

Some basic documentation for TAMS Analyzer v. .6.5

TAMS Analyzer (TA) is basically a coding and code extraction program which uses the TAMS system described elsewhere. With TA a researcher can code and search across multiple documents and save the output in a format compatible with spreadsheets and database programs. There is a standalone, command line version of the extraction part of the software that is available for both Unix and Macintosh. That version is flexible but less interactive than TA. If the TAMS system meets your needs but you feel like there are some output options such as whether to output headers or not, whether to output record numbers or not, or whether to add a field for the paragraph number of the found code, you may want to turn to the command line version for extraction.

This documentation basically explains the three windows that users need to juggle in TA.

I. Work bench window

The user is given exactly one work bench to use. The work bench is for the management of multiple files. There are three things that the work bench can be used for.

A. *Establishing a centralized code file*

Normally in TA the code list is generated from the list of codes used already in the file. Usually, however, the list of project codes should be kept separately. The file that contains the codes that all other documents should draw from is here called the code source. Make sure that you have opened your code source, it should be listed in the files list on the work bench. Select it and click code source. To turn off centralized coding click the check box towards the bottom of the work bench. Once you have set up centralized coding the code lists on all open windows reflects the codes in this code source, not in their own document. Clicking the New button adds the code to the bottom of the code source file as well as applying it to the active window. Note that the list of codes on the work bench is only available if there is a central code source.

B. *Searching across multiple files*

By using the Add, Add all, Remove, and Remove all buttons the user can move files from the file list on the left side of the work bench to the search list on the right side of the work bench. This is the list of files that tams will use

to search if the work bench search button is used. If there is a centralized code list the user can enter a code to search just by double clicking the name of the code in the code list window. You can enter the codes to be searched for in the search field, clear that string with the clear button, and control the output by using the three check boxes provided:

1. *raw: the output shows the codes embedded in the text*
2. *exact: a search for a string “mystring” will have to match it exactly. Subcodes will not be returned (e.g., mystring>subcode)*
3. *empty: a blank record marked “*EMPTY*” will be produced for every {!END} that does not have any of the matching codes.*
4. *simple: simple is a search that is between a limited and unlimited search. In essence it does an unlimited search on a single code name (including family). Limited search results cannot be recoded so if you need to recode chunks of data, you will need to learn how to do simple searches: see the section on search types below.*

C. Establishing an Init file

The init file is simply the first file searched for in a multi file search. This is a file that might have a list of universal and repeat codes that hold over all of the documents. It might also have meta codes to trigger various functions, though most of these have been disabled other than the three check boxes that appear in the search window. In v. .6.5 the clean flag can be turned off through the {!dirty} metatag as well.

II. Data entry or document window

This window actually has your raw data file, which might be an interview or documents or observations. On the left hand side of the window is a list of existing codes which might come from the document or from a central code source. Which is used depends on whether you have set-up a central code source using the work bench (see § I.A). The general way to code data, using a code that already exists is to select a passage from your file and double click the name of the code. Alternatively, you can click the name of the code one time, the name of the code will then appear in the box above the code list; then click the code button. If you want to add a new code, type it in the box above the code list and click the new button. If you want to create a new code which is a family member of an existing code (you want to add a code hot>very>water when there is already a hot>very code) you can click the existing family button (hot>very) one time, and then add your extension (>water) and click new. When you click new you will be

prompted for a definition of the new code. If you are using centralized coding this is added as text coded by the new code at the bottom of the centralized coding file. If you are not using centralized coding, this definition will be put in as a comment for the passage you are coding.

Other features of note in document windows:

A. *Bookmarks*

Often it is helpful to mark a place in the file. I use this mostly to note where the last spot I worked on in the file before quitting. Insert into the file the following metatag, not available in the command line processor, {!bookmark name}. A typical bookmark might be {!bookmark Wed. night}. You can then pick “reload” from the Bookmark submenu (on the TAMS menu) and it will go through and locate all of the bookmarks. A list of the book marks will then appear underneath the “reload.” Picking a bookmark will scroll the document to that point and select the bookmark. When a file is opened, this list of bookmarks is automatically compiled. Adding additional bookmarks, however will not appear unless you pick reload.

B. *Diagnostics*

If a search seems to do nothing, there is probably a mistake in the codes. Two typical mistakes involve not having a proper close for an open tag (a {a} but no {/a}) or nested tags which mean some of the coded text will not be found: {a} this {a} is {/a} a problem {/a}. A search of the code “a” will only pull up to the first end tag in this last sentence. The words “a problem” will not be returned. There are syntax checkers for both of these problems under the TAMS menu. Picking these items will scan the document and select problem tags. Run repeatedly to find all the potentially dangerous points. If no problems are found the current selection will not change.

C. *Code list integration*

Often it is helpful to generate a list of all the codes in a file. Pick “Generate code list” from the TAMS menu and a list of the existing codes (in tag format) will be appended at the end of the file. You could then copy and paste these where you need them.

More often you might want to see if you have created any new tags that don’t appear in a centralized code file. Perhaps you went off coding and forgot to use the Workbench to indicate the central code file, or you hand coded, i.e., actually typed in the new codes without using the buttons like new and code (or double clicking). No problem, indicate which file is your central code file

(using the Workbench: see I.A) and then pick “Integrate with central code list” from the TAMS menu. This will append any new codes to the tams menu with a date stamp.

D. *Moving through the document from code to code*

Selecting codes can be tricky (for deleting them, for instance), so to help you, you can move through the document selecting codes through the document with the “Find current code” and “Find next code” options.

E. *Removing codes from part of the document*

While you could step through a section of the document using the features described in II.D., sometimes you may just want to remove all the codes in one section. After selecting a portion of the text pick “Remove codes from selection” from the TAMS menu, and all codes, end codes and metacodes will be removed from that portion of the document.

F. *Searching*

Of course, data entry is only half the ethnographic fun, the other part involves analysis. To extract data click on the search tab and a whole new set of options opens up. Codes are listed on the lower left, and to search for a particular code double click on the code name. Subsequent double clicks will append additional names to what is in the “search for” form field. To start over you can hit “Clear search” and the search form field will be emptied. You can type in connective and other information as well: ‘+’ for a logical and and ‘,’ for logical or. There are three prefixes to a code that you can use as well:

1. * = substring. Searching for *hat will find passages that have any code with the letter sequence h-a-t in it including codes such as “what”, “madhatter”, and “chat”
2. > = any level. The phrase represents a code name at any level of the total code name: >hat will find hat>top>level and top>hat>level and top>level>hat. This is not a substring so it will not find text that is coded only by top>level>what. The actual level name needs to match exactly (no, prefixes can’t be combined)
3. ' = exact. Normally searching for a code returns the entire family tree: searching for hat returns hat, hat>top, hat>top>level (though not top>hat). Searching for 'hat returns only hat not the rest of the hat tree.

There is also a form field for indicating the name of the window that the results are going into. Be warned, as of .6.5 this and other issues concerning saving results are dicey, buggy and in transition. Yes you can save your results data, as described below, but you have to be careful! This is at present the most problematic part of TA.

As for the four search flags see the description of each in section I.B.

III. Results window

When you do a search the results appear in a results window. Results windows are basically browsers for results. A large text view appears at the top which shows the searched for text. The user can click on different rows in the large table shown in the bottom 3/4 of the window and the different found text will pop up in the top text view. The text in the top pane can be selected and copied, but it can't be modified. Columns in the results table (the bottom 3/4) can be resized by clicking on the lines between columns and dragging, and the columns can be re-ordered by dragging them to different positions.

A. *What the results show*

The table of results shows one row for each found record. What constitutes a found record depends on the type of search. See section IV. The first column is simply a row number. It also shows whether the row has been "marked" for recoding or for adding a code by post-fixing a "+" after the number. A column then appears for each universal code and then for each repeat code. For simple and unlimited searches the next column "_code" reveals the code of the returned text. For limited searches that column shows the criteria that the returned text matches. This is a subtle and important difference. See section IV for more details. The next column "_data" has the text that meets either the code or the criteria described in the "_code" column. If a comment was attached to the code (see TAMS syntax documentation) it then appears in the _comment window. The last column "_begin_loc" is the character position of the brace for the code that started that passage.

B. *Sorting results*

Columns can be sorted by clicking on the column head so that the whole column is selected and clicking the sort column button. Note this is not a widely tested feature!

C. *Finding the text in the original document*

If you want to look at the found text in its original context, possibly for additional coding or just to see additional contextual information, select the row in the results pane (that bottom 3/4 of the results window) that you are interested in and click the “Find record” button, the document window will be moved to the front and the text selected. **NOTE THAT ANY CHANGES MADE IN THE DOCUMENT WILL MAKE THE RESULTS WINDOW OUT OF DATE.** If you add text, for example, all of those `_begin_loc` numbers will be off, so if you go back and try to look at a different record, the program will be off in the text that it selects, so when you return to the results window you may want to refresh the results.

D. *Refreshing the results*

Any change in the main document window: additional text, additional codes, changes in the names of the codes, etc. makes this results window invalid. To rerun the search with the same flags and limit string, click the Refresh button, and the data will be replaced based on the changes in the document.

E. *Recoding and adding additional codes to the main window using the results*

One of the main functions of TAMS analyzer is to allow interaction between the document and the results. There are 2 main types of interaction. The first involves adding additional codes to the original window based on results, the other is actually changing the codes in the main window. Both of these are easily done, but the user needs to beware of some complications that can come up.

1. *Adding codes*

To take a selection of the results records and to each of them add some common code, start by marking the records you want to add the code to. Do this by picking Mark from the Recode submenu of the TAMS menu. You can also just pick shift-apple-M to mark records (and shift-apple-U to unmark them). Then pick “Add code” from the Recode submenu of the TAMS menu. A sheet will drop down allowing you to type in the code you want to add. Click OK and you can go back to the document and check the added codes. The way the adding works is as follows: adding the code “eggs” to the following passage: `{ham}breakfast{/ham}` results in `{eggs}{ham}breakfast{/eggs}{ham}`.

2. *Recoding*

A similar process works for changing the codes. This only works with simple and unlimited search results. To change the codes of found passages mark them using the Mark and Unmark menu options from the Recode submenu of the TAMS menu. Pick Recode from the Recode submenu, and again a sheet drops down allowing you to type in the new code you want (at some point a list of codes may be available, for now you have to type) and click Ok. Recoding the {ham}breakfast{/ham} to “eggs” results in {eggs}breakfast{/eggs}. But problems can result! One that plagues recoding more than adding codes is that the recoded passages will disappear from the results window. This is not a bug. You have changed the passage so it no longer matches the simple search criteria used to make the window. It’s no longer a matching result! Other problems are described in section 3

3. *Problems that can result from adding and recoding data*

It is not unusual for the effect of adding and changing codes to result in nested codes so that you get {a} some {a} text {/a} here {/a}. This sort of code within the same code (it would be fine if it was {a} and {a>b}) causes all sorts of problems. After adding or recoding go back to the document and run the “Check for nested” menu option from the TAMS menu.

F. *Saving results*

TAMS was never meant to be last word in data analysis. For many more complicated tasks, you will want to move the extractions to a database program such as Panorama, 4D or Filemaker, all of which can do sophisticated searches and calculations based on the results you get in TAMS. Unfortunately TAMS Analyzer’s ability to save really shows my novice status as a programmer. Be careful with saving. The best thing to do is to pick “Save to” from the file menu, and pick DocumentType as the file format to save in (this appends a .txt to the file name). If you save your results as a ViewType it appends .tam to the file name which many programs will then fail to recognize. No major problem here, just rename the file in finder with a .txt ending and say yes when the finder asks if you’re sure about changing the file type, extension, ending...

Hitting the close box will give you a moment in which you can save the file to .tam file, but if you pick cancel at this point the window goes away until you quit, then you’ll be asked if you want to save the file; I’m not sure how much of the file is even there at that point, SO BE VERY CAREFUL. The

safest thing to do is to use Save to save your results, and tell the program to review changes before quitting, but answer don't save to anything other than document windows. Sorry, at some point this will all be cleaned up!

IV. Unlimited, simple, and limited searches

Extracting information from a coded file is a tricky business. Depending on the way that you set up the search very different results will be obtained, and different options regarding recoding will be available. The key difference is whether the program searches for codes or searches for text. In simple and unlimited searches, the program looks for codes that match some criteria and return the text associated with those codes as separate records, one for each passage that has a code that matches. For limited searches, the program goes through and returns sections of the document that match a given criteria. Consider this passage:

{a}{a>b}This is the text{/a}{/a>b}

An unlimited search or a simple search on "a" will return 2 records: one for {a} and one for {a>b} from this passage, this is because the program is looking for codes that match the criteria and there are two such codes here. ON THE OTHER HAND, a limited search will return only one record, since it is just going through and asking for each character it finds whether that character is in a "zone" that matches the criteria of the search. Because the computer is just returning sections of text and is doing minimum keeping track of the codes attached to that text, you cannot recode a limited search.

Using a logical and "+", logical or ",", or a wildcard "*" automatically makes a search a limited search. To recode based on a limited search use the "Find record" button and edit the text by hand. I recommend that you work through the list of records you want to recode backwards so you don't have to hit the refresh button each time.

If you do not have a plus, comma or asterisk in your search, you have the choice of making it a limited or simple search by clicking on the simple check box in the search view of either the workbench or document.

V. Troubleshooting

A. I do a search and nothing happens...

You probably have mismatched begin and end tags somewhere. This can happen for all sorts of reasons, but you can find the missing bits by picking

“Check for Pairs” from the TAMS menu. This will move you to the starting tag of a pair that has no end: {a}this needs an endtag; rather than {a}this doesn't need an endtag{/a}. Note that endtags without a start tag pose no problem to TAMS though may give unwanted results.

- B. I save my results window and my database can't read it

Sometimes programs need the file to have a “.txt” filename end. Change the file name in finder so that it has the .txt end. Also you can use “Save to” under the file menu and pick “DocumentType” as the file format. This will automatically give it the .txt end.

- C. I save my results window and my database reads it as one really long record

That means you need to have the old (pre OSX) carriage returns/line ends. To get that pick “Preferences” from the TAMS Analyzer menu and make sure that “Use old Mac new line character for results” is checked. This should be less of a problem for more recent database programs (ones written for OSX).

- D. I save my results window and my database reads it but chops off quotes or even leaves them out

This is certainly a problem for Panorama database program and maybe others. Insert at the top of the file the following metatag: {!noquote}. This turns quote characters (") to \Q and apostrophe (') to \q.

Work bench window	1
<i>Establishing a centralized code file,</i>	<i>1</i>
<i>Searching across multiple files,</i>	<i>1</i>
<i>Establishing an Init file,</i>	<i>2</i>
Data entry or document window	2
<i>Bookmarks,</i>	<i>3</i>
<i>Diagnostics,</i>	<i>3</i>
<i>Code list integration,</i>	<i>3</i>
<i>Moving through the document from code to code,</i>	<i>4</i>
<i>Removing codes from part of the document,</i>	<i>4</i>
<i>Searching,</i>	<i>4</i>
Results window	5
<i>What the results show,</i>	<i>5</i>
<i>Sorting results,</i>	<i>5</i>
<i>Finding the text in the original document,</i>	<i>6</i>
<i>Refreshing the results,</i>	<i>6</i>
<i>Recoding and adding additional codes to the main window using the results,</i>	<i>6</i>
<i>Saving results,</i>	<i>7</i>
Unlimited, simple, and limited searches	8
Troubleshooting	8
<i>I do a search and nothing happens...,</i>	<i>8</i>
<i>I save my results window and my database can't read it,</i>	<i>9</i>
<i>I save my results window and my database reads it as one really long record,</i>	<i>9</i>
<i>I save my results window and my database reads it but chops off quotes or,</i>	<i>9</i>