

# Tagger

from



Disclaimer: I have tested this software, and TRIED to make it crash. I couldn't. That does not mean it's foolproof. There's no way to predict or test all the possible hardware and software configurations on Windows machines.

Nor can I guarantee that you won't make mistakes. You should always make backup copies of important files before making changes to them.

Use of this software is entirely at your own risk.

The Tagger software and accompanying documentation is  
© Copyright 2004, Paul Myers and TalkBiz, Inc.

## Introduction

Tagger is a very simple program for computers using the Windows operating system. It's designed to add unique identifiers and extra fields to databases. For the purposes of this program, a database can be any text file consisting of one or more lines.

Tagger works on text databases only. While this may initially seem to eliminate its use for Access, Approach, FoxPro and other formats, this limitation is easily gotten around.

Most users of the program will need it initially only for email lists. These are normally text files which consist of, at most, a first and last name and an email address. Regardless of the format of the database, you'll need to understand...

## DataBasics

What Is A Text Database?

A text database is a very uncomplicated thing. It consists of a text file in which each line contains fields. If more than one field exists on each line, they are separated by delimiters. An example:

```
First Last Email
Paul Myers paul@talkbiz.com
Freddie Farkel farkel@example.com
```

This is a tab-delimited database, with two entries. Each entry has three fields: First, Last and Email. These are called by the unoriginal name "Field Names." The Field Names, and the fields themselves, are separated by tabs, one between each pair of fields.

Simple.

Any character can be used as a delimiter, and in most cases, the delimiter exists only between fields, not at the beginning or end of lines. There's one very common exception to this formatting rule. The quoted comma separated database. The above example, in quoted comma-separated format, would look like this:

```
"First","Last","Email"
"Paul","Myers","paul@talkbiz.com"
"Freddie","Farkel","farkel@example.com"
```

The same data can be used in a comma separated list without the quotes. Enclosing fields with quotes allows commas to be used in the fields. Tab-delimited databases can contain any punctuation within fields other than tabs. This is another reason I recommend using the tab-delimited format.

If you're going to use this software for testing copy in emails, that allows you to worry about your writing, rather than eliminating normal punctuation.

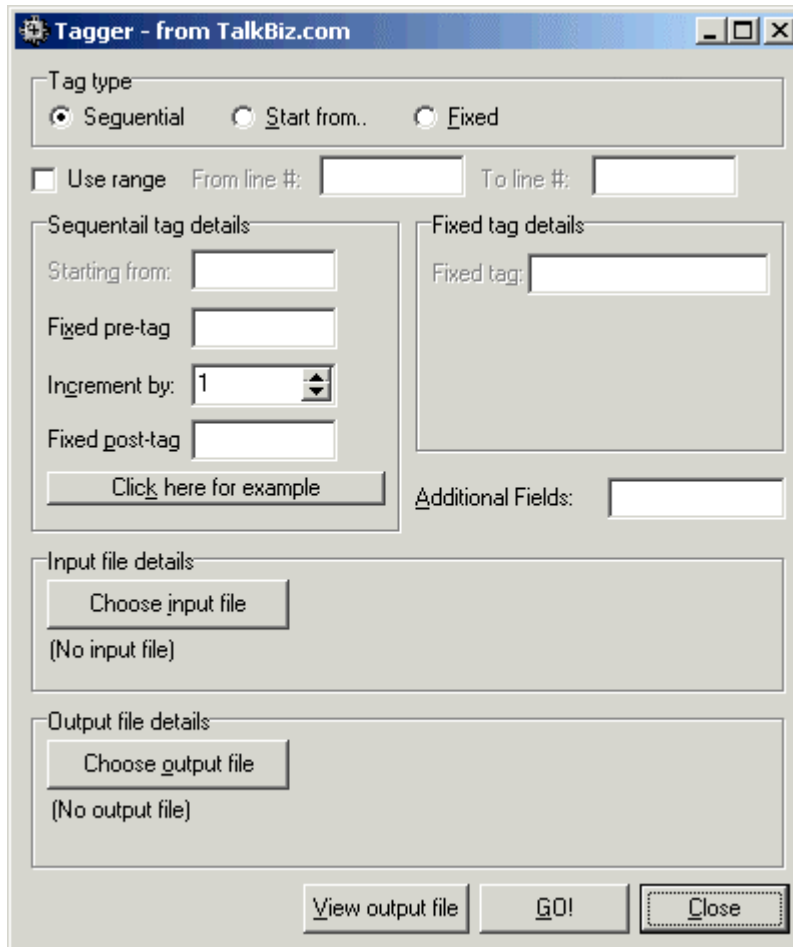
You have just learned virtually everything you need to know about text databases to use this program. After this, it's a matter of counting spaces and fields, and entering the info into the (very simple) program interface.

The only other thing you need to do before getting to work is prep your database file. To do this, cut the top line, which includes the Field Names, and save it to another file. The reason for this is explained in the next section.

Now you're ready to start...

## Using The Program

The program has exactly one screen, which looks like this:



You'll note that certain options are ghosted out (unavailable), depending on which type of tag you want to add to your database. The three types of tags are:

**Sequential:** In this mode, the program will begin the sequence of numbers for the tags at 1, and increase each succeeding entry by whatever value you choose in the "**Increment by**" option. I recommend leaving this value at one, but you can choose any number you want.

**Start From:** In this mode, you enter the numeric part of the highest ID code already in the database, and the program will start the numbering of new entries one higher than that number. For example, if the highest numbered code is 2500,

you would enter that in the “**Starting from**” field, and the first entry in the new additions would be 2501.

**Fixed:** In this mode, there are no numeric IDs added. This is for adding identical fields to all new entries in the database. Handy for a lot of purposes which you’ll see as you use the program.

## Tag Details

In each mode, there are different options for using the Tag Details.

The numbering differences are explained above. In both **Sequential** and **Start From** modes, you can use the **Fixed pre-tag**, **Fixed post-tag**, and **Additional Fields** options.

**Fixed pre-tag:** This is where you enter whatever you wish to be inserted between the end of each line that currently exists in your database and the new fields you’re adding.

If you are adding fields to a comma delimited database, this needs to start with:

”  
,

to indicate that a new field is starting. If you also wish to have some text added before the numeric portion of the unique ID, you would include it in this area, after the comma and quotation mark.

As an example, let’s say you’re adding a list of email addresses you got from the ListBuilder pay-per-subscriber service. You might want to be able to identify the source of the subscriber later, in which case you’d make the entry in the **Fixed pre-tag** section

,”LB

as shown in the screen shot above.

If the database is tab delimited, you need to tell the program to insert a tab character before any new fields. Because the program allows you to navigate between areas by using the Tab key, you’ll need to use a special sequence to indicate anywhere that you want to include a tab character in the output. To do this, use the sequence:

^T

That's a carat (shift 6) and the letter T. Anywhere this combination appears in any of the fields, the software will output an actual Tab character. So, the above example would look like:

```
^TLB
```

For most email address lists, which have only one field per line, this is the easiest way to handle the addition of new fields.

There will be cases in which you need to add a number of fields BEFORE the unique ID. This will be most common when working with existing databases. To do this, simply add the appropriate fields, as many as needed, in the **Fixed pre-tag** section.

Let's say you wanted to add three blank fields before the unique ID field. In a comma delimited file, that would look like (assuming you wanted the LB that was used in the above example to be part of the numeric ID):

```
","","",""LB
```

For a tab-delimited database, it would be:

```
^T^T^T^TLB
```

This is much simpler than it looks, when you actually sit down and do it. Honest.

**Increment by:** This is just the number you want the numeric ID to increase by for each entry in the database. I recommend leaving it at 1.

**Fixed post-tag:** This is useful for adding whatever you wish AFTER the numeric ID. If you're adding one field to a tab-delimited database, this should be left blank. If you're adding fields to a comma delimited database, it should contain a closing " (quotation mark.)

You might want to add other data to this section. For example, you might want to add something to indicate that the new entries are people who have purchased from you, perhaps the letter C. Or you might include a code for the date they were added to the database.

Whatever you add, remember that for comma-delimited databases, the closing quotation mark must be the last character in this area.

**Tip:** If you are going to be adding the same fields to multiple lists (for example, regular batches of new subscribers), you may want to save the entries for each field in the program in a small text file. Then you can just copy and paste them in

where they belong later, changing only the **Start from** value as needed. You can add as many fields as you need in both the **Pre-tag** and **Post-tag** sections.

**Click here for example:** Clicking on this button will pop up a small screen that will show you how the combination of **Pre-tag**, **Increment by** and **Post-tag** sections will look when put together. Note that ^T will not show as Tabs in this display. Don't worry. The ^T combinations will be rendered properly in the final output.

Note also that any fields included in the **Additional fields** area won't show in this view. They will be properly rendered in the final output.

**Additional fields:** This area of the program is for adding (aww shucks, you guessed) additional fields. While not strictly necessary, it provides a different way to organize the new fields for those who may work better with this sort of layout. The same rules apply for fields added in this area as for the others.

This area works in all three modes. That is, no matter what type of tags you're adding, entries in the **Additional fields** area will be added to the final output.

**Choose Input File:** This is the text file you want to add fields to. Uses the standard Windows file selector.

**Note:** I recommend that you always work with a backup, rather than the original file.

**Choose Output File:** This is the file you want the database with the new fields added to be written to. If the file doesn't exist, just navigate to the directory in which you want it created, and type in the name you want it to have. The program will create the file when it's done adding fields.

**Note:** DO NOT make the output file and input file the same. If you make a mistake and alter the original file, correcting it will be a pain in the... Well, it won't be fun.

**GO!:** When you click on this, the program will add the tags and write the output file. In most cases, this will be a very fast process. Adding unique sequential IDs and a number of other fields to a list of 27,000 addresses takes less than 10 seconds on my machine. Very large files on very slow computers may take substantially longer.

When the process is done, a small alert box will pop up saying "Tagging complete."

**View Output File:** Clicking on this button after you've added the required fields will open the new file in your default text editor. Look carefully to make sure the fields are set up the way you want them, and the delimiters are all correct.

If they are, you're done. If not, adjust the settings to correct any mistakes in the layout and click on **GO!** again, without changing the input and output file names. The program will overwrite the output file with the corrected version. View the output file again, to make sure it's right, and then you're done.

**Close:** Shuts down the program. 'Nuff said.

## Finishing The Database

As mentioned in the beginning of these docs, you needed to remove the first line of the text database, which contains the Field Names, before adding fields.

Now you need to modify that line, to add the appropriate titles for the fields. For example, let's say you started with a comma-delimited database, with this as the first line...

```
"First","Last","Email"
```

...and added fields for Unique ID, Source, Date, and Publication. The new first line would look like this:

```
"First","Last","Email","Unique ID","Source","Date","Publication"
```

If it were tab-delimited, the same line would look like this:

```
First  Last  Email Unique ID  Source      Date  Publication
```

Now paste that in as the FIRST line in the modified file, and save it to disk. Your new database is ready to use. You can use it as-is for programs that handle text databases, or you can import it into Access, Approach, FoxPro, or any other database program that supports text delimited formats. All the good ones do.

## What About My Existing Databases?

Excellent question!

And yes, I have an excellent answer. You CAN use this program with Access, Approach, FoxPro, and other database formats. You just need to do a very small extra bit of work.



You'll need to export your existing database into a text format. Virtually all database programs will export to at least tab-delimited and comma separated formats, which are both pure text files, as described above.

You then add the fields you want, (along with adding the appropriate field name) and import the new database back into the program.

An easier way to do this is to open the database in your main database software, and then open a text editor, with Word Wrap turned off in the editor. Click in the screen of the database software that shows the full list of entries, and select Highlight All. Holding down the Control key and clicking A will usually do the same thing. Then select Copy, or hold down the Control key and tap the C key once.

Now click in your text editor, hold down the Control key, and tap V once. This will paste the database into the editor, in tab-delimited format.

Now save the file to disk, and make the changes just as described in the section "Using The Program."

**NOTE:** I do not provide support for the various database programs in use. Explanations of the functions described here for them can be found in the documentation for your specific database software.

As always, I ***strongly recommend*** making backup copies of your databases before making any changes to them. This ensures that correcting minor mistakes doesn't become a time consuming (and expensive) project.

With the use of these small safety precautions, this program should prove a significant time saver, and help you to better track your promotional efforts. And testing, as they say, is the mother of success...

Your comments and questions are welcome.

Paul Myers  
TalkBiz News  
[paul@talkbiz.com](mailto:paul@talkbiz.com)  
(888)576-1755