Moving Pervasive SQL Speedbase Databases

1. Introduction

This document describes two techniques that can be used to copy Pervasive SQL (formerly known as Btrieve) format Speedbase databases between servers (or between different folders on the same server).

In addition to restoring the actual Pervasive SQL (Btrieve) database files a number of registry options on the new server **MAY** have to be modified to agree with the settings on the server from which the data files were copied.

2. Pervasive SQL Database Overview

A Pervasive SQL (Btrieve) Speedbase database consists of 6 parts:

- A Global format Data Dictionary (e.g. DIDEMON);

- A Global format Schema File (e.g. DBDEMON) on the same unit. Note that the Schema File for a Pervasive SQL Speedbase database is unlike the Index File, with the same name, for a Global format Speedbase database. The Pervasive SQL Speedbase Schema File contains pointers to the BDCF and Pervasive SQL data files. These pointers contain the Server name and the path to the BDCF file;

- The BDCF file. This is very similar to the Global Data Dictionary file but resides on the Server in the same directory as the Pervasive SQL data files;

- The Pervasive SQL data files, one per Speedbase record type. Each file contains the data and indexes;

- Various entries in FILE.DDF, FIELD.DDF and INDEX.DDF files. These are held in the same directory as the BDCF and Pervasive SQL data files. These files provide external products, such as Pervasive Scaleable SQL, a view of the database;

- An entry in DBNAMES.CFG, which normally resides in the Windows directory. This file is needed by Pervasive SQL to provide an SQL database name for the directory that contains the .DDF files. The Pervasive SQL Speedbase Gateway does not currently maintain DBNAMES.CFG, the Pervasive Control Centre is used for this task. The file, DBNAMES.CFG, is itself a Pervasive SQL file. **This file is only required if Pervasive SQL is going to be used to access the database.**

3. Restoring a Pervasive SQL Speedbase Database, Method 1

This simple technique can be used to restore any Speedbase Btrieve database:

The following files MUST be available:

- The Global schema file;
- All the *.dat or *.mkd data files;
- All the *.ddf files;
- The BDCF from the same directory as the *.dat or *.mkd files. **NOTE THAT THE BDCF FILE IS VITAL - SEE BELOW.**

What you need to do to restore the data:

1. Copy the Global schema file to an appropriate unit;
2. Move the entire Windows directory (i.e. *.dat or *.mkd files, the BDCF file and the various .DDF files)
3. Use the $BADN/$BN32 "Change Path to Database" option to update the Global schema file.
4. Use the Pervasive Control Centre (PCC) to add a Database Name to point to the new Windows directory. **This step is only required if Pervasive SQL is going to be used to access the database;**

The Gateway will operate with the current database settings (see section 5). A simple database rebuild will retain the current database settings. However, a database conversion will result in the current database settings being changed to reflect the current registry settings.

IMPORTANT NOTE: Although this data restore procedure is the simplest to implement it does have the disadvantage of overwriting any existing DDF files in the selected Windows directory. **DO NOT USE THIS OPTION IF THE WINDOWS DIRECTORY CONTAINS AN EXISTING PERVASIVE SQL SPEEDBASE DATABASE**.

4. Restoring a Pervasive SQL Speedbase Database, Method 2

Method 2, described in section 3, cannot be used if the existing Windows folder already contains one, or more, Speedbase Btrieve databases. In these circumstances the technique described in this section must be used.

The following files MUST be available:

- A Global format database of the same generation number;
- All the *.dat or *.mkd data files;

What you need to do to restore the data:

1. Use $BADN/$BN32 option 1 to create an empty database on the selected Windows directory. This option will create the required DDF BDCF File relationships, without destroying the existing DDF files in the destination directory. However this option will create the new database in the format now specified by the current registry settings.

2. Copy the supplied database files (.dat or .mkd) files over the empty ones that have just been created in step-1.

IMPORTANT NOTE: USE OF METHOD-2 WILL RESULT IN AN UNUSABLE DATABASE IF ANY OF THE SETTINGS IN THE REGISTRY ARE DIFFERENT FROM THE CURRENT DATABASE SETTINGS EMBEDDED IN THE .DAT OR .MKD FILES. GREAT CARE MUST BE TAKEN TO ENSURE THAT THE REGISTRY SETTINGS ON THE "DESTINATION" SERVER/GATEWAY ARE THE SAME AS THE SETTINGS ON THE "SOURCE" SERVER/GATEWAY.

5. Current Database Settings

As explained in document IN274 a number of registry settings are available to configure the Speedbase Pervasive SQL and Microsoft SQL Gateways. Some of these registry settings just apply to the current Gateway session and can be ignored when transferring databases between locations. However, several other registry settings affect the layout of the data in the database files, the BDCF files or the DDF files and **MUST BE IDENTICAL BETWEEN THE SOURCE AND DESTINATION SERVERS/GATEWAYS OTHERWISE THE TRANSFERED DATABASE WILL BE UNUSABLE.**

The following settings **must** be identical:

PervasiveTrueNullDate (see IN274 section 2.18)

PervasiveIncludeIdentity	(see IN274 section 2.19)
PervasiveUseMKDExtn	(see IN274 section 2.20)

The LongNames, LongNameType and UpperCase settings (see IN274 sections 2.15, 2.16 and 2.17) do not affect the Pervasive SQL .DAT or .MKD files. The various long names options only affect the .DDF files. When using Method-1 to transfer a Pervasive SQL Speedbase database the LongNames etc. options travel with the imported .DDF file(s). However, when using Method-2, if you require 3rd party reporting tools such as Crystal Reports to continue accessing the database the LongNames, LongNameType and UpperCase settings must be identical between the source and destination servers/gateways. If you do not require 3rd party reporting tools such as Crystal Reports to continue accessing the database, the various long names settings are not important.