Brief guide to new features in .8.7

I. Signed codes

One of the serious limitations of previous versions of TAMS Analyzer was that there were no way of separating the codes of different coders. If several people are coding documents it would be nice to have a system for keeping their codes separate. TA .8.7 introduces a way of “signing” tags. Know that there are many little features NOT implemented for managing such tags. There is not a way, for instance, of deleting just my tags but not yours, at present. There are probably lots of other small corrections that will need to be made to fully integrate this new syntax. As of this version TA will insert signed codes and search for signed codes; but that is all.

A. The new syntax

TA supports both signed and unsigned tags. Unsigned tags look like the codes TA used in previous versions: {a} and {/a} would be an unsigned tag pair. If my coder ID is mgw, the signed version would be {a [mgw]} and {/a [mgw]}. The signature consists of an identifier in braces after the code. If I wanted to add a comment it would be placed after the the closing tag signature: {/a [mgw]: this is a close tag}. Spaces between the code and the signature are ignored.

B. Coding

For data entry, coding signatures are enabled through the TAMS Analyzer preference dialogue:

Select “Use coder id” and fill in a series of letters (no spaces or punctuation) in the field provided. This will insure that when you code in TA your id will be placed in the tag.

C. Searching

1. Filtered results

TA provides two separate mechanisms for finding coder specific results. The first gets all the results of a search and then filters them for the coders you have provided. To implement such a filter just enter the id’s of the coders you are searching for in the “Coder id” field of the work bench or search pane. coders should be separated by commas so to look for my code (mgw) and the codes of someone named bob (bob) enter into the Coder id field “mgw, bob” (without the quotes).

To look specifically for unsigned codes use “*” (asterisk). So to look for codes that are either signed by me (mgw) or are unsigned enter “mgw, *” into the Coder id field.

2. Coder id as a search criteria
   What if I want to not filter results but to look for a specific instance of a code (alpha) signed by bob OR a different code (beta) signed by me. TA uses a different syntax for such searches. They are entered into the Search or Search for fields rather than the coder id fields. I would enter into the Search field the following string:

   ```
   alpha[bob], beta[mgw]
   ```

   If I want to consider alpha codes done by bob or me I could enter the following:

   ```
   alpha[bob; mgw], beta[mgw]
   ```

   (Notice that the coder ID's are separated by semicolons not commas!)

   Or I could split up the list

   ```
   alpha[bob], alpha[mgw], beta[mgw]
   ```

   Again, I could search for unsigned codes using an asterisk:

   ```
   alpha[bob;*], beta[mgw]
   ```

   I could also search for anything done by bob or mgw

   ```
   [bob], [mgw]
   ```

   I could also search for where our coding overlapped. The following returns all passages coded as both alpha by bob and beta by mgw:

   ```
   alpha[bob]+beta[mgw].
   ```

   Again, all of this would be entered in the Search for or search fields, not the coder id fields.
3. Advanced coder identity issues
   One advanced feature in the use of coder id as a search criteria feature is the use of partial name matching. TA can check the prefixes of a coder name so if I name one whole group of coders as groupAmgw and groupAbob etc. I can search for all groupA codes by searching using a tilde ("~" to indicate that the code is a prefix not the entire code name:

   ```
   [~groupA]
   ```

   This will return all coded passages coded by people whose IDs start with “groupA”. To look for all exzamples of passages coded “alpha” by groupA you would enter the following search string:

```
alpha[~groupA]
```