AUTOCODE Function

n = **Autocode**(Alpha_Var, Separator-List, Dif-Number, Inverted-File-Name, Codes-Description,
Exclusion-List[, History-File-Name] SET array-name);

[ ] indicates that this part is optional.

The function returns the following codes:

>0: The code description specified by Alpha-Var renders only one code and is automatically returned by the function.

-1: The code description entered didn't match any of the original gloss list specified in the "Code-Description" parameter (too vague).

-2: When the number of alternatives found by the algorithm is more than one but less or equal the maximum specified by the SET AUTOCODE command, the dialog box displaying each of the alternatives is open; if the operator is not satisfied with any of the alternatives can press the <Esc> key and the function will return -2.

-3: The number of alternatives rendered by the matching process is higher than the maximum specified by the SET AUTOCODE command. In this case, there is no dialog box displaying the various alternatives and the only choice left to the operator is to further refine the code description entered.

This function hides all the complexity of the automatic coding process, allowing the user to pass the text to be coded, and receiving from the function the corresponding code, provided that it is a valid description of the field to be coded.

The parameters list passed to the function is as follows:

- **Alpha_Var** is the text or description of the item to be coded; it can be in capital letters or upper and lower case; it can contain any type of ASCII characters but before it is submitted to the matching process, it will be **normalized**. By normalization we should understand the following operations: (i) conversion from lower to upper case; (ii) special characters like 'á', 'é', 'í', etc are converted to 'A', 'E', 'I', etc.; (iii) special characters passed in the **Separator-List** –second parameter- are ignored; (iv) excludes from the text all words that are found in the **Exclusion-List** –sixth parameter-; (v) the remaining words are sorted in ascending order creating a new list of independent words that will be submitted to the matching process; the first matching process step will be carried out over the **Historic-file** –seventh parameter- to check if that description exist –only if the parameter is present-.
- **Separator-List** is a list of special symbols (character string) that should be considered as words separators but otherwise, they should be ignored at the time the text is scanned. This list will probably include symbols like ',', '-', '(', '}', etc. This parameter is more adequate when the function is used in batch rather than in data entry since the DE operators and interviewers can be instructed not to use these special symbols when entering the text.
- **Dif-Number** is a numeric constant/variable that tells the systems the number of different characters that are acceptable -in the word matching process- in the event that a given word is not found in the data base (inverted file). Normally, this constant is one or at the most two, allowing for misspellings, plural/singular, masculine/feminine, etc. The algorithm used is the Levenhstein algorithm that has the advantage of being independent of the language. Essentially, given a word (keyword), a list of words or dictionary (the inverted file), and a

constant N, it will give a list of all the words in the dictionary that have at the most N differences (in characters) with the word being searched (keyword). It's clear that using 2 differences might lead to wrong conclusions and we highly recommend you test thoroughly the algorithm before applying it to the real process.

- **Codes-Description** is the original ASCII file where each possible code is fully described. The file is used first to produce the inverted file and, later on, to display different alternatives when the text entered to auto-coding renders more than 1 option. In these cases, all valid alternatives are displayed for operator's selection.
- **Inverted-File-Name** is the name of a special file that has to be generated prior to the use of the AUTOCODING function. The file is produced by a stand alone utility described later in this documentation. This file and its organization is the essence of the automatic coding strategy and algorithms.
- **Exclusion-List** is a list of words (alpha array) to be excluded in the text normalization procedure and it should be identical to the one used in the stand alone utility to generate the inverted file. Usually, this list includes words like articles, prepositions, conjunctions, and generally speaking, words that have importance from the grammar point of view but don't add meaning to the text.
- **History-File-Name** is the name of the file were the normalized text and the corresponding code either automatically imputed or assigned by the interviewer or DE operator will be stored. Conversely, before starting the matching process, the normalized text is used as index to this file and if the key is found, the corresponding code is assigned to the current text.
- **SET array-name** tells the system the array name where the codes of the different alternatives will be stored when more than one option was rendered by the matching process.
- In addition to the above mentioned parameters, this function works in conjunction with the SET AUTOCODE command to define the maximum number of possible alternatives the user wants to display. If the description entered renders more than the maximum specified by the SET command, the function will return a -3 code, meaning that the description is vague and therefore needs to be revised.

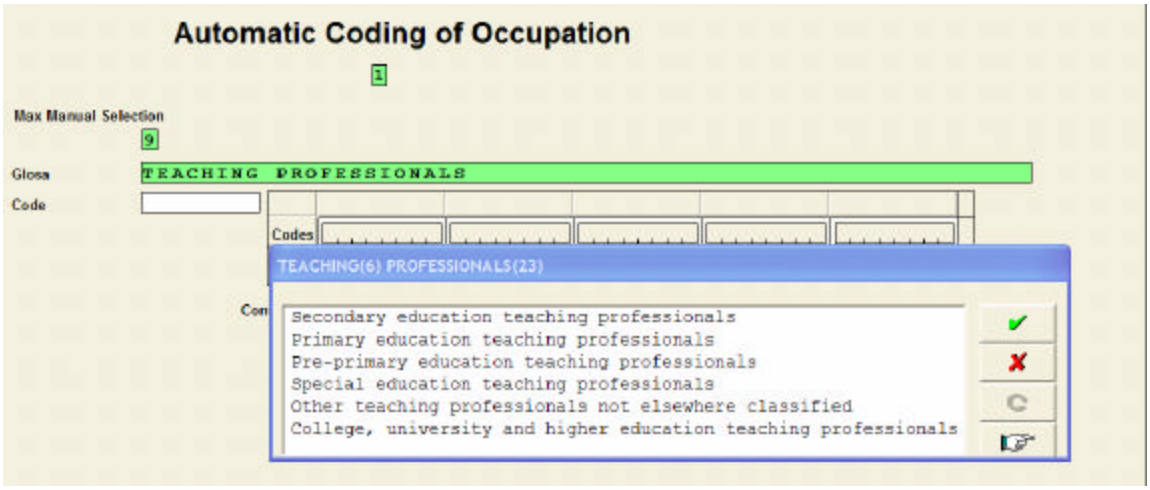
Operation Description:

As it was mentioned before, the text to be coded is normalized and converted into a list of independent and sorted keywords (in ascending order) that typically will produce three to four words (there is no limit to the maximum number of keywords). The first step is to check if the concatenation of keywords used as a key already exists in the historical file. If it's found, the corresponding code is returned by the function. Note that the historical file is optional and is created based on similar studies (surveys, censuses) previously carried out. Given the role importance this file plays, it is crucial to make sure that it's free of errors or they will contaminate the current operation. Another consideration regarding the historical file is that the code description should be sufficient to determine a unique code without the need to use one or more dependent variables in the process.

The second step consist in searching each keyword in the inverted file, keeping track of (i) the total number of hits or different original codes where that word was found; (ii) the list of actual codes where the word was found. If a keyword is not found in the inverted list, it will be omitted from the third step explained below, and there will be no automatic imputation even in the event when the other keywords all have in common one code.

The third step consists in the determination of the common code(s) of all the lists generated in the second step (intersection of code sets). Three different situations can be faced when the final code set is generated: (i) the code set is an empty list, in which case, the system will require assistance from the DE operator to further refine the description entered or in selecting the most adequate option; the possible options will be taken from the union of the code sets of

each keyword, displayed by number of matches –those codes that had more words matching will be displayed first-, with the matching words highlighted; (ii) the code set has only one code, in which case the function automatically returns that code; (iii) the intersection list has more than one code, in which case, again the system will require the operator's assistance to identify the most likely one from the different alternatives. In this case –as in (i) above-, the options will be displayed in a box having as heading the original gloss entered by the operator.

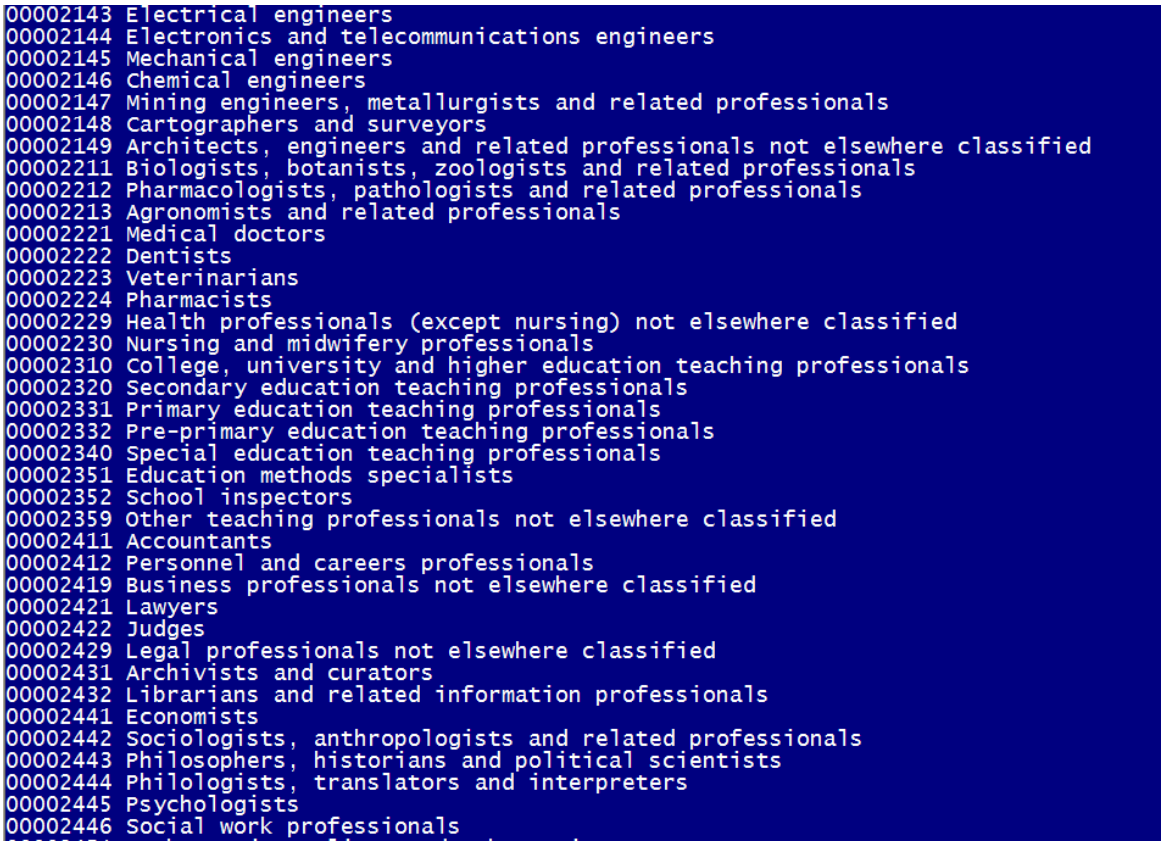


In the example above, the occupation description entered "TEACHING PROFESSIONALS" is clearly vague since there are six equally possible cases that match the specification. It's important to point out that the occupation description used in this example is not suitable for an automatic coding process since the wording used in the description is more academic than the every day wording. Probably, the normal answer for the specific occupation would be "HIGHSCHOOL TEACHER" or "UNIVERSITY PROFESSOR" rather than teaching professional. The results that will be obtained in the automatic classification depend strongly on how suitable the original list is for this process. In the particular case of occupation, it is highly advisable to have a detailed list of all occupations even if the same code has to be repeated several times with different descriptions. It is clear that a list description aimed to do manual coding is not suitable for automatic coding. Let's consider the following descriptions: "OTHER PROFESSIONALS NOT ELSEWHERE CLASSIFIED". This description might mean something for a person reading it in a specific context but in our case, it doesn't mean anything.

Whenever the dialog box is displayed, the DE operator can either select one option, in which case the function will return the corresponding code, or press the <Esc> key refusing to select one, in which case, the function will return the code -2.

In the rare event that even the union of all the individual keyword codes lists is empty, the function will return the code '-1'.

One crucial file for all this process is the original "Code_Description" since from there the inverted file is generated. The following image shows part of this important file:



Although as it has been anticipated, the code description presented here is not adequate for an automatic coding process, it shows some characteristics that have to be pointed out.

- The code (first column shown in the figure above), has to be numeric and of fixed length; if there are codes of different length, they should be padded with zero(s) on the left. This is important since at the time the file is scanned to produce the inverted file, the code length is defined. The code length is calculated using the first line only. Thus the file needs to be homogeneous (all codes have the same length). The first BLANK or space following the code is used as delimiter denoting the end of the code string.
- The text following the numeric code string is the code description and should be as concise and thorough as possible. However, this text is the one that will be displayed when more than one alternative is rendered by the matching process. Therefore, it should be clear enough for the operator to decide between the various alternatives.
- The file should be an ASCII file.