Establishing Multiple Projects with Independent Libraries

by Dick Cleaveland

The design of Ultimate Family Tree presupposes that there is significant value in having a single library of place, source and media data supporting multiple projects. For most of us this makes sense in that it removes the necessity of entering and maintaining the same data in different places. There are some circumstances, however, where it would be useful to have projects totally divorced from each other.

One such case is that of the professional genealogist who would like to treat his or her customer-related projects independently, packaging and filing away each customer's data at the end of a task, without the residue (library) cluttering up other client's data bases. Another case for many of us is that which arises when we receive a GEDCOM file and would like to inspect it without taking the chance that it would clutter up ("contaminate" might be a better word) our data base.

Given the design of UFT, there are several ways of handing these circumstances. I shall describe a few. All of these methods assume a version of UFT up to and including version 2.9, distributed (only, so far) with the "Platinum" package. There are hopes (only, at this point) that future versions of UFT may internally accommodate these desires better.

A warning about choosing to use independent libraries is in order: if you have separate libraries, it will not be possible to use the UFT-provided project merge capability. The only way to merge two projects having separate libraries is to GEDCOM one of them out and import the GEDCOM into the other. Unless place and source data (for the same places and sources) is entered identically in the two projects there is liable to be a resultant need to combine them after the import. The same holds true for individuals which might appear in both projects.

The techniques described here are for establishing new projects with their independent libraries. A technique to split an existing project off from the common library will be the subject of a future article.

Multiple installations of UFT: The simplest way to handle the problem is to make a new, separate installation for each project you wish to have. When the installation routine asks you to identify the drive and folder for installation, simply give it something different than the default C:\UFT. For example, call the second one C:\Jones if you want to put the Jones project there. If you have more than one hard drive, you can use it too: D:\Smith, for instance. Then you can get the project you want, with its independent library, simply by starting the correct copy of UFT. This has the disadvantage, of course, that the program files not specifically related to the project are replicated for each installation, consuming space on the hard drive. In past years this extra space requirement has been more important than it is these days, with hard drives getting larger and cheaper, and so this technique may come into more frequent use since it is by far the simplest.

Ping-Pong using backups: The next simplest way is to have your two or more projects operating out of the same installation of UFT, but swapped back and forth via backups. To get started for this you need a "skeleton" data base, consisting of just one dummy individual and only the initial library that UFT provides upon initial installation. (I shall use the term "folder" rather than "directory" in deference to Windows standards, although it sure takes a bit of retraining on my part!) Make a fresh UFT installation, opening a new project and naming the project "Skel".

Put in a dummy individual and make a backup, being sure to include the library and the support files. Call the backup "skeleton.sqz". Then start your first project using File - New, let's say "Jones". After you've filled in the project info form make a backup of the Jones project (call it jones.sqz). Now start the second project using File - New ("Smith"), fill in it's project info and back it up as "smith.sqz". At this point you can reopen and delete the Skel project if you wish. Now you can restore either project, work on it, back it up and switch to the other project as desired. To add a third project you simply need to start by restoring the skeleton.sqz and proceed as you did with Jones and Smith.

This process has the disadvantage of significant time delays in switching from one project to another, since a backup and restore process is involved for each switch, but has the advantage of having only one installation of the program and it's non-data-related files.

Multiple installations of UFT with program material removed: This is a variation on the first scheme mentioned above, where all but one of the installations is crippled by removing data-independent program files to save space. This takes a bit of detailed knowledge of the structure of UFT files, and is not for the faint-of heart. Prime candidates for removal are the .exe and .dll files, the accessory folders such as GAZ, rr, and SSDI, and those in the sub-folders of the image sub-folder (BACKGND, BARS, BULLETS and ICONS).

One must start a project in each of the installations before removing the extraneous material. Henceforth all that is required to access it is to start the remaining complete UFT version and use the File - Open menu item to navigate to the proper project by changing folders and/or drives as needed.

This has the advantage over the first method of requiring less disk storage space, but the risks associated with monkeying around with files in the various folders.

Using a skeleton project backup on either a real or simulated second hard drive: This next technique makes use of a second hard drive - but wait, you can use a simulated second hard drive if you don't already have a handy real second hard drive.

To create a simulated second hard drive is really very easy in any Windows system (can't say about NT or OS2). First, create a folder on one of your drives (I'll assume C) and call it, for instance, "DriveX". Next, using Notepad, edit your C:\AUTOEXEC.BAT file and add the following instruction to the end:

```
SUBST X: C:\DriveX
```

This tells your system that you’d like to have the folder C:\DriveX treated just like it was a hard drive. Reboot your system at this point; after you reboot, you’ll see that you have a drive X on your system.

Now install a fresh copy of UFT onto drive X, using the default folder name of UFT. Add one dummy individual, and then back up that project as Skeleton.sqz. Now delete the entire new folder X:\UFT.

Establishing a new project: Using your regular UFT installation, open the Skeleton.sqz backup, and when it says "Please select which drive this backup will be restored to:" select X. While reading the backup UFT will re-create the UFT folder on the X drive. You will find your dummy individual there and if you look at the location of your files you will find that the library and project files are all in the folder X:\UFT. Now use File - Save As to create a new project with the name you want for this new project, and after you have filled out the project information form use File - Open to reopen the Skel project and delete it. You now have a project in X:\UFT which has it’s own library and no extraneous files in that folder.

Now rename the X:\UFT folder to something else even the name of the project (Jones, for instance). Then (this sounds silly, but is needed) close out of all your programs, exit Windows and reboot your machine. After it is rebooted, use your regular UFT program to open the project you have just created by using File Open and navigating to X:\Jones\Projects\Jones. There you have it.

But you want more than one, how to manage that? Easy. Just redo the sequence of steps from "establishing a new project" to this point. You can repeat this process as often as the space on your hard drive will allow.

This technique is more complex than the others, but has some advantages. Switching between projects is rapid (use the project list at the bottom of the File menu) and you have only one installation of the UFT program itself. Aside from that, if you are involved in using more than one machine, and if these machines have different disk drive arrangements, transfers of your data between machines using backups is quite uncomplicated assuming you have defined a simulated disk drive X on both machines.

I’ve been using "X" as the identifying letter of the fake drive in this article but you may use anything you want which will not duplicate any drive on any of the machines you might want to use. Or of course if you have one you can use a real second hard drive.

In closing, a warning and admonition: I’ve spent a good deal of time helping RUG members who have managed to get into trouble with their UFT systems. The vast majority of the problems I have seen were related to the user being unaware of the location of their project and library files. Please, if you elect to use one of these techniques, and especially if you decide to be creative with some variation, either look at Help About... - Project info and the tabs for project and library files or print the report under Reports - Summary - Project with the File List option selected to see where your files actually are.

The admonition is to consider seriously whether you really want separate libraries. The advantages to a single library are significant, and "working around" the basic design of UFT is fraught with a hidden danger: a future version of UFT could come along which would make upgrading much more difficult for those who have been creative in stepping outside of the bounds of what the system designers had in mind.