Loading the Transducer Database

Importing Data into the Transducer Database

Jon Drnek

V-Note # 0003

Abstract

Instructions for loading the transducer database with data from an external system.

Question

How can I load the transducer database in VibrationVIEW with data from my existing transducer database?

Answer

VibrationVIEW uses SQLite (www.sqlite.org) as the database engine behind the transducer database. The details on how to use SQLite are beyond the scope of this document but the documentation on the SQLite website is quite good.

There are three tables that are used in VibrationVIEW, the **transducer** table, the **settings** table, and the **security** table.

The primary table is the **transducer** table. Although the fields in the table match the field in the Transducer Database dialog box in VibrationVIEW, there are some items to note.

- The GUID field is the key for the table. This should be a globally unique key. This is not visible in VibrationVIEW
- The Axis field in VibrationVIEW is called Direction in the database.
- The "Differential Input" field in VibrationVIEW is the inverse of the "Single Ended" field in the database
- If there is a 1 in the Deleted field, the transducer is considered logically deleted and will not show up in VibrationVIEW.
- CalibrationDate stores the number of seconds elapsed since midnight on December 31, 1899
- Triggers are used to update the ModifyDate field when a row is changed.
- The following fields are not currently used and should not be updated by external systems.
    - Label
    - Assigned System
    - Assigned Channel
    - Create User
    - Modify User

Transducer Table Layout

Field Name	Field Type
GUID	TEXT PRIMARY KEY
TransducerType	TEXT
TEDSEnabled	INTEGER
Manufacturer	TEXT
Model	TEXT
SerialNumber	TEXT
LabID	TEXT
CalibrationDate	TEXT
SensitivityAtFref	FLOAT
Direction	TEXT
UserData	TEXT
Label	TEXT
PoweredAccel	INTEGER
CapCoupled	INTEGER
SingleEnded	INTEGER
DCReading	INTEGER
AltUnitEnabled	INTEGER
AltUnitName	TEXT
AltUnitLabel	TEXT
CreateDate	TEXT DEFAULT CURRENT_TIMESTAMP
CreateUser	TEXT
ModifyDate	TEXT
ModifyUser	TEXT
Deleted	INTEGER DEFAULT 0
AssignedSystem	TEXT
AssignedChannel	INTEGER
LowBiasVoltage	INTEGER

Import

A simple way to import transducers is to use a .csv file.

1. Create a .csv file from the source database. There must be one column of source data for each column in the database. The columns can be blank.
2. Start the sqlite3 command line program with a parameter of your database

```
C:\bin\SQLite>sqlite3.exe c:\path\to\db\xdr.db
SQLite version 3.6.13
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite3>
```

3. Set the separator to match the field separator in your .csv file.

```
sqlite> .separator ,
```

4. Import the data using the import command.

```
sqlite> .import sourcefile.txt transducer
```

Synchronization

Simple synchronization between an external database and the transducer database can be done with the following steps on a periodic basis.

1. Create a .csv file from the source database
2. Delete the **all** transducers from the transducer database using the following commands.

```
C:\bin\SQLite>sqlite3.exe c:\path\to\db\xdr.db
SQLite version 3.6.13
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite3>delete from transducer;
```

3. Import the .csv file created in step 1 using the steps described in the Import section.

These steps can be automated using a windows batch file

Samples

Sample files for the above examples can be downloaded from http://www.vibrationresearch.com/dbsync.html

DDL

The actual DDL used for the transducer database can be downloaded from http://www.vibrationresearch.com/downloads/TransducerDBDDL.sql

Other Tables

The **settings** table is for internal use only and should not be updated by external tools.

The **security** table identifies which computers have access to update the **transducer** table. This is only enforced in VibrationVIEW, not in the database itself. If your interface would need to replicate this security, you would have to implement it yourself.

The key field for the Security table is “computer”. The value would be found in one of the following registry locations. The first entry takes precedents.

```
\HKLM\SOFTWARE\Vibration Research Corporation\8.0\ComputerID
\HKCU\SOFTWARE\Vibration Research Corporation\8.0\ComputerID
```

If there is an entry in the table for the computer ID with the “permission” value set to 1, the computer can update the database. The user level security is handled completely within VibrationVIEW and does not use the security table.