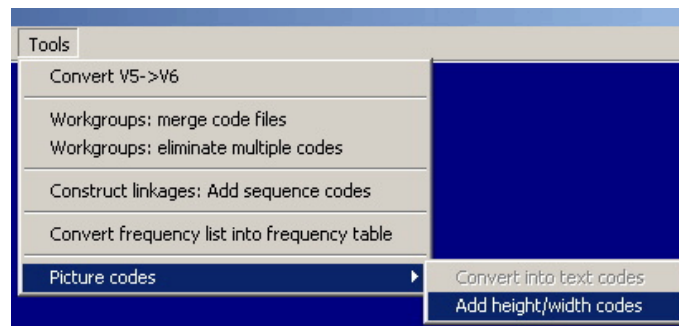


Chapter 15

What are the AQUAD tools good for?

The last group of functions in the main menu is labeled "*Tools*." The screen shot below shows, which options are available. The option labels as well as the windows occurring after clicking are more or less self explaining. Therefore we describe only briefly the purpose of each function and list stepwise how to handle its possibilities.



15.1 How to convert codes from AQUAD 5 into the format of AQUAD 6

Aquad 6 cannot handle directly code files produced with the preceding version, because earlier codes were only up to 22 characters long. Now the limit are 30 characters. This conversion routine adapts your old code files to the conditions of version 6.

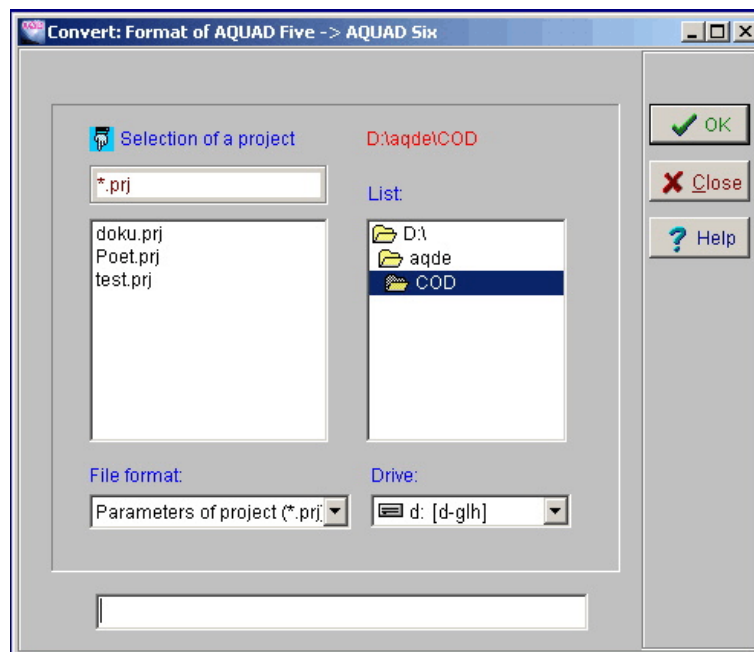
You will experience no problems, if you observed the conventions valid for coding in AQUAD 5. The conversion routine expects that

- all names of text files have an extension ".txt" or ".rtf";
- all code files were saved (automatically) a sub-directory "..\COD";
- in this sub-directory the master code list ("*.amc") of the project can be found.

Clicking on the conversion routine in the menu opens the window presented below. There (in the right part) you care first about the settings that tell AQUAD where to find your old code files. That is, you select appropriate drive and path settings (in the example below: drive "d:" and path "d:\aqde\COD"). In the box on the left you will be presented with the names of available project settings. If these names do not occur automatically, just double click on the marking "COD". If still no project names are visible in the box on the left, you may have moved your old project settings to some other location ...

In our example we find the familiar project "*poet.prj*" among two other project files. We click on its name and the name will be copied into the selection slot above (the blue hand is pointing to it). Now AQUAD has all information it needs, and we can click on "OK" and wait for the results.

The routine effects that we find afterwards our old project listed among the projects of AQUAD 6; correspondingly all (internal "*.atx") text files, memos, and code files belonging to this project were converted and copied into the matching sub-directory of AQUAD 6.

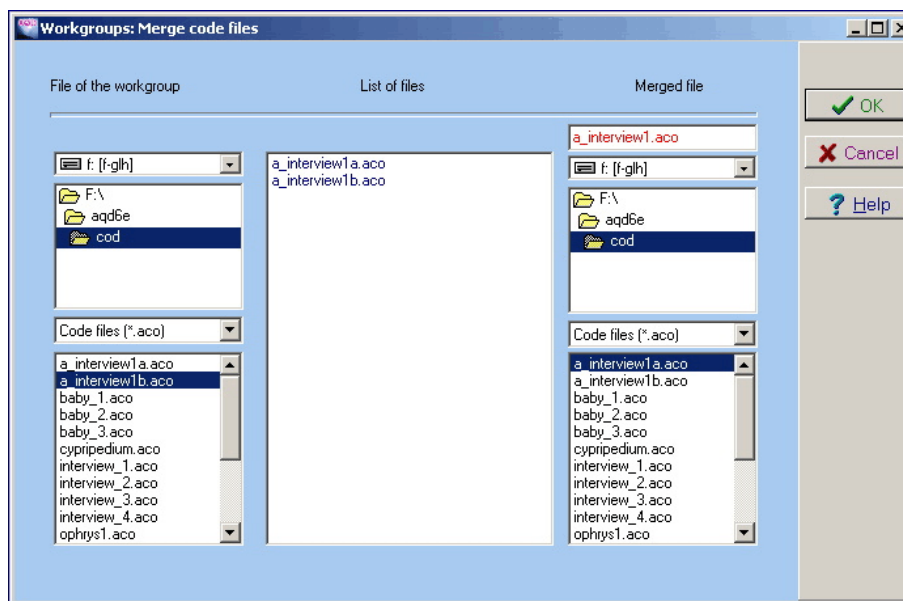


15.2 Tools for work groups: How to merge code files

A group of researchers working together at a common project often decides to divide the work among the members. Particularly during coding this strategy both helps to save time and to intensify the analysis: For instance, part of the team could look in the interview recordings (or texts) on your program CD only for what the teacher says about social relations (to students, colleagues, parents, superiors), another part for statements about teaching and learning, a third sub-group for reflections on the relation of theory and practice in everyday tasks in classrooms, etc. As a variation of cooperation in research teams, the files of a project could be distributed among subgroups or individual members once they have elaborated a common system of categories for coding.

Both strategies are not only well suited to speed up the common work, but also to introduce beginners into concrete applications of qualitative methods. Comparing ways of access, problems, and results between groups sets off highly intensive and fruitful debates about approaches of qualitative methodology. At least, we experienced these effects in university seminars for beginners and for doctoral students.

However, there is always the same problem: various code files referring to the same data file have to be merged into a single code file for all team members. The following analyses will then be based on this file, which contains all interpretative perspectives elaborated by the whole group. Already when we distribute the work tasks, we should take care that the groups use different names for saving code files attached to the same data file, otherwise one code file will overwrite the other when we copy them from the teams' PCs to the project manager's computer. Let us see, how the "merge"-tool is operated:

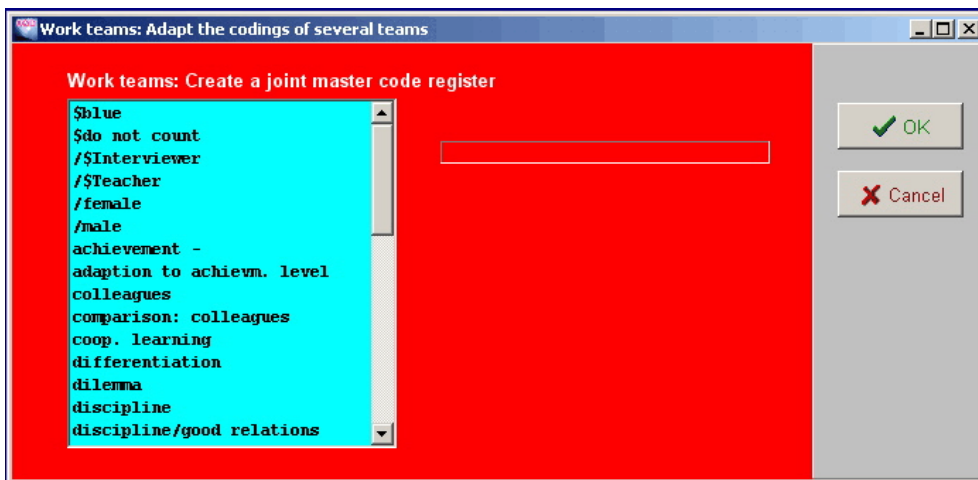


After clicking on the function in the main menu, the above window occurs – with mostly empty boxes, of course. If necessary we change the settings in the upper left part, where drive and directory of the actual code files (after copying them to the project manager's PC) are determined. On top of the file box (lower left side) we find two code files "a_interview1a.aco" and "a_interview1b.aco." These names tell us, that a group A and a group B coded separately the audio recording of interview 1 (on your CD: "a_inter1.mp3"). Now we intend to merge these two code files, therefore we select them by clicking, which copies their names into the box labeled "List of files" in the middle. What we need now is a name for the merged files. The box "Merged files" on the right contains at the moment the same list as the box on the left. We can click on one of the already selected files, in the example the file "a_interview1a.aco;" this copies the highlighted name in red color into the name slot right on top. One of the selected code files after the other (here only on file more) will be added to the file with this name after clicking "OK." Because we need a file name corresponding to the initial data file "a_interview.mp3" we erased the "a" from the selected file name and thus get a merged code file "a_interview1.aco."

15.3 Tools for work groups: How to eliminate multiple codes

If we merge code files related to the same data file (see section 15.2), we usually find multiple codes, in this case several identical codes attached to the same data segments. In our interview example each work group will use the speaker codes *"/\$Interviewer"* and *"/\$Teacher"* to characterize corresponding data segments. Of course, we have to eliminate multiple occurrences of identical codes, otherwise we would produce erroneous results when we have our codes counted, when we have linkage hypotheses tested, etc.

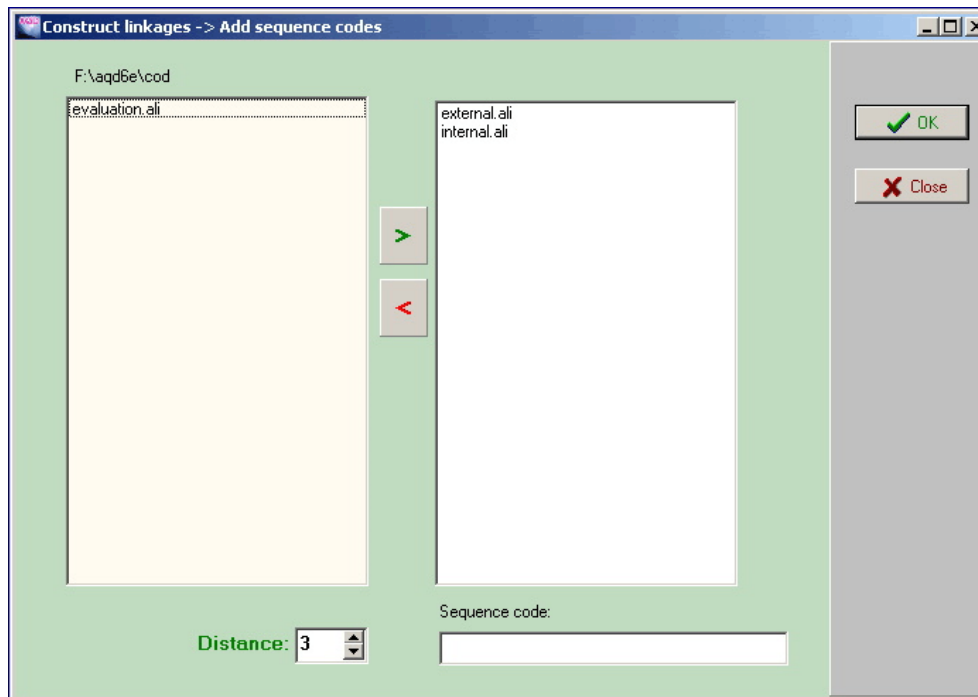
This job is done by the function *"Work groups: eliminate multiple codes."* Based on the master code file of your project, which is shown also, AQUAD looks for redundant code entries and adapts the merged files. Clicking on *"OK"* starts the job.



15.4 How to apply automatically linkage constructions and insert sequence codes

In large projects with a great number of data sets and codes it has turned out to be effective, if researchers first concentrate on constructing linkage hypotheses under theoretical points of view. These constructions are saved (see chapter 12, section 12.1), and we can apply them later and have AQUAD additionally insert sequence codes in our code files wherever the hypothetical linkage is true.

However, if we construct a number of hypotheses, it will be boring as well as a waste of time to go through all the steps of linkage testing – selecting the function from "Tools," selecting a linkage construction, writing a sequence code, saving the results – again and again. Therefore All these steps are automatized in a sort of batch function:



In the left box all linkage constructions ("*.ali" for AQUAD Linkage) available in the code subdirectory (here: "F:\aqde\cod") are listed. There were three of them, but we already selected two by marking them and we moved them into the selection box on the right side by clicking on the green arrow. If you assume that the red arrow will reverse the operation (for marked items, of course), you are absolutely right! Additionally we may modify the default value of 3 units of distance (lines or seconds).

What we cannot enter in this batch mode is determining a specific sequence code for each of the selected linkage hypotheses. Sequence codes will be determined automatically, that is, AQUAD takes the name of the actual linkage hypothesis and uses it as code name. In our example all positive findings according to the linkage "*external.ali*" will be coded as "*external*" and all findings based on the linkage construction "*internal.ali*" as "*internal*," and so on. During the operation the actual sequence code is shown in the corresponding slot at the bottom of the window, however, we will be able to read it only if the program has to process a huge amount of data and needs much time.

15.5 How to convert frequency lists into tables (export as CSV files)

This function was already describe in detail in the section on combinations of qualitative and quantitative methods. Please, read chapter 14, section 14.4 to find out, what this function is good for and how to use it.

15.6 How to modify codes of graphic files

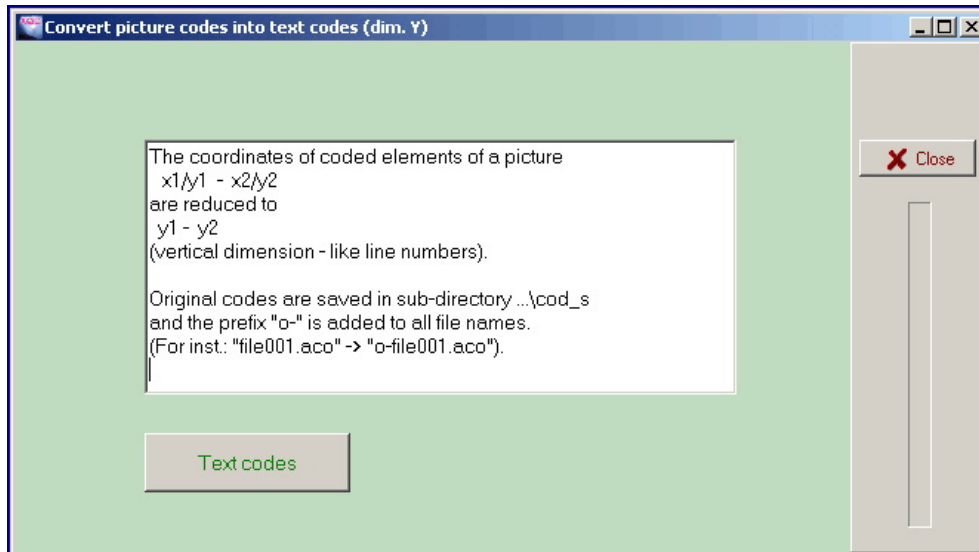
15.6.1 Converting picture codes (x/y) into text codes (y)

In chapter 6, section 6.9.2 we described how to analyze scanned text files, that is texts available in form of picture files. The only problem is that we have to find a way how to localize coded text segments without given line numbers.

When we code texts, we mark relevant segments by drawing a rectangular frame around them. In our code files the program saves the coordinates of the upper left corner (x_1/y_1) and the lower right corner (x_2/y_2) of this frame.

With these data it is no problem to retrieve any coded segment irrespective of what its content is – some scanned lines of text, part of a photo, etc. However, if we intended to determine the sequential order of graphic data segments like we do with segments of text, audio and video recordings (for instance: What occurs before a specific segment? What is included in a specific segment? Which segments overlap each other?), we would run into trouble. A sequence is defined as a one-dimensional, linear order of elements, but now we have two-dimensional objects and corresponding coordinates. Well, there is a simple solution, which works well for "texts as pictures," in our case texts, which were scanned or photographed and saved in JPG format (see chapter 4, section 4.1): we eliminate the x-components from each segment's coordinates of position on the graphic surface. What we keep are the y-coordinates, that is information about beginning and end of each segment on the (vertical) y-axis – or a functional equivalent of line numbers in text files.

The initial code files are saved and automatically activated again, if a function of analysis (for instance, coding again later) demands graphic coordinates. In this case, that is, if you interrupt a sequential analysis of graphically represented data by editing your initial graphic codes, you have to repeat the conversion into text codes afterwards.



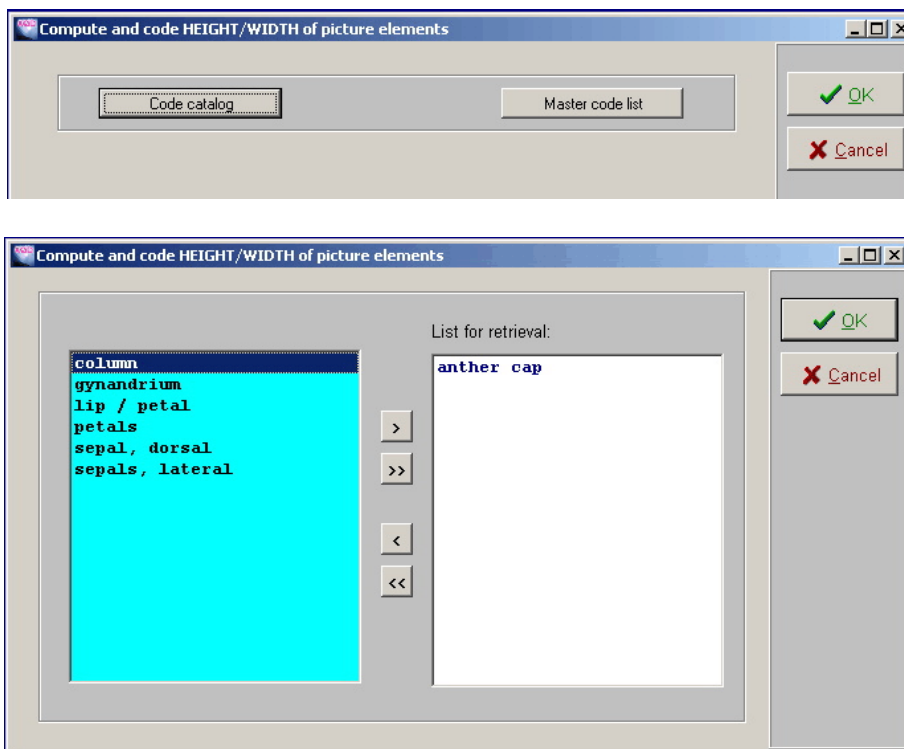
This option is available only if you actually work with the data of a picture project, that is, if an appropriate project is "opened." Clicking on the button "Text codes" starts the conversion like described above in all files of the open project. Don't worry: Your initial picture codes are saved automatically in a sub-directory "..\cod_s" of AQUAD and written back into "..\cod" if necessary. An additional button "Original codes" appears, if you start this function again. Clicking on it permits to undo the conversion on demand. Text codes are *not* saved in this case, that is they are not copied into "..\cod_s", but overwritten in "..\cod" by your original code files. However, you can repeat this conversion in almost no time whenever you need text codes again.

15.6.2 Computing and coding width and height of picture segments

To answer some specific research questions we need a function to compute the dimension of selected picture segments. It may be interesting, for instance, to find out the sizes of particular picture elements and to compare them when analyzing children's drawings. How big was the father drawn as compared to mother and siblings in this child's family drawing – and did all children produce similar differences in size? Or: Which proportion of area does a specific object consume in the drawings of different children?

The function "Add height/width codes" computes height and width of selected picture segments (selected by their corresponding codes) as differences of vertical and of horizontal pixel-coordinates and inserts the results in form of new codes in the available code files. Consequently, you have to code critical picture segments *before* you can use this function.

Either by loading an already created code catalog or by selecting actually from the master code list you determine those elements in your pictures, whose height and width are interesting in your analysis. The computation is quite simple: The difference of coordinates $y_2 - y_1$ gives the height, the difference $x_2 - x_1$ the width of a segment (in pixels). The procedure begins – like counting codes (see section 10.5) – with selecting critical codes:



Clicking on "OK" after selecting codes produces two code entries added to each of the selected codes. For instance, if we select in the example files of our project "blossoms" the code "anther cap" (see screen shot above), the routine computes height and width (in pixel-units) of this part of the blossoms in our photo files and adds two codes (below the additional entries to the code file "ophrys4.aco")

178/205	256/245	anther cap	
178/205	256/245	anther cap	H: 40
178/205	256/245	anther cap	W: 78

Remember for later statistical analyses that the symbol for the dimension ("H:" or "W:") is always placed on positions 46 and 47 of the code-"string", and the measure (in pixels) on positions 48-50.