MAXQDA and Chapter 9 – Coding Schemes

Chapter 9 discusses how the structures of coding schemes, alternate groupings are key to moving forward with analysis. The nature and structures of the coding scheme in a software application and how it can be used can make a difference to the processes of analysis. There are two key aspects – in order to feel confident about the potential of a software to support particular working styles the inherent nature of the structures is significant; following on from that the ability to manipulate the coding scheme and extract the most from it will increase confidence in the software. *See all coloured illustrations (from the book) of software tasks and functions, numbered in chapter order.*

Sections included in Chapter 9

Project related factors

Is ‘coding scheme’ the same as ‘theoretical framework’?

Separating aspects of analysis

Escaping the confines of a scheme

Hierarchical

Non hierarchical

Better ways to collate or express theory

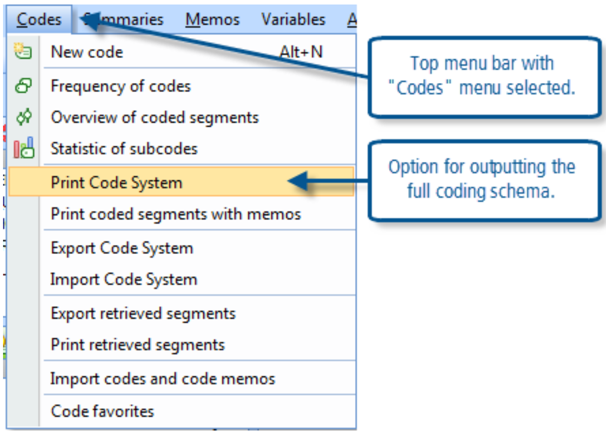
Sets and alternative grouping

MAXQDA11 and Case Study B – Output code lists

This task was covered in Chapter 7 Exercise 8, but is briefly repeated here.

The coding schema can be printed out at any stage with the menu option “Codes > Print Code System”. Figure 9.1.1 illustrates the “Codes” menu from the top menu bar with this option highlighted.

Figure 9.1.1 – Codes main menu

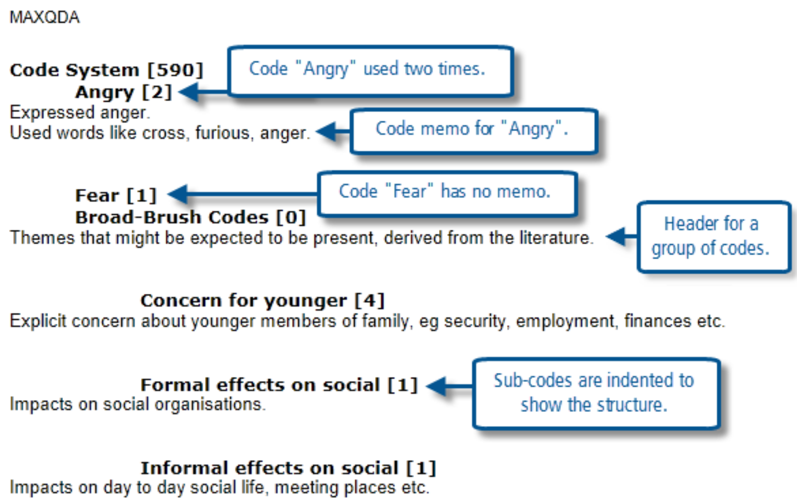


Top menu bar with "Codes" menu selected.

Option for outputting the full coding schema.

When you select the option to print the code system there is one supplementary dialog which asks if you want to include the code memos – it asks "Visualize code memos?" with "yes" and "no" buttons. Including the code memos may extend the length of the print-out, especially if you have stored many long definitions, but will generally be a useful part of the output.

Figure 9.1.2 – Code System Print-out



MAXQDA

Code System [590]

Angry [2]

Expressed anger.
Used words like cross, furious, anger.

Fear [1]

Broad-Brush Codes [0]

Themes that might be expected to be present, derived from the literature.

Concern for younger [4]

Explicit concern about younger members of family, eg security, employment, finances etc.

Formal effects on social [1]

Impacts on social organisations.

Informal effects on social [1]

Impacts on day to day social life, meeting places etc.

Code "Angry" used two times.

Code memo for "Angry".

Code "Fear" has no memo.

Header for a group of codes.

Sub-codes are indented to show the structure.

Figure 9.1.2 above, shows part of such a report. Note how the code names appear in a bold font, with the number of times they have been applied in square brackets. The code memos, if you have answered "yes" to the dialog about visualizing them, are beneath the code names and in a lighter font. The names of sub-codes are indented as a way of showing the hierarchical structure.

MAXQDA11 and Case Study B – Rename codes and edit their parameters

This task was covered in Chapter 7 Exercise 1 but is briefly repeated here.

Editing code parameters

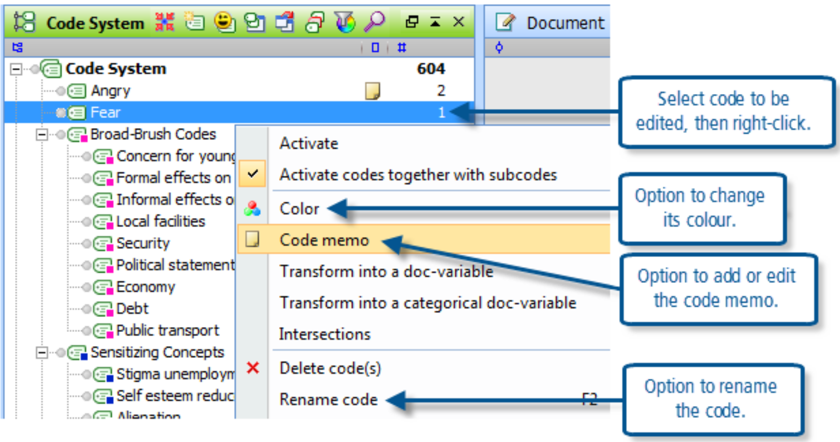
All of the details that were entered in the New code dialog can be altered at any time should the need arise. Simply select the code to be edited in the Code System window with a right-click and select one of 3 relevant options from the context menu:

- Rename code (F2)
- Color
- Code memo

Rename the code

The renaming of a code takes place on the code label where it appears in the Code System window. This uses techniques just like renaming a file in the MS Windows operating system. When a code has been renamed the new name will appear wherever the old name might have been seen, the only things that will not be altered are previously printed or stored reports.

Figure 9.2.1 – Context menu on a code name in the Code System



Select code to be edited, then right-click.

Option to change its colour.

Option to add or edit the code memo.

Option to rename the code.

TIP: An alternative short-cut to rename a code that has already been selected in the Code System window is to use the function key F2. (Note how a reminder of this is shown in the context menu, as illustrated in Figure 9.2.1 above).

Change a code's colour

As you refine your coding scheme you may find it necessary to change the colour of some codes at the same time as they are renamed. Selecting the "Color" option from the code context menu allows you to select any colour for that code. If you select this option for a group header you will see an additional dialog at the end of the process asking if you want to apply the new colour to the whole group, so it is possible to change an entire group's colour in a single operation.

Edit the code definition in the Code Memo

Selecting the "Code memo" option from the code context menu opens up a standard memo dialog. This has a different appearance from the "New code" dialog in which you may have started the code definition, but it is the same dialog as you use for other memos in the project. Any existing text in the code memo will be present in the editing field of the memo dialog, where it can be edited, deleted or added to in the usual ways. All edits and changes are saved when you close the memo dialog.

TIP: As suggested in the chapter, it is possible to keep the existing definition as part of your audit trail while adding a new definition for the re-named code. In view of the way that the first part of a code memo can be displayed at any time by holding the cursor over its 'post-it note' symbol in the Code System window, you should insert the new definition above the previous one and probably use the date button in the memo window toolbar to add the current date.

There is a short-cut to edit a code memo that is always available and that is to double-click on the memo symbol beside the code in the Code System window. This immediately opens the memo window with the existing code definition showing.

MAXQDA11 and Case Study B – Merge codes

This task has not been covered in earlier exercises. In MAXQDA11 the principle of merging two codes is based upon the idea that it is the coded segments that are affected by such a change (rather than the codes themselves).

To illustrate this point imagine a coding scheme with two codes, say “Code A” and “Code B”, which are to be merged into a code to be called “Code AB”. This will be achieved in 2 main steps and 3 supplementary steps as follows:

1. Open the context menu on Code B and select the option “Move coded segments”
2. Open the context menu on Code A and select the option “Move coded segments from Code B”
3. Rename Code A to “Code AB” (use the context menu or F2)
4. Edit the code memo for Code AB to record the details of the procedure you have just carried out (use the context menu or double-click on the code memo icon)
5. Either delete Code B (which now has no segments attached to it), or move it to another part of the code system reserved for redundant codes

At the end of this procedure Code AB will be linked to all of the segments previously coded as either code A or Code B, and Code B will not be linked to any segments. A segment that was previously linked to both Code A and Code B will be linked just once to Code AB.

TIP: Note that this procedure cannot be reversed or undone directly, so you should probably make a back-up copy of the project file immediately before taking these steps. It will then be possible to go back to that back-up if you change your mind about the merged codes.

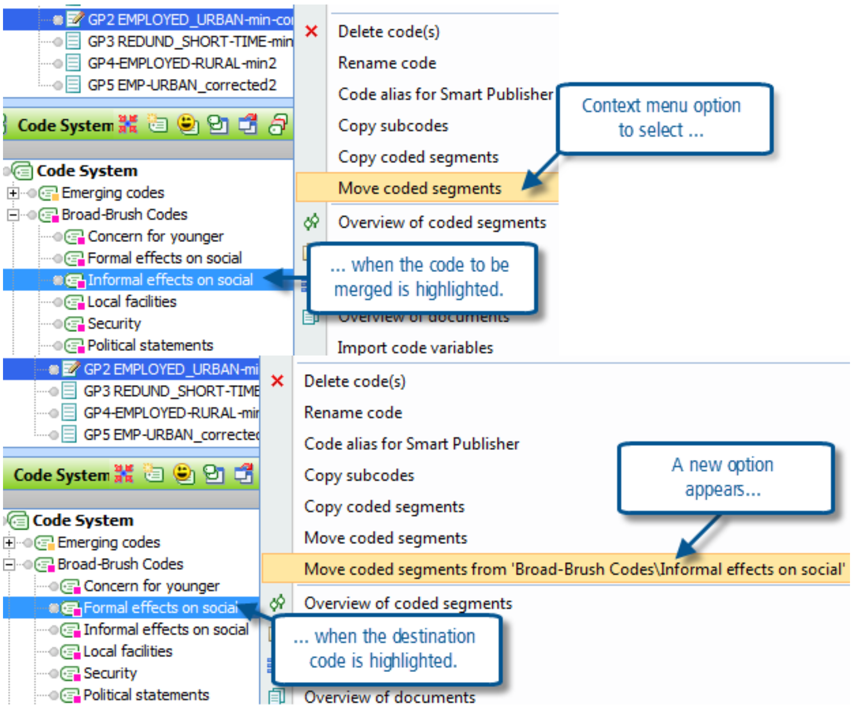
Figure 9.4.1 compares the context menus that appear at steps 1 and 2 when the codes “Formal effects on social” and “Informal effects on social” are merged into a single code “Effects on social”.

TIP: Note that the options in the context menu when you right click on a code label will change between steps 1 and 2 above, because the program expects that the next operation after you have selected the option to move the segments from Code B will be to put them into another code.

At step 1 the code “Informal effects on social” is selected and a right-click on it opens the context menu shown in the upper part of the figure. The option “Move coded segments” has been selected (as shown by the brown highlighting). (Note that only the relevant part of the context menu is shown in this illustration).

At step 2 the code “Formal effects on social” is selected and a right-click on it opens the context menu shown in the lower part of the figure. Now a new option has appeared “Move coded segments from …” with the label for the code selected at step 1 inserted into the context menu. A click on this option brings up a confirmation dialog asking you to confirm that you want to go ahead with this operation, if you click on “Yes” then the merge is completed and the number of segments attached to the “Informal ..” code drops to 0 while the number attached to the “Formal ..” code increases.

Figure 9.4.1 – Changing context menus during code merge



Context menu option to select ...

... when the code to be merged is highlighted.

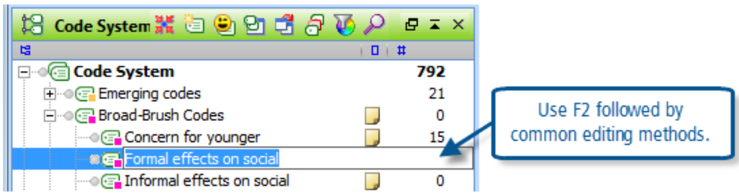
A new option appears...

... when the destination code is highlighted.

Now the supplementary steps should be followed.

First, re-name the code linked to all of the segments that have been merged. It should still be highlighted from the previous operation so the F2 short-cut would be a good option to use (the alternative is the right-click click context menu option “Rename code” which is visible in both parts of Figure 9.4.1).

Figure 9.4.2 – Renaming a code



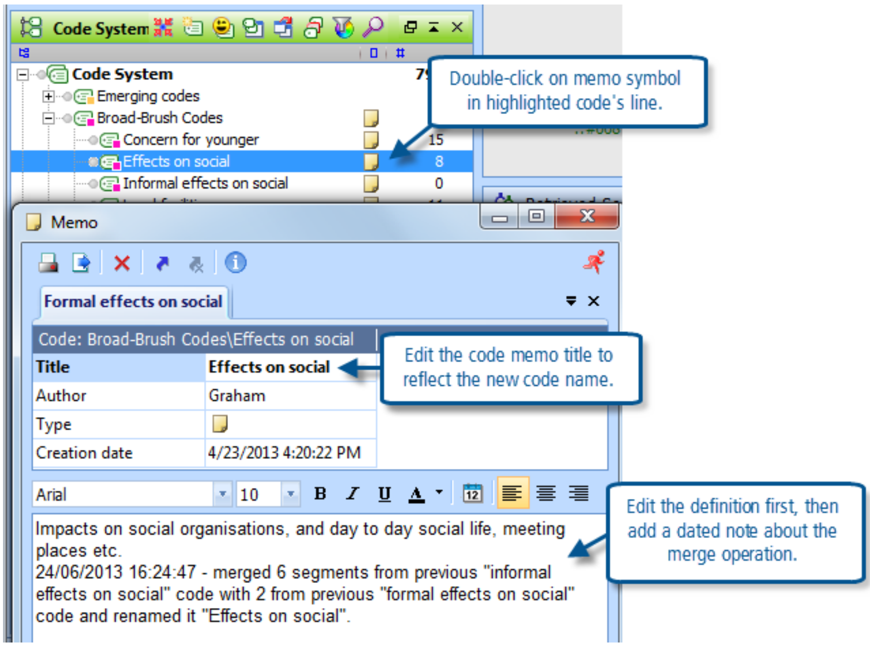
Use F2 followed by common editing methods.

In Figure 9.4.2 the F2 key has been pressed and now the label for the combined code can be edited. To keep most of the existing label hit the “End” key first so that the cursor jumps to the last character in the label, you can then use backspace to delete from the end or use the arrow keys to move around the text string and insert or delete letters as necessary. Here, we will delete the first word so that the new label for this code is “Effects on social”.

Next, edit the code memo for this combined code to reflect the changes that have been made to it. Double-click on the code memo symbol beside that code to open the memo editor (or use the right-click context menu option “Code memo”). See Figure 9.4.3 for an illustration.

TIP: Note in Figure 9.4.3 how the old code “Informal effects on social” now has no segments attached to it, as shown by the zero on that row in the frequency column. Also note that the code above it in the list has been renamed and now has 8 segments attached to it.

Figure 9.4.3 – Edit the code memo for the merged code



Code System
- Emerging codes
- Broad-Brush Codes
  - Concern for younger 15
  - Effects on social 8
  - Informal effects on social 0

Double-click on memo symbol in highlighted code's line.

Formal effects on social

Code: Broad-Brush Codes\Effects on social

Title	Effects on social
Author	Graham
Type	
Creation date	4/23/2013 4:20:22 PM

Edit the code memo title to reflect the new code name.

Arial 10

Impacts on social organisations, and day to day social life, meeting places etc.
24/06/2013 16:24:47 - merged 6 segments from previous "informal effects on social" code with 2 from previous "formal effects on social" code and renamed it "Effects on social".

Edit the definition first, then add a dated note about the merge operation.

Finally, either delete the redundant code by using the “Delete code(s)” option from the right-click context menu on its line, or drag it to a suitable section in your coding scheme such as “Codes no longer in use”.

MAXQDA11 and Case Study B – Group codes into hierarchical listings

This task was introduced in Chapter 7 Exercise 1 as part of the process of creating codes, but it is worth re-examining it here in the context of re-organising your coding schema.

The two main advantages of using some hierarchical structure in the coding system are to facilitate activation for queries and output tables later in the analysis, and to allow some codes at times to be reduced to just a heading in the Code System window leaving more screen space for the other codes currently being used.

When the context menu option “Activate codes together with subcodes” has been ticked for a code group header, then the whole group of codes can be activated with a single Ctrl+click on the group header code.

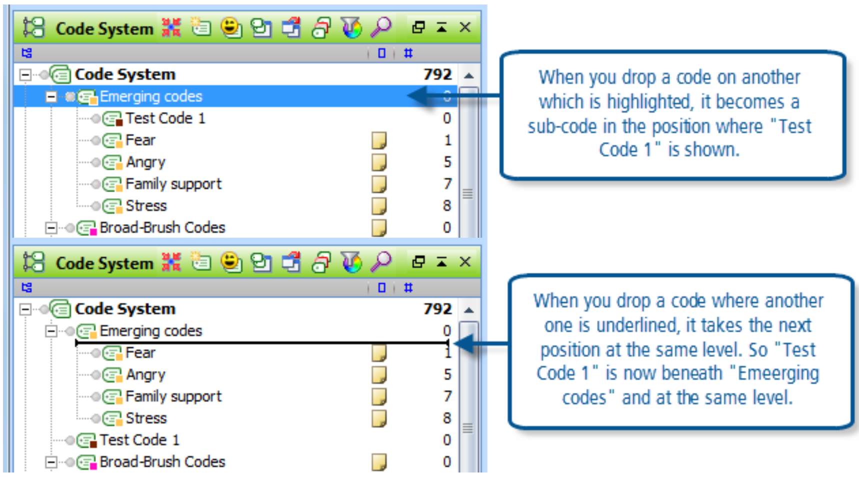
A click on the “-“ box beside the group header code will minimise that group down to the header alone, while a further click on the “+” box beside that lone header will expand the group to show all of its members once again. This means, for example, that you can have more space to see the survey question codes while you work on coding the survey responses without having to scroll past the long group of focus group speaker codes.

Moving codes around within the code system

The trial codes “Angry” and “Fear” may now be used to experiment with moving codes around the Code System. Drag and drop is the basic method to use. Click on a code, say “Fear” and, holding the mouse’s left button pressed down, move the cursor until it is in the place where you want that code to be shown, then release the button to drop it there.

There are two different effects that appear as the cursor approaches a new location to drop a code. If the code at the destination is highlighted with a blue background as you release the button then the code that you dragged will be dropped as a sub-code of the highlighted one. Alternatively, if a thin black line appears to underline the destination code as you release the button then the code will be dropped in the next position beneath the destination code with both codes at the same level. This is illustrated in Figure 9.5.1 using a new code called “Test Code 1” which has been dragged onto the group header code called “Emerging codes” using each method.

Figure 9.5.1 – Using drag & drop to move codes around the Code System



When you drop a code on another which is highlighted, it becomes a sub-code in the position where "Test Code 1" is shown.

When you drop a code where another one is underlined, it takes the next position at the same level. So "Test Code 1" is now beneath "Emeerging codes" and at the same level.

If you wanted “Test Code 1” to appear in the list between “Fear” and “Angry” you should drag it and release the button when “Fear” is underlined. In this way you can move any code to any required position in a single operation, it doesn’t matter how many hierarchical levels it is moving up or down.

TIP: The best way to learn this is to try it several times. Use two codes for practice and make them appear in any order at the same level and then alternately each as a sub-code of the other. You may also move them in and out of the other groups.

You can also drag a whole group and drop it in a different place by dragging the group header. The same effects work for the group. So, if you drag the “Sensitizing Concepts” header over the “Broad-Brush Codes” and drop it when the latter is highlighted with a blue bar then you will find that the first group has become a sub-group of the second. If this was a mistake it can be corrected by dragging the sub-group to another place in the code system and dropping it beneath an equivalent level group while the black line is showing beneath its name.

There is one code heading that works slightly differently and that is the overall header for the whole schema, the “Code System” code. You cannot make this show a blue highlight when you drag another code to it, but nevertheless any code or group of codes dropped on it will always appear as a sub-code or a sub-group of it. You may want to drag something to this header to make it appear at the very top of the coding schema.

Create new code groups

A new code group is created in the same way as a new code. Select the place in the Code System where you would like the new group to be shown and use the “New code” icon, the right-click context menu option “New code”, or the keyboard short-cut “Alt + N” to open the New code dialog. Type in an appropriate name for the new group and select a colour for its members (this is optional).

Once the new group header has been created you can drag existing codes to it and drop them under it when the header code is highlighted in blue. You can also create other new codes under it by starting the new code process while the header code is highlighted.

There is no significant difference between the header code of a group and the sub-codes 'beneath' it. You can apply a header code to data segments in the same way as the sub-codes, so it can be an actively used code like any other. Some operations can be carried-out quickly on a whole group by applying them to the group header, this includes activation and changing the colour of code brackets, although you still have the option of doing either of these operations on the group header alone.

Sorting codes in a group

You can sort the sub-codes in a group into alphabetical order by selecting the group header code with a right-click and using the option "Sort sub-codes" from the context menu.

TIP: Although it is possible to sort all the codes into alphabetical order, this is difficult to undo if you do not like the resulting effect. As you work with a coding system you become familiar with the location of the codes in the list and generally the manual adjustments using drag and drop are quite sufficient to keep the list in a user-friendly sequence.

Colour in groups of codes

Don't be afraid to experiment with using colour as a way of identifying an emerging hierarchical structure of codes in the code margin of your documents. If you have different groups for different types of meaning then having a common colour for a group makes it easier to pick out its members as you scroll through a coded document. In MAXQDA11 you can change the colour of a single code or a group of codes as many times as you like, the current colour will always be shown in the code margin (unless you have used the option to hide all code brackets of that colour).

It is sometimes necessary to use codes for structural purposes in MAXQDA11, in contrast to thematic meanings. In these examples some codes have been used to identify the speakers in the focus groups and others have been used to identify the responses to the separate questions in the survey data. These have been set up using the default green colour as a way of visually separating them from the various alternative colours being used for the thematic codes.

TIP: MAXQDA11 can work with up to 10 different levels in the Code System hierarchy. However, it is best to keep to 3 or 4 levels at the most until you are familiar with all of the effects and uses of the multiple levels.

Delete codes and code groups

It is a fairly drastic step to delete a code in the Code System. This can be achieved by using a right-click and selecting the option "Delete code(s)" from the context menu. You will be asked to confirm that you really do want to proceed with this operation, because once effected it cannot be undone in any simple or direct way.

If you have decided that a code is no longer helpful or relevant to your analysis, so that you are considering deleting it, then it might be just as valid to move it to a section at the bottom of your coding schema which you might call "Redundant codes". Having done so you might also change its colour to something that will not catch the eye when you see it in the code margin, or better still change it to a specific colour that you can then exclude from the display in the margin (right-click in the code margin and untick that colour in the list for display). In that way you can easily bring that code back to life again if you change your mind at a later stage.

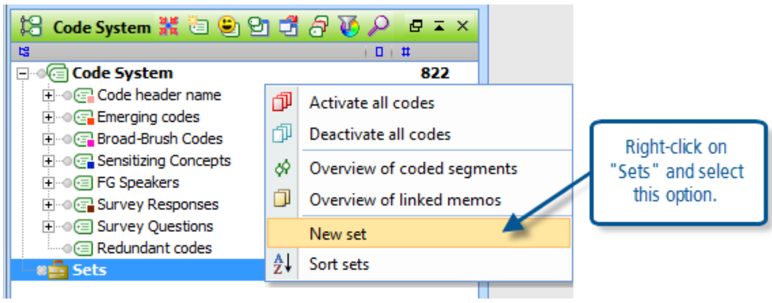
MAXQDA11 and Case Study B – Create short-cut groupings of codes – "Code Sets"

You may have noticed the heading "Sets" at the bottom of the list in the Code System window (and a similar item in the Document System window). This is to be the topic of the next exercise.

A code can only exist at any time in one place in the coding schema, but there may be occasions when you might wish it could be shown in more than one hierarchical group. By using "sets" you can create flexible groupings of codes which are simply groups of short-cut links to the codes. The various sets that you create are held at the bottom of the Code System and look quite similar to the hierarchical groups of the codes. A code may be in any number of sets simultaneously. You might be generating a small theory and in your mind several codes for different areas of the coding scheme contribute to it.

As a first step, create a set called "Codes I'm not sure about yet". To do this use a right-click on the "Sets" label just beneath the list of codes in the Code System window, and select the option "New set" – as shown in Figure 9.6.1.

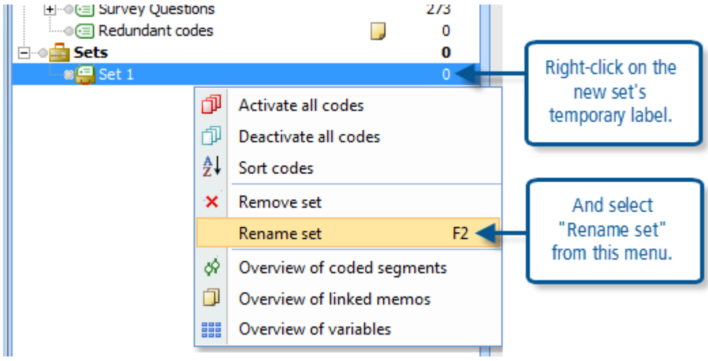
Figure 9.6.1 – Creating a new code set



Right-click on "Sets" and select this option.

The new set will appear just below the heading with the temporary name "Set 1". You can now rename it with a right-click context menu option on that label "Set 1".

Figure 9.6.2 – Rename a new set



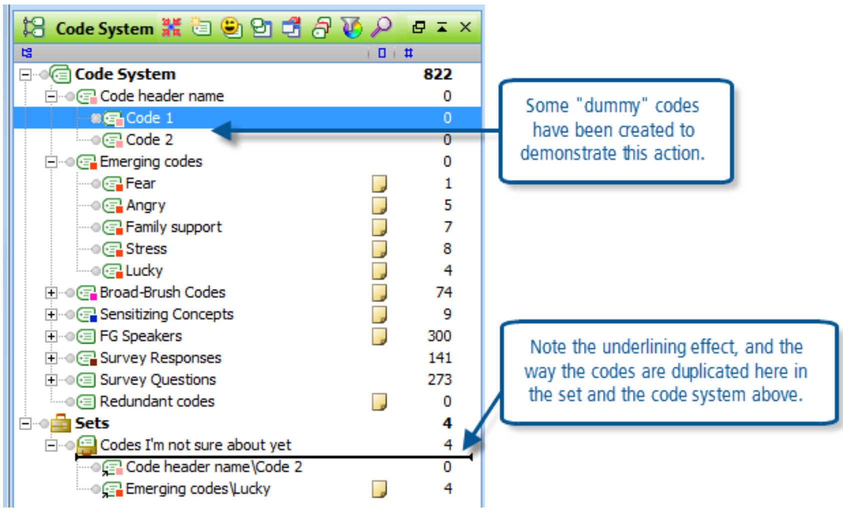
Right-click on the new set's temporary label.

And select "Rename set" from this menu.

You can now type in the suggested label, "Codes I'm not sure about yet", in place of the temporary label.

To add codes to this set, you can simply drag them from anywhere in the Code System and drop them on the appropriate set. What you will see is that the code remains in its original place, when you drag it and drop it on a set, while a copy of the code appears in the set. You will also find that you do not see the highlighting effect just before you release the mouse button to drop the code onto the set, only the underlining effect appears. This is illustrated in Figure 9.6.3.

Figure 9.6.3 – Adding codes to a set



Code System	822
Code header name	0
Code 1	0
Code 2	0
Emerging codes	0
Fear	1
Angry	5
Family support	7
Stress	8
Lucky	4
Broad-Brush Codes	74
Sensitizing Concepts	9
FG Speakers	300
Survey Responses	141
Survey Questions	273
Redundant codes	0
Sets	4
Codes I'm not sure about yet	4
Code header name\Code 2	0
Emerging codes\Lucky	4

Some "dummy" codes have been created to demonstrate this action.

Note the underlining effect, and the way the codes are duplicated here in the set and the code system above.

As can be seen in Figure 9.6.3, the full path is shown for each code in the set including the group header under which the code is found in the code system. A set can include codes from several different groups in the code system above it, that is the main advantage of using sets as you can create temporary groupings that cut across other structures.

Further sets can be created in the same way as the first one. Each will appear beneath the other sets in the lower part of the Code System window. Just like the code groups above them, code sets can be minimised by clicking on the "-" box to the left of the set name, and sets can be moved around with drag and drop procedures.

TIP: Code sets can be used for structural purposes as well as thematic ones. If you expand the group of codes that identify focus group speakers in this example you could create different code sets for the male and female speakers in those focus groups. These sets can then be used in analysis tables such as the Code Relations Browser (to be discussed in Chapter 13 exercises) to compare how males and females contributed to the focus group discussions.

Related to the ideas we mention in talking about virtual, alternative groupings in the coding scheme (above) is the need to keep on top of the management of analysis. In Chapter 10 managing interpretations efficiently is helped by creating a system which helps to keep track of associations and ideas. Chapter 10 is connected in a similar way to Chapter 11 where mapping tools provide other methods of expressing or visualizing associations.

Graham Hughes and Stefan Radiker 2014