Forcing the Requester Version with PSQL

A White Paper From

GOLDSTAR
SOFTWARE
www.GoldstarSoftware.com

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

There is a Btrieve API call that allows a developer to determine the version of the database requester, known as the "Version" call. Normally, this call will return the database requester version, and it can be very useful for the application to know which version is running, so that it can use the supported features of that version.

However, sometimes, application developers use this call incorrectly. Instead of finding out if the database version was GREATER OR EQUAL TO a given value (such as 6.15), they instead checked for EQUAL TO the value. While this worked when we had only Btrieve 5.x, 6.10, and 6.15, this logic often breaks when users upgrade to Pervasive.SQL V7 or newer releases. Of course, the Btrieve API has an incredible level of backwards compatibility, so these applications will usually run on the newer databases without any problem at all -- and it is only these shortsighted version checks that cause problems.

Luckily, Actian created a workaround for this problem. With every version starting with Pervasive.SQL 2000i, the client software includes a special feature that allows you to change the information that is provided back to an application by the Version call. This change does not affect every application, but rather you indicate for which application executables you wish to change the version, and the client then lies to that application to provide the expected response for the application in question.

To make this change, you will need to manually edit your registry, so this is not stuff for the uninitiated -- messing up the registry can trash your computer. As such, make sure you know what you are doing if you attempt this. To that end, since only experienced people should be doing this, we'll keep this description on a highly technical level.

Start RegEdit and find the registry key for the Microkernel:

HKLM/Software/Pervasive Software/Microkernel Router.

Create a new key (it will probably not be there already) called "Settings Override" and open it up.

Create a new key (inside Settings Override) for the application name (e.g. peachw.exe) and open it up. Note that this is the name of the controlling EXE or DLL file. You may have to experiment to find the exact controlling file, or you may need multiple keys in here to handle some applications. The FileMon and RegMon utilities from SysInternals (www.sysinternals.com) are quite helpful in this regard, or you may simply be able to right-click the startup icon and find out which EXE is referenced therein.

Create a new String Value inside the application name key with the name of "Version Level", then set the value of the entry to the database version you want, such as "6.15".

Create another new String Value inside the application name key with the name of "Version Type" and assign a value of the database type from the following chart: "S" = NetWare Server, "T" = Windows Server, "W" = Win16 Workstation Engine, "N" or "9" = Win32 Workstation Engine. There may be other valid values, but S and T are usually the safest, if client/server databases are supported by the application.

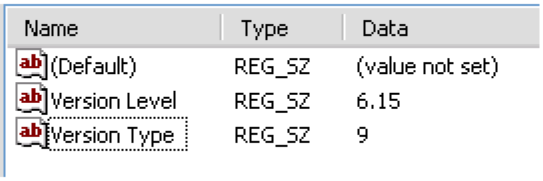
For the PeachTree example above, the application name is “PEACHW.EXE”, so the registry settings will be:

> HKLM\Software\Pervasive Software\Microkernel Router\Settings Override\peachw.exe\Version Level = "6.15"
> HKLM\Software\Pervasive Software\Microkernel Router\Settings Override\peachw.exe\Version Type = "T"

As another example, for a version of the Lytec Medical software (LMXE.EXE) that requires Pervasive.SQL 2000i, it will be:

> HKLM\Software\Pervasive Software\Microkernel Router\Settings Override\lmxe.exe\Version Level = "7.90"
> HKLM\Software\Pervasive Software\Microkernel Router\Settings Override\lmxe.exe\Version Type = "T"

When you are done, it should look like this:



Name	Type	Data
(Default)	REG_SZ	(value not set)
Version Level	REG_SZ	6.15
Version Type	REG_SZ	9

Now, try the application and see if the new version is being seen by the application. If so, then the application will probably work correctly! If not, you probably got the wrong application task name. You may have to use the RegMon utilty (from SysInternals) to watch the registry for all accesses to the "Settings Override" key, and you should see the exact task name that the requesters are looking for.

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