

Applying Previous Updates After Merging Additional Data.

Modifying separate copies of the same database will always causes problems when these databases are merged. The merge functions of K9-Ped do a good job of letting you know when there is different data for any particular dog but can not prevent reintroducing previously deleted dogs or a dog with a previously corrected spelling.

When you merge a database that contains a dog you have deleted that name is added to your database just like any other new entry. If you have changed a name then the old spelling has been "deleted" and the new spelling is "added". Merging a database still containing a dog with the old spelling will result in the "deleted" name being added again as a new dog.

The latest version of K9-Ped now keeps track of changes and deletions in two files, NameChng.db and Obchange.db. These files only keep the last action for any particular name. Deleting a name will overwrite a previous change. Adding a name back in will remove any previous deletion or name change.

These files can be applied to any "breed" database and will reinstate your changes and deletions.

Please be aware that there is no perfect way to combine two separately updated databases. For example, someone adds a new dog to their database and uses a dog you have deleted as the sire. Their new dog will be added during a merge but, since the sire will be deleted by applying your previous actions, the link to the sire name will either missing or the old spelling of the sire will be added as a new dog.

To avoid this problem it is recommended that two spellings of the same dog always be corrected by changing the wrong spelling to the right one, even if their is no difference in any of the data and the wrong spelling has no progeny. By changing the name rather than just deleting the wrong one you will then provide a name change link correction for the above example.

Using the "Apply Previous Updates" Feature

Since the NameChng.db and Obchange.db are new files they will not contain any old information. K9-Ped has been keeping track of your deletions and recent changes. There is a feature to capture some of your previous actions. Since the program did not overwrite this information with later inserts the results may not be 100% accurate. It should, however, give you a good start, on producing the "changes" database files.

Go to "Tools | Database Tools | Apply Previous Updates". Click the "View Name Change Database" button. This will tell the program what directory you want to use for the NameChng.db and Obchange db. Select the correct directory, click on the file name visible and then "Open". Click the "Add Data From Changes List" button. Don't worry that you only see one file. This is also just to select the directory for the changes.txt and deletelist.txt files. After you select the directory and click the visible file name, click "Open". This will load the "changes" files with the data from these files. In both cases you will most likely want to use the default, current breed directory. This will be referred to as the "original" directory in the next step. You only have to do this step once as the program will record all future changes and deletions.

The next step is merging the databases. When doing this type of merge, either with the restore function or the Share Data function, it is a good idea to do a trial run with your existing database isolated from unintended errors and deletions. The easiest way to do this is to make a backup of your existing database, add a new "breed" and restore your backup to this new breed.

Using this new "breed", restore the backup file to be merged or Read the Shared Data file. If the merged data contains previously deleted dogs or wrong spellings you have since corrected they will appear in this combined database.

Go to "Tools | Database Tools | Apply Previous Updates".

Click on "View Name Change Database". Select the ORIGINAL directory (Your normal breed directory, the directory you used in the first step.) Click on the file name and then "Open". The contents of NameChng.db will be displayed. If you have previously deleted and then re-entered dogs you will want to find these dogs' names and delete the entries by selecting the incorrect item and clicking the box with the minus sign.

Select the type of data merge to use. This will depend on the confidence level you have for the merged data. If you suspect that there is a combination of corrections you will want to use the "Prompt on Differences" selection.

When you have finished removing unwanted deletions click the "Apply Name Changes" button. This will remove previously deleted dogs and reinstate previous name changes. If you selected "Prompt on Differences" you will be asked to resolve these differences during the process.

Again, there is no perfect way to merge two databases that have been separately modified. Even diligently changing incorrect dog names rather than deleting them does not address all of the potential problems. Consider the case of the shared database having the wrong call name. You delete the name because it is incorrect but don't know the correct one. Person "A" makes a change in a different field so the dog's record is marked as updated but does not change the call name. Person "B" changes the call name to the correct value.

Merging A's database will add the call name you deleted as the merge function will always overwrites blank data fields and A did not delete it. You notice the change and delete the call name. Then you merge B's database and the correct call name is added. If the program had tracked your deletions on a field by field basis and applied these changes after each merge it would delete the incorrect call name added when A's database was merged but would also delete the correct one added by B's database.

This is just one example of the problem. I hope it explains why no program can correctly handle merging two databases. Even a system with a single database is subject to similar problems. You send an address change request to a company. After sending it you realize that the new zip code you provided is incorrect and immediately send another request with the correct zip code. The change requests are received and they enter the changes into their single, master database. Unfortunately your second request gets processed before your first one. You see the problem.

Even with the unavoidable problems this new feature should save a lot of time that would be spent repeatedly deleting the same names after every merge and making the same changes over and over.