurred

This status code is obsolete in MicroKernel versions 7.0 and later.

There is a malfunction that the MicroKernel cannot specifically detect or from which the MicroKernel cannot recover. Retry the operation. If the error persists, there may be system corruption; try to clear the system by rebooting, and then try the operation again.

80: The MicroKernel encountered a record-level conflict

The MicroKernel did not perform the Update or Delete operation because of a record-level conflict. For example, station A reads a record, station B reads the same record and updates it, and then station A attempts to update the record. The application should reread the record prior to resending an Update or Delete operation. Alternatively, the application can employ record locks to avoid conflicts.

In key-only files, you receive this status code if the record is moved in the file b-tree after being read and before being updated or deleted. A record can move as a result of other records being inserted, updated, or deleted.

81: The MicroKernel encountered a lock error

The MicroKernel returns this status code in one of the following situations:

- The application tried to unlock a record that is locked with a multiple record lock, but the record position stored in the data buffer does not correspond to any record locked in the associated file.
- The application tried to unlock a single-record lock with a multiple-record lock or vice-versa.

82: The MicroKernel lost positioning

When performing a Get Next or Get Previous operation on a key with duplicates, the application tried to retrieve a record that was deleted or whose key value was modified by another application. Use a Get Equal or a Get Direct/Record operation to re-establish positioning. (See Status Code 44: The specified key path is invalid for a related positioning problem.)

83: The MicroKernel attempted to update or delete a record that was read outside the transaction

This status code is obsolete in MicroKernel versions 7.0 and later.

The application tried to update or delete a record within a transaction, but it did not read the record within the transaction. The application must read the record within the transaction before attempting to modify the data.

84: The record or page is locked

- An Insert, Update, or Delete operation attempted to lock an index page to insert or delete a key value. Have your application check for this status code and retry the operation if the status code is returned.
- The application tried one of the following:
  - Applied a no-wait lock on a record that is currently locked by another application
  - Tried to access a file in a no-wait transaction while another application holds one or more active record locks in that file
  - Tried to update or delete a record locked by another application.

  The application can use either of the following recovery methods:

  - Retry the operation until it is successful. This can be the simplest and quickest solution for a network with light to moderate use.

  Note: Applications should limit the number of retry attempts when status 84 is received inside a concurrent transaction. Otherwise, the application might enter a deadlock situation with another transaction. If status 84 is still received after a few retries, abort the transaction and then attempt the transaction again.

  - Use the **wait** option (+100/+300) instead of the **no-wait** option (in versions that support the **wait** option).

85: The file is locked

The MicroKernel returns this status code in one of the following situations:

- While one user has a file locked in an exclusive transaction, another user attempts to lock all or part of that file.
- The workstation MicroKernel has a file open, and a client machine that has the Requester loaded tries to open the same file via the server MicroKernel. The server MicroKernel cannot open the file because it cannot obtain exclusive access. The client machine that has the Requester loaded receives this status code.
- When opened by a MicroKernel, two data files have the same filename but different extensions (for example, INVOICE.HDR and INVOICE.DET). One file is open and in Continuous Operation mode, causing the MicroKernel to generate a delta file (for example, INVOICE.^^^). The MicroKernel returns this status code when you attempt to open the second file. For this reason, we recommend naming your files with completely different names, not just reusing the same name with different extensions.
- Without any pattern of occurrence, you may receive a status 85 when the file is closed because Anti-Virus software opens and locks the file to scan causing the next database operation to fail. To resolve, set the Anti-Virus software to not scan Pervasive PSQL data files. Consult your Anti-Virus software manual for instructions on how to exclude files.
- A delete action was attempted on a data file that is in continuous operations. A data file put into continuous operations is locked from deletion through the relational interface and the transactional interface. In addition, the file is locked from any attempts to change the file structure, such as modifying keys and so forth.

86: The file table is full

An application may receive this status code if the database engine is unable to allocate memory to represent a Btrieve data file. The database engine also has a limit of 65,535 files that it can handle at the same time. The status code is also returned if that limit is reached.

One possible cause of this status code is an application that continues to open additional files without closing the data files that it has already opened. View the active files with the Monitor utility to examine whether this may be happening. See Viewing Active Files in *Advanced Operations Guide*.

In addition, consider status code 87: The handle table is full. Status code 86 is for files and status code 87 is for handles. Multiple handles can be open for a given file. If the database engine cannot allocate memory for a Btrieve handle, then the application may receive a status 87. The handles allocated to client applications are limited by memory. (Older types of handles that are allocated to DOS applications using the DOS requesters are limited 65,535.)

87: The handle table is full

This status code is obsolete in Pervasive.SQL 2000i and later versions. The handle table is managed dynamically by the engine.

You have either attempted to open more handles than the MicroKernel is configured to support, or the MicroKernel attempted to open more files than the operating system allows.

- To configure your operating system to allow more handles, refer to your operating system documentation.

It is helpful to know the following details regarding the MicroKernel requirements for handles from the operating system. When the same file is opened multiple times, the MicroKernel uses only one operating system handle. However, if the file is in v6.*x* or later format and the file is shared via MEFS mode, the MicroKernel opens a second handle for the associated .LCK file.

If the file is in v5.*x* format, the MicroKernel might request a second handle, for the .PRE file. Also, if the file (in any format) is placed in Continuous Operation mode, the MicroKernel requests another handle for the delta file. If the file is extended, the MicroKernel requests an operating system handle for each of the extension files.

- In the Btrieve v6.15 DOS or Microsoft Windows NT 4.0 environments, you may received this status code when opening the 16th file in a DOS application running under Windows NT. There may two solutions:
  - Btrieve File handle configuration may be set incorrectly in BTI.CFG. Check BTI.CFG for file handle setting (/h: and /f:) and increase those values.
  - Check the file= setting in CONFIG.NT. This file is in the WINNT\SYSTEM32 directory and raise the values.

NOTE: default value is 20.

88: The application encountered an incompatible mode error

The MicroKernel returns this status code in one of the following situations:

- If an application opens a file in Exclusive mode, all other applications receive this status code when they try to open the same file in any mode.
- If an application opens a file in any mode other than Exclusive, all other applications receive this status code when they try to open the same file in Exclusive mode.
- While using the MicroKernel Continuous Operation mode:
  - You attempted to remove a file from continuous operation, but the file is not in continuous operation mode.
  - You attempted to remove a file from continuous operation, but a different client placed the file into continuous operation.
  - You attempted to include two files in continuous operation that have the same name but different extensions.
  - You attempted to include a file in continuous operation, but the file is already in continuous operation.
  - The files were previously in continuous operation and the server crashed. Now, when you attempt to take the files out of continuous operation, a Status Code 88 is returned.

In the last case described above, once the server has gone down, the Btrieve engine does not know which files were in continuous operation. Status Code 88 is returned because of this condition.

In order for you to take the files out of continuous operation, you must open the files before trying to end continuous operation mode. When the file is reopened, the Btrieve engine detects that the continuous ops flag is set and looks for the delta file. At that point, the delta file roll-in occurs.

To initiate the roll-in of an existing delta file, the associated data file must be opened. You can open the file with a utility such as the function executor or the application that uses the file. If the application that uses the file repeatedly opens and closes the file, you are advised to open the file with the function executor. This recommendation is made because the delta roll-in is a low priority task. The roll-in was designed in this fashion so that the file can still be used while the roll-in is occurring. If the application closes the file and the roll-in has not finished, the roll-in is initiated again when the file is re-opened. As a low priority task, the roll-in process may take some time. Once the engine completes the roll-in, it deletes the delta file.

89: A name error occurred

This status code is obsolete in MicroKernel versions 5.0 and later.

BSERVER was loaded before you specified the short name to which the device was redirected. You must specify all short names that you want to share with the NET SHARE command before you start BSERVER.

90: The redirected device table is full

This status code is obsolete in MicroKernel versions 6.0 and later.

The DOS Requester redirection table or server routing table is full. This occurs if you attach to additional servers or map additional drives after loading the Requester. Reload the Requester, specifying a larger value for either the **Number of File Servers (/S)** option or the **Number of Mapped Drives (/R)** option. This status code also occurs if you detach from a server and attach to a different server. Once a client has attached to a server, the Requester does not remove its name from the server routing table.

91: The application encountered a server error

Note: Please see our Pervasive PSQL Knowledge Base for new and updated articles on troubleshooting this status code. You can access the Knowledge Base at the Pervasive Software website.

The MicroKernel returns this status code in one of the following situations:

- The Requester cannot establish a session with the server. Either the client/server MicroKernel is not loaded or the server is not active.
- The SPX drivers are not installed or are outdated.
- The MicroKernel has reached the maximum limit for the number of sessions it can open at one time. To avoid receiving this status code, increase the value for the **Number of Sessions** configuration option. Beginning with Pervasive.SQL 8, the MicroKernel dynamically manages the number of sessions, and it cannot be manually increased or decreased.
- An application specified a path for a file and did not include the volume name in the path.
- The MicroKernel Router has not been loaded, and the following situation has occurred: an application that uses both the MicroKernel Router and the MicroKernel to make remote calls (and which therefore includes the server and volume name when performing an Open operation) has attempted to open a remote file. Because the MicroKernel Router does not interpret the server name, the MicroKernel attempts to do so but cannot.
- A communication or network addressing problem exists in your network environment, so the MicroKernel requests never reach their destination server address. Ensure that your client and server network components are up to date and certified for your network environment.

► **To adjust the Receive Packet Size:**

1. Access the server properties in PCC (see To access configuration settings in PCC for a local client in *Advanced Operations Guide*.
2. Click **Communication Buffer Size** in the properties tree.
3. Ensure that the **Receive Packet Size** value is appropriate for your environment.

92: The transaction table is full

This status code is obsolete in MicroKernel versions 7.0 and later.

The application exceeded the maximum number of active transactions. Use the configuration properties to specify a higher value for the **Number of Transactions** configuration option.

93: The record lock types are incompatible

The application tried to mix single-record locks (+100/+200) and multiple-record locks (+300/+400) in the same file at the same time. You must release all locks of one type before you can execute a lock of the other type.

94: The application encountered a permission error

The MicroKernel returns this status code in the following situations:

- The application tried to open or create a file in a directory without the proper privileges. The MicroKernel does not override the network privileges assigned to users.
- The designated server is in the server routing table, but your particular client is not logged into that server.
- The system data source name (DSN) on the server has an error in the pathname to the data files.

95: The session is no longer valid

Note: Please see our Pervasive PSQL Knowledge Base for new and updated articles on troubleshooting this status code. You can access the Knowledge Base at the Pervasive Software website.

The server MicroKernel returns this status code for one of the following reasons:

- The previously established session is no longer active due to an error at the client machine, at the server, or on the network. Verify that the client machine is still attached to the server, and then unload and reload the Btrieve Requester.

If you are using the SPX protocol:

- The server MicroKernel has reached the maximum number of SPX sessions. Use the Monitor utility to check this statistic. To avoid receiving this status code, add more memory. The number of sessions is managed dynamically up to available memory.
- This may be a time delay problem if the client machine does not receive a response back from the server in an appropriate time frame or after an appropriate number of retries. Refer to your network configuration documentation for information about increasing timeout and retry parameters. This is often necessary in a WAN environment or a LAN configuration with heavy network traffic.

► **To adjust the Receive Packet Size:**

1 Access the server properties in PCC (see To access configuration settings in PCC for a local client in *Advanced Operations Guide*.

2 Click **Communication Buffer Size** in the properties tree.

3 Ensure that the **Receive Packet Size** value is appropriate for your environment.

- Ensure that the SPX timeout parameters are set as follows in both the client machine NET.CFG file and the server SPXCONFG.NLM file:

  ```
  SPX VERIFY TIMEOUT=54
  SPX LISTEN TIMEOUT=108
  SPX ABORT TIMEOUT=540
  ```

  These three values must have a 1:2:10 ratio. You can increase these values to at most three times the default. If you continue to receive this status code after increasing these values, the problem is most likely not related to these settings.

- For Windows servers, verify that the **Maximum Packet Size** registry setting is 576 decimal or 240h. The path to the MaxPktSize registry setting is HKEY_LOCAL_MACHINE\System\currentControlSet\ Services\NwlnkIPX\NetConfig\MaxPktSize.

  If you continue to receive this status code after increasing the network timeout parameters, this status code usually indicates a problem with network communications. Verify that you have up to date network cards and drivers; for example, incompatible LAN card drivers can also cause this status code to occur. Consult your LAN administrator for network communication troubleshooting.

1 Choose **Start** ▸ Run.

2 Type Regedit and click **OK**. The Registry Editor opens.

3 Make the following CHANGES to the server's registry using Regedit:

HKEY_Local_Machine ▸ System ▸ CurrentControlSet ▸ Services ▸ NwLnkIpx ▸ <Network Card> ▸ MaxPktSize = 240 Hex.

HKEY_Local_Machine ▸ System ▸ CurrentControlSet |
Services ▸ NwLnkIpx ▸ <Network Card> ▸ NetworkNumber = <Non-Zero Value>

4 ADD the following registry entry:

HKEY_Local_Machine ▸ System ▸ CurrentControlSet |
Services ▸ LanManServer ▸ Parameters ▸ MinClientBufferSize regdword = 500 decimal.

96: A communications environment error occurred

The MicroKernel returns this status code for the following reasons:

- You tried to attach to the MicroKernel on a server, but the SPX connection table or the MicroKernel client table is full. To avoid receiving this error, add more memory. Both of these resources are managed dynamically up to available memory.
- An application that calls the MicroKernel can return this status code if the DBNAMES.CFG file contains a named database definition specifying a data location on a different server.

97: The data buffer is too small

The application either tried to read or write a record that is longer than the current allowed settings for the MicroKernel or the Btrieve Requester, as follows:

- For an Update, Insert, or Create operation, the application receives this status code if the data buffer length it specifies for the record exceeds the message buffer length.
- For a Get, Step, or Stat operation, the application receives this status code if the message buffer is shorter than the length of the data the MicroKernel would return, regardless of the data buffer length specified in the application.
- For a Get Chunk or Update Chunk operation, the total size of the retrieved or updated chunk exceeds the message buffer length.
- *DOS Requesters only:* Reload the Btrieve Requester and specify a higher value for the message buffer size. This is done using the /D parameter which is documented in *Getting Started With Pervasive PSQL*.
- For Windows servers, verify that the Maximum Packet Size registry setting is 576 decimal or 240h. The path to the MaxPktSize registry setting is HKEY_LOCAL_MACHINE\System\currentControlSet\Services\NwlnkIPX\NetConfig\MaxPktSize.

100: No cache buffers are available

This indicates that the MicroKernel has used all the cache buffers it allocated at load time.

You may encounter this status code if your application uses a large number of write operations (insert, update, and delete) in a user transaction. The current implementation of the MKDE requires all modified pages to reside in cache until the transaction completes.

NOTE: On a machine with limited available memory, you may not be able to successfully complete very large transactions with thousands of write operations.

If you are a developer, you can modify your application to commit transactions more frequently, so that fewer modified pages remain in cache. The more common approach to this problem is to increase **Cache Allocation Size** in the configuration options and then reload the MicroKernel.

On Windows, the first time the database engine starts it initializes the cache allocation size to 20% of physical memory and writes that value to the Windows Registry. After that, whenever the engine starts, it reads the value from the Registry and does not re-calculate the setting. If you add or remove memory from the system, you must modify the cache allocation size setting to take best advantage of the new amount of memory available.

► To increase Cache Allocation:

1 Start Pervasive PSQL Control Center (see Starting PCC on Windows in *Pervasive PSQL User's Guide*).

2 Expand **Engines** and find the desired engine name.

3 Right-click on the engine name and select **Properties**.

4 Click **Performance Tuning**.

5 In the right hand frame, adjust the **Cache Allocation Size** by entering the amount of memory to allocate for the cache. See Cache Allocation Size in *Advanced Operations Guide*.

6 Restart the engines for the new setting to take effect.

101: Insufficient operating system memory is available

This indicates that there is not enough operating system memory available to perform the requested operation. To fix this problem, perform one or more of the following:

- Go to Performance tuning in the server configuration and decrease the value for the **Cache Allocation** configuration option.
- Add memory to the server.

102: Insufficient stack space is available

This indicates that the MicroKernel has run out of stack space. To increase the amount of stack space available to your application, re-link the application, setting the stack size to a higher value. The MicroKernel returns this status code only to Windows-based applications that call WBTRCALL.DLL, or applications that call the Btrieve interface on the local server.

103: The chunk offset is too big

The MicroKernel returns this status code in one of the following situations:

- A Get Direct/Chunk operation specified an offset beyond the end of the record, either explicitly or using the next-in-record bias to the subfunction value. Unless the MicroKernel returns this status code while processing the first chunk, the operation was partially successful. Check the data buffer length parameter immediately after the call to see how much data was retrieved (and therefore how many chunks).
- An Update Chunk operation specified an offset that is more than one byte beyond the end of the record. This status code indicates that the MicroKernel has made no changes to the record.
- An Update Chunk operation with an Append subfunction causes a record length to exceed its operating system file size limit. The MicroKernel has made no changes to the record.

104: The MicroKernel does not recognize the locale

During a Create or Create Index operation, the operating system was not able to return a collation table for the country ID and code page specified. Ensure that the application specified the locale's country ID and code page correctly and that the operating system is configured to support the country ID and code page.

105: The file cannot be created with Variable-tail Allocation Tables (VATs)

An application tried to create a file with Variable-tail Allocation Tables (VATs) but without variable-length records (a precondition for files to use VATs). This status code applies to key-only files as well as to regular data files.

106: The MicroKernel cannot perform a Get Next Chunk operation

An application called the Get Direct/Chunk operation to retrieve a chunk from a record and used the next-in-record bias on the descriptor subfunction. However, after the application established its positioning in the record (and prior to this call), the target record was deleted.

107: The application attempted to perform a chunk operation on a pre-v6.0 file

An application tried to use either a Get Direct/Chunk operation or an Update Chunk operation on a file in pre-v6.0 format.

109: An unknown error was encountered either creating or accessing a semaphore

The Windows platform of the workstation MicroKernel attempted an operation using incompatible versions of the DLLs. Shut down the MicroKernel and make sure that you are using the most recent versions of the DLLs.

110: The MicroKernel cannot access the archival logging configuration file

The archival logging configuration file (BLOG.CFG) contains entries for the data files on the drive for which you want to perform archival logging. The MicroKernel returns this status code for the following reasons:

- The MicroKernel cannot find the BLOG.CFG file. Ensure that the file is in the \BLOG directory in a real root directory of the physical drive that contains data files you want to log. (That is, do not use a mapped root directory.) If your files are on multiple volumes, you must create a \BLOG directory on each volume.
- The MicroKernel cannot open the BLOG.CFG file. Either the file is locked or it does not exist.
- The MicroKernel cannot read the BLOG.CFG file. Either the file does not use the correct format or it is corrupt. Refer to *Advanced Operations Guide* for information about the format of the BLOG.CFG file.

111: The specified filename was not found in the archival logging configuration file

The MicroKernel cannot find the specified file in the BLOG.CFG file. The file must be specified in the BLOG.CFG file on the same physical drive. By default, the MicroKernel names the archival log file the same as the logged file, but with a .LOG extension. However, you can specify a different filename for the archival log file in the BLOG.CFG file. Ensure that the BLOG.CFG file indicates the correct filename for the archival log and ensure that the archival log file exists.

112: The specified file is in use by another client

Before the MicroKernel can perform a roll forward, the file must be in the same state it was in when it was last backed up. If another client changes the file, you must restore the file again before rolling forward.

113: The MicroKernel is unable to open the archival log for the specified file

The status code can result from different situations. In one case, the database engine cannot find the archival log file associated with the specified file. By default, the MicroKernel names the archival log file the same as the logged file, but with a .LOG extension. However, you can specify a different filename for the archival log file in the BLOG.CFG file. Ensure that the BLOG.CFG file indicates the correct filename for the archival log and ensure that the archival log file exists.

Another cause can be that archival logging is still turned on for a file and your are trying to roll forward a log file of the same name. Because the database engine is actively logging to the log file, you cannot use it to roll forward.

114: The archival log for the specified file is invalid

The archival log associated with the specified file is not a valid archival log file. By default, the MicroKernel names the archival log file the same as the logged file, but with a .LOG extension. However, you can specify a different filename for the archival log file in the BLOG.CFG file. Ensure that the BLOG.CFG file indicates the correct filename for the archival log and ensure that the archival log file exists.

115: The MicroKernel cannot access the archival logging dump file

The MicroKernel cannot access the archival logging dump file for one of the following reasons:

- The filename indicated for dumping entries in an archival log is not a valid filename. Be sure this filename does not contain a volume specification. The dump file is created on the same volume as the log file.
- The caller does not have access rights to the dump file.
- The MicroKernel cannot open the file because another user has opened the file using an exclusive operating system lock.

116: The file is owned by another MicroKernel engine acting as a Gateway

The MicroKernel cannot contact the engine running on the gateway computer even though it can read the locator file. This might occur the following reasons:

- The name of the gateway computer cannot be resolved. To solve this problem, try one of the following:

- Ensure that the gateway computer is registered with your name resolution service, such as DNS.
- If you are not using a name resolution service, you must provide name resolution manually. Locate the file named HOSTS on your current machine. Add a line in this file associating the TCP/IP address of the gateway computer with the network name of that computer. For example, if the gateway computer is named “mycomp” and its IP address is 125.1.4.245, then you should add the following line to the file:

  ```
  125.1.4.245 mycomp
  ```

- When the two computers are separated by a router so they can both see the server, but cannot see one another. Try the following:
  - **a.** Use the Gateway Locator utility to identify the owner of the gateway.
  - **b.** Use Pervasive System Analyzer (PSA) to test the network connection to that computer.
- You may have attempted to open a file with two different Workgroup engines that are mapped to the files using different share names. The MicroKernel attempts to correct this, but cannot do so in all cases. Make sure each computer is mapping to the same share name.

117: Invalid content detected in Continuous Operations delta file

An operation that caused the delta file to be read during Continuous Operations encountered invalid content. (See also Using Continuous Operations in *Advanced Operations Guide*.)

Note: Invalid content in a delta file is a condition that is extremely unlikely to occur. However, if it does, operations on the data file made while backup was in progress may be lost. Transactions that occurred during the backup **can** be recovered **if** Transaction Logging or Transaction Durability is set to “on” for the Server configuration and the data file was not opened in accelerated mode. If you require integrity of multiple data files, open all data files that were modified during backup even if their delta files are valid.

No user action is required if this status code occurs during a start of the Pervasive PSQL engine. That is, Pervasive PSQL was in Continuous Operations for one or more files when an outage occurred. The database engine does not attempt to roll in any invalid delta files.

In the scenario just described, the database engine logs a message reporting this status code. The OPEN operation that caused access to the bad delta file will still succeed even though the delta file contains invalid content. If a transaction log exists, lost transactions are rolled forward as normally occurs during an OPEN operation after an outage.

After the start and OPEN complete and any transactions are rolled forward, **move** the delta file with invalid content to a different directory. This allows you to perform Continuous Operations on the same data file again and retains the delta file with invalid content if you need technical support.

If an operation returns this status code while Continuous Operations is active, the operation did not successfully either read from or write to the delta file. In this case, the delta file may no longer be usable for Continuous Operations. Your best option is to do the following actions in the order indicated:

- Close **all** OPENs of the data file.
- Cease all further operations directed at the data file.
- End Continuous Operations by issuing a `butil -endbu` command (or, if you use Pervasive Backup Agent, you can issue the command `pvbackup -off`).

  Note that if the delta file is damaged the roll-in may not complete successfully when ending Continuous Operations.
- If Transaction Logging or Transaction Durability is active for the data file, perform a single OPEN of the data file to roll forward the transaction log.
- If you require integrity of multiple data files, apply these same actions to all data files that were written during the backup.
- After completing these corrective actions, **move** the delta file with invalid content to a different directory. This allows you to perform Continuous Operations on the same data file again and retains the delta file with invalid content if you need technical support.

120: Maximum number of B-Tree index levels reached

This status code may result if you use a large key size for an index and small page sizes. Index keys can fill the B-Tree to the allowable depth even though the B-Tree is not completely full.

To prevent this status code, try one or both of the following:

- Rebuild the data file with a larger page size to increase the number of keys stored per page.
- Turn on index balancing to maintain a better distribution of index keys (performance decreases somewhat with index balancing on). See Index Balancing in *Advanced Operations Guide*.

130: The MicroKernel ran out of system locks

This status code is obsolete in MicroKernel versions 6.15 and later.

This status code can indicate a temporary condition in which no system locks are currently available. The following are example cases:

- A single client is performing a very large transaction, in which thousands of records are being modified.
- Many clients are performing large transactions concurrently.

A client can receive this status code whether or not it is in a transaction. In some cases, a client can simply retry the failed operation. If other clients have released system locks in the interim, the retried operation may succeed. If a client in a transaction receives this status code, end or abort the transaction. If the transaction is very large, consider breaking it into multiple, smaller transactions. You can also use the Setup utility to lower the number of system locks devoted to explicit locking. To do so, lower the values assigned to the **Number of Locks** and/or **Number of Sessions** configuration options.

132: The file has reached its size limit

The MicroKernel returns this status code in one of the following situations:

The MicroKernel returns this status code in one of the following situations:

- The file size reached its maximum limit. This limit depends on the file version, the page size, and the number of records per page. See File Size in *Advanced Operations Guide* for a complete discussion.
- An operation attempted to allocate more than the maximum number of page allowed for a data file.
- A data file has remained in continuous operation for a lengthy period of time, causing its delta file to exceed 4GB in size.
- A single data file segment has reached the operating system file size limit.

If the file uses a page size smaller than 4,096 bytes, you can rebuild the file using Rebuild utility and set the page size to 4,096 bytes, to take advantage of the larger file size limit.

133: More than 5 concurrent users attempted to access the same data file

This status code is obsolete for all versions of Pervasive.SQL 7 and later. In the Pervasive.SQL 2000i SDK for a workstation environment, you attempted to access a data file with more than five MicroKernels at the same time. The Pervasive.SQL 2000i SDK for a Workgroup environment limits the number of concurrent users of a file to five engines.

134: The MicroKernel cannot read the International Sorting Rule

The MicroKernel returns this status code for the following reasons:

- The ISR is not found in the COLLATE.CFG file.
- The COLLATE.CFG file is missing or corrupt.
- The MicroKernel cannot read the ISR from the COLLATE.CFG file.

135: The specified ISR table is corrupt or otherwise invalid

The MicroKernel found a readable COLLATE.CFG file, but the specific International Sorting Rule table is invalid.

136: The MicroKernel cannot find the specified Alternate Collating Sequence in the file

The MicroKernel returns this status code in the following situations:

- You tried to create an index that uses an Alternate Collating Sequence (ACS), but the MicroKernel cannot locate an ACS with the specified name in the file.

- You called a Step Next Extended, Get Next Extended, Step Previous Extended, or Get Previous Extended operation and specified an ACS name, but the MicroKernel cannot locate an ACS with the specified name in the file.

138: The NULL indicator position is invalid

- In order to ensure accessibility to your data from all of the Pervasive PSQL access methods, the NULL indicator segment (NIS) must appear immediately before the data segment that the NIS indicates.
- A NIS cannot be indicated by another NIS.

139: The MicroKernel has detected an unacceptable value in the key number

Certain operations either use, or reserve the use of, the key number parameter as a subfunction number, rather than as a means to specify a file index to be used with the operation. (Note: This is also done in the GetEqual operation). This status code is returned if an application does not specify a valid subfunction number (via the key number parameter) to one of these operations:

- You issued a Begin Transaction operation with an invalid key number.
- You issued an End Transaction operation with an invalid key number.
- You issued an Abort Transaction operation with an invalid key number.
- You issued a Start Extended operation with an invalid key number.

143: The MicroKernel cannot allow unauthorized access to files in a secure MicroKernel database

- You attempted to open a data file bound to a MicroKernel database that has security enabled. The MicroKernel does not allow access to such files, except through the MicroKernel.
- The MicroKernel also returns this status code if you are *not* using the MicroKernel and all of the following are true:
  - You attempt to create a file with the **Replace** option.
  - A bound MicroKernel data file with the same name and location already exists.
  - The database to which the existing file is bound has security enabled.

146: Duplicate system key

The same key number was generated by two different threads generating system keys.

147: The log segment is missing

The MicroKernel cannot find a log segment that is necessary for rolling at least one file forward.

148: A roll forward error occurred

The MicroKernel encountered an error while rolling a file forward. Depending on the operating system, the MicroKernel reports an error message as follows:

- The Windows workstation MicroKernel displays the message in the console message window and writes the message to the Pervasive Event Log (PVSW.LOG), which is located in the WINDOWS\SYSTEM or WINNT\SYSTEM32 directory.
- The Windows server MicroKernel does not display a message, but writes the message to the Pervasive Event Log (PVSW.LOG) in the WINNT\SYSTEM32 directory.

149: SQL Trigger

While using the Btrieve API to alter database tables or entries, the system encountered SQL restrictions placed on the database by the SQL layer.

151: Chunk Offset Too Small

You cannot insert or delete chunks within the fixed portion of a record.

160: Invalid parameters passed to MicroKernel

The MicroKernel detected a corrupt parameter used by the SRB (Service Reply Block) due to one of the following:

- Network glitch occurring in the process of transporting the SRB across the network, somehow damaging the parameters in the SRB.
- The most likely cause for this error is a mismatch between legacy Scalable SQL components on your system and Pervasive PSQL components.

To resolve this problem, reinstall Pervasive PSQL to restore consistency among the installed components. If you still encounter the problem after reinstalling and rebooting, contact Technical Support.

161: A key has reached a maximum limit for user count, session count, or data in use, or has changed state to expired or disabled

Several scenarios can result in this status code being returned.

- This status code is also returned after a temporary license has expired. The state of the key, which can be verified with the License Administration utilities, changes to “expired.” Commonly after a temporary license expires, all users receive this status code and are unable to access the database engine. The solution is to authorize a permanent license key. Contact the application vendor or Pervasive Software to purchase a permanent license key.
- The maximum limit allowed by your license agreement for user count, session count, or data in use has been reached. You can try closing one or more sessions or files, which may reduce the value for user count, session count, or data in use below the maximum limit. (A “session” is defined as a client ID used by the transactional engine interface or a connection to the relational engine interface.) Another solution is to authorize an increase key for user count, session count, or data in use.

  See also Monitoring Database Resources in *Advanced Operations Guide* for details on determining the current, peak, and maximum values for user count, session count, and data in use.
- You may also receive this status code if your machine configuration changed since you authorized the Pervasive PSQL product. Product authorization is tied to your machine’s hardware configuration. After you have applied a product key on a machine, changes to certain hardware configuration items could disable the key. The state of the key, which can be verified with the License Administration utilities, changes to “disabled.”

  If you need to change hardware configuration, deauthorize the key first using the License Administrator utilities. Deauthorizing the key disassociates the product key from the unique hardware configuration. After you complete the hardware configuration changes, you can again authorize the product key using the License Administrator utilities.

162: The client table is full

This status code is obsolete in Pervasive.SQL 2000i and later versions. The related configuration setting is managed dynamically by the engine.

You may receive this status code due to one of the following:

- You have run out of memory.
- Your number of Active Clients has exceeded 64K.

163: The NULL indicator cannot be the last segment

The NULL indicator segment (NIS) cannot be the last segment of the key descriptor.

169: Protocol mismatch between client cache and remote engine

This status code indicates that your client software is not up-to-date with your remote database engine. You should only receive this status code if you are running V8 pre-release client software against a V8 general-release remote engine.

The solution for this issue is to uninstall your client software and install the latest V8 client.

170: Database login required

Authentication to the database failed due to a wrong or missing username.

171: Database login failed

Authentication to the database failed due to a wrong or missing password.

172: Database name not found

Specify a valid database name for the machine.

173: Already logged in

A Btrieve login request failed because the client is already logged into the specified database.

174: Logout failed

A logout can fail if you are not logged in to a database or there are remaining open file handles to the database when the logout is attempted.

175: Wrong database URI format

The URI connection string was formatted incorrectly. The first five bytes should be “btrv:”.

176: File or table not specified in URI

An Open or Create was issued using a URI connection string that contained neither a file name nor a table name.

177: Table not in database

An Open was issued using a URI connection string that contained no file name, and a table name that does not exist in FILE.DDF.

178: Directory not in database

An Open was issued using a URI connection string that contained a full path file name that references a directory that does not exist as one of the data directories for the database.

Add the directory to the database using the database properties dialog in the Pervasive PSQL Control Center (Windows) or the `dbmaint` utility (Linux).