

## Analysing Open-Ended Survey Question Data in ATLAS.ti

### 1.0 Data Preparation for ATLAS.ti (v5 & v6)

The following material provides step-by-step guidance for preparing the texts, collected from a large number of respondents answering several open-ended questions in a survey situation, for qualitative analysis using ATLAS.ti software. These instructions assume that the decision has been made to organise these texts into a separate document for each question, with each respondents' answer to one question included in each document. (For a discussion of the arguments for and against such a decision please see the material [here](#)). This way of organising data is not the mainstream procedure in ATLAS.ti and so these instructions may be seen as a "workaround" to achieve a satisfactory basis for analysis of open-ended survey data.

There are many stages in this procedure, some of which may not be relevant in certain circumstances. Some users may have alternative methods or short-cuts, in which case please use your own judgment as to which elements from these instructions to apply. Our purpose here is to provide a comprehensive guide, which has been tested and proved to work, for the benefit of those who have not achieved this task successfully before. These instructions refer to Microsoft Excel and Microsoft Word and are illustrated with screenshots from those programs, however the operations involved are not particularly sophisticated and almost any equivalent spreadsheet and word-processing program would probably be usable as alternatives.

#### Outline

- 1.1 Locate the set of response texts and a unique identifier (ID) for each respondent and copy them into Microsoft Excel.
- 1.2 Select the socio-demographic attribute variables that will be required to be available in ATLAS.ti to inform the qualitative analysis and copy them into Microsoft Excel.
- 1.3 In Microsoft Excel create structured alpha-numeric strings that incorporate the ID and variable data for each respondent.
- 1.4 Combine the ID strings from (1.3) and the associated response texts from (1.1) in adjacent columns of a spreadsheet.
- 1.5 Copy the spreadsheet data from (1.4) to Microsoft Word for formatting, converting to text, and saving in Rich Text Format (RTF).
- 1.6 Assign the RTF files to an ATLAS.ti Hermeneutic Unit (HU) as Primary Documents.
- 1.7 Create a thematic code in the HU for each value of each attribute variable in the ID strings from (1.3).
- 1.8 Run the ATLAS.ti Autocode routine for each thematic code from (1.7) for all of the Primary documents in the HU.
- 1.9 Check the accuracy of the autocoding process and investigate any discrepancies

#### Detailed Steps

- 1.1 Locate the set of response texts and a unique identifier (ID) for each respondent and copy them into Microsoft Excel.**

It is almost impossible to create separate instructions for every possible format in which open-ended survey data may be found, but most programs have capabilities to export such data in a format which can be read by Microsoft Excel.

**TIP:** Before carrying out any data processing, try to examine data in their most original format in order to identify some of the longest individual responses and to look for unusual characters in the texts. The longer responses may get truncated in some conversion processes (for example on being brought into SPSS if the 256 character default was not changed) and it is useful to be able to check that you have the fullest versions of all responses before you carry out analysis. Unusual characters, other than basic alpha-numeric and punctuation, sometimes affect conversion processes so it is a good idea to check that these have been copied faithfully before proceeding.

Each separate response must be linked to the relevant respondent identifier and some careful thought may be helpful at this point. Generally in quantitative surveys this is a serial number, not necessarily in a completely unbroken sequence as it may originate in a sampling procedure. However, qualitative researchers may be more used to using forenames, pseudonyms or initials to identify informants. For the purposes of the sort of analysis envisaged here it is very important that each identifier (ID) is unique, because it will be used to connect the texts with the socio-demographic attributes of the speakers and later to link the responses to different questions made by each speaker. Serial numbers are probably more reliable for this purpose.

The goal of this first step is to create a worksheet in Microsoft Excel for each question whose responses are to be analysed, with a row for each respondent and two columns of data – one with the respondents' IDs and the other with their response texts in full. It may be helpful to use the *word-wrap* and *auto-fit row-height* commands within Microsoft Excel so that the longest answers can be seen in full and checked for completeness.

It is likely that there will be some missing data; respondents who did not answer some questions, and so be aware of this at each data preparation stage. Some tables will include a row for every informant and others will only include those who answered that question. The vital principle to remember is that every response must be identified by the correct ID or serial number.

**TIP:** When you have completed this stage of data preparation, make sure that you have saved back-ups of the spreadsheets to a secure location so that you can return to this point if all goes wrong at a later stage.

## **1.2 Select the socio-demographic attribute variables that will be required to be available in ATLAS.ti to inform the qualitative analysis and copy them into Microsoft Excel.**

This is a crucial step in preparing data in this way for ATLAS.ti because when the whole process has been completed it will be extremely laborious to change your mind and add a further variable. (It would be laborious to add another variable because of the particular way we are recommending this data to be prepared; in conventional qualitative analysis using ATLAS.ti, with separate documents for each respondent, attribute variables can be added quite easily at any stage using Document Families. However, that will not be an effective organisational device at the end of this process.)

In undertaking this step, you need to be sure that you include all of the socio-demographic attribute variables that you may need in the analysis of any of the open-ended questions.

**TIP:** Be aware that the more variables and values that you include the longer it will take to prepare the data for analysis – so it is generally not a good idea to include every possible attribute variable. For more advice on these decisions please see the discussion on [this web page](#).

When you get to step 8 in this series you will have to carry out a separate autocoding routine within ATLAS.ti for each value of each attribute variable that you have selected at this step. So, for example, if you choose 7 attribute variables with an average of 6 possible values for each, you will have to carry out  $7 \times 6 = 42$  autocoding procedures for all of the data to be usable within ATLAS.ti. This provides a further justification for being very careful with this selection.

It is quite likely that the variable data is available to you in a statistical program like SPSS, in which case there is a straightforward procedure for transferring the data to Microsoft Excel. However, before you start the transfer it may be worth considering some recoding within SPSS in order to simplify the variables, shorten the value label strings and remove ambiguities or inadvertent duplications.

It is possible to alter some strings in Microsoft Excel, or even in Microsoft Word later on by using *find and replace* commands, but it is certainly easier and less error prone to do this in SPSS. In SPSS terminology, it is the “Value Labels” that you will be exporting and so these are the items that need to be made short and clear. For example, change gender labels to “M” and “F” rather than “male” and “female” and limit place names to their first 6 letters provided this doesn’t create duplication. However the number of different values each attribute variable can take may be significant so, for example, rather than use a numerical attribute variable showing actual ages in years bring them together in groups of 10 years span and label these “age15\24yr” etc. (Using this particular approach for organising data in ATLAS.ti there does not appear to be a way to use numerical age values in a sorting function).

The procedure for exporting data from SPSS to Excel uses the *Save As* command in SPSS.

**TIP:** This can be done in a single step but a two stage process provides a mid-point to return to if something goes wrong.

Having completed the selection and recoding decisions and processes and saved the full dataset, start the *Save as* command to create a new data file with all of the cases but only the particular variables that are wanted for analysis in ATLAS.ti. Click on the Variables button to open the dialogue screen that allows you to select a subset of the variables into a new dataset.

It will probably be best to click on the *Drop All* button first, to uncheck every variable box, and then manually tick the box for each variable that you have decided to use (in its recoded form if you made that a new variable). Make sure that you include the ID or serial number variable in the list and, if the question texts have already been brought into SPSS it would be very useful to include those ‘variables’ as well (see step 4 below).

When this selection operation is complete, click on the *Continue* button, enter an appropriate new name and path for the file to be stored at (you don’t want to overwrite the master data file and thus delete valuable data), leave the file type as “\*.sav”, and save the file.

Now in SPSS open the new data file that you have just created and close the master data file (which will still be open if you are using SPSS v15 or higher), and check that you have all of the data that you expect to bring into ATLAS.ti.

If necessary repeat the last steps to correct any omissions or errors.

When you are satisfied that all is correct, start another *Save as* command from within SPSS. This time on the main dialogue screen change the *Save as type* field to “Excel 97 through 2003 (\*.xls)” by using the pull-down menu. When this is selected, two further options below come into play. You should tick both of them – *write variable names to spreadsheet* and *save value labels where defined instead of data values*. The first of these will provide helpful identification of each attribute as column headers in Microsoft Excel, the second makes sure that you can interpret the attribute in a qualitative environment (“Wrkpart” makes more sense than “3” in the context of work status, see Figure 1.).

Finally enter a file name, which can be similar to the one used for the temporary SPSS file because it will have a different extension (.xls instead of .sav) and hit the Save button. (If you have MS Office 2007 available you can use that version to save to “\*.xlsx”).

### 1.3 In Microsoft Excel create structured alpha-numeric strings that incorporate the ID and variable data for each respondent.

The next step makes use of the “concatenation” function in Microsoft Excel. This function combines the contents of two or more separate cells in a single string. Doing this will result in a unique identifying text string for each survey respondent comprising the relevant attribute variable information for use in autocoding responses in ATLAS.ti.

The basic syntax of the command is `=(A2&B2)` to combine the contents of cell A2 followed by those in B2. When additional characters within inverted commas are included they will be added as further text.

Thus `=(A2&“-“&B2)` will place the contents of cell A2, followed by a hyphen, and then the contents of B2 in the cell where this logic is located.

Open the spreadsheet created in step 2 above. After adjusting column widths it should look something like the screenshot shown in Figure 1, below. It is a good idea to have the respondent IDs (serial numbers) in the first column but the order of the other columns is less significant. Note how, in this screenshot, each variable has its own column (“age” is in column C and “work” is in column E etc.).

Figure 1: Microsoft Excel screenshot after transfer of variable data from SPSS

	A	B	C	D	E	F	G	H	I
1	Respondent	sex	age	areax	work	ghflowarn	ghwhere	qlongs	
2	RESP.04402	M	age18\24yr	Worces	Student	fNwN	Notfl	Prp05\10yr	
3	RESP.04403	F	age55\64yr	Worces	Retired	fYwY	Outer	Prp20\99yr	
4	RESP.04407	F	age18\24yr	Worces	Wrkfull	fYwN	Outer	Prp03\05yr	
5	RESP.04408	F	age35\44yr	Worces	Wrkfull	fYwN	Outer	Prp05\10yr	
6	RESP.04409	M	age55\64yr	Worces	Wrkfull	fNwY	Notfl	Prp03\05yr	
7	RESP.04410	M	age55\64yr	Worces	Retired	fYwN	Outer	Prp20\99yr	
8	RESP.04417	M	age55\64yr	Worces	Wrkfull	fNwN	Notfl	Prp20\99yr	
9	RESP.04418	F	age25\34yr	Worces	Lkghome	fNwY	Notfl	Prp01\03yr	
10	RESP.04419	M	age65\74yr	Worces	Retired	fNwY	Notfl	Prp01\03yr	
11	RESP.04420	F	age45\54yr	Worces	Wrkfull	fNwN	Notfl	Prp20\99yr	
12	RESP.04421	F	age65\74yr	Worces	Retired	fNwN	Notfl	Prp10\20yr	
13	RESP.04422	M	age55\64yr	Worces	Wrkpart	fYwN	Outer	Prp10\20yr	
14	RESP.04423	M	age45\54yr	Worces	Wrkfull	fYwN	House	Prp20\99yr	
15	RESP.04424	F	age65\74yr	Worces	Retired	fYwN	House	Prp20\99yr	

Next, in a blank column to the right on the same spreadsheet page, create the concatenation logic to combine all of your variables in a single text string for the first respondent, using hyphen symbols to separate each element. In the example shown below in Figure 2 the following logic was typed into cell J2:

=(A2&"-"&B2&"-"&C2&"-"&D2&"-"&E2&"-"&F2&"-"&G2&"-"&H2)

That logic was then copied and pasted into all of the rows of column J where data existed, to give the screenshot shown in Figure 2. As these are relative references they will adjust in the pasting process to refer to the equivalent cells on each row.

Note in Figure 2 how all of the text is held in a single column (J in this example).

*Figure 2: Combine attributes into a single text string*

I	J	K
	Atlas Identifier	
	RESP.04402-M-age18\24yr-Worces-Student-fNwN-Notfl-Prp05\10yr	
	RESP.04403-F-age55\64yr-Worces-Retired-fYwY-Outer-Prp20\99yr	
	RESP.04407-F-age18\24yr-Worces-Wrkfull-fYwN-Outer-Prp03\05yr	
	RESP.04408-F-age35\44yr-Worces-Wrkfull-fYwN-Outer-Prp05\10yr	
	RESP.04409-M-age55\64yr-Worces-Wrkfull-fNwY-Notfl-Prp03\05yr	
	RESP.04410-M-age55\64yr-Worces-Retired-fYwN-Outer-Prp20\99yr	
	RESP.04417-M-age55\64yr-Worces-Wrkfull-fNwN-Notfl-Prp20\99yr	
	RESP.04418-F-age25\34yr-Worces-Lkghome-fNwY-Notfl-Prp01\03yr	
	RESP.04419-M-age65\74yr-Worces-Retired-fNwY-Notfl-Prp01\03yr	
	RESP.04420-F-age45\54yr-Worces-Wrkfull-fNwN-Notfl-Prp20\99yr	
	RESP.04421-F-age65\74yr-Worces-Retired-fNwN-Notfl-Prp10\20yr	
	RESP.04422-M-age55\64yr-Worces-Wrkpart-fYwN-Outer-Prp10\20yr	
	RESP.04423-M-age45\54yr-Worces-Wrkfull-fYwN-House-Prp20\99yr	
	RESP.04424-F-age65\74yr-Worces-Retired-fYwN-House-Prp20\99yr	

There is one further step to carry out before these text strings can be used: a copy and paste special operation to remove the logic that created the strings and convert them to pure text.

Highlight the whole of the column you have just created (in this example column J), click on the *Copy* button (or press Ctrl + C), move the cursor to a blank cell (say K2 in this example) and select the *Paste Special* command. From the choice of options that appears, select *Paste – Values* and then click on *OK* to effect the command. The text strings should be copied across to the new location. These new versions of the text strings can be copied to any other location with simple copy and paste commands, whereas if you tried to copy the version in J2 to another cell you would probably get a “#REF” result because in the new location the concatenation logic would not find sensible data to combine.

Save the workbook to secure the text string data you have now created.

#### **1.4 Combine the ID strings from (1.3) and the associated response texts from (1.1) in adjacent columns of a spreadsheet.**

The next operation could take one of many forms, depending on the particular data and circumstances that you face. The objective is to create a spreadsheet that looks something like the example shown in Figure 3, where each response text (in column B) is associated with the correct attribute string for



that respondent (in column C) on the same row. (Not shown in this screenshot is column A, where the serial numbers brought in with the response texts at step 1 were matched-up with their equivalents in the attribute strings).

It is important to emphasise at this point that the attribute strings are placed to the right of the text responses, even though this may not be what you might expect. This is necessary so that when the data eventually is used in ATLAS.ti the coded quotations will be much more easily interpreted in various screen and printed outputs.

**TIP:** We have tried this in both ways and our conclusion is that having the attribute strings **after** the text responses produces more usable data.

The likely problem is that you have probably got many more attribute strings than you have response texts for any one question, because some respondents did not provide an answer to that question. If you have a large number of respondents (and this advice is based on the assumption that you have) you will want to avoid individual copy/paste operations to bring the attribute strings in one at a time because this will be a very slow process.

A quicker solution to the problem would be to use an empty column adjacent to the master list of attribute strings created at the end of step 3 (say column L in this example) and manually enter a "1" in each row for which you have a response text, then highlighting the two columns (K and L) use the Data-Sort command to sort them according to the data in column L.

Provided the numerical order was not disturbed, by selecting just the cells in column K which have a "1" in column L, you should now have a block of attribute strings which can be copied into the response text spreadsheet in a single operation and which should match up with the serial numbers already there.

**TIP:** There may be shortcuts possible in some circumstances depending on the nature of your data. We recommend you experiment on a small number of cases to find a procedure that works for your data.

*Figure 3: In Microsoft Excel, combine attribute strings with texts for analysis*

	B	C	D
1	QMORE		
2	step by steps approaches "in case of flooding" to do this that or	RESP.04408-F-age35\44yr-Worces-Wrkfull-fYwN-Outer-Prp05\10yr	
3	the other i.e. A checklist		
4	needed someone to advice of help available p. she heard	RESP.04424-F-age65\74yr-Worces-Retired-fYwN-House-Prp20\99yr	
5	nothing at all		
6	knowledge of electricity is dangerous for elderly people and they	RESP.04443-F-age65\74yr-Worces-Retired-fNwY-Notfl-Prp05\10yr	
7	needed to know what to do		
8	something! Respondent had no advice at all	RESP.04454-F-age65\74yr-Worces-Retired-fNwN-Notfl-Prp20\99yr	
9	somebody from the council physically coming around during the	RESP.04808-M-age35\44yr-Woodfo-Wrkfull-fYwN-House-Prp01\03yr	
10	day advising, and telling you what could happen, flood line not		
11	specific, when it said essex we thought of rural areas,		
12	information before and not after	RESP.04822-F-age55\64yr-Woodfo-Retired-fYwY-Outer-Prp20\99yr	
13	what was likely to happen, we did not have a clue	RESP.04823-F-age25\34yr-Woodfo-Lkghome-fYwN-Outer-Prp03\05yr	
14	all the advice we received afterwards, e.g. Move car, wash cuts	RESP.04824-M-age45\54yr-Woodfo-Wrkfull-fYwY-Outer-Prp10\20yr	
15	with disinfectant		
16	more help from the police, wrong attitude, the firebrigade very	RESP.04825-M-age35\44yr-Woodfo-Wrkfull-fYwN-Outer-Prp05\10yr	
17	helpful		
18	more warning in advance, use of a siren, personal contact	RESP.04830-M-age55\64yr-Woodfo-Wrkfull-fYwY-Outer-Prp20\99yr	
19	being informed earlier, the police did not knock, did not find out	RESP.04832-F-age25\34yr-Woodfo-Wrkfull-fYwN-Outer-Prp05\10yr	
20	till 30 mins before.		
21	we think that somebody cocked up river with flood gates, and	RESP.04833-F-age55\64yr-Woodfo-Wrkfull-fYwY-Outer-Prp10\20yr	
22	we keep getting warnings but in the 15 years i have lived here		
23	we have had nothing like it before or since		
24	once it happened, nobody came around, if somebody official	RESP.04834-F-age55\64yr-Woodfo-Wrkfull-fYwY-Outer-Prp10\20yr	
25	came to the door and informed us of what was happening and		
26	what to do		

Immediately after such an operation, select a few rows of data to check that the correct attribute string is associated with each response text, particularly at the bottom of the list.

**TIP:** Note in Figure 3 how the *wrap text* alignment and *auto fit row height* command have been used to display the varying lengths of response texts in full within the limitations of screen size.

Finally, save the workbooks to secure this stage of your work.

### **1.5 Copy the spreadsheet data from (1.4) to Microsoft Word for formatting, converting to text, and saving in Rich Text Format (RTF).**

Before starting the process of copying a spreadsheet containing response texts and attribute strings to Microsoft Word, make sure that there is an empty column to the right of the texts (column D in Figure 3 above). Also, open Microsoft Word and prepare an empty document with landscape layout, due to the width of the materials you will be copying.

In Microsoft Excel, highlight the two columns shown plus a third (blank) column to the right of the response texts. This extra column plays an essential role later in the autocoding process within ATLAS.ti. Copy the whole block of three columns with all of the data and paste it into the empty Microsoft Word document (a simple Paste command is sufficient). Wait for the document writing icon at the bottom of the word screen to indicate that the process has completed (which may take a few minutes if the document is large).

**TIP:** It is visually useful to change the Respondent Identifier information to a different colour (as shown in Figure 4) in order to distinguish it visually from the respondents' words. (See figure 6 below for a further illustration of this, note how the black text can be read separately from the blue attribute strings). You can simplify this task, if you have several sets of texts to format, by setting a style set to match your requirements and then using the style commands as shortcuts.

Figure 4: After pasting into Microsoft Word, format table elements in columns

QMORE	
step by steps approaches "in case of flooding" to do this that or the other i.e. A <u>checksheet</u>	RESP 04408-F-age35/44yr-Worcester-Wikfull-FYwN-Outer-Prp05/10yr
needed someone to advice of help <u>available</u> <u>p.she</u> heard nothing at all	RESP 04424-F-age65/74yr-Worcester-Retired-FYwN-House-Prp20/09yr
knowledge of electricity is dangerous for elderly people and they needed to know what to do	RESP 04443-F-age65/74yr-Worcester-Retired-IRwY-NotFI-Prp05/10yr
something! Respondent had no advice at all	RESP 04454-F-age65/74yr-Worcester-Retired-IRwN-NotFI-Prp20/09yr
somebody from the council physically coming around during the day advising, and telling you what could happen, flood line not specific, when it said <u>essex</u> we thought of rural areas,	RESP 04808-M-age35/44yr-Woodfo-Wikfull-FYwN-House-Prp01/03yr
information before and not after	RESP 04822-F-age55/64yr-Woodfo-Retired-FYwY-Outer-Prp20/09yr
what was likely to happen, we did not have a clue	RESP 04823-F-age25/34yr-Woodfo-Lkghome-FYwN-Outer-Prp03/05yr
all the advice we received afterwards, e.g. Move car, wash cuts with disinfectant	RESP 04824-M-age45/54yr-Woodfo-Wikfull-FYwY-Outer-Prp10/20yr
more help from the police, wrong attitude, the <u>firebrigade</u> very helpful	RESP 04825-M-age35/44yr-Woodfo-Wikfull-FYwN-Outer-Prp05/10yr
more warning in advance, use of a siren, personal contact	RESP 04830-M-age55/64yr-Woodfo-Wikfull-FYwY-Outer-Prp20/09yr
<u>being</u> informed earlier, the police did not knock, did not find out till 30 <u>mins</u> before.	RESP 04832-F-age25/34yr-Woodfo-Wikfull-FYwN-Outer-Prp05/10yr
we think that somebody cocked up river with flood gates, and we keep getting warnings but in the 15 years i have lived here we have had nothing like it before or since	RESP 04833-F-age55/64yr-Woodfo-Wikfull-FYwY-Outer-Prp10/20yr
once it happened, nobody came around, if somebody official came to the door and informed us of what was happening and what to do	RESP 04834-F-age55/64yr-Woodfo-Wikfull-FYwY-Outer-Prp10/20yr
for us would have been useful to understand the route of the flood we presumed it would flood from the front of the house where as it didn't it flooded from the back where we weren't prepared	RESP 07202-F-age45/54yr-Woodfo-Wikfull-FYwY-Outer-Prp03/05yr

When the colours and formats have been set to your satisfaction you should convert the table layout to a single column of text.

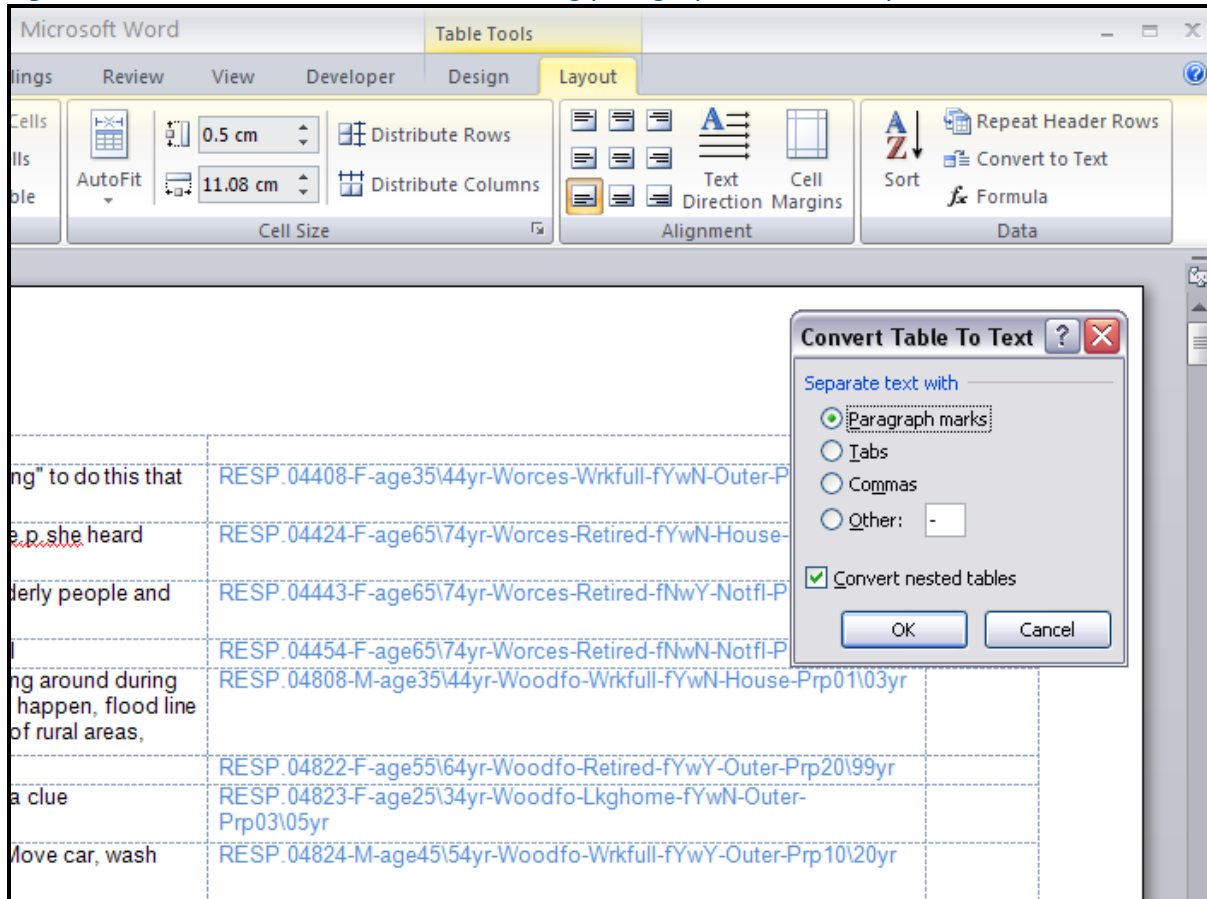
If you are using Microsoft Word with the MS Office 2007 version, to locate the command for converting the table to text, position the cursor in the table and select the *Layout* ribbon under *Table Tools*, *Convert to text* will be visible at the right hand end of the ribbon. As shown in Figure 5, select the option to separate text with "Paragraph marks".

Once again, wait for the text writing icon to indicate the completion of the procedure.

In Microsoft Word 97-2003 position the cursor somewhere in the table containing your data, select the *Table* command, then the *Convert* options, and finally click on the *Table to Text* command. As shown in Figure 5, select the option to separate text with "Paragraph marks".



Figure 5: In Word, convert table to text using paragraph marks as separators

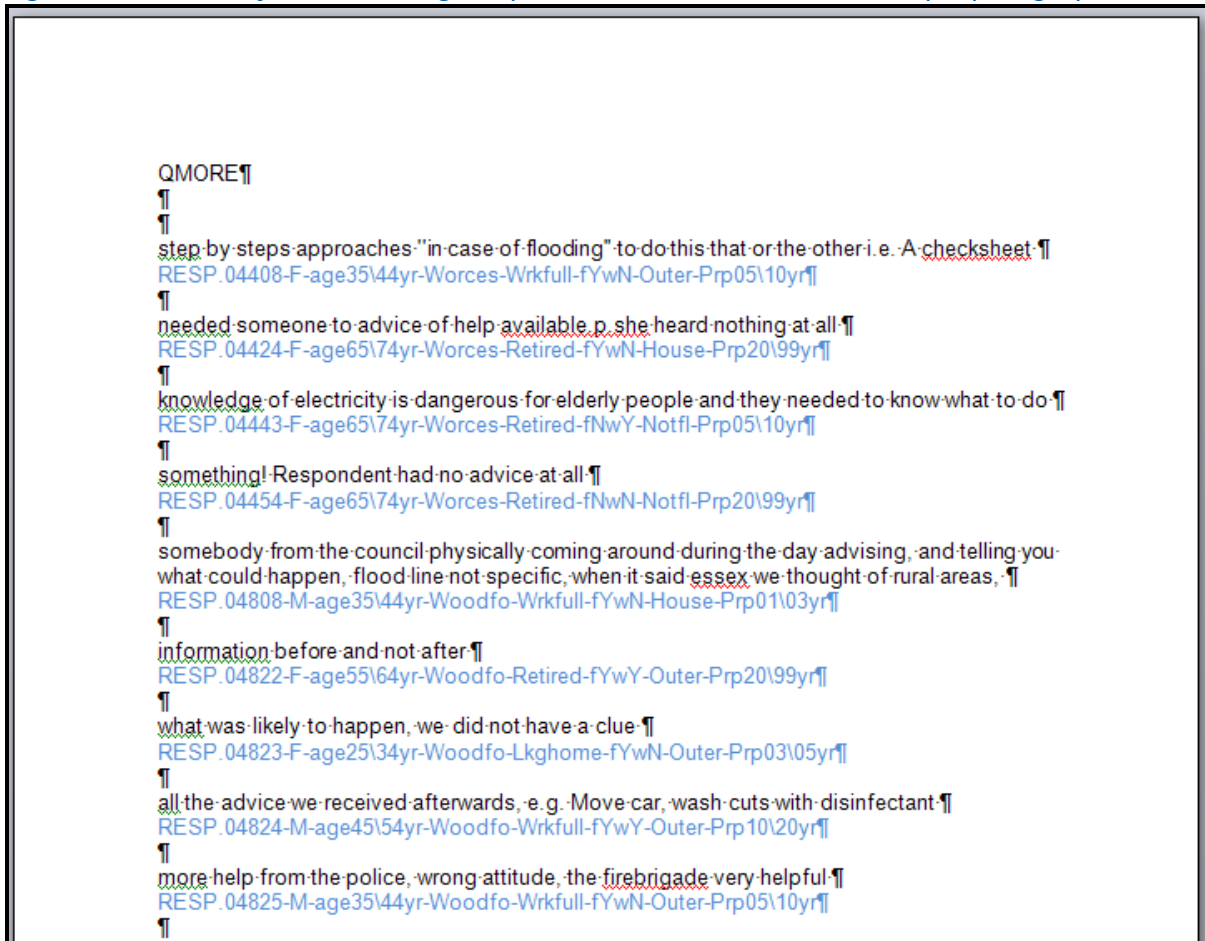


Switch on the *Show/Hide* icon ("¶") in order to check the allocation of paragraph marks in your document. It is essential that there are two such marks between each attribute string and the following response text. This can be seen in Figure 6 below where there is a ¶ symbol at the end of each (blue) attribute line and a second one on a blank line before the next (black) text line.

**TIP:** The third (blank) column that was copied from Microsoft Excel (see the instructions between figures 3 and 4 above) has triggered the extra ¶ markers, so if these are not present in your data you have probably missed that particular instruction.

Provided that all is well with the data format, switch the orientation to Portrait (with the command *File / Page Setup*, and then click on the Margins tab to see the orientation options) to achieve sensible line lengths for longer text strings. Once again check the longest texts to make sure that no data has been lost in the processes outlined above.

Figure 6: In Microsoft Word, change to portrait orientation, check multiple paragraph marks

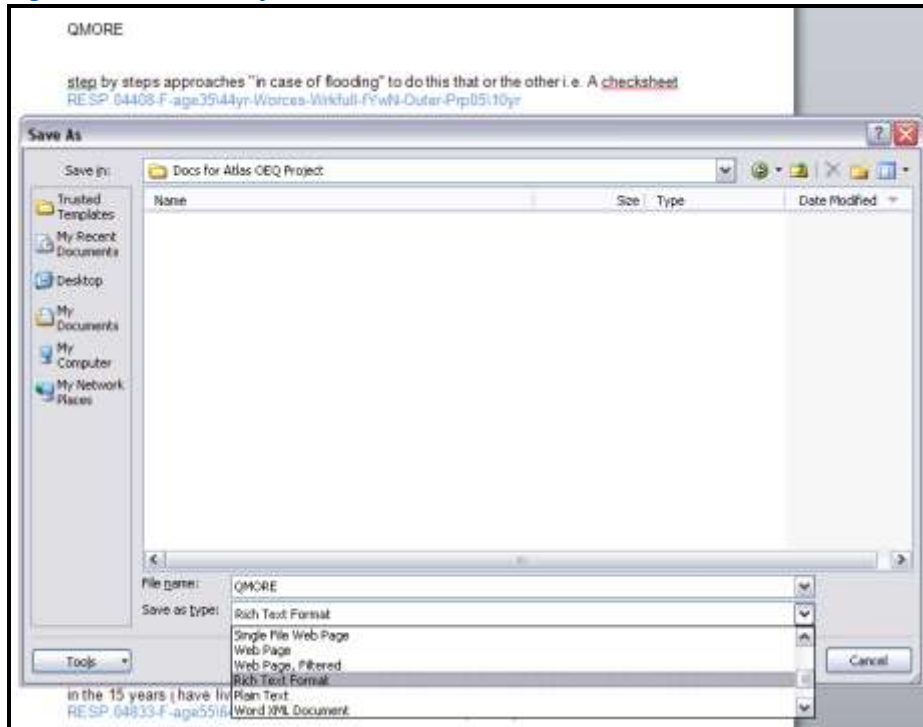


Note in Figure 6 how the (blue) attribute strings have been placed **beneath** the (black) response texts to which they relate. This is achieved by placing them to the right of the texts in Microsoft Excel, as discussed at Step 4 above. As mentioned before, it is our considered opinion that this sequence will produce the most usable format of the data in ATLAS.ti, even though it may seem counter-intuitive to some researchers.

TIP: We recommend that you save your final documents in *Rich Text Format* for ATLAS.ti. Other formats are usable, but RTF appears to work best with this sort of data. The colour formatting is preserved and it is possible to edit the texts within ATLAS.ti (for example to correct spelling mistakes which may otherwise reduce the effectiveness of text searches).

It is also good practice to save the final versions in a secure location where they will not be altered in any way outside the ATLAS.ti program (because the documents are not stored within the Hermeneutic Unit in ATLAS.ti they may otherwise be vulnerable to subsequent 'external' editing which would corrupt the quotation links and make the data unusable). This is shown in Figure 7 below.

Figure 7: In Microsoft Word, Save As - Rich Text Format



Save back-up copies of the final versions in a separate secure location should you need them for other purposes.

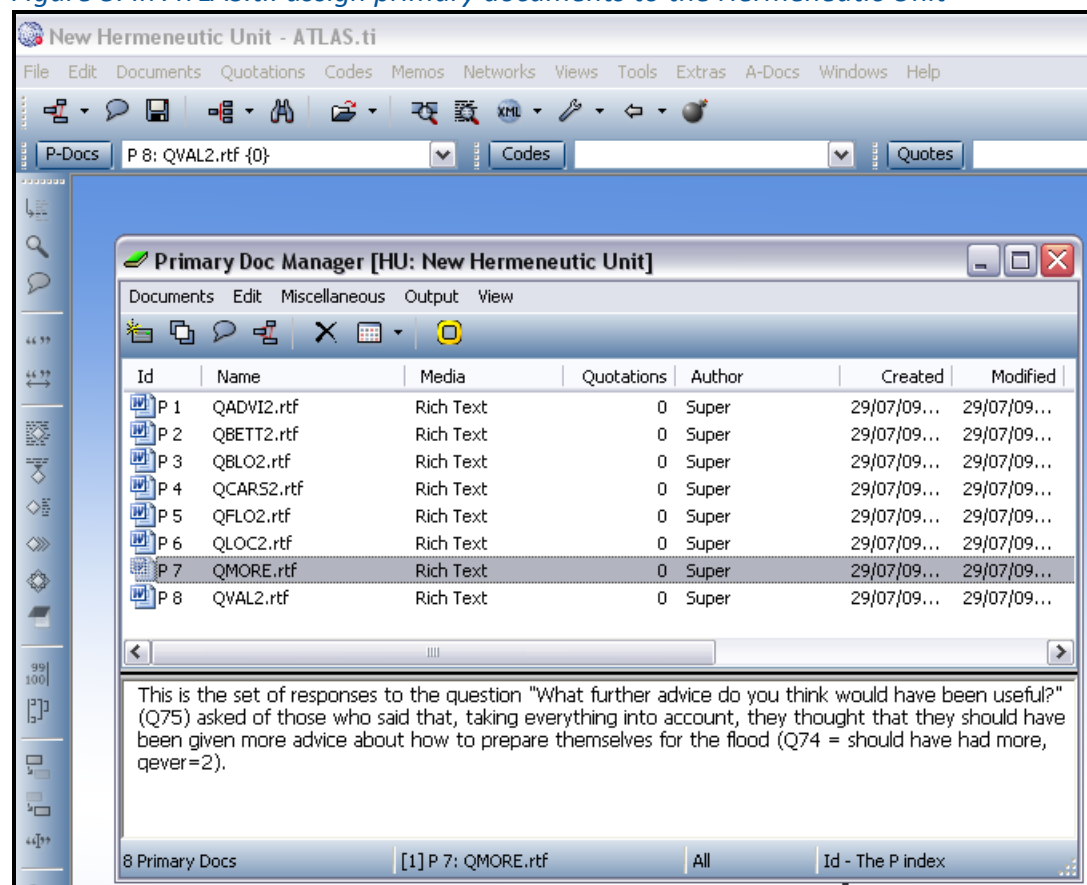
### 1.6 Assign the RTF files to an ATLAS.ti Hermeneutic Unit (HU) as Primary Documents.

In this example there are 8 separate documents, each holding the set of responses to a single open-ended question in a survey about flooding. These have all been created as indicated above and then saved in Rich Text Format. The ATLAS.ti (v6) program has then been opened and a new Hermeneutic Unit created for this project.

Using the *Documents / Assign* command it is straightforward to select and assign all 8 documents in a single procedure. Remember that ATLAS.ti uses an “external database” structure, so the documents are not copied into ATLAS.ti, rather the assignment process creates links in the ATLAS.ti Hermeneutic Unit to the location where the documents are stored.

**TIP:** After completing the assignment process it is recommended that information about the nature of each document is typed into its properties field. Figure 8 below shows the information recorded for primary document 7, “QMORE.rtf”.

Figure 8: In ATLAS.ti. assign primary documents to the Hermeneutic Unit



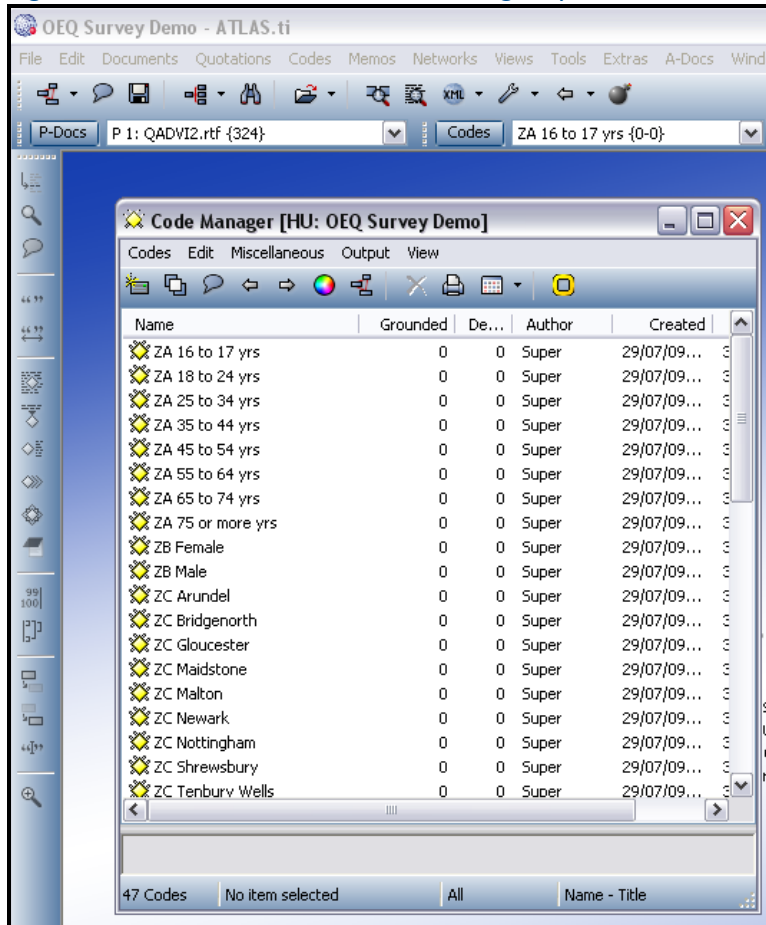
### 1.7 Create a thematic code in the HU for each value of each attribute variable in the ID strings from (1.3).

It is now necessary to create a code for each attribute value identified at step 2 above. In this example we have 7 sets of attributes (excluding the respondent ID) which take between 2 and 12 values each, making a total of 47 possible attribute-type codes.

**TIP:** In order to keep related codes together in the alphabetical code list it is suggested that each code name is prefixed with a grouping letter. In Figure 9, below, it can be seen that the "age" codes have been prefixed with "ZA", the gender codes with "ZB", and the location codes with "ZC". The Z prefix pushes all of these socio-demographic variable codes to the end of the coding list, while the A, B, and C codes separate the variable groups.

Later, when thematic codes are created for the main analysis it will be advisable to give them suitable prefixes to keep them in groups separate from these attribute codes in the code Manager listing.

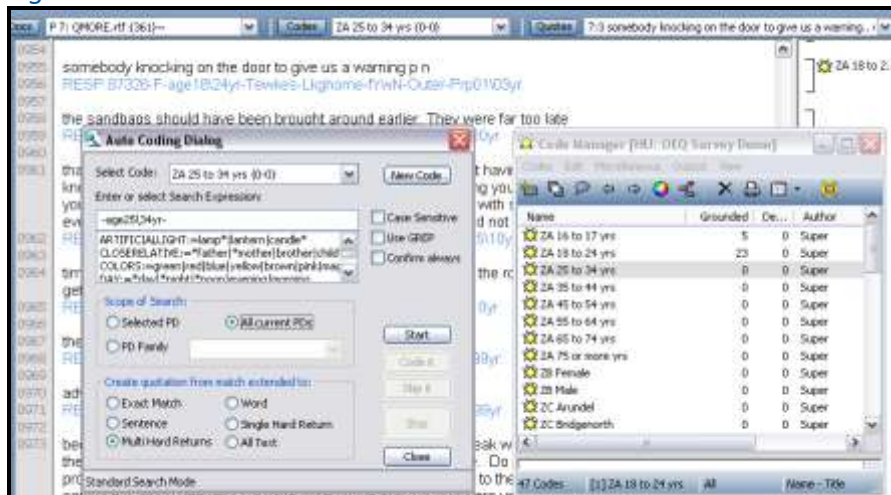
Figure 9: Create all attribute codes, in groups



## 1.8 Run the ATLAS.ti Autocode routine for each thematic code from (1.7) for all of the Primary documents in the HU.

This is the stage where all of the work carried out in accordance with the instructions above comes together and, hopefully, makes sense. The screenshot in Figure 10 below is taken from ATLAS.ti v6 (but would be almost identical in v5).

Figure 10: Autocode routine screens





The autocode routine will have to be run separately for each code set up in the previous step, so in this example it had to be run 47 times.

Select *Codes / Coding / Auto Coding* from either the Main Menu commands or the Code Manager window to open the dialogue box shown on the left in Figure 10.

Field: *Select Code:* – use the drop-down menu to choose the first code from the alphabetical list. (In the above example two codes have been run and the third is now selected). But see the comment below about which attribute group to autocode first.

Field: *Enter ... Search Expression:* – here you use the code names that were set-up at step 2 along with the hyphens that were inserted into the attribute strings at step 3. In the above example the string “-age25\34yr-” is being used. Now you should see why the hyphens were so necessary so that this routine can distinguish “-F-” for female from the initial letter of “Flooded”. If you have used consistent structures for each set of codes it should be a simple matter to edit the string as each subsequent code in the set is run.

Field: *Scope of Search:* – set this to “All current PDs” so that the codes are attached to all of the documents in one operation.

Field: *Create quotation from match extended to:* – set this to “Multi Hard Returns”. This is where the two paragraph marks between each response becomes significant (see step 5). The code will be applied to all of the response text and the whole attribute string for each applicable respondent (this can be seen at the top of figure 10 where the code “ZA 18 to 24” and its bracket are visible).

The three fields on the right of the dialogue box, *case sensitive*, *use GREP*, and *confirm always* should not be applicable if you have set everything up correctly. *Confirm always* is often checked by default so you may need to uncheck this option. Press *Start* and wait for the process to be completed.

It is tempting to try to start the next autocode routine too early because the number of hits for the current code appears very quickly in the Code Manager window (see “23” opposite “ZA 18 to 24 yrs” in Figure 10). However, the sign that the process has completed is when the open document behind the dialogue box refreshes and shows the last text given that code, so make sure that you can see a highlighted string containing the attribute you have just run in the active primary document before you start the next one. The routine may take some time, depending on the amount of data you have.

Keep repeating the routine until you have run it for every code in your attribute list, as mentioned above this was 47 times in this example. At this point you may appreciate the need for economy and caution when you selected and simplified the attributes at step 2 above.

**TIP:** Consider which attribute group to code first, as this will influence the way in which code output reports appear in the analysis stage of your work. As each block of text is coded, a “quotation” is created and these are automatically numbered. When an output report is created the default order for showing quotations is this numerical sequence, so the first attribute group that is autcoded here will determine the order in which these quotations are reported. (We recommend for open-ended survey data that all thematic coding uses the same quotations as the attribute coding where possible, so that codes co-occur). Using an attribute with fewer values (such as gender) as the first coding variable will minimise the grouping effect that this creates, whereas an attribute like working status may generate some confusion if it has been used as the first coding variable.

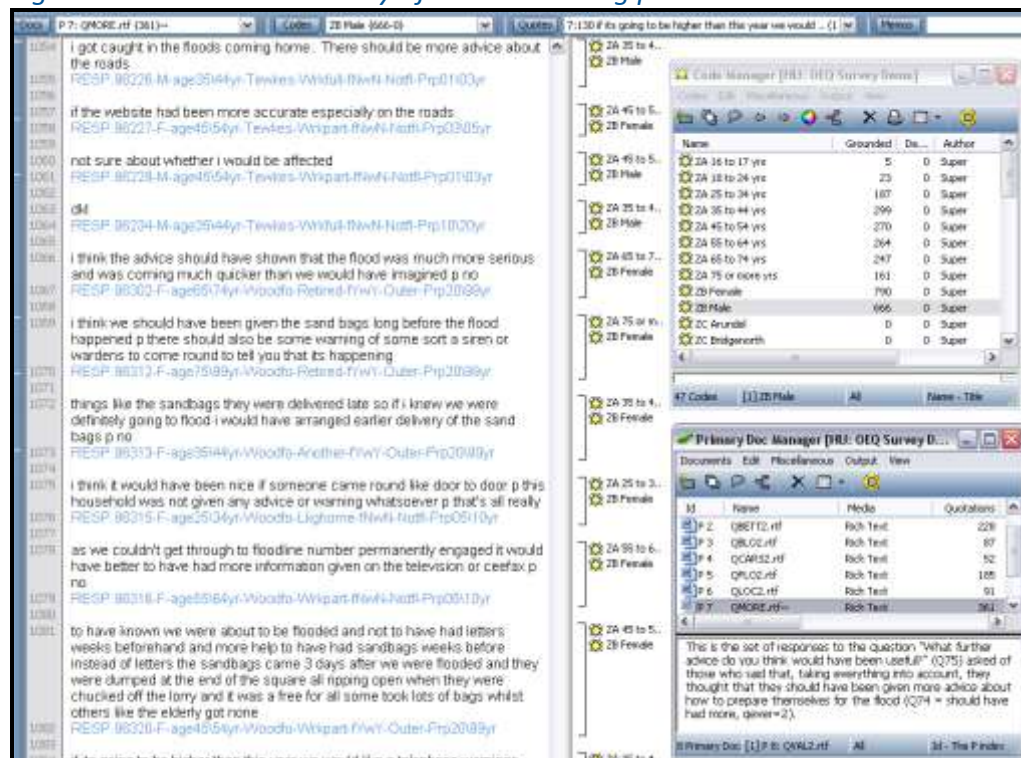
## 1.9 Check the accuracy of the autocoding process and investigate any discrepancies

Figure 11, below, shows the example at the stage when just two of the attribute variables had been fully coded. This allows some points to be demonstrated more clearly than would be possible when all of the attributes had been coded.

In Figure 11 one document has been opened (P7: QMORE.rtf) and the Code Manager and Primary Document Manager windows have been positioned so that the code brackets are visible in the margin. Autocoding has been completed on the Age and Gender variables. It should be apparent that each text in the open document has exactly two codes applied to it, a “ZA” code and a “ZB” code. But this would provide an extremely tedious method of visually checking the completeness of the process with several documents and many respondents.

However a little work with a calculator will provide more effective reassurance. In ATLAS.ti terminology each attribute string and its associated text is a separate “quotation”. Any further autocoding of the remaining variables in this example will not result in any more quotations, it will merely add coding density to the set of quotations that has now been created. In Figure 11 if you add up the number of quotations indicated in the Primary Document Manager window for all 8 documents (324+228+87+52+185+91+361+128) it comes to 1,456. If you add up the values in the “Grounded” column of the Code Manager Window for each variable set you should get the same total (for Age it is 5+23+187+299+270+264+247+161=1,456 and for Gender it is 790+666=1,456). As a final check, if you open the Quotations Manager window it should display the same total in its lower left corner.

Figure 11: Check the accuracy of the autocoding process



**TIP:** As you work through your variables keep a running total of the grounded density figure for each set of codes and check that it comes to whatever the total number of quotations is in your data. Please note that this total will not be the same as the number of respondents in your survey, or even the number of respondents multiplied by the number of questions, because each respondent may have answered varying numbers of questions.

You are now ready to begin the qualitative analysis of the texts. At any stage, should you wish to look at the responses made by a subset of your sample, say by males over the age of 65, the query tool can be used with a careful selection of the attribute codes to display those texts.

**TIP:** Some users may be tempted to start thematic or conceptual coding before they have completed the tedious autocoding of the socio-demographic variable data, but we would advise against this for two reasons. Firstly, the numerical checks on the accuracy of the autocoding process just explained may not be so straightforward and clear if some differently specified quotations have been created for thematic purposes. As a result, an error may go undetected. Secondly, it would be a good idea to create a back-up of the project with all of the autocoding complete and checked, as this would be a useful point to return to if something went wrong during the thematic coding.

## 2.0 Analysing Open-Ended Survey Question Data in CAQDAS Packages: Initial Coding Approaches for ATLAS.ti

There are, of course, many different ways to analyse responses to open-ended questions. This page is not a step-by-step guide on how to conduct analysis, it is rather a series of observations about how the features of ATLAS.ti might interact with a particular type of dataset. This page should be read in the context of the related materials concerning the use of ATLAS.ti for the analysis of open-ended survey question data, accessible from the main *Analysing Survey Data* page.

The tools discussed below are illustrated with examples from the same post flooding event survey that was used to illustrate the data preparation processes. For a summary of the project from which this data derives see [here](#). This data is characterised by a fairly large number of short statements.

This webpage was edited in December 2010 when the current version of ATLAS.ti was 6.2.16. If you cannot find some of the facilities mentioned it may be because you are using an earlier version of the software.

### Summary:

- 2.1 Reading the texts – by respondent or by question?
- 2.2 Developing a coding scheme – manually or by using word frequency tools?
- 2.3 Text searching and autocoding.
- 2.4 Coding – data indexing versus data reduction.
- 2.5 Checking summarising codes – consistency and omissions.
- 2.6 Looking for similarities or differences?

### Details:

#### 2.1 Reading the texts – by respondent or by question?

In ATLAS.ti the decision as to which way to read the texts had to be taken at the data preparation stage. If the texts were organised in the way suggested in the data preparation instructions on this website, that is to say grouped by the question to which they refer, then there is no effective way of displaying them in the opposite way once they have been assigned to a Hermeneutic Unit (HU) and coded by the attribute variables. All that can be done to read all of the responses made by any one respondent is to carry out a text search across all of the documents using that respondent's unique ID string as the search term. This will locate their responses one at a time, within the source context of the question documents and so will prove to be a tedious method to use other than very occasionally.

If the texts have been imported using the "Survey Import" routine that was added to ATLAS.ti in version 6.2 then a separate Primary Document will have been created for each respondent and a thematic code will have been applied to identify each separate question within those documents. By using the output for a single question code it will be possible to read through all of the responses for that question isolated from the other question responses and this may be an effective basis for manual coding of the themes within a single question. However it will not be possible to use the automation tools (to be described below) effectively with the data in this form. So, where the number of respondents is very large and a degree of automation is desirable, it is still worth considering



preparing the data on a document per question basis and using the procedures described below to speed-up the work of analysis.

The remainder of this document will look at the strategies that can be used in ATLAS.ti to analyse a large number of response texts that have been organised in the document per question format. For a discussion of the merits of this approach versus the document per case approach see here. For this approach it is a simple matter to open a primary document containing all of the answers to a single question and read through them systematically by scrolling the document in the main screen

Figure 12: A basic working screen layout in ATLAS.ti

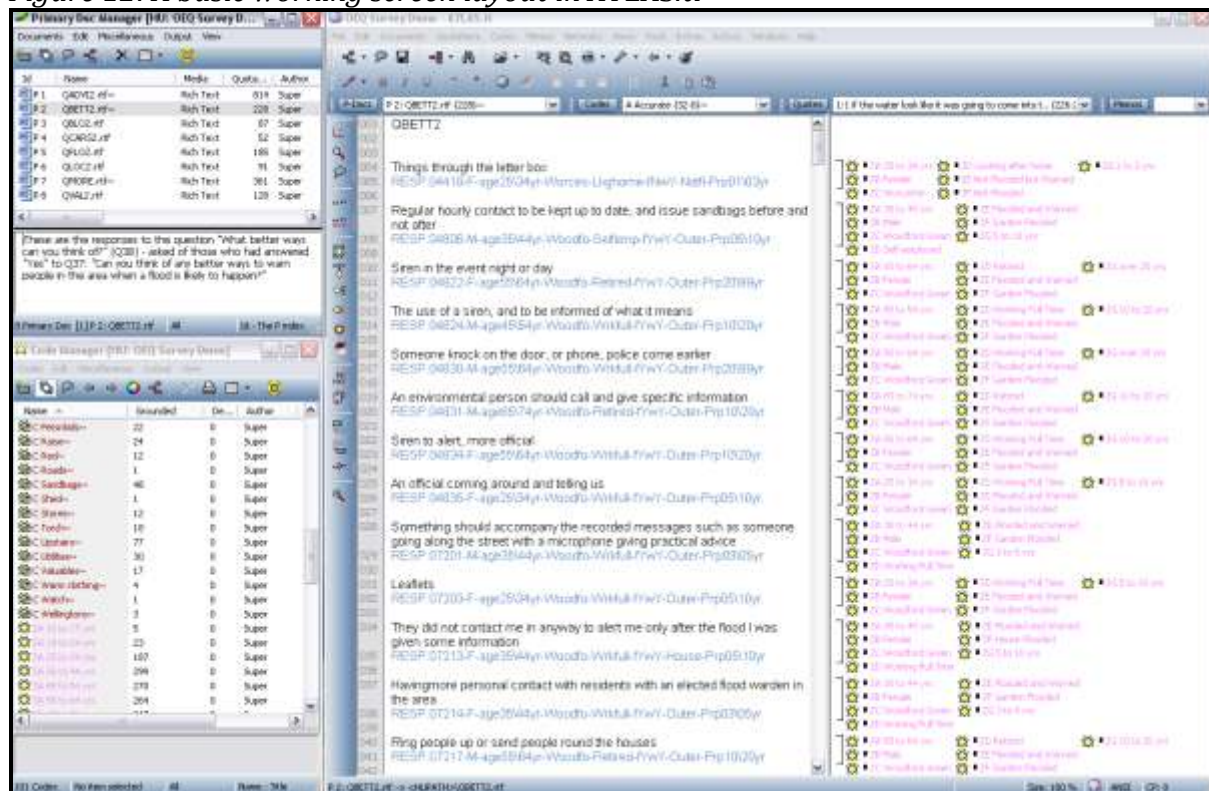


Figure 12, above, shows an initial layout of window panels in ATLAS.ti as the analysis work begins. The central panel displays a primary document containing response texts (in black font) and the ID/socio-demographic strings associated with them (see the data preparation instructions for guidance on how to create this document structure). To the left two of the object manager windows have been opened and arranged. At the top is the Primary Document Manager showing the set of documents containing the responses to eight separate questions. Below that is the Code Manager showing some thematic codes and some socio-demographic codes (using colour to distinguish these). To the right is the “Margin” area of the display, which scrolls with the central panel, here showing the quotation brackets and codes for the seven socio-demographic codes which have been attached to each response (coloured pale pink).

**TIP:** There is a potential difficulty with the socio-demographic codes, which will be needed later for investigating patterns of response amongst various sub-groups of the survey respondents, because they may obscure clear sight of the thematic codes as these are applied to the response texts. Two alternative suggestions may be helpful here.

As shown in Figure 12, use may be made of the facility in ATLAS.ti to set different colours for code labels. In the Code Manager window highlight a block of codes using the mouse and then click on the circular colour icon in the Code Manager toolbar to select an appropriate colour for that set of codes.



To see those colours used in the Margin area it is necessary to right-click in the Margin area and select “Use object colours” from the context menu that will be displayed there. Selecting pale colours for the socio-demographic codes and bold colours for the thematic codes will help the latter to show up more clearly.

A second suggestion is to hide the socio-demographic codes with a filter command so that only the thematic codes will be visible in the Margin area. This involves several steps which may not seem straightforward to those who are not experienced users of ATLAS.ti, so more detailed guidance on those steps may be found here.

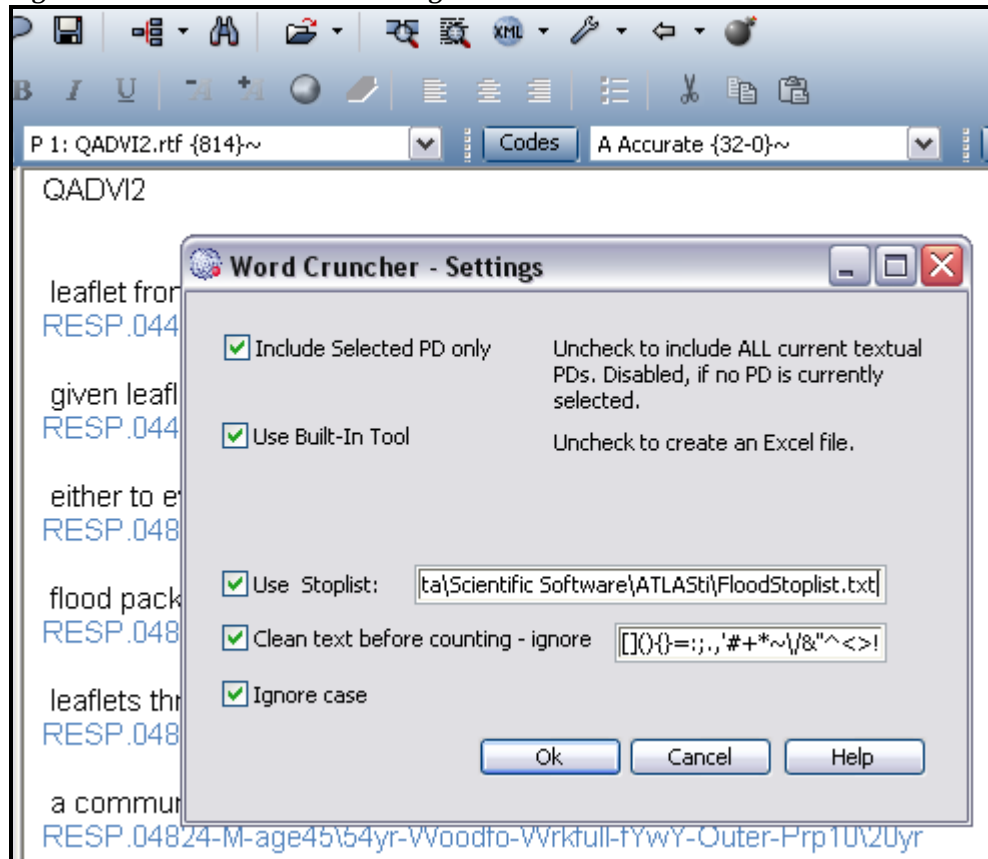
## **2.2 Developing a coding scheme – manually or by using word frequency tools?**

The nature of the analytic strategy will affect whether it is appropriate to develop a coding scheme manually or by using word frequency tools. If working deductively a coding scheme may be derived from, or informed by, existing (theoretical) frameworks; in these situations the following comments will not really be relevant. If, on the other hand, you are working inductively and therefore intend to generate coding categories from the ideas mentioned in the response texts themselves then you have a choice as to how to proceed. You may work ‘manually’ by reading the texts and choosing categories that seem to be mentioned in those texts or alternatively let the software help by creating a list of the most frequently used words in the texts and allow code development to be informed by this list.

The manual, or maybe that could be termed “human”, method will be required at some stage if really accurate coding is needed, because only human readers can detect all of the subtleties of human expression involving multiple ways of phrasing any particular idea. However to get started, particularly in a large dataset, it may be worth trying the word count method to get an early idea of the range and salience of words used. The most frequently used words of interest (ie ignoring trivial words) may be expected to provide indications of the most frequently expressed concepts, although multiple possible meanings for some words can complicate this assumption.

In ATLAS.ti the word count function is found in the *Tools / Word Cruncher* menu option or by using the icon from that menu which can be found on the Main toolbar. The dialog box for this command is illustrated in Figure 13.

Figure 13: Word Cruncher dialog box



Taking the check boxes shown in Figure 13 in turn, the following comments may be helpful:

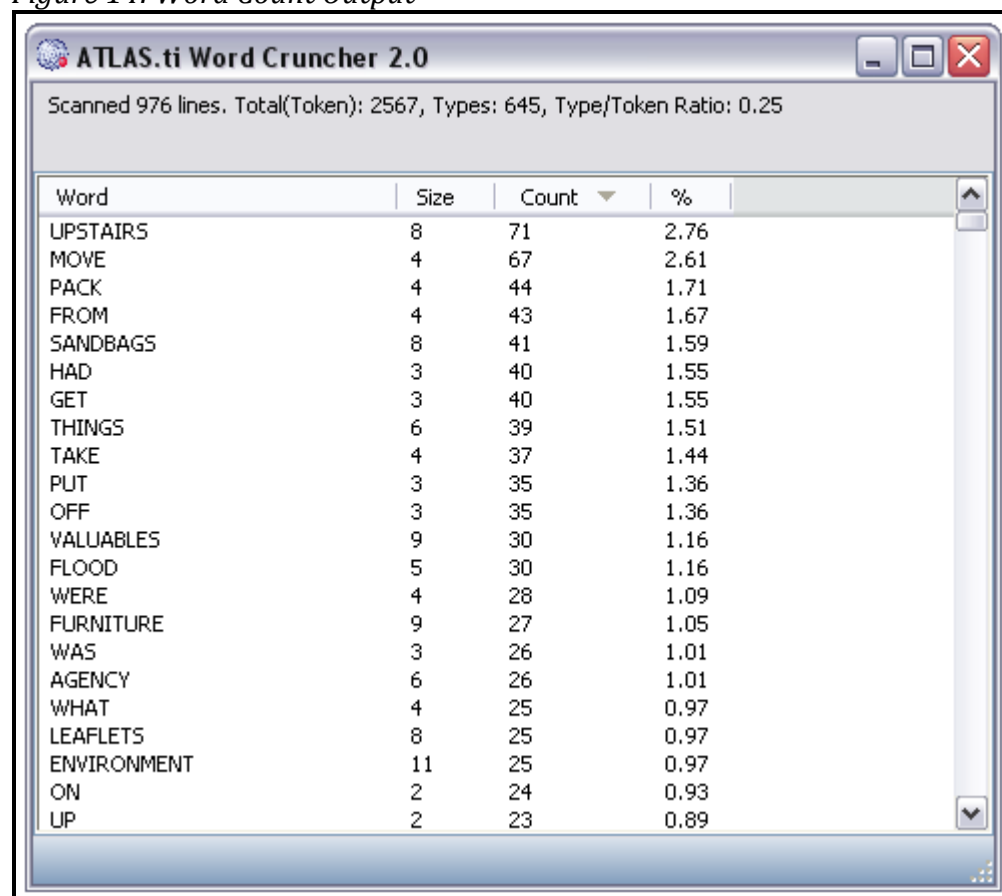
- Include Selected PD only – this governs the extent of the material to be included in the count, a single document (the document currently on view), all of the documents in the project, or the set of documents limited by a current filter. The approach taken in this example is to just analyse the responses to a single question, QADVI2, so that document has been opened (behind the dialog box), and this option is ticked.
- Use Built-In tool – this option is a matter of judgement. By leaving it ticked you will stay within ATLAS.ti and the limited facilities to sort and display the output from this process, whereas by unchecking this box you will create a spreadsheet outside the program where you may be able to exploit many more sorting and displaying functions. We would recommend staying within ATLAS.ti at first and only exporting to Excel when you are comfortable that the word counting approach is suitable and you need more sophisticated tools to process the output. However, if you have selected more than one document in the choice above, you will be forced to export the results because the built-in tool can only handle a single document at a time.
- Use Stoplist: - this function allows you to exclude certain selected trivial words from the calculation and should help you to ‘see the wood for the trees’ more clearly. The default stoplist is very short, and hardly excludes anything, so, if you choose to use it, you will probably need to set up your own list and edit it for each document that you need to analyse. To edit the stoplist, first cancel this command and then select *Extras / Explorer / User System Folder* to open a folder window where you should be able to see the file “Stoplist.txt”. This can be edited in Windows Notepad by typing additional words to exclude in capitals at the end of the list. If you save the amended stoplist file under a different name you will need to amend the file name in

the Word Cruncher Settings dialog to match it. In this illustration it has been saved as "FloodStoplist.txt".

- Clean text before counting – this shows a set of characters to be ignored in the counting process, in most cases this seems likely to be a sensible option to use.
- Ignore case – again this is likely to be a useful option so that a word which sometimes starts with a capital letter will be counted along with the occasions when it is all in lower case.

An example of the output from the Word Cruncher calculation is shown in Figure 14. The table can be re-sorted by clicking in the header area of each column, in this illustration this has been set to decreasing order of the count column, so the most frequently used words are at the top of the list. It may often be useful to sort alphabetically by clicking on the "Word" header in order to see misspellings and plurals of words side by side. In this example it can be seen that the word "sandbags" appeared 41 times, but the alphabetical list showed that "sandbag" appeared twice, "sanbags" once, and "sand bags" a further 7 times. This information will be invaluable if text searches or autocoding routines are used to locate and code the responses with a code for this theme.

Figure 14: Word Count Output



Word	Size	Count	%
UPSTAIRS	8	71	2.76
MOVE	4	67	2.61
PACK	4	44	1.71
FROM	4	43	1.67
SANDBAGS	8	41	1.59
HAD	3	40	1.55
GET	3	40	1.55
THINGS	6	39	1.51
TAKE	4	37	1.44
PUT	3	35	1.36
OFF	3	35	1.36
VALUABLES	9	30	1.16
FLOOD	5	30	1.16
WERE	4	28	1.09
FURNITURE	9	27	1.05
WAS	3	26	1.01
AGENCY	6	26	1.01
WHAT	4	25	0.97
LEAFLETS	8	25	0.97
ENVIRONMENT	11	25	0.97
ON	2	24	0.93
UP	2	23	0.89

The data which was "crunched" for Figure 14 related to advice respondents had been given about preparing for being flooded. In this context several potential coding themes can be seen in this extract. Sandbags have already been mentioned but "upstairs", "pack", "valuables", "furniture" and "leaflets" all look worthy of further exploration. Knowing all of the actual spellings used for each word from this output helps the researcher to investigate the uses of these words in an efficient way.

This illustration may also demonstrate why it pays to be really thoughtful about adding words to the stoplist. In Figure 14 it can be seen that the word "off" appeared 35 times. Whilst it would be

understandable to exclude such a short word in many situations, in this case it was found that many of these 35 uses were next to words like “electricity”, “gas”, “mains”, or “power” and that this little word was a useful way of locating most of these different phrases which referred to the single important idea of turning off utility supplies in the event of a flood.

Another inductive method of developing coding themes from the content of the texts themselves is for the researcher to read the texts systematically, noting ideas that appear to be important or relevant, and then creating codes to represent those ideas. There is no reason why open ended survey questions should not be analysed with this frequently used qualitative approach. However, where the number of responses to a single question is very large, and it is suspected that there is considerable repetition of ideas amongst those responses, then this approach may be found to be unnecessarily time consuming. ATLAS.ti has tools which, if used with imagination and skill, can provide powerful and thorough assistance for such tasks.

As you develop the ideas for your coding scheme you should also consider two other aspects of this work. Do you want to create any form of structure for your codes so that they can be located in the Code Manager in groups? (Note that it will be possible to make multiple groupings of codes later, with the Code Families tool, to reflect developing theories about relationships between the themes represented by the codes.) And, secondly, do you want to create a separate set of codes for each question in your data? There are no firm solutions to these questions, it depends on your own preferences and working practices, the nature of the data and your analytic approach. But, as ATLAS.ti has no visible structures for hierarchies of codes in its main code listing, you may need to consider using a system of letter prefixes in order to create some visible structure for yourself.

It is possible to create a ‘cosmetic’ coding structure in ATLAS.ti by prefixing groups of codes with a common letter. In this example we placed an ‘A’ in front of all codes used in the analysis of document QMORE and a ‘C’ in front of all the codes used in the analysis of document QADVI2. This makes it easier to locate the correct code when allocating them manually whilst reading the document on screen.

If the survey questions were largely unrelated to each other, perhaps because they were widely spaced apart at different points in the questionnaire, then a separate code group for each question may be suitable. However if you expect to be interested in the way some themes arose across multiple questions, then having a more unified coding structure may suit you better.

### **2.3 Text searching and autocoding.**

Before carrying out any coding activity, some thought should be given to the decision of exactly how much text should be coded, or in the terms used by ATLAS.ti, how much text should be included in each quotation. The significance of this decision will only become fully apparent when you start to retrieve sets of texts which have been allocated to a specific code or combination of codes. The issue to be decided is whether to include the whole response with the socio-demographic ID strings or just the phrase which is relevant to the concept being coded.

In simple terms, if you only code the minimum text or just the relevant phrase, it may be easier to check an output for consistency (because there will be less material to read) but it may be harder to locate an omitted code, an incorrectly coded passage or to consider a quotation in the light of its speaker’s characteristics (because there will be less contextual material available in most outputs). If the majority of the texts in your document are very short it would probably be better to include the

whole response and the full ID string (i.e. the same quotation as was used for the attribute coding when the data was first prepared for CAQDAS), because then all of the attributes will always be readily accessible in any set of quotations that are output. On the other hand if you have quite lengthy responses and you want to consider nuances of meaning within them then more precise coding and quotations may be appropriate. As far as the program is concerned there is no difficulty with shorter quotations because ATLAS.ti will be able to identify the attribute codes within which they are nested using the Query Tool.

Although many qualitative analysts may naturally prefer to do all coding work manually it is quite reasonable in some circumstances to use ATLAS.ti's autocoding process. For example quite a lot of responses may be simply "don't know" because the open-question has not triggered a specific response. Coding such material individually would be tedious work but by checking with a *Text Search* for that phrase (probably first with and then without the apostrophe, or both together by searching for "dont|don't") and then using the *Autocode* function these can be allocated to a code quickly and efficiently. With practice more positive common concepts may be identified and also rapidly coded in this way. It should be easier to follow this procedure and then eliminate any incorrect codings, following a consistency check, than to code a large number of very similar statements manually.

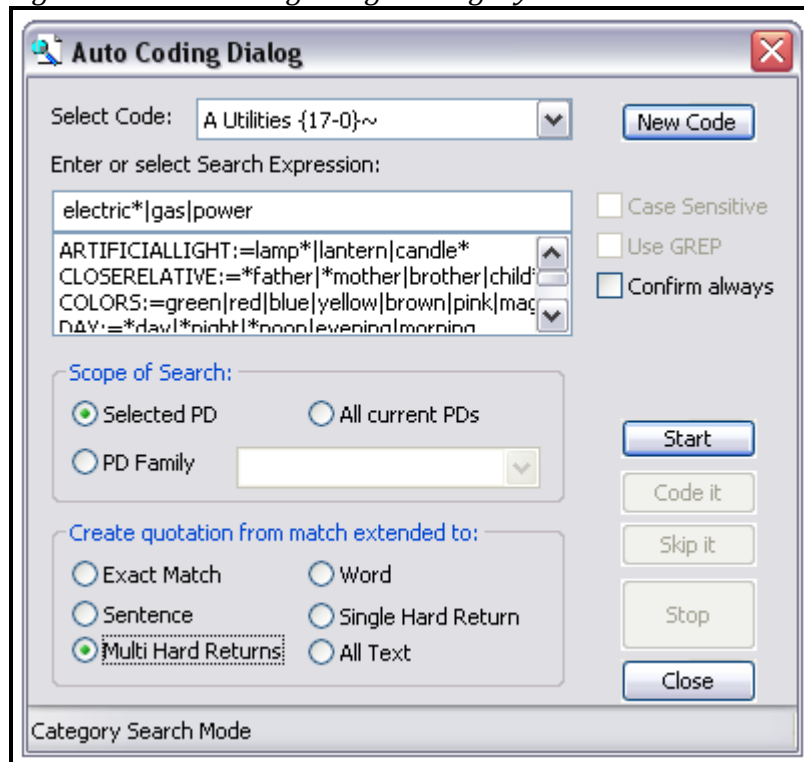
The text search function can be initiated in any one of several ways. The option *Edit / Search* can be used from the main menu bar, or else the shortcut keystroke Ctrl+F or the toolbar icon (both of which can be seen in the Edit menu) can be used. This brings up a simple dialog box where the required string(s) can be entered and the currently open document searched forwards (using the "Next" button) or backwards (using the "Previous" button). With each click on Next or Previous the document will be scrolled and a further instance of the search word will be highlighted in the text where it can be read in context and its meaning can be assessed.

If you decide that a word or phrase has been used with sufficient frequency and consistency to justify an automated coding procedure then the Autocode routine can be used. This is the same routine as the one used to allocate codes for socio-demographic attributes in the data preparation stage. However in this context it can be used a little differently because several variations of words and spelling can be used in a single coding pass. The procedure can be started with the *Codes / Coding / Autocoding* option, and new codes can be created within the dialog box, but it is probably a good idea to create the required code beforehand and then concentrate in the dialog box on the variations for the search string.

Following on from the example used above, it was noted in the Word Cruncher output that there were a variety of references to gas, electricity and power. The first few text searches indicated that many of these referred to advice that these utilities should be switched off or disconnected before floodwater got into the house. These could all be autocoded in a single operation. As shown in Figure 15, below.



Figure 15: Autocoding using a category search



In Figure 15 it can be seen that the search expression is “electric\*|gas|power”. This will locate and code all sentences in the selected document which include any of these words, and the asterisk at the end of the first word means that “electrical” and “electricity” will also be located and coded. The word Cruncher output had shown that there were no variations on the spelling of “gas” or “power”.

Other points to note in Figure 15 are as follows.

- The scope of the search has been restricted to a single selected primary document, *i.e.* the one that is currently open; this fits with a working practice (used here) of coding one question’s responses at a time so that the meaning of a phrase is interpreted in the context of that single question.
- The quotations coded with this command will be extended to include the full response and the respondent’s ID and socio-demographic characteristics (indicated by selecting the “Multi Hard Returns” setting).
- The Case Sensitive check box has been left blank so that instances where the first letter has been capitalised will also be located and coded.
- The Use GREP check box has been left blank because this search is not based around character structure in the texts.
- The “Confirm always” box has been left blank so that all of the ‘hits’ will be coded automatically and the procedure will run quickly. Alternatively, by ticking this box it is possible to review each located ‘hit’ in turn and take a separate decision to ‘code it’ or ‘skip it’ using the buttons below the ‘Start’ button, and even to vary the size of the quotation coded each time, but these options will slow the process down. The choice between fully- and semi-automatic working will depend on personal preference and the nature and quantity of data to be analysed.

- It is possible to save an auto coding search expression for re-use later, by adding a name or label for the search, a colon, and the equals sign in front of the expression. Examples of this can be seen in Figure 15 and are offered by the program when you use this function.

**TIP:** When this option was run it was noted that 28 quotations were added to the code “A Utilities”, this was fewer than expected as the Word Cruncher output had shown 38 occurrences of the variations of the words in all. A check on the actual quotations revealed that there were 10 responses where both electricity and gas were mentioned and this explained the difference.

The autocoding process will not complete the task if data reduction to accurate quantities of references is the goal of the analysis (see section 4 below). The number of different ways in which a concept may be expressed will frequently exceed the number of ways the analyst expects to find it. So a combination of autocoding and human interpretation is needed to achieve a high level of accuracy. But time can undoubtedly be saved through the use of well-directed search and autocoding routines.

## 2.4 Coding – data indexing versus data reduction.

The actual techniques of manually applying codes to segments of text are not discussed here. They are common to all applications of the program and are clearly explained in ATLAS.ti's help manual and in other sources. However, the possible uses to which the analysis of responses to open-ended survey questions may be put is a matter worth discussing further.

As a coding scheme is developed and applied to textual data, the analyst will inevitably encounter uncertainty and doubt. Does one particular text represent something different from others read before which mentioned a particular keyword? A common solution to this is to be generous and inclusive, applying specific codes to a range of comments that initially appear to be connected to those concepts, with the good intention of returning later and checking the work. This activity may be described as “data indexing” as it facilitates the retrieval of various passages that appear to relate to a particular topic.

When open-ended questions have been asked in survey situations it may be anticipated that the analyst will often be asked to generate numerical summaries of the data, probably in the form of statements of the type “X% of responses to this question mentioned Y”. The obvious source of the numbers for this output is the coding of concept “Y”. However the statement will only be valid if the use of that concept in every one of the responses allocated that code is consistent and equivalent, because the code that is used in this way has effectively replaced the words recorded for each respondent. The original textual data has been reduced to the code label.

When put this way it should be apparent that work needs to be done by the analyst to refine the inclusive indexing codes before they can be safely used as ‘summarising reducing’ codes. In the example used above “upstairs” was the most frequently used word in the document, relating to advice to move valuable items to a higher part of the house in order to protect them from flood damage. It would be straightforward to autocode all of the responses that included this word with a code to reflect that theme. However a closer check revealed that, while most of the responses were indeed along the lines of “Take everything important upstairs”, one response was actually “I have a file upstairs which gave various useful tips as to what to do”. It would appear that this latter comment is mentioning ‘upstairs’ in a different context to the others and so should probably be excluded from that coded group. Another respondent added a different dimension to the theme with “move stuff upstairs

! but no upstairs here -single floor flat”, apparently acknowledging the common advice but drawing attention to its irrelevance in his particular circumstances. For some interpretations at least, this response also should probably be excluded from the main group to avoid counts of the occurrence of the word ‘upstairs’ being misleading.

## 2.5 Checking summarising codes – consistency and omissions.

There are a variety of tools in ATLAS.ti to assist with the refinement of codes when they have to be reduced to summarise what was originally said. Two particular aspects should be considered, firstly confirmation that all of the passages connected to any one code are all sufficiently similar to be treated as equivalent, and secondly confirmation that no other passages that are also equivalent have been omitted from that code.

The first step in confirming consistency or equivalence is to extract all of the passages that have been allocated to a code and read them carefully looking for differences of meaning that might justify exclusion from this code group. An obvious way of doing this in ATLAS.ti is to generate an output report through the Code Manager window. Open this by clicking on the “Codes” button in the Object Managers toolbar, select the code of interest in the Code Manager, then from the options list in that window select the option *Output / Quotations for selected code(s)*, and then choose where to send the report from the subsequent dialog box (“Editor” for screen display and maybe subsequent printing, or “Printer” for immediate hard copy).

Figure 16: Output – Quotations for selected code

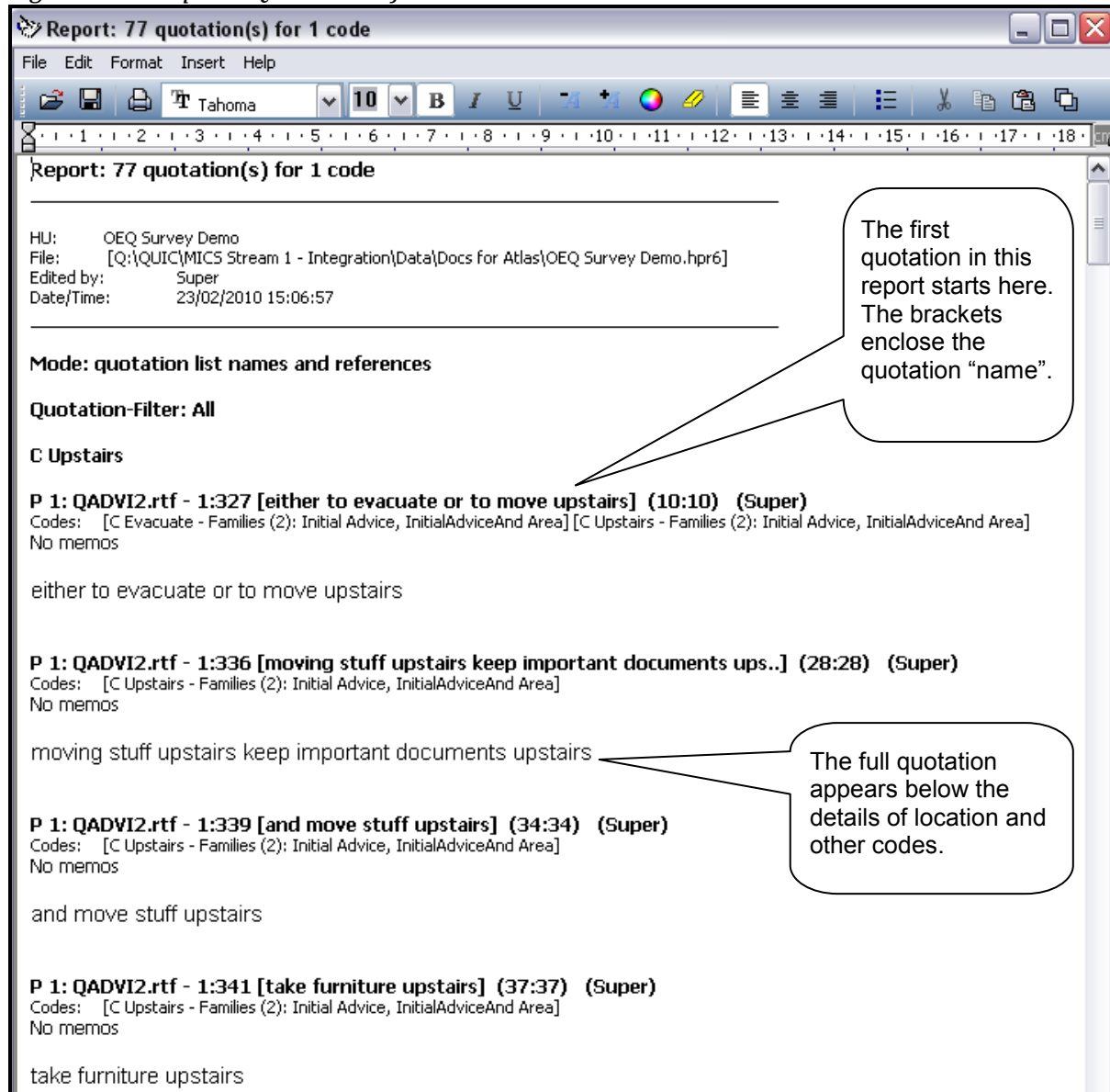


Figure 16, above, shows the first four quotations listed for the output of code "C Upstairs" (to continue the example). It may not be immediately apparent as to what is the significance of each element in this report. There are several lines of header information before the first quotation. At first glance it may appear that the quotation in square brackets in this line is identical to the similar text 3 or 4 lines below, but this is not quite correct. The text in the first line is the shortened quote name used in the Quotation Manager list, while beneath that is the full quotation. When anything longer than a brief phrase has been coded then the full quotation will be longer than the list name. (The number of characters included in the list name is controlled by a setting in *Extras / Preferences / General Preferences* – General tab and has been set here to 50).

It is important that quotations are examined in full in order to maximise the probability of identifying inconsistent or incorrect applications of the code. So you should read the plain text versions to make that judgement because it is easier to compare quotations in this format. If you find a code that has been incorrectly applied you should note the paragraph number on which it occurs (the "(10:10)" section after the list name for the first reference in Figure 16) in order to locate it manually in the document. (ATLAS.ti does not have an interactive link between the report editor and the document to which it refers).

The second line for each quotation in this report is about the other codes that have been applied to this particular quotation and the code families to which those codes belong.

**TIP:** Please note that this only shows the codes for this precise quotation, if it is nested within a longer quotation then the codes applied to the larger quotation will not be shown in this report.

Whilst checking all of the quotations linked to a single code it should be possible to write a concise definition of that code. There is a place for this in the lower half of the Code Manager window. Highlight the code of interest in the top half, click the cursor into the bottom half and type the definition there. If you find it difficult to write a concise definition of a code then it may be inferred that you should not refer to the number of references to that code in any data reducing statements.

**TIP:** Another way of checking the consistency with which a particular code has been applied involves using the network tool. Open a new network, probably with the label of the code as its name, and drag the code in. Then right click on the code in the network and select “Import neighbors” from the context menu, this will bring in all of the quotations for that code as a cascading series of tags. You may need to alter a display setting to read the full quotations, this is controlled in the network window by the command *Display / Quotation Verbosity / +Full Text*. An advantage of this method is that the tags can be moved around in the network display and grouped according to similarities or differences, this may be helpful if a code theme is still being developed or the code needs to be split into sub-themes. This will not be so practical if a code has a large number of quotations. Also, by right-clicking on any quotation in the network and selecting *Display in Context* it is possible to jump to the source document (although in survey data this would only be helpful if the quotations are short extracts from the responses, and not the full responses).

It is more difficult to search for code omissions; passages which are closely equivalent to those already allocated to a particular code but which have not yet been allocated themselves. One possible method for doing this is to filter-out all of the segments which have been allocated the code and then carry out a series of text searches using key words connected to that code on the remaining text passages. This is not a simple operation in ATLAS.ti but it is possible to do it quite efficiently using the following set of procedures.

In outline, the suggested method is to build a complex query (using the Query Tool) to identify all the potentially relevant responses which have not had the code of interest allocated to them, save the output from that query as a new document, assign that document to the project, run a variety of text searches looking for words associated with the code of interest, and investigate any results which appear to indicate that the response has not been coded despite containing an associated word. Finally, the temporary document can be “disconnected”, to remove it from the project in order to prevent unnecessary clutter accumulating.

The specific query structure will depend crucially on whether you have uniformly applied thematic codes and relevant socio-demographic codes to exactly the same quotation, or at times have coded shorter phrases to thematic codes (see section 3 above). Some queries for the former coding method will be simpler than for the latter.

First, let us consider the simpler situation, where there exists only one quotation per response with both thematic and socio-demographic variables applied to it (i.e. the former of the two cases above). Open the Query Tool, using the binoculars icon in the main toolbar or the menu option *Tools / Query Tool*. The query needs to extract just the responses in the relevant document that have not been allocated to the code of interest. In the example that follows the code of interest we are testing is

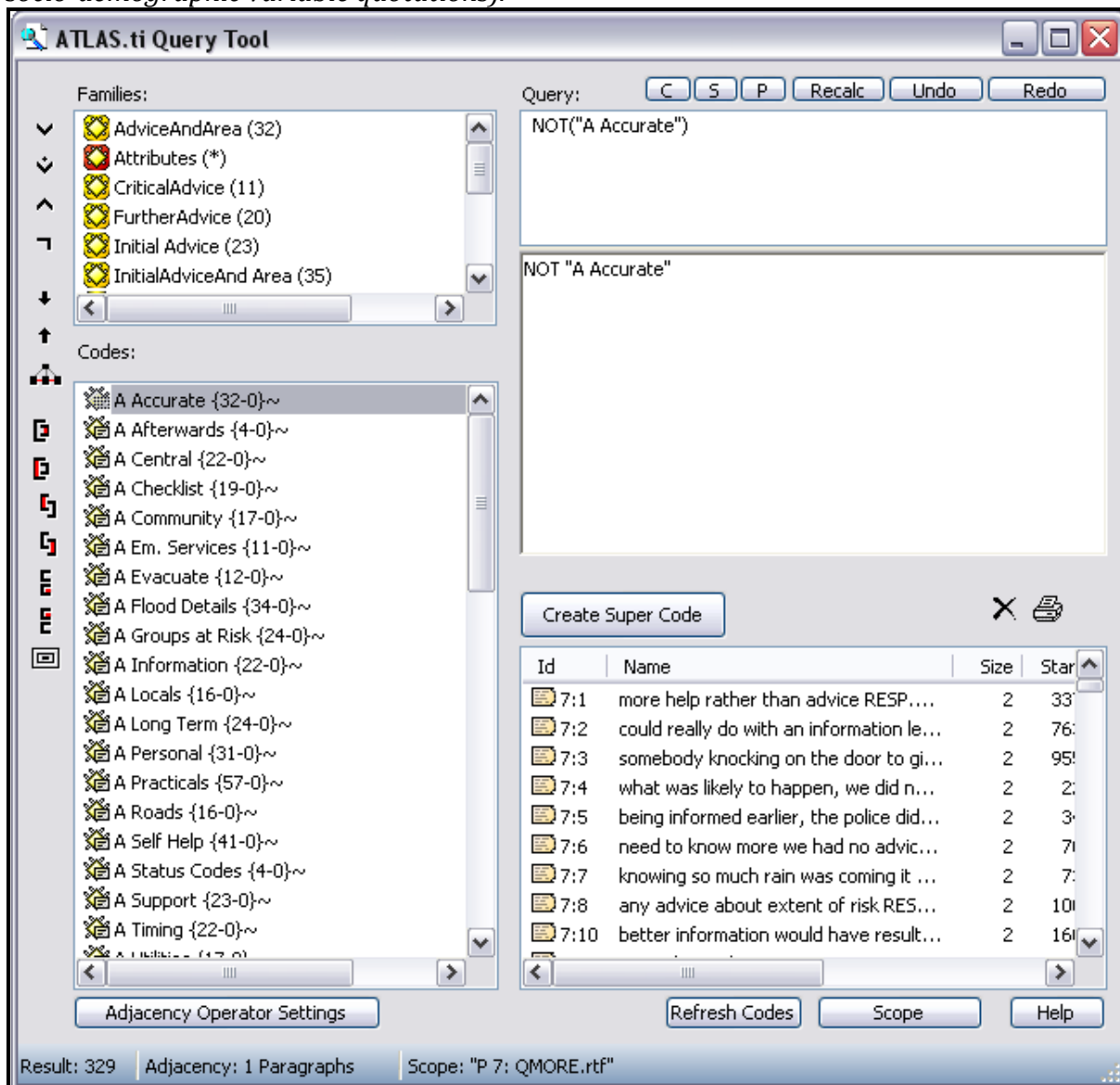


called “A Accurate” and the texts are stored in Primary Document “P7 QMORE”. So we will be looking for all of the quotations in P7 QMORE that are not coded with A Accurate.

If the “accurate” code has consistently been applied to the same size quotations as the socio-demographic codes then the query should proceed as follows:

- Using the “Scope” button at the bottom of the query screen, limit the query to just the document(s) that you require by double clicking on the relevant document (in this case just QMORE) and close that dialog with “OK”.
- Select the A Accurate code in the lower left panel with a double click and see it also appear on the right.
- Select the “NON” operator (the fourth icon down the left margin in Figure 17, a rotated L symbol) with a click.
- The query in the upper panel should now read NOT(“A Accurate”), the list of quotations in the bottom right panel should now be everything in the selected Primary Document(s) which has not been allocated to the specified code of interest, in our example ‘A Accurate. The number of these quotations is shown in the bottom left corner of the Query Tool window and should be the total number of responses to this question less the number coded to the code of interest ( $361 - 32 = 329$ ).

Figure 17: Using the Query Tool to check for omitted codings (thematic code quotations exactly match socio-demographic variable quotations).



Pick up the instructions to proceed from this point below Figure 18 (after the more complicated notes on handling shorter thematic quotations).

Next, let us consider the more complex situation, where only short extracts from the responses have been coded with the code of interest.

The first step, if it has not already been done for another purpose (such as hiding some code labels as outlined at the end of Section 1 above), is to create a Code Family for one of your socio-demographic variables. This will be used to define the full set of responses to any one question in a way that can be used by the Query Tool. Probably the variable with the fewest number of members (values) will be that for Gender, so that should be the quickest to create (but it is important that there are no cases in the data with missing values for the selected variable). From the Code Manager window click on the yellow ring icon, or from the main menu line select *Codes / Edit Families / Open Family Manager*. Click on the *Create New Family* icon, enter a simple name (e.g. "ZB Gender" – using the same prefix letters as were used in the codes that will be its members) and hit OK, then select each of the codes that you have set up for that variable (i.e. 'ZB Male' and 'ZB Female') in the right hand panel below and move

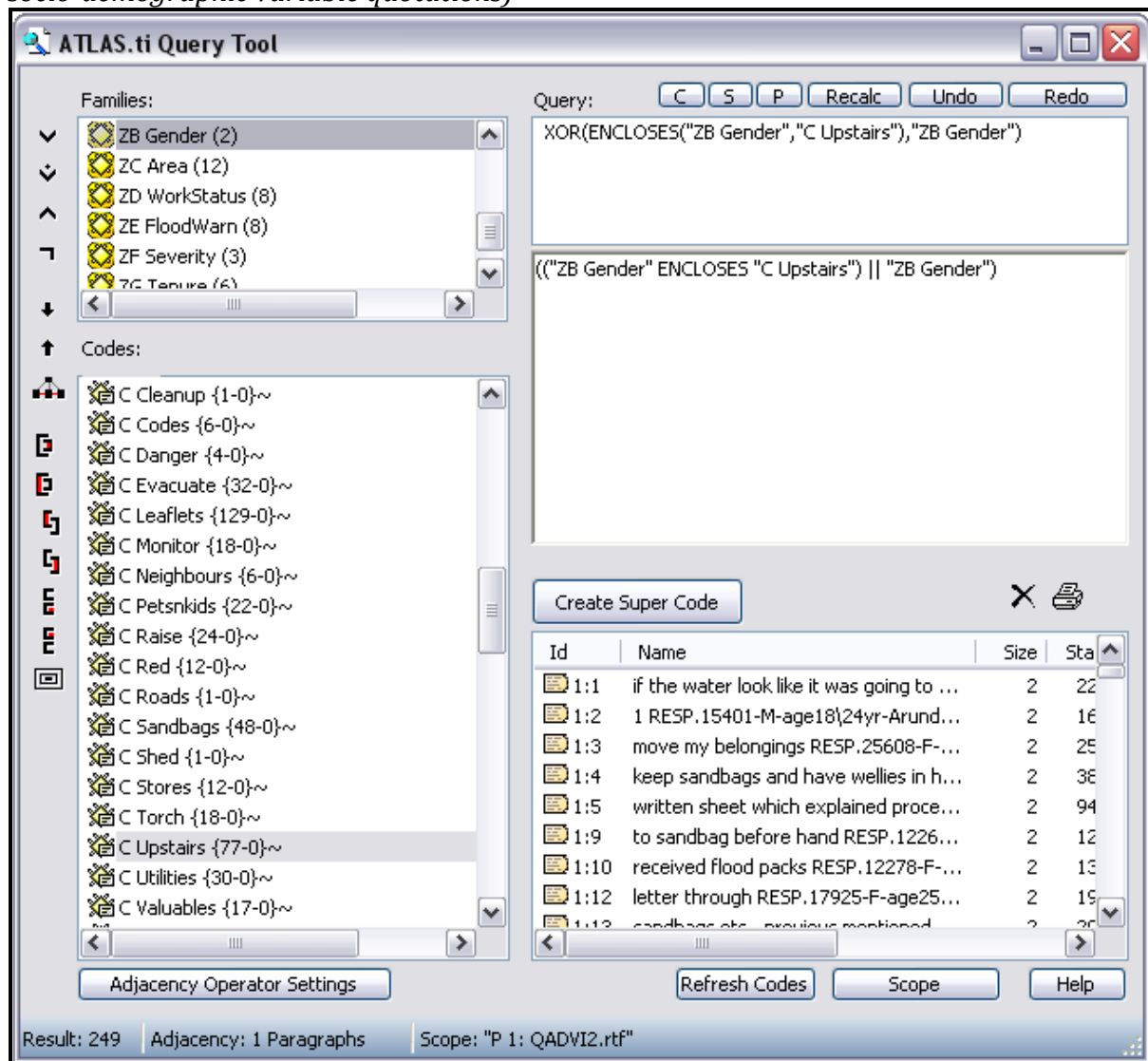
them to the left hand panel using the arrow button in the middle. No special save command is required to complete this process.

Next open the Query Tool, using the binoculars icon in the main toolbar or the command *Tools / Query Tool*. The query needs to extract just the responses in the relevant document that have not been allocated to the code of interest. In the example that follows the code of interest we are testing is called "C Upstairs", the socio-demographic code family is called "ZB Gender" and the texts are stored in Primary Document "P1 QADVI2". So we will be looking for all of the quotations in P1 QADVI2 that are in family ZB Gender (which should be all of them) but excluding those that are also allocated to code "C Upstairs". See Figure 18 below for an illustration of the Query Tool window at the end of this process.

- Using the "Scope" button at the bottom of the query screen, limit the query to just the document(s) that you require by double clicking on the relevant document (in this case just QADVI2) and close that dialog with "OK".
- Select the ZB Gender family in the top left panel with a double click and see that name appear in the panels on the right.
- Select the C Upstairs code in the lower left panel with a double click and see it also appear on the right.
- Select the "ENCLOSES" operator (9<sup>th</sup> button down on the left of Figure 18). The Query expression in the top right panel should now read  
ENCLOSES("ZB Gender", "C Upstairs")  
the order of the terms is important with this operator.
- Select the ZB Gender family by double clicking in the top left panel again – this may seem counter-intuitive but trust us, it is correct!
- Select the "XOR" symbol (a "V" with a dot in the middle, 2<sup>nd</sup> button down on the left of Figure 18). The expression in the top panel should now read  
XOR(ENCLOSES("ZB Gender", "C Upstairs"), "ZB Gender") .
- The list of quotations in the bottom right panel should now be all of the complete responses in the selected Primary Document which do not include anything that has been allocated to the specified code.

See Figure 18, below, for an illustration of the Query Tool screen at the end of this latter procedure. In the bottom line of the panel you can see "Result: 249"; this is confirming that 249 quotations are selected by the current query. It should be possible to use this information to confirm that the query is delivering what you expect, so in this example with 324 responses to question QADVI2 and 77 quotations coded to "C Upstairs" we would expect  $324 - 77 = 247$  quotations in the result – the difference of 2 items is accounted for by the fact that in two places the upstairs code has been applied to two separate phrases within a single response. As you build the query using steps like those outlined above this result figure adjusts at each step to count the hits found by the latest step in the process. It is also useful to observe that the restricted scope of the query is confirmed in this bottom bar.

Figure 18: Using the Query Tool to check for omitted codings (thematic code quotations shorter than socio-demographic variable quotations)



Whatever the length of your thematic quotations, from this point there are several possible ways to proceed.

- The simplest may be to click on each item in the results window within the Query Tool panel in turn and examine the interactively linked text in the source document that will be highlighted in the main working window.
- Alternatively you may choose to print the list of quotations in order to read the hard copy search result for references that have been omitted from the relevant code – to do this click on the printer icon in the Query Tool panel. Next you will see a small menu of choices for the content of the report, experience will help you in the future but for now we would recommend the “Full Content – No Meta” option as the first to try as this will give you the report that is easiest to read. This particular option brings up another dialog, the “Poor Man’s Reporter” with five options – of these you should uncheck “Clip quotation contents” because it is important to get the full quotations in the report, it is probably also useful to uncheck “Include source references” as this should save you unnecessary clutter in the report. Finally you get another dialog asking where to send the output to: - “Editor” sends it to screen, from where you can print it if you are happy with its appearance, “Printer” sends it direct to hard copy. (TIP: Note

that reports generated from the query tool may not appear in the same sequence of responses as in your primary documents. The responses will in fact be grouped according to the order in which the first socio-demographic variable was autocoded during the data preparation phase, because that process created the initial quotations within the document.)

- The final option brings you the possibility of more computer assistance. Follow the “print” notes above until the choice of output destination is reached, then choose the “File” option. This brings up a dialog for you to name the file and choose a location to save it in. You should edit the suggested name to uniquely identify the report (in this example, say, “QADV12 Not C Upstairs.rtf”), save it in rich text format, and put it in the same location as the rest of your data files for the current project. It will then be a simple operation to Assign that file as a new Primary Document in the project.
- The advantage of this is that you can run text searches as many times as you like on this document, searching for keywords related to the concept behind the code, until you are satisfied that no relevant responses have been omitted. Remember that a nil result on a text search of this kind is a useful result as it confirms that you have already coded all instances of the search term.
- Then it is possible to remove the temporary primary document by using the *Document / Disconnect* command as it has no further use in your analysis.
- Note that if you do observe a response where an appropriate code has been omitted, you will need to use the information from the report to locate that text in the main Primary Document in order to apply the code there – it will be no use applying the code to the temporary report document which you are going to disconnect later.

Each of these processes may seem to involve a lot of work, so judgement will be necessary to decide how much is appropriate. These checks are important if you are going to use the code frequencies in any statistical analysis or reporting, they are not so significant if you are merely indexing the ideas in your data. If you started with a clear coding scheme and precise definitions of the codes before you began interpreting the texts then you may be more confident that you have applied the codes consistently and accurately. However, if you have worked more inductively, gradually refining the meanings and uses of the codes with ideas found within the texts, then you are more likely to have inconsistencies between the coding you did on earlier readings and those you did later. It is also important to check for these types of error if more than one person has been involved in coding any particular set of texts.

## 2.6 Looking for similarities or differences?

When analysing the responses to open ended survey questions it may well be easy to slip into the expectation that the most frequently used codes, or rather the concepts to which they refer, are the most important. After all, these are the items that seem to have the most interpretive and statistical ‘weight’. However, it should always be worth looking out for contributions which are different from the common ideas. One-off comments will never feature in the quantitative tables because, by definition, they lack numerical support. But a small number of individuals may well take the opportunity of an open-ended question to add an unexpected thought and these contributions represent a challenge and an opportunity for the analyst.



It is worth considering what the purpose behind the inclusion of an open ended question in the survey was. In many situations previous research will have revealed the most likely answers and these will have been included as response categories in closed questions asked elsewhere in the survey, but then an open question has been included to pick up other ideas. In these situations it is the unusual answers which may be of most interest. It is for this reason that it is worth analysing the open-ended questions systematically.

For instance, in the data used as an example for these instructions the question QADV12 was used to ask respondents what advice they had been given in order to prepare themselves for an impending flood about which they had been warned. It may be interesting to note that out of 324 responses to this question, just three people mentioned warm clothing or blankets. Now it may be the case that for most people the need for warm clothing as you sit out a flood in an upstairs room is too obvious to be worth mentioning, but this may also be a clue that there was a potentially significant gap in the advice actually given to the flood victims in the incidents under consideration. It seems that the value of a detailed qualitative analysis of the responses to such a question is an opportunity to pick up the unexpected ideas which would be so easily overlooked in a statistical analysis.

### Conclusions:

There are many lengthy step-by-step instructions in the above materials. These have been included to help those who are not familiar with certain aspects of the way ATLAS.ti works. However this is not intended to imply that these are the correct/only/best ways of analysing the responses to open-ended survey questions in ATLAS.ti. These are merely examples of procedures that do work, particularly with data of the sort shown in the examples. However it will always be the case that different data may require different procedures, but we hope that these examples will help some analysts to get over the problem of using unfamiliar software or of using familiar software in an unfamiliar way.

Readers who are interested in comparing these processes across different CAQDAS packages should note that the centrality of the “quotation” in ATLAS.ti can be the source of complexity in these procedures. It is for this reason that we recommend that serious consideration be given to the decision to adopt the whole response as the standard quotation for thematic coding, unless the responses are particularly rich and full of meaning.

### 3.0 Quantitative Analysis Strategies for analysing Open-ended Survey Questions in ATLAS.ti

In common with other pages in this section of the website, this page is a series of observations about how the features of ATLAS.ti might interact with a particular sort of dataset. This page should be read in the context of the related materials concerning the use of ATLAS.ti with Open-ended Survey Questions, in particular the Data Preparation Instructions and the Qualitative Analysis Strategies, since the quantitative strategies outlined below can only be effected after the data have been imported and coded systematically in an ATLAS.ti project.

The tools discussed below are illustrated with examples from the same post flooding event survey that was used to illustrate the data preparation processes. For a summary of the project from which this data derives see [here](#). This data is characterised by a fairly large number of short statements.

#### Outline:

- 3.1 Codes-Primary-Documents-Table
- 3.2 Codes Co-Occurrence Tool
- 3.3 Query Tool
- 3.4 Comments on Survey Import routine
- 3.5 Summary conclusion

#### Detailed Guidance:

##### 3.1 Codes-Primary-Documents-Table

In many situations the most important output required from the analysis of some open-ended survey question responses will be a simple summary of the number of times each thematic code has been applied to the set of responses to a single question. When the data has been prepared in the way that we have suggested (here) then this will be completed in the simple operation of generating the Codes-Primary-Documents-Table. This option can be found most obviously in the Code Manager window under Output, or from the main menu options at *Codes / Output / Codes-Primary-Documents-Table*.

Following selection of the above menu option you will be faced with four further options: quotation count or word count and, for each of those, a direct report or an export to an Excel spreadsheet. In the circumstances of analyzing survey data it is most likely that you will need to use the quotation count option and, unless you particularly want to manipulate the table further, the direct report is likely to be sufficient – so the top option “Quotation Count Report” is the one to try first. In common with other ATLAS.ti outputs, you then get further choices over whether to send it to screen (“editor”), direct to the printer, or to a text file. Figure 19, below, shows an example of such an output.

Figure 19: Output Codes-Primary-Documents-Table example

**Frequency table: codes by primary documents**

File Edit Format Insert Help

Courier New 10 B I U

CODES-PRIMARY-DOCUMENTS-TABLE (CELL=Q-FREQ)  
Report created by Super - 25/07/2011 15:03:14  
"HU: [C:\Documents and Settings\scslnh\My Documents\Scientific Software  
\ATLAS.ti\Te...\OEQ Survey Demo.hpr6]"

Code-Filter: Code Family "Initial Advice" [23]  
PD-Filter: All [8]  
Quotation-Filter: All [1946]

CODES	PRIMARY DOCS								Totals
	1	2	3	4	5	6	7	8	
C Bagdocs	31	0	0	0	0	0	0	0	31
C Cars	7	0	0	0	0	0	0	0	7
C Cleanup	1	0	0	0	0	0	0	0	1
C Codes	6	0	0	0	0	0	0	0	6
C Danger	4	0	0	0	0	0	0	0	4
C Evacuate	32	0	0	0	0	0	0	0	32
C Leaflets	129	0	0	0	0	0	0	0	129
C Monitor	18	0	0	0	0	0	0	0	18
C Neighbours	6	0	0	0	0	0	0	0	6
C Petsnkids	22	0	0	0	0	0	0	0	22
C Raise	24	0	0	0	0	0	0	0	24
C Red	12	0	0	0	0	0	0	0	12
C Roads	1	0	0	0	0	0	0	0	1
C Sandbags	48	0	0	0	0	0	0	0	48
C Shed	1	0	0	0	0	0	0	0	1
C Stores	12	0	0	0	0	0	0	0	12
C Torch	18	0	0	0	0	0	0	0	18
C Upstairs	77	0	0	0	0	0	0	0	77
C Utilities	30	0	0	0	0	0	0	0	30
C Valuables	17	0	0	0	0	0	0	0	17
C Warm clothing	4	0	0	0	0	0	0	0	4
C Watch	1	0	0	0	0	0	0	0	1
C Wellingtons	3	0	0	0	0	0	0	0	3
Totals	504	0	0	0	0	0	0	0	504

INS NUM

To generate Figure 19 a Code Family filter was first applied to restrict the table to the set of codes labeled "Initial Advice" – this was the set of codes used to analyse the responses to the first question in this data set, and the table shows that none of these codes has been used with any of the other seven documents. Without that filter, the table would have been much longer as by default it would have listed every single code in the code system. It would have been possible also to apply a Document Family filter to limit the report just to the single applicable document for an even simpler table.

Provided that you can be confident that no code has been applied more than once to any single response then this table provides the basic quantification data for interpreting how many respondents

have mentioned each identified theme. If the option to export it to Excel is used then it is a straightforward operation to convert it to percentages if required.

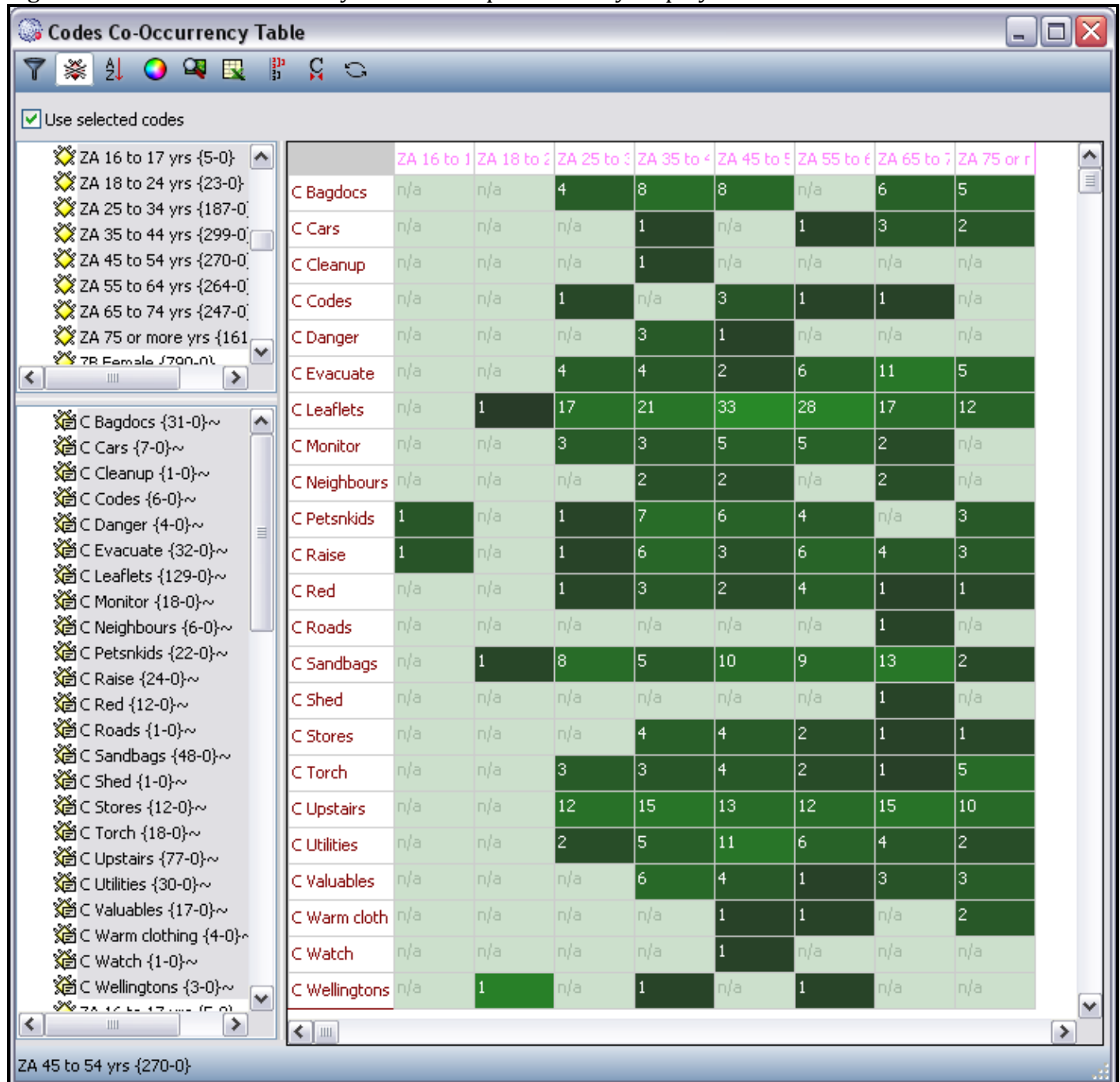
### 3.2 Codes Co-Occurrence Tool

If the data has been prepared and coded in accordance with the advice provided in the pages linked to this section then the quantitative summaries available will only be effective at the aggregate data level. This is clearly demonstrated with the co-occurrence tools, which are found in the Tools menu at *Tools / Cooccurrence Tools / Table Explorer* (ATLAS.ti v6).

**TIP:** When this option is first selected the program opens a new window to display the table and begins calculating the count of each potential combination of overlapping or co-occurring codes. If you have a large number of codes this may take some time. We have observed that applying a code family filter before starting this option makes no difference, the program will still calculate all of the code combinations. However, the Primary Document Family filters do have an effect so, by limiting the data to a single question document, it is possible to speed up the calculation process considerably.

Figure 20, below, shows an example of the basic output obtainable with this tool. However several steps were taken to achieve this display. Firstly, in order to limit the amount of data on which calculations were performed, a filter was set to a single question document by using the Primary Document Family Manager. Then, after the first iteration of calculations (which cannot be prevented from happening, but see the TIP below) the code groups were selected in the left margin area of this window so that a single set of respondent attributes (here the age groups have been used) and a single set of thematic codes (here those prefixed with a "C") would be displayed when the check box "Use selected codes" just below the toolbar was ticked. Note that the top part of the margin controls the codes to be displayed in the columns, and the lower part controls the display of the rows in the table.

Figure 20: Codes Co-Occurrence Table – Frequencies only displayed



TIP: Each change in the settings appears to set the program off recalculating the values in the table, which can be frustrating at times. However, we have observed that you only have to wait for the cells which are visible within the current frame of the window to be calculated, so if you keep the window quite small until you have achieved the desired combination of settings then you can speed up the process.

It is worth experimenting with some of the icons in the toolbar in this window to observe the effects they produce on the display. The colour wheel icon toggles between three heatmap styles, in Figure 20 the green setting has been chosen over red or blue. The 8<sup>th</sup> icon from the left toggles the display of c-coefficients along with the frequencies, while the 7<sup>th</sup> icon from the left performs a cluster analysis which affects the coefficients. At the time of writing there is not much information available about the precise ways in which these functions work so it may be safest to stick with the basic frequencies as



shown in Figure 20. Finally, the 3<sup>rd</sup> icon from the left is another toggle, this time switching between ordering the columns alphabetically (as shown here) and groundedness (ie frequencies).

If you want to explore how the thematic coding might relate to a different attribute variable, select the new set of attribute codes in the upper portion of the left margin and click on the last icon to recalculate the table with the new settings.

The table is interactively connected with the underlying data. So, when any cell in the table is selected a list of the quotation names is displayed and selecting any of these will display that quotation in full in its context in the Primary Document. Thus, to use the example of Figure 20, we might explore how the comments about Sandbags made by the 8 people aged 25 to 34 differed from those made by the 13 people aged 65 to 74 who mentioned that topic.

Note that if you have applied thematic codes to smaller segments of the data than the full responses to which the attributes were coded, then this routine will link you to the larger quotations within which the thematically coded quotations have been made. It does not appear to make any difference which type of code is set in the columns, the output will always be the fullest possible context.

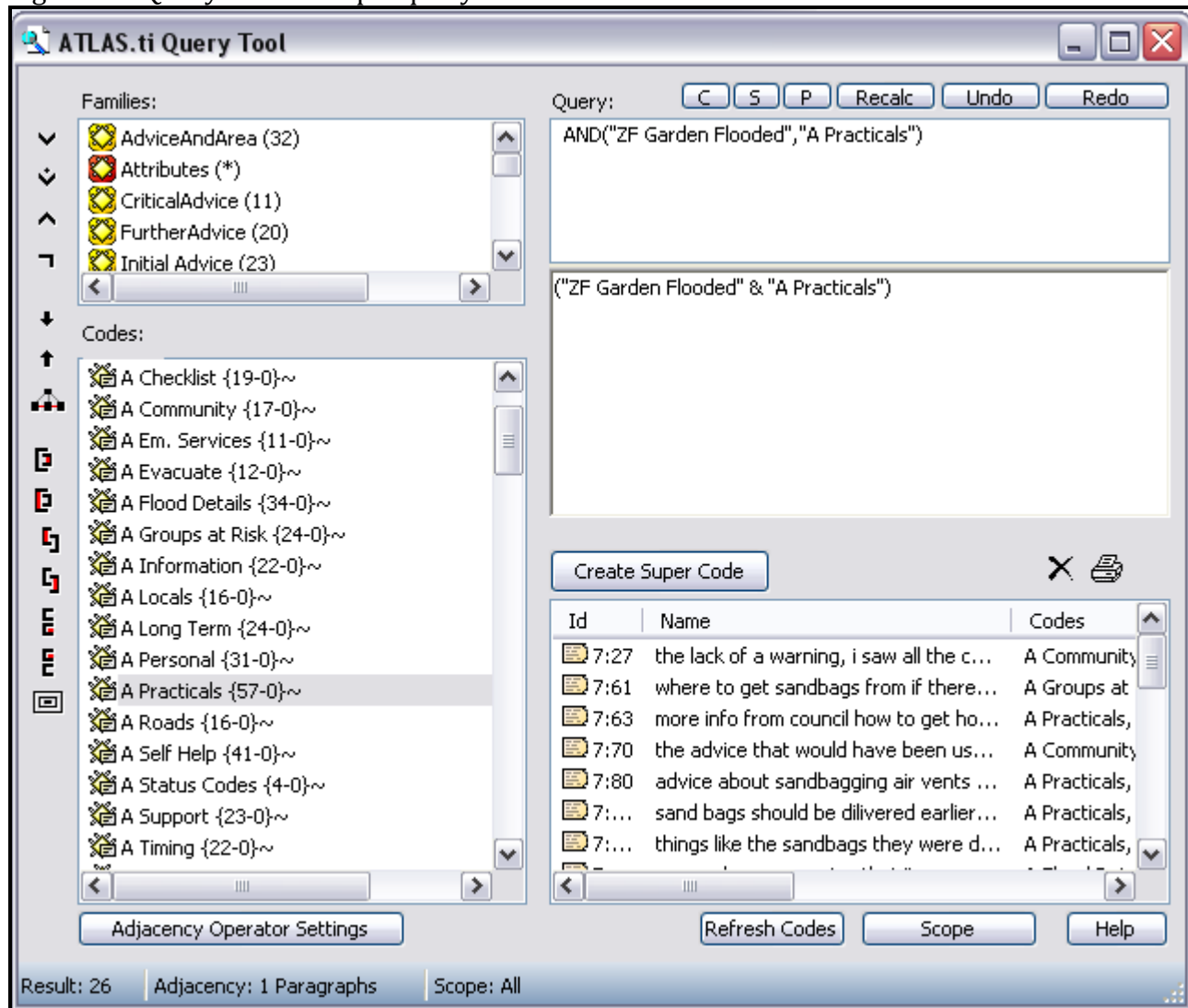
This table is not as sophisticated as those in the other CAQDAS programs discussed on this website in that it does not have facilities for displaying row or column percentages in place of the basic frequencies. However there is a function for exporting the table to a spreadsheet, the 6<sup>th</sup> icon from the left, and it should be a straightforward matter to add calculations there to achieve those displays. Of course in that situation there would be no direct links back to the quotations and contexts, that would still have to be done through the ATLAS.ti program.

### 3.3 Query Tool

The Co-Occurrence tool described above counts the simple co-presence of each combination of pairs of codes and displays them in a table. The Query Tool is a more sophisticated way of searching for and counting the set of quotations that match a specific combination of conditions. In particular there are several different ways of defining “co-presence” in the query tool and it is possible to search for combinations of three or more codes all being co-present in the same quotations. However in quantitative terms this tool only provides one count at a time, although it could be used to populate a table one cell at a time. The Query Tool is initiated either with the menu option *Tools / Query Tool* or by using the binoculars icon on the main toolbar.

Figure 21, below, shows a simple query being set up. When the dialog box is first opened the three panels on the right hand side of the screen are blank. As codes are selected from the list on the left with a double click they are shown in the panels on the right. The full query is built in the top panel, the latest addition to the query is shown in the middle panel, and the set of quotations matching the top line of the top panel are listed in the lower right hand panel.

Figure 21: Query Tool – simple query



Here in Figure 21 the three basic elements of a query have been introduced. First the code “ZF Garden Flooded” was selected (that has 559 quotations linked to it), then the code “A Practicals” was selected (with 57 quotations), and finally the operator “AND” or intersection was selected using the “^” symbol in the upper left margin of the dialog box. This has generated a list of 26 quotations, some of which can be seen in the scrollable panel on the lower right part of the window – the precise number of hits is displayed in the extreme bottom left corner of the window as “Result: 26”. This is telling us that there are 26 quotations which have been coded with both the “Garden flooded” and “Practicals” codes. At each stage of the process of building a query the dialog box changes its display and some practice is recommended to develop an effective understanding of how this powerful tool can be used successfully.

Of course, the result of 26 shown above could also have been read off a Co-Occurrence table that had been set up to plot co-presences amongst the “A” and “ZF” sets of codes in this data. But what the Query Tool offers is greater flexibility to check specific combinations of codes quickly and more power to examine more complicated combinations of codes. It is probably not so useful for purely exploratory work where quantitative patterns are sought.

The extreme left hand margin in Figure 21 contains all of the operators which can be used to combine the codes in the query expression. Many of these are unlikely to have much relevance with survey data, such as we are using in these examples, but there are some important distinctions to emphasise. The intersection or “AND” operator as used in Figure 21 will only be effective in circumstances where you have applied the thematic codes to the whole of each applicable response, as with the attribute codes (ie. including the ID and attribute strings as well as all of the response text). This is because the “AND” operator searches for instances where both of the codes have been applied to a single quotation. If you have applied thematic codes to shorter parts of the responses you will get no results for “AND” queries, instead you will need to use the “ENCLOSES” operator (6<sup>th</sup> up from the bottom) having selected the (larger) attribute quotations first and the (smaller) thematically coded quotations second. Alternatively you can use the operator at the bottom of the list (which is labeled “COOCCUR” when you hover the mouse-pointer over it) which is essentially similar to the implied operator in the Co-Occurrence Table, as this is less sensitive to differences in the exact quotation boundaries.

You may find it interesting to combine three codes in a query, for example to compare how Female respondents whose houses were flooded have been coded thematically against the Male respondents whose houses were flooded. This might lead you to construct a query expression like the following:

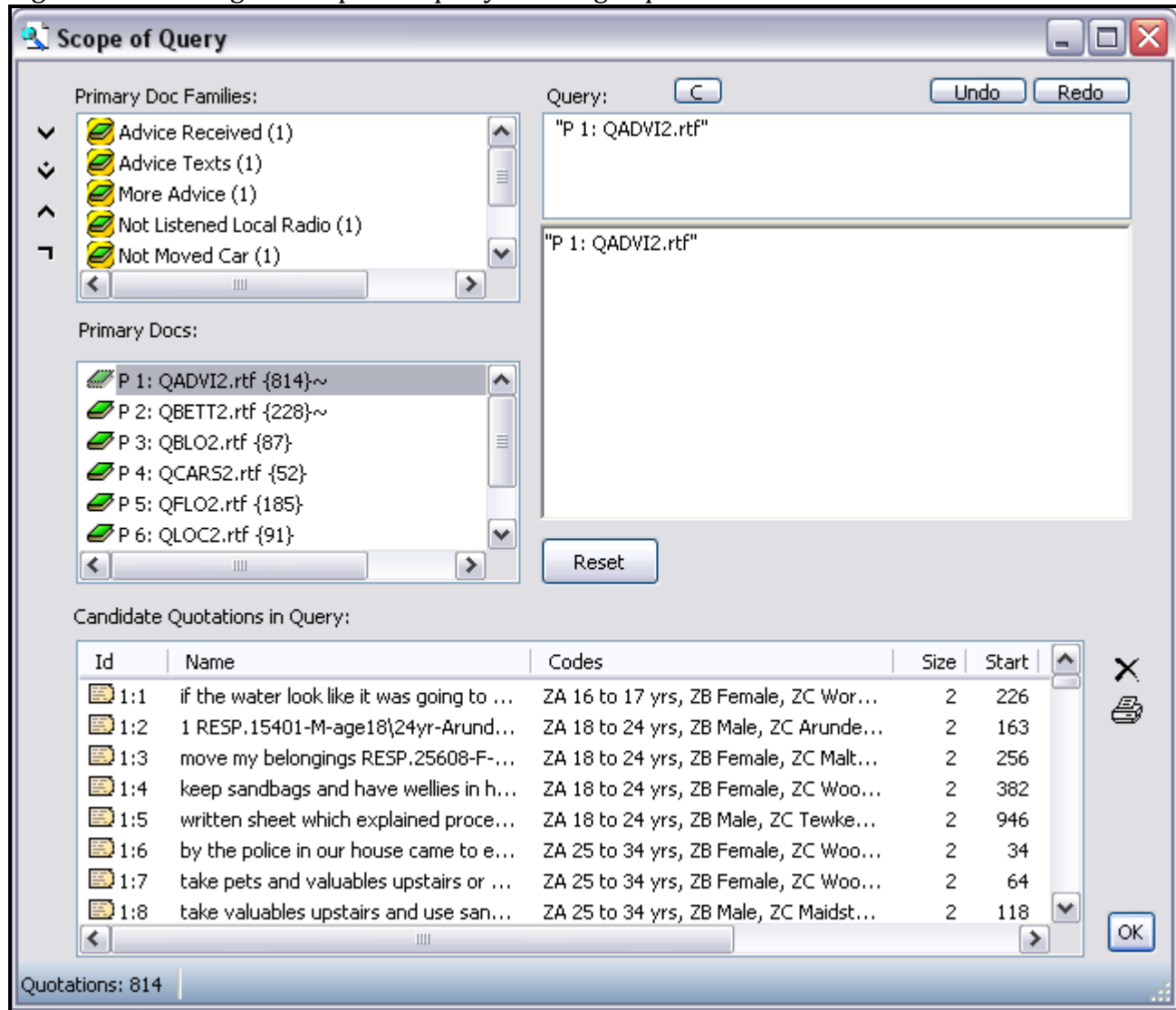
ENCLOSES(AND(“ZB Female”, “ZF House Flooded”), “C Evacuate”)

And then by undoing the last two steps (with the “Undo” button at the top of the dialog box) replace the evacuate code with another in the same group. As each query is effected you may either note the count in a manual grid, or alternatively by using the printer icon you can choose to print the set of quotations that match that query.

We have observed that the Code Family filters work with the Query Tool so that, if you apply a filter before opening the Query Tool you will only be offered the filtered set of codes in the left hand panel. This may save some scrolling up and down the list if you need to carry out a lot of queries.

It is not necessary to apply a Primary document family filter in order to restrict the Query Tool to the data for a single question, in fact the tool will ignore such a filter. Instead there is a facility within the Query Tool to limit the documents, as required, and this is initiated through the “Scope” button at the bottom of the window. Figure 22, below, shows an example of the dialog box that opens when you click on the Scope button. This looks very similar to the Query Tool dialog box but it deals in Primary Documents instead of codes and has a much simpler list of operators. It is possible that you may want to include two or more specific documents in the query, in which case you will need to use the “OR” operator to link them (the “V” symbol at the top of the list). When you have completed the selection of documents, click on the “OK” button at the bottom right hand corner of the dialog box to return to the main query dialog and you will see your selection identified in the very bottom bar of the dialog where it will show something like Scope: “P1: QADVI2.rtf” (Note in Figure 21 where it shows “Scope: All”).

Figure 22: Limiting the scope of a query to a single question document



### 3.4 Comments on Survey Import routine

ATLAS.ti v 6.2 introduced a new data import routine specially designed for the responses from open-ended questions asked in surveys. This takes data from a spreadsheet layout, where each row represents a separate respondent or case and each column represents an attribute variable or a question variable. When the correct distinguishing identifiers have been used, ATLAS.ti will create a separate document in the Primary Documents Manager for each case, it will group these into Document Families to represent each possible attribute, and it will identify each question with a code in the Code System. If this routine is used there will clearly be significant differences over the preparation of quantitative outputs compared with the guidance given above because the data will be in different forms and locations. The following brief notes may help users who have applied this new routine to their data.

The Codes-Primary-Documents-Table will now generate a matrix of codes against cases (instead of against questions as in our example above). This will be useful if you wish to transfer the code data back into a statistical program to carry out more sophisticated calculations, such as regression

analysis, using the coding data generated in your analysis of the open-ended questions. The table will have a column for each case and a row for each code, so it may well be necessary to use a transposition routine in the spreadsheet program to switch these around as most statistical programs are likely to import data on a row per case basis.

In order to generate a table of codes per question after using the Survey Import routine it will be necessary to use the Codes Co-Occurrence tool and set one of the column to show the code for the question you need to quantify whilst selecting the appropriate thematic codes for the rows. There is no direct printing function in that tool and so it will probably be best to use the icon (6<sup>th</sup> from the left in Figure 20) to export that table to Excel and print it from there.

The Query Tool will be extremely important in this style of analysis because each question's responses can only be identified through the coding applied to it during the Survey Import routine. So it is likely that more use will be made of this tool to identify combinations of thematic codes and question codes. Of course with the respondents' attributes being stored in Document Families, it will also be necessary to make use of the Scope function within the Query Tool to investigate how subgroups of respondents have answered the open-ended questions.

The reason why we do not firmly recommend users to make use of the Survey Import Routine is that at the current time it seems impossible to use the semi-automation tools like Word Cruncher and Autocoding effectively when the responses to each question need to be analysed separately. The Code Family filters do not prevent those tools interacting with data that has been coded to filtered-out codes. Thus users seem to have a choice with v6.2, either to set the data up with a separate document for each question and use a planned autocoding scheme to apply attribute codes directly to the data, with the consequence that everything can be done except outputting codes per case in a format that can be exported to a statistics program, or to use the Survey Import routine and be able to do everything except apply the semi-automation tools to one question's responses at a time.

### 3.5 Summary conclusion

Many qualitative analysts, who would be familiar with these CAQDAS programs, may be reluctant to use quantifying statements about the data that they analyse. This would be for the very good reason that most often the size of the sample that they are working with is too small to justify a numerical conclusion. However, when one is working with survey data then there is more likelihood that the sample size is sufficiently large and sufficiently random to support some quantified statements.

At the same time many quantitative analysts may be wary of applying statistical techniques to data which has been coded separately from the data collection process. However, provided the data collection process was sufficiently robust to capture the data accurately, then the fact that the codes have been applied by research analysts rather than the respondents themselves should not be a reason to water-down the analysis processes. Because the coding process is transparent and replicable, the coded data may be used with more confidence than that derived from the 'amateur' coders of the respondents to closed questions.

Thus, if quantitative conclusions are being drawn from the rest of the survey data, then there should be no reason why quantitative conclusions cannot be drawn from the open-ended question responses as well.



## 4.0 Export Data to Statistical Packages after analysing Open-ended Survey Questions in ATLAS.ti

In common with other pages in this section of the website, this page is a series of observations about how the features of ATLAS.ti might interact with a particular sort of dataset. This page should be read in the context of the related materials concerning the use of ATLAS.ti with Open-ended Survey Questions, in particular the Data Preparation Instructions and the Qualitative Analysis Strategies, since the export strategies outlined below can only be effected after the data have been imported and coded systematically in a ATLAS.ti project.

The tools discussed below are illustrated with examples from the same post flooding event survey that was used to illustrate the data preparation processes. For a summary of the project from which this data derives see [here](#). This data is characterised by a fairly large number of short statements.

Unfortunately, when the responses to open-ended questions in a survey have been set-up in the way that we have recommended for ATLAS.ti, there is no way that numerical data can be extracted automatically on a case by case basis for transfer to a statistical package. This is a consequence of the way the case identifying number is merely a piece of text within the response documents. It is possible to run a text search manually for a specific case number and then to make manual notes of coding frequencies observed for that case, but this is clearly not an effective way of processing a large number of cases, as is likely to be the requirement after a survey.

If the export of code frequencies for further statistical analysis is an important aspect of your project then you should consider the use of the Survey Data routine that was introduced in ATLAS.ti v6.2. This has the potential to set up a separate primary document for each respondent or case in the data, and thus to create the possibility of using the Codes-Primary-Documents-Table combined with Code Family filters to generate the output that you require. The disadvantage of using the Survey Data routine is that it then becomes impossible to limit any of the semi-automated data processing tools, such as word frequency calculations and autocoding to the responses for a single question in the data. So our final work-around suggestion here would be to consider setting up a separate ATLAS.ti Hermeneutic Unit for each open-ended question in the survey, using the Survey Data routine to import the data for each, and you would then be able to use the automation tools and subsequently export the code frequencies at the case level.