Databases and backups

Ken Zook

August 17, 2010

Contents

1 Database introduction

FieldWorks uses the Microsoft SQL Server 2005 Express database engine to store language project data. Versions prior to FieldWorks 4.2 used Microsoft SQL Server 2000 Desktop Engine (MSDE). The free express version of Microsoft SQL Server 2005 has some reduced capabilities (e.g., it can only make use of 1 Gb of memory). FieldWorks uses a named instance of SQL Server called SILFW. This avoids conflicts from other applications that might also use SQL Server.

The main executable files for Microsoft SQL Server 2005 Express are normally installed in the c:\Program Files\Microsoft SQL Server directory. Within this subdirectory, the SILFW instance is normally installed in MSSQL.1\MSSQL directory, however the number could vary if there are other instances of SQL Server installed by other programs. Also, if SQL Server 2005 Express was installed over an older MSDE 2000, the instance is normally in the c:\Program Files\FieldWorks\MSSQL$SILFW directory. Each database instance has a number of master database files. These are held in the data subdirectory under the instance directory. One way to tell where your version of SQL Server is located is to use Start…Programs…Microsoft SQL Server 2005…Configuration Tools…SQL Server Configuration Manager. In the right pane, right-click SQL Server (SILFW) and choose Properties. Click Advanced tab and check the Install Path value.

1.1 Database files

The actual FieldWorks projects you create are SQL Server database files stored in %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data, where %ALLUSERSPROFILE%\Application Data is c:\Documents and Settings\All Users\Application Data on Windows XP and c:\ProgramData on Vista. Each database file consists of two physical files. For the Sena 3 FieldWorks project this is

- Sena 3.mdf (the actual data), and
- Sena 3_log.ldf (the log file).

The .mdf file contains the relational tables used to store your user data, as well as system tables, stored procedures, views, and functions that are used by the program to access your data.

The log file is normally used to keep a history of every change in the database so you can roll back to an earlier version if you want to. If FieldWorks used this feature, the size of the log file could get very large. To avoid this, FieldWorks uses a simple recovery option so the log file is only used temporarily while longer transactions are taking place. Once the transaction is done, all data is in the .mdf file and the log file is freed up for the next transaction. FieldWorks also sets the databases to “auto_shrink on” in order to keep the file sizes smaller. Normally when FieldWorks backs up and moves database files both files are used, but it is possible to restore a database from just the .mdf file. When the files are zipped, the log file takes very little space.

Note: SQL Server cannot attach database files that are compressed. In some user installations, the All Users directory is compressed. If this is true, the FieldWorks 5.4 or later installer and programs will detect the compressed data directory (%ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data) installed by FieldWorks and will warn the user that they must uncompress this directory before the databases can be attached. In FieldWorks 5.4 any file in the data folder that is compressed will cause this error warning and you must manually uncompress any files that are compressed. FieldWorks 6.0 only indicates the error if a .mdf or .ldf file is compressed, and it offers an option to automatically correct the problem.

1.2 Project and database names

A FieldWorks project has an internal LanguageProject name, but because of performance problems in SQL Server 2005, we no longer use this name. Instead, the database name is simply the file name of the database (without the .mdf extension), so the characters in the project name are limited to valid file names.

As of FieldWorks 5.4 if you want to change a project name, you can

- restore a FieldWorks backup to a different name, or
- use the db program to change the name of the database.

1.3 Cautions with database files

For SQL Server to use a database, the database files must be attached to the server. When you attach them, SQL Server locks the files so you cannot move, copy, or back them up with backup software. The files are not locked when SQL Server is not running, but you still should not try to move or rename these files, because SQL Server maintains information on these files that it uses when it restarts.

The only time it is safe to copy, rename, move, or back up database files is when the database files are detached. SQL Server must be running to detach the files. The db utility discussed below provides a way to attach and detach database files, as well as to perform various other database file operations. Users should back up their databases by using the backup and restore dialog discussed below.

As a result of this locking, for off-computer backups, the user should back up the .zip files produced by a FieldWorks backup created from File…Project Management…Backup and Restore. These files are typically in My Documents\My FieldWorks\Backups.

1.4 Database file backups

SQL Server provides a standard way to back up a database. This operation generally takes a few seconds and can be done at any time, even with programs open. The result is a single .bak file that you can copy, rename, archive, etc. The .bak file combines the .mdf and .ldf files into a single file that is slightly larger than the .mdf file. A SQL Server restore option will restore this .bak file to a new database or will overwrite an existing database. It can only overwrite an existing database if that database is not in use at the time.

FieldWorks provides a dump method that dumps all the project data to a FieldWorks XML file. It also has a load method which will load a FieldWorks XML file into a SQL Server database that has all the tables, stored procedures, views, and functions, but no actual data.

2 Installing and uninstalling SQL Server

When SQL Server is installed for FieldWorks use, there are various settings that need to be made to the database in order for it to work properly with FieldWorks. As a result, you should not directly install the SQL Server 2005 Express files from the FieldWorks downloaded installation directories (in Utilities\SQL Server 2005 Express directory) or downloads directly from Microsoft. Instead, you should always us the FieldWorks master installer, setup.exe to install SQL Server. This normally happens after FieldWorks is installed. If FieldWorks is already installed but SQL Server fails to install for some reason, you can run setup.exe again and it will skip installing FieldWorks and go directly to installing the missing SQL Server. After the initial installation with special parameters, a special script is run to set some properties that FieldWorks needs for normal operation. For example, if the xp_cmdshell option is not enabled, FieldWorks will fail to migrate databases from older versions. If you ever need to check these setting manually, here's how:

1. Run…All Programs…Microsoft SQL Server 2005…Configuration Tools…SQL Server 2005 Surface Area Configuration
2. Click Surface Area Configuration for Services and Connections
3. In the dialog that comes up, select SILFW…Database Engine…Remote Connections on the left and make sure Local and remote connections and Using TCP/IP only are checked, then click OK.
4. Click Surface Area Configuration for Features
5. In the dialog that comes up, select SILFW…Database Engine…OLE Automation and then xp_cmdshell on the left, and make sure the options are checked in the right pane, then click OK.
6. Click OK to close the main configuration dialog.

When you uninstall SIL FieldWorks from Add or Remove Programs, FieldWorks project database files are detached, but not deleted. You must uninstall the Microsoft SQL Server 2005 as a separate step, if desired. When you install FieldWorks, it installs sample databases for Sena and Lela-Teli, and sets an InitMSDE registry flag to 1 in

HKEY_LOCAL_MACHINE\SOFTWARE\SIL\FieldWorks.

Whenever a FieldWorks program starts, it checks this flag, and if set, it sets up some user accounts and stored procedures in the master SQL Server files. Then it looks for any detached database files in the %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data directory and attempts to attach these databases. In a few rare cases, this process failed in some way and the databases were left unattached. The db utility can both force this database initialization process and manually attach the databases if the automatic process ever fails. This process is also called by the master installer after FieldWorks is installed.

The version of SQL Server Express delivered with FieldWorks 5.2 and later will install and run on a 64-bit Windows XP or Vista operating system.

3 Creating new FieldWorks project databases

The c:\Program Files\SIL\FieldWorks\Templates directory holds various files used for creating new FieldWorks project databases. The key files are BlankLangProj.mdf and

BlankLangProj_log.ldf. These files represent a detached database file that has all the tables, stored procedures, views, and functions, but no data in the user tables. Several XML files contain the beginning data to load into this blank project.

Basic steps when users create a new FieldWorks project:

1. Copies BlankLangProj files to the FieldWorks\data directory and renames it to the project name given by the user.
   Note: Users can type any characters in the internal project name, but file names have character limitations, so when there is a conflict, FieldWorks substitutes underscores for the invalid characters.
2. Attaches the renamed database files.
3. Loads NewLangProj.xml, SemDom.xml, and POS.xml into the database.
   Tip: These files contain the default data including various lists that FieldWorks programs expect to be in the database. The Semantic Domains and Part of Speech lists are loaded separately, making it easier for an entity to customize, if desired.
4. Makes several changes to the database to set up the writing systems specified by users.
   Note: The first time users run Flex or Data Notebook, they must choose between anthropology lists. OCM.xml and OCM-Frame.xml are two options provided by the standard installation for this list.
5. Flex and TE both load current copies of stylesheets from XML files, and TE loads other lists for key terms.

4 FieldWorks backup and restore

FieldWorks provides a backup and restore mechanism to save and restore databases from a zip file. Go to File…Project Management…Backup and Restore. You can configure automatic backups for scheduled times or use periodic reminders.

Users should *back up* their project(s) *frequently* and *copy* them off their computer to more permanent media and safe off-site locations. It is especially wise to back up databases *before* any major operation such as import and large bulk edit operations. If anything goes wrong in the database—and this is extremely rare—restoring a backup is the easiest way to recover work.

A FieldWorks backup is a zip file that contains a SQL Server .bak backup file of the database, and optionally a FieldWorks XML dump of the database. It also has a header file. The name of the file is the database file name followed by a backup time in the YYY-MM-DD HHMM form. For older backups where the internal name of the project differed from the database file name, the backup name had the internal name *first*, followed by the database file name in parentheses (e.g., Kalaba (TestLangProj) 2006-10-17 0938.zip). When restoring an older file, the current version of FieldWorks just uses the file name portion.

The Backup and Restore…Backup tab lists the user's FieldWorks databases and checks the currently open database(s). The default directory is My Documents\My FieldWorks\Backups. Users can choose another directory and it will remember the last setting. Click Start Backup, and the SQL backup will back up the checked databases, one file per database. This does not take very long. However, if users check the XML option, it also adds a FieldWorks XML dump file to the zip file and takes a lot longer. From this dialog users can also control reminders, set automatic scheduled backups, and add a password to protect their backups.

In the Backup and Restore…Restore tab, users can select a directory and a project name and it will list all the backups for that project. They can then select one and have FieldWorks restore that version. It defaults to the backup file name, but if that file already exists, it asks whether to overwrite the existing database or to restore to a different name. Users can set a checkbox to force FieldWorks to restore from the XML file if it exists, rather than from the .bak file. After restoring a file, it opens in the current FieldWorks program. During this process, if the restored file has an older version number, it migrates to the current version. This migration process is discussed later in this document.

Note: Migration only works from .bak files. XML backups from older versions may or may not restore successfully.

The restore process currently looks for a .bak file inside the zip file with the same file name (minus date/time). To avoid problems, *do not* intentionally or unintentionally change the name of a zip file (but see note below)! For example, if the backup was Kalaba (TestLangProj) 2006-10-17 0938.zip, and it is renamed to Kalaba (TestProj) 2006-10-17 0938.zip, FieldWorks looks for TestProj.bak inside the zip file. When it cannot find that file, it displays this error message: “The selected backup version contains no data or XML files. Restore has failed.” Fix it by either changing the name of the zip file or the name of the .bak file inside the zip file. This restriction may be removed in future versions.

Note: Since current versions of FieldWorks ignore the old project name portion of a backup file name, it is now possible to rename the backup file by enclosing the file name inside parenthesis and putting some other comment ahead of this. For example TestProj 2006-10-17 0938.zip could be renamed to My original version of Kalaba (TestProj) 2006-10-17 0938.zip and it would load normally.

Users can manually construct a FieldWorks backup file by zipping a .bak and/or .xml file and giving the zip file an appropriate name, including a date and time in the proper format. To get the restore to work, add a header file (can be empty) to the zip file. Although FieldWorks cannot create a backup file without a .bak file (a design decision), it *will* restore a backup file that only contains a zip file and a header.

It is easy to extract the .bak or .xml file from a FieldWorks backup file and then use the db program to restore either file, unless the XML file is an older format (see the data migration section).

4.1 Additional files

Backup and Restore only saves the database. For a complete backup, you need to consider other files outside of the database that need to be backed up separately. Here are ones to consider.

- Picture files used in the lexicon or scripture are currently in %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Pictures.
- Media files used in the lexicon are currently in %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Media.

- Language definition files in %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Languages (Most of this is stored in the database, but not quite everything (e.g., PUA definitions, related language).
- Any encoding converters. The actual files are in %ALLUSERSPROFILE%\Application Data\SIL\Repository, but these files can reference CC tables, TECkit files, and other converters that could be located anywhere, although a default location is %ALLUSERSPROFILE%\Application Data\SIL\MapsTables.
- Keyman tables you are using
- Fonts you are using.
- Settings files in %USERPROFILE%\Local Settings\Application Data\SIL\Language Explorer where %USERPROFILE%\Local Settings\Application Data is C:\Documents and Settings*\Local Settings\Application Data on Windows XP and c:\Users\*\AppData\Local on Vista where * is your Windows logon name. These files will be recreated automatically from default settings if deleted, but you could save your current settings if you want.

4.2 Subversion archiving

JAARS has provided access to a Subversion repository for fields to archive their data outside the country. If you desire to use Subversion for archiving backups, there are two possibilities.

- Store your FieldWorks backup .zip files into Subversion. Since this is a binary file, the diff process would not be of any use, and it would require uploading the entire file (5-10+Mb) each time. The only advantage here is storing a backup off site. Also, being a FieldWorks backup file, it will migrate to newer versions of FieldWorks.
- Store a FieldWorks XML dump file into Subversion. This file is quite large, but being a text file, the diff process will likely dramatically reduce the actual data that needs to be uploaded, and take less overall space on the Subversion system. For long-term archiving this is the best format, but FieldWorks does not provide automatic migrations for newer versions of FieldWorks. If you tried to restore an older version, you'd either need to load it into the version of the program from which it was originally saved, or do some special transforms to the data in order to load it into a newer version. If you decide to do this, use the following process which could be done by calling a single batch file.
  1. Use "db dump…" to dump the database to XML.
  2. Use "svn commit…" to upload the changes to Subversion.

5 DB utility

Db.exe is a command-line utility that performs many common database tasks. To run this program, open a DOS box on your FieldWorks directory and execute commands from this DOS/CMD window. One way to open a DOS window is through Start…Run, and then type "cmd" in the Run dialog. This brings up the Cmd window. The db program is installed in c:\Program Files\SIL\FieldWorks, but since this is on the path, you can run the db command from any location.

If you type "db" by itself, it will list a summary of the available commands.

When you do not specify full path names, db reads and writes files to the path specified by the DbDir registry key in

HKEY_LOCAL_MACHINE\SOFTWARE\SIL\FieldWorks.

During a normal installation this is set to %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data, so it expects to find the files in this directory unless you include a full path specification.

If database names or file names contain spaces, they need to be enclosed in double quotes.

Db is challenging to use if your filenames have non-ASCII characters. You can paste the characters into the command line using Right-click Paste.

Except for the Start command, all others assume that SQL Server is running, and will fail if it is not.

Typical uses of the db program

- Attach and detach database files (mdf/ldf or just mdf)
- When detached, move the database files to a new machine or give them a different name and then attach them
- Get a list of project databases
- Copy, rename, or delete project databases
- Backup or restore to/from SQL Server .bak files
- Start, stop, and initialize the database (as done after an install)
- Dump a database to a FieldWorks XML file, validate a FieldWorks XML file, and load a Fieldworks XML file into a project
- Execute a SQL script from a file
- Check the database version of a project
- Shrink the size of database files

5.1 Summary of DB commands

5.1.1 attach

Looks for an mdf file with the given name in DbDir and attaches it. If there is an associated .ldf file, this is used as well.

Example

db attach “Sena 3”

5.1.2 backup

Backs up the database in a SQL Server .bak format.

- If a destination file is not given, it will be backed up to the DbDir directory with the database name and a .bak extension.
- If a destination name is given without a path, it will go in the DbDir directory.
- If a destination is given with a path, the backup will be stored as specified.

Examples

db backup “Sena 3” (Produces Sena 3.bak in the DbDir directory.)

db backup “Sena 3” “Sena 3a.bak”
db backup “Sena 3” “c:\my backups\Sena 3.bak”

5.1.3 copy

Makes a copy of the given database with a new name. If you are running on Vista or Windows 7 with User Access Control on, you’ll have to run this command from a Cmd prompt that is started by right-clicking the cmd icon in the Start menu and choosing Run as Administrator. Otherwise you’ll get an access error message.

Example

db copy “Sena 3” “Sena 3a”

5.1.4 dbnames

Returns an alphabetized list of names of all attached database files in the SILFW named instance, including system files.

Example

db dbnames

5.1.5 debug

Turns debug tracing on or off. This requires a database name to execute a stored procedure in that database, but once tracing is turned on, it will log commands on any FieldWorks databases. When tracing is on, SQL writes tracing information to c:\FwDebug.trc. You can open this file in the SQL Profiler to get information on all the SQL commands executed during the time the trace was on. You *cannot* install SQL Profiler on an end-user machine, but this file can be useful to a support person who has a full version of SQL Server. You can also open the file in ZEdit in UTF-16 mode to see the queries that are being executed by FieldWorks.

Examples

db debug on “Sena 3”
db debug off “Sena 3”

Caution: Be sure you turn off debug mode when done as it can use a lot of disk space.

5.1.6 delete

Deletes the specified database, automatically deleting the mdf/ldf files.

Example

db delete “Sena 3a”

5.1.7 detach

Detaches the specified database, freeing it for tasks such as copying, renaming, deleting, and backing up.

Example

db detach “Sena 3”

5.1.8 dump

Dumps a database to a FieldWorks XML file in the DbDir directory, silently replacing any existing file. The XML file will be named the same as the database.

Example

```
db dump “Sena 3” (Produces Sena 3.xml in the DbDir directory.)
```

After dumping this file, the program will leave a file (Sena 3-Export.log) in the DbDir directory listing some statistics and any errors encountered during the dump.

5.1.9 exec

Executes all SQL queries in a given file on the specified database. If a path is not given, the SQL script *must* be in the DbDir directory. This provides a convenient way for a support person to send a file with SQL commands to fix a database and have the user execute the script. The script can have multiple batches separated with GO as in the data migration files.

Examples

```
db exec “Sena 3” AddIndonesianGlosses.sql
db exec “Sena 3” “c:\scripts\Fix problems.sql”
```

Note, the SQL script can be an ANSI file if there are no special characters in the queries. If Unicode characters are needed, the file must be UTF-16LE for the exec command to work. If you are saving the file in Word or Notepad, save as Unicode encoding, not UTF-8 or Unicode big endian.

5.1.10 fwnames

Returns an alphabetical list of names of all FieldWorks databases that are attached.

Example

```
db fwnames
```

5.1.11 initialize

- Runs the ExtInitMSDE method on DbAccess.dll.
- Copies a couple stored procedures in the master database (they should show up in db procs).
- Sets the “sa” password and creates a FWDeveloper login
- Then attempts to attach any mdf files found in the FieldWorks data directory. This is not DbDir, but is the data directory under the path specified in the RoodDir variable in registry key
  HKEY_LOCAL_MACHINE\SOFTWARE\SIL\FieldWorks.

For standard installations, this is the same directory, but may be different on developer machines. This command does the same thing that usually happens the first time a FieldWorks program is started after a FieldWorks installation. When done, the user’s databases in %ALLUSERSPROFILE%\Application Data\SIL\FieldWorks\Data should all be attached. It doesn’t hurt to run this more than once.

Example

```
db initialize
```

5.1.12 load

Loads a database from a FieldWorks XML file, silently replacing anything that was already in this database. The database name will match the file name (without the extension). The XML file must be in the DbDir directory. A load will only work if there are no connections to the database being replaced.

FieldWorks 5.0 Bug: The db program has a bug in FieldWorks 5.0 that keeps the load function from working. You'll see an error message, "Cannot load test because the template file c:\Program Files\FieldWorks\templates\BlankLangProj.mdf does not exist." This can be fixed by adding a RootDir string value to HKEY_LOCAL_MACHINE\SOFTWARE\SIL\FieldWorks. The value should be c:\Program Files\SIL\FieldWorks, or identical to the RootCodeDir string value in the same registry key.

Example

db load "Sena 3" (Loads Sena 3.xml from the DbDir directory.)

After loading this XML file, the program will leave a file (Sena 3-Import.log) in the DbDir directory listing some statistics and any errors encountered during the load. If the load process fails, this log file is often helpful in determining the reason. It is best to use the validate command before loading a suspect XML file to catch any errors ahead of time.

5.1.13 logins

Lists the current logins for the database. When Fieldworks is properly installed, the login names are sa, BUILTIN\Administrators, and FWDevelper.

Example

db logins

5.1.14 procs

Lists the name of the special FieldWorks stored procedures in the master database. When properly installed, it should list sp_GetFWDBs and sp_DbStartup.

Example

db procs

5.1.15 rename

Changes the name of a database to a new name. If the destination database already exists, it will issue an error and ignore the command. If you are running on Vista or Windows 7 with User Access Control on, you'll have to run this command from a Cmd prompt that is started by right-clicking the cmd icon in the Start menu and choosing Run as Administrator. Otherwise you'll get an access error message.

Example

db rename "Sena 3" "Sena 3a"

5.1.16 restore

Restores a database from a SQL Server .bak file, silently replacing an existing database. A restore will only work if there are no connections to the database being replaced. If a simple name is given, it will look for the given name with a .bak extension in the DbDir directory and restore to a database with that name. If a file name is given after the database name, it will use this .bak file and restore it into the specified database.

Examples

db restore “Sena 3” (Restores from Sena 3.bak)
db restore “Sena 3” “Sena backup.bak” (The .bak file must be in the DbDir directory.)
db restore “Sena 3” “C:\My Backups\Sena 3.bak”

5.1.17 shrink

Compresses the size of the database mdf/ldf files to the minimum size.

Example

db shrink “Sena 3”

5.1.18 sources

Writes out the SQL sources for stored procedures, functions, triggers, views, and constraints.

- If a destination file is not given, it will be saved to the DbDir directory using the database name with “Sources.txt” appended to it.
- If a destination name is given without a path, it will go in the DbDir directory.
- If a destination is given with a path, the file will be saved as specified.

Examples

db sources “Sena 3” (Produces Sena 3 Sources.txt in the DbDir directory.)
db sources “Sena 3” “Sena 3a.txt” (Stored in DbDir directory.)
db sources “Sena 3” “c:\my backups\Sena 3.txt”

5.1.19 start

Starts SQL Server if it is paused or stopped; otherwise it is ignored. It may take a few seconds to start.

Example

db start

5.1.20 stop

Closes any open connections and stops SQL Server. It may take a few seconds to complete. If any FieldWorks programs are open at the time, they will crash the next time they try to access the database.

Example

db stop

5.1.21 structure

Writes out the contents of Class$, Field$, ClassPar$, and then dumps field, key, and constraint information for each user table.

- If a destination file is not given, it will be saved to the DbDir directory using the database name with “Structure.txt” appended to it.
- If a destination name is given without a path, it will go in the DbDir directory.

- If a destination is given with a path, the file will be saved as specified.

Examples

db structure “Sena 3” (Produces Sena 3 Structure.txt in the DbDir directory.)
db structure “Sena 3” “Sena 3a.txt” (Stored in DbDir directory.)
db structure “Sena 3” “c:\my backups\Sena 3.txt”

5.1.22 tune

Turns tune tracing on or off. This requires a database name to execute a stored procedure in that database, but once tracing is turned on, it will log commands on any FieldWorks databases. When tracing is on, SQL writes verbose tracing information to c:\FwPerform.trc. You can open this file in the SQL Profiler to get information on all the SQL commands executed during the time the trace was on. You *cannot* install SQL Profiler on an end-user machine, but this file can be useful to a support person who has a full version of SQL Server. You can also open the file in ZEdit in UTF-16 mode to see the queries that are being executed by FieldWorks.

Examples

db tune on “Sena 3”
db tune off “Sena 3”

Caution: Be sure you turn off debug mode when done as it can use a lot of disk space.

5.1.23 version

Returns the current version of the FieldWorks database (see Data Migration section for details).

Example

db version “Sena 3”

5.1.24 validate

Validates a FieldWorks XML dump file based on FwDatabase.dtd. The XML file and FwDatabase.dtd must *both* be in DbDir. The output goes to temp.err in the DbDir directory. If no errors are detected, the file has a length of 0. The validation process catches many types of problems, but not everything. The actual load process may catch additional errors, so always check the log file after the load command. If you make any changes to a FieldWorks XML file and plan to load it into a database, run this validation first. It will catch many kinds of errors before trying to load the data. If there are any errors, temp.err lists the problems, giving the line number in the XML file where the problem is located.

Example

db validate “Sena 3” (Validates Sena 3.xml in DbDir.)

6 Data migration

During ongoing FieldWorks development the conceptual model is sometimes changed. Sometimes lists are updated to new versions such as Ron Moe’s semantic domains. The FieldWorks database contains stored procedures used by programs to access and manipulate the database. As programs change, sometimes these stored procedures change or may require new ones. All these things are stored in the Fieldworks project database with end-user data.

FieldWorks programs need to stay in sync with the stored procedures, tables, and database data. To accomplish this, each database has a version number (DbVer in the Version$ table). FieldWorks programs record the version of the database they expect.

User databases need to be current so they work with FieldWorks programs as they evolve. When developers change anything that makes existing databases invalid, the database version number is incremented and a SQL script (and possibly additional code) is written to migrate user databases from the previous version to the new version. These SQL scripts are stored in the c:\Program Files\SIL\FieldWorks\DataMigration directory along with occasional XML files that get loaded during various steps.

Every time a FieldWorks project opens, the program checks the version of the database to make sure it matches the version it needs. If the versions match, the program continues normally. If the database version is higher than the program, the program is unable to open that database, so an appropriate error message is given. If the database version is lower than the program, the program automatically migrates the database to the current version prior to opening the project. The migration process

1. Backs up the project file to a .bak file in the FieldWorks data directory and appends the old version number to the file name (e.g., Sena 3-v200129.bak)
2. Finds a file in the DataMigration directory to migrate the data to the next version (e.g., 200129To200130.sql) and executes the script on the current database. A few migration steps require loading xml files or doing something outside of SQL scripts. It increments the database version and updates the progress indicator to show the next version number it is working from.
3. Loops back to step 2 and continues migrating the data if the incremented database number has not reached the program version number.
4. Opens the database as usual when the versions match.

When migration is complete, the old backup file stays in the data directory. If you want to go back to that version, you can use the db program to restore the original database

```
db restore “Sena 3” “Sena 3-v200129.bak”
```

If the migration process fails for some reason, the program restores the original database and issues an error message indicating the failure. If this ever happens, contact LSDev for help.

Most migration steps are quick and take a few seconds per migration. A few may take 5 minutes or more, such as updating the semantic domain list. During long migration steps, the progress bar does not move, so just wait patiently until it is done.

Data migration only works with .bak files. Although XML is good for long-term backup and is safe from changing file formats, there is currently no plan to provide XML data migrations.

7 Version issues

Because of installation complexities, it is very difficult to have more than *one* version of FieldWorks on one computer. If you *must* do this for some reason, here are the options:

A. Uninstall one version and install the other version whenever you need to switch.
B. Have separate Fieldworks directories and re-register *all* the dlls and change registry settings when switching.
Because of the FieldWorks version situation, users *cannot* go back to an older version after opening a database on a newer version. The *only* option is to go back to an older backup prior to migration and lose any work done after that backup.

You may install a newer version and migrate your data, then decide to return to an older version. You should *always* back up your database regularly! If you did not make a backup, you can use the renamed backup prior to migration.

Another version difficulty is if you prepare a database for someone using an older version. After you work on the database with a current version, you *cannot* give your backup to them unless they *also* upgrade to the same version you used.

If you maintain older database versions on your computer, you can accidentally migrate them if you open them in a newer program. You can get back to the earlier version by restoring the renamed backup file.

If you know what changes took place in database versions between your version and an older version, you may be able to back up your data as an XML file and have the person with the older version restore from the XML. Many times migrations *only* affect stored procedures or add new classes you did not use. In these cases, your XML dump file is identical to the dump file in an older version. The database version number is stored in a FieldWorks XML dump file, but other than logging differences in the log file, the load process will always load the data into the current database version.

8 Summary

SQL Server is an extremely robust way to store data. It recovers gracefully if power fails and also recovers well from program crashes. Unless users understand these topics, they should *not* touch the mdf/ldf files in their FieldWorks\Data directory. They can freely make backups and restore from the backups as needed. The FieldWorks UI *only* provides the option to back up and restore via the FieldWorks zip file. The db program provides other useful alternatives to those who know how to use it.

Here are relative file sizes and back up times for a medium size FieldWorks project (11,000 lexical entries, 1,000 wordforms, 6 texts, and a few Scripture or data notebook entries) in their different forms.

Yi.mdf	60.9 MB
Yi_log.ldf	10.2 MB
Yi.bak (using db backup)	61.0 MB (8 sec; 21 sec to restore)
Yi 2006-10-17 0938.zip	17.9 MB (65 sec)
Yi 2006-10-17 0938.zip with XML	22.4 MB (324 sec)
Yi.xml (using db dump)	31.9 MB (302 sec; 558 sec to load)
Yi.zip (Yi.xml zipped)	4.6 MB (302 + 7 sec)

Times always depend on a lot of things. These times resulted from

- a 2.8 GHz machine
- with 512 MB memory

- with 27 GB free disk space, and
- without recent defragging.

You can choose to use whatever version best meets your time and space criteria.

For long-term archiving, an XML version is highly recommended because it is a standardized text format that will be readable far into the future. It also works best with a Subversion repository. For shorter-term backups, the .bak format is preferable because it takes much less time, and because FieldWorks developers are committed to migrating this form to newer versions of FieldWorks. At this point, FieldWorks will not migrate XML files when the conceptual model changes.