



Working with data

In: The NVIVO Qualitative Project Book

By: Patricia Bazeley & Lyn Richards

Pub. Date: 2011

Access Date: May 25, 2020

Publishing Company: SAGE Publications, Ltd

City: London

Print ISBN: 9780761970002

Online ISBN: 9780857020079

DOI: <https://dx.doi.org/10.4135/9780857020079>

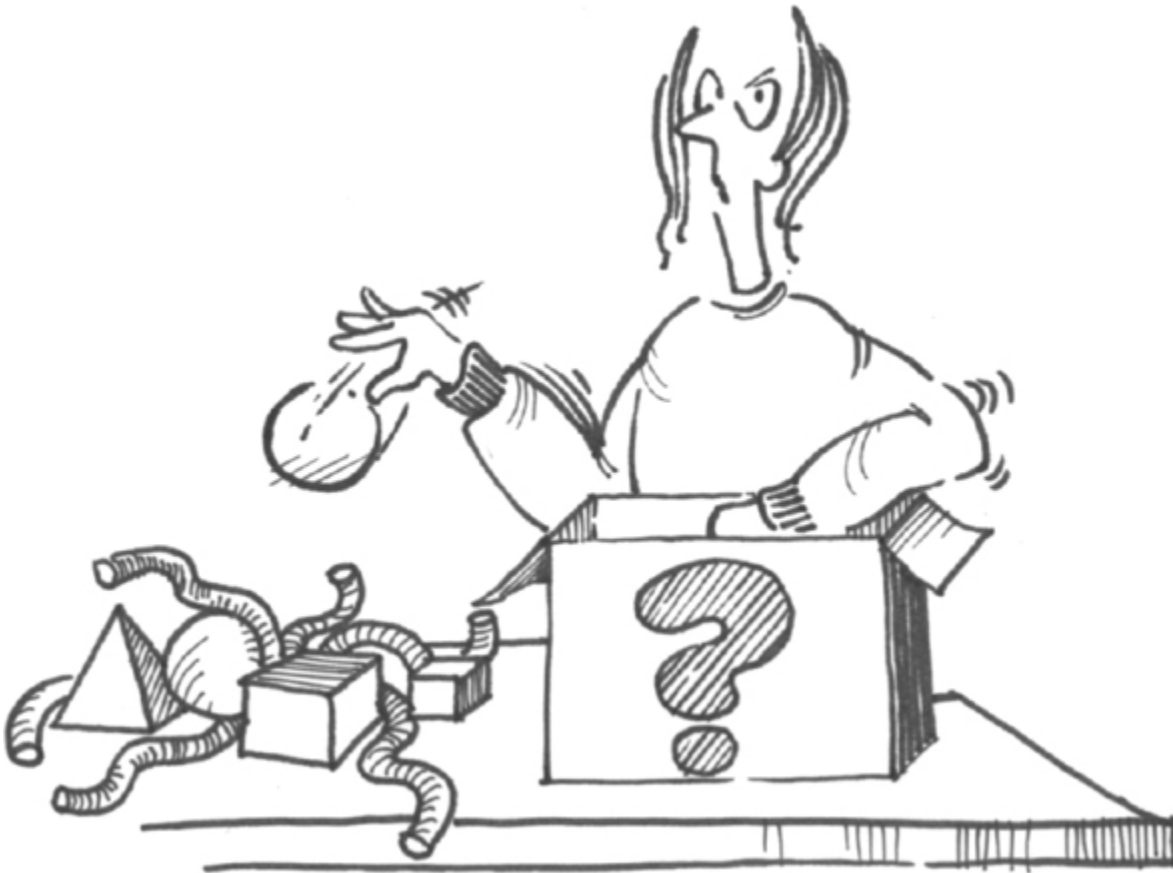
Print pages: 49-68

© 2000 SAGE Publications, Ltd All Rights Reserved.

This PDF has been generated from SAGE Research Methods. Please note that the pagination of the online version will vary from the pagination of the print book.

Working with data

This Part describes processes of thinking “up” from the data. Central to qualitative method is the combination of ways of recognizing, recording and exploring ideas discovered in data. Central to the design of NVivo is the goal that many ways of recording of insights and ideas can be integrated. If you wish to edit or annotate, you do not have to stop coding and go into another mode. If you code, it is not invalidated by subsequent editing. Memos happen as ideas form, and can be coded. New nodes happen as a result of coding, and the nodes you have made already can be revised and enriched as you move around a document. You can rethink them as you use them and code memos that discuss them.



Reflecting and recording ideas

Annotating text, memoing specific points in the text and writing memos about documents or nodes are different ways to record ideas about the data, typically undertaken concurrently with coding, not as different stages. As you become familiar with these tools, reach for the appropriate one to record an idea.

Annotating the text

Some reflections on the text are brief notes, aside comments. Others are reflections on the detail of discourse or the use of a term, or reminders about a related item. As you read, minor, brief insights can be inserted “behind” the text of the document, using DataBites. DataBites are the DataLinks that take you either to data external to the project or to annotations.

Internal annotations are fast to prepare and fast to access. Click on the anchor and see the annotation. They will appear “live” in text retrieved by coding; they can be printed as endnotes on a text report of your document, but their text can’t be separately coded or searched.

Link an annotation



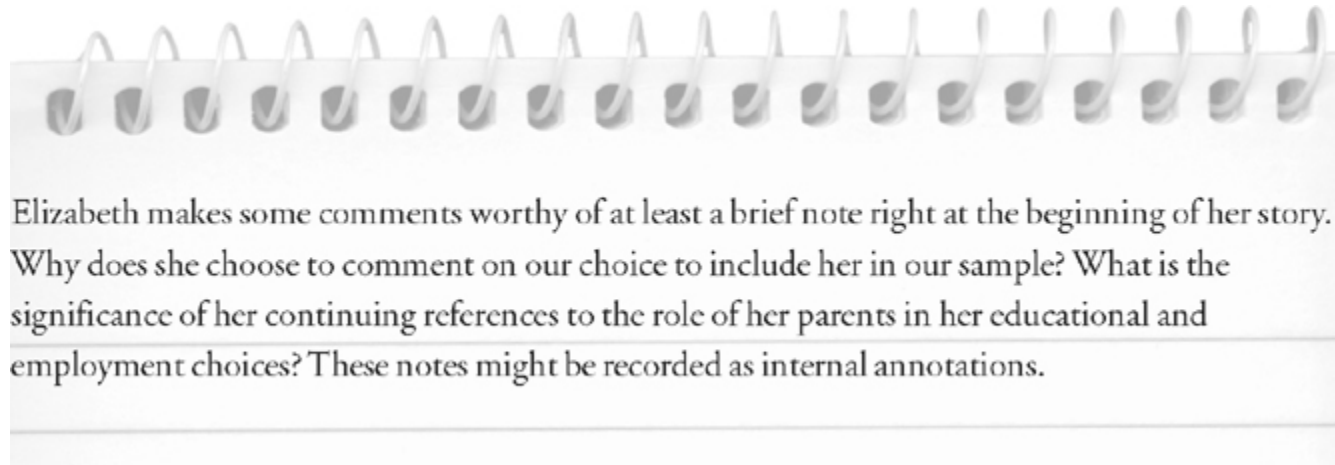
- Highlight the text you wish to annotate.
- Click on the DataBite symbol in the Browser's toolbar.
- Confirm that you wish to make an internal annotation.
- Type your note in the box provided, then click Save & Close.
- Note that the text anchor for the DataBite has been marked.

To access your DataBite:

- With your cursor in the marked text, click on the DataBite symbol, or choose Inspect DataBite from the context menu.



DataBite anchors are text, not an inserted icon. This allows you to highlight just the text being annotated (the expression discussed, for example) or to insert a note which is developed in an internal annotation (“argument followed”) or a description of the linked external file (“this is a photo of Annie”). The anchor should generally be quite short (a word or phrase rather than a whole sentence or paragraph).

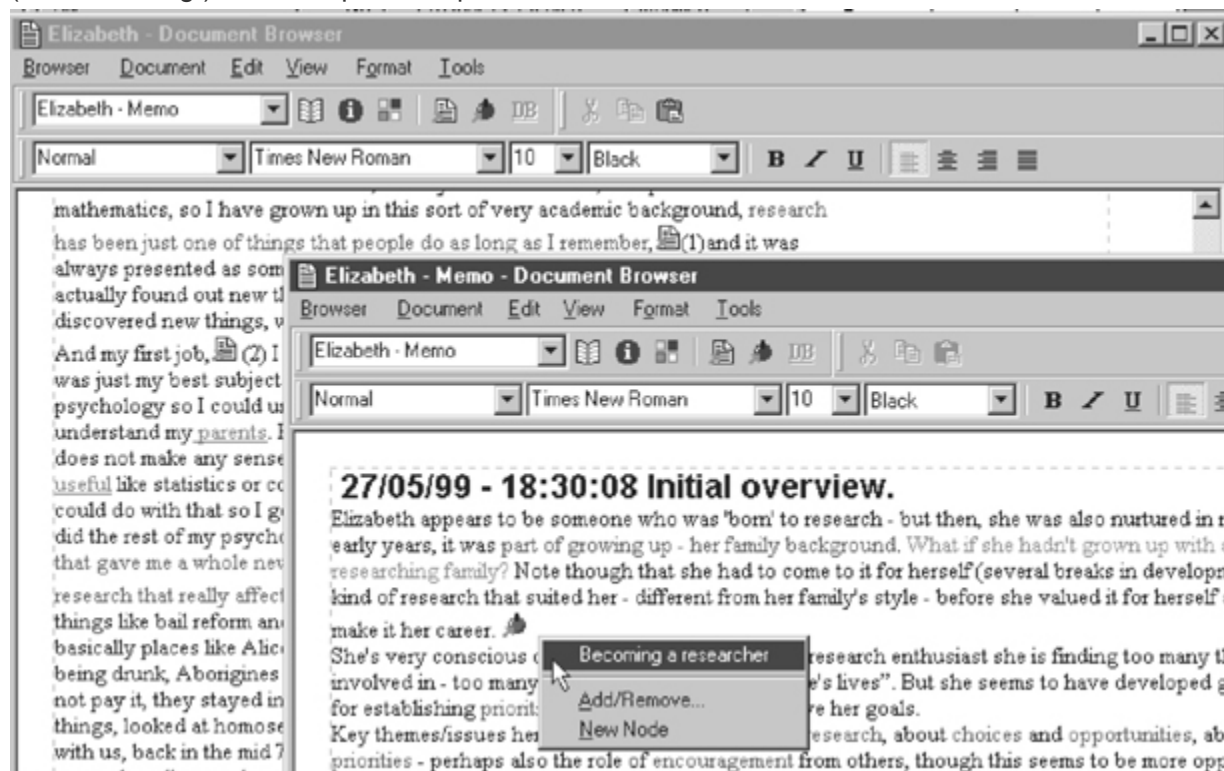


Elizabeth makes some comments worthy of at least a brief note right at the beginning of her story. Why does she choose to comment on our choice to include her in our sample? What is the significance of her continuing references to the role of her parents in her educational and employment choices? These notes might be recorded as internal annotations.

Memoing detail

There are times when a specific comment in the text prompts reflection—something that is said gives clues to the process being studied, or spins thinking off in a new direction, worthy of a memo linked to that text. You can create new memos, or keep adding to a memo created for the document. If it will be helpful, the memo can be coded or further links made as you write.

If the reflections you wish to record are more about the category (node), you can link the memo directly to that node as well as to that point in your document. Or you may wish to make a new memo for the node. Text (with its coding!) can be copied and pasted from one document to another.



Link a memo



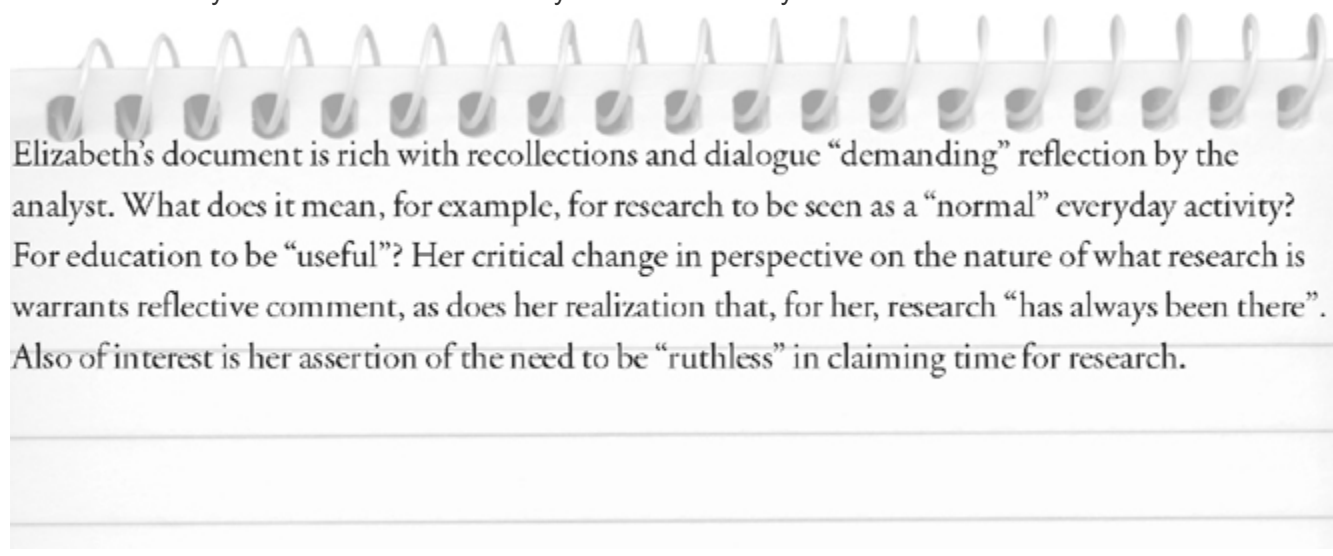
While working through the text, when a word or phrase or sentence prompts reflective thinking:

- Place your cursor at the *end* of the passage of interest.
- Click on the DocLinks icon in the Browser's toolbar.
- The DocLinks dialogue appears, set to Show DocLinks from the selected text position. You can now link to a new or existing memo.



NVivo has many ways of logging the history of your ideas. Insert the date and time in your memo (select from the Edit menu). Use a subheading to identify each addition to an existing memo. Use the ability to edit a document to insert an identifying number or letter into the text immediately after the DocLink icon (your cursor will already be there), and identify with that number the specific entry in the memo about that text. Or the text that prompted the comment can be copied from the Document Browser and pasted into the memo, for detailed analysis there.

For many researchers the key to qualitative coding is to combine it smoothly with reflecting on data. Coding becomes an analytic process, rather than a mechanical sorting of text. Coding memos as well as the original text will redirect your attention to the issues you've raised and your reflections on them.



Coding and what it does

Coding has both an organizational and analytic role. Some knowledge about the content of documents will simply be descriptive. Which of the research questions is this about; to whom were the speakers referring; what event were they describing? But some of what you “know” will be interpretive, concepts that are derived from the material in the documents—labels to express how this person is coping; what this comment is telling me about the context of what's happening here; the impact of that event.

In what follows, we start with some advice on coding for researchers unfamiliar with qualitative coding. Then we outline how to do it in NVivo, coding either from the Document Browser (as you focus on a document) or from the Node Browser (as you review a topic).

Thinking about coding

Coding is a way of expressing thinking “up” from the data: it makes nodes, and gathers references to the material about these topics. You can then review all that material, think more about it and code on from it.

Coding is also a way of thinking “down” from categories to all the material about them. You can access that descriptive and conceptual knowledge, refine it, check back to the original context, and (as will be shown later) explore or test conclusions by checking for associations between those concepts, descriptive categories and/or ideas. So, organizationally, coding is a useful way of keeping track of topics or ideas, and of documents about those topics. But more than that, creating a node and selecting text for coding are interpretive, analytic processes in which thinking about the data is extended beyond the descriptive to a more abstract level. These processes lead you in different ways—to thinking about the shape of your categories, to thinking about the many sorts of material at that category, to asking questions about relations of categories.

Initially coding is likely to be of four (not mutually exclusive) types:

- a) In Vivo coding in which you pick up words, phrases and ideas directly from the text. In Vivo codes are likely to be much more about detail and some will later be dropped or merged with other codes.
- b) Descriptive coding which captures information provided in the text, such as where or when an event may happen, what someone reports about how they are feeling. While some of these descriptive categories may be further explored in terms of the conditions and consequences applying to them, minimal interpretation is required at the initial point of coding.
- c) Broad-brush coding gathering material on wider categories. These might be topics of questioning or observations, whole responses to structured questions, or key themes or issues arising from the data. Later, as the relevant dimensions become clearer, you can return to these nodes and use the live Node Browser to “code on” this material into more discrete categories.
- d) What we will call concept coding, where you are using the code to pick up and “open out” an idea that you might want to explore. At this stage these concepts might be broad or detailed, descriptive or interpretive. Memos are often useful in association with this type of coding.

Approaches to coding

In the introduction to this book, we highlighted the range of qualitative methods, and warned that the researcher must find a relevant method for their research question. Coding will be very different for different approaches to analysis. Some work through data in broad-brush mode first, then concentrate on more detailed coding and analysis of particular themes or issues. For example, a field researcher may first identify the formal groups or institutions in the setting studied, then return later to everything coded at this “obvious” topic, more finely coding evidence of informal structure, cultural patterns, etc. In other methods, researchers work very slowly and reflectively through documents, coding as a way of developing concepts. In grounded theory studies, for example, researchers may work by “open coding,” going through early documents line-by-line and memoing extensively as a way of “breaking open” the data and the concepts held therein.

Your approach will be shaped by the nature of the project and the structure of the data, and these in turn by your methodology and training. But almost all methods involve coding, and many ways of coding. If you are

uncertain about how to code, or, more critically, why you would want to do this, return to do more general reading on methods. Once you locate your study methodologically, the uses of the variety of techniques will become clearer. But whichever approach you take, it is crucial to remember that coding is just one way of working qualitatively with your data. As you code, use other ways of stimulating and storing ideas—visual clues (highlighting the text), linking, memos and models also to capture the rich ideas prompted by your data.

Broad-brush coding

Given the live Node Browser, there is no need ever to treat coding as permanent. Rather, treat it as a first step to thinking something out. As you read through the document you will find that there are larger passages (perhaps whole paragraphs or speaking turns) you can code at broad topic areas (themes, issues, such as you identified in your project journal or the document memo). Possibly these are already identified as sections in the document, because there was some structure to your data gathering. It may be useful to code (name) these topics/themes/issues simply as a first pass. This will bring some level of organization, particularly important if the volume is increasing. It will give access to all the data coded at one topic, e.g. all the answers to Question 3, or everything Jim said. A review of the codes developed this way will give an overview of the range and depth of topics covered. Furthermore, this kind of coding allows examination of the context in which a code is used. For example, does it make a difference to how the respondent discusses a particular issue if it is in response to the question asking about that issue, compared to when it comes up spontaneously during discussion of another topic?

It is quite possible, too, that early coding proliferates categories. Broad-brush coding will allow you to identify text that is particularly relevant to the area/s you need to focus on for now. Once you have conquered those areas (and delivered the thesis or report!), you can return to analyse and report on those topics set aside.

One of the tutorials in NVivo, *Get on with living*, shows broad-brush coding at work. The researcher, Lynn Kemp, coded her interviews with people with spinal injuries first at very broad categories (e.g. community services, employment, education). She then “coded on” from the community services text (which was her immediate focus), capturing the detail of what it was that people with spinal injury were seeking from life, what services were offering, and how these might be redesigned to allow their clients to fulfil their “plan of life”.

Concept coding

By contrast, for some methods, analysis involves detailed exploration of early ideas, with thinking about all of a concept's possible meanings recorded in codes and memos. What results is a highly integrated process combining development of codes, coding, memoing and reflective journaling. Concept development happens fast in this approach and the first documents coded will strongly influence the categories developed to capture that thinking. So methods which involve creating categories “up” from the data require ways of checking the recurrence of early categories and of ensuring that new categories can be generated from later data, with the possibility of checking back through earlier data for those later concepts.

A pragmatic approach

If you are unfamiliar with these techniques, a useful in-between way to start generating codes is to identify interesting passages in your text, and create a code (or multiple codes) that “tag” it. The node title might be descriptive, only later being changed to something more abstract and of use also in coding other documents.

Try the following three steps which Lyn developed for undergraduate teaching: many of us have found they lead to generating useful codes. (If you are very unsure, print the document and work first on paper.)

Identify: What's interesting? Highlight the passage.

Ask: *Why* is it interesting? This may generate a useful descriptive code or perhaps an interpretive code. Make a node (or on paper, write it in the margin).

Then ask: Why am I interested in *that*? This will generate a more abstract concept that will be of general use in your coding system, particularly when one starts to organize the concepts/categories into a more structured system. Make a node, or a paper note. And so on to the next interesting passage.

The immediacy and convenience of doing this on screen will soon win you over from paper coding! But either in NVivo or on paper, what happens is that nodes are “emerging” from your thinking about the data—and at the same time you are connecting them to the data about that concept.

As you create a code, get into the habit of documenting what it is. Name it carefully, and if the title doesn't provide clear identification, add a description. If the code is at all interpretive or abstract, record in a memo what made you see this as a category, and what you were seeing in the data you coded there.

You can print a report of all your nodes and their descriptions at any stage, providing a detailed “codebook” to review and rethink, and to use as you code.

Coding in NVivo

Once you start coding, many things happen at once. It is important to see them as research acts— as part of the process of analysis. One is making nodes (saying you are interested in *that*), a second is putting at the nodes references to selected text (just *this* is interesting, and about *that* topic). A third is visiting the node to see what's there, maybe recoding it, or coding on. While this early momentum is building, it's easiest to use Free Nodes for holding coding.

The process of thinking about a code will sometimes also prompt a memo, and similarly, the process of writing an annotation or a memo can assist in clarifying what codes are needed—so the processes work best when they are integrated. At first it will feel that progress is slow, but soon the project is growing simultaneously in several areas, and they are linking in a web of data, categories and thinking.

In what follows, we focus on the coding processes in this web of interpretation. We describe coding from the Document Browser, but note, exactly the same processes apply if you are coding from the Node Browser.



Develop ways of recalling your research questions (e.g. in the Project Journal). It is amazingly easy to become beguiled by fascinating “red herrings”. But also be aware that red herrings may indicate rich fishing fields for concepts as yet undeveloped—perhaps this “irrelevant” idea should be stored, just in case it becomes relevant later?

Coding at new nodes

When something is interesting, decide *why* it is interesting and why you are interested in *that*. These questions will tell you whether you want to annotate or code. If you want to make a category for *that*, think coding.

Using the Speed Coding Bar



- Highlight the passage of interest, and think about the categories required to do justice to this passage. It probably needs several.
- Create new nodes either by In Vivo coding or by typing a new node title. For now, allow them to be Free Nodes: in Part 5 we will begin to shape them into a more efficient system.
- Think carefully about what the passage is saying: *what* it is about (descriptive type codes), *why* it may be important (analytic type codes).
- Consider also whether it might be useful also to write a memo related to the nodes you have created (especially the analytic ones).



New nodes can proliferate in this early stage, but as themes recur in further documents, it is more common to be re-using existing nodes (see below). If re-use doesn't happen, worry! The categories may be too specific to apply to more than one document. Or the data may be too atomistic to give a general picture and synthesis of what is being studied.

Coding at existing nodes

If the node needed already exists, there are several ways to find it quickly and select it to take the coding of a highlighted passage.

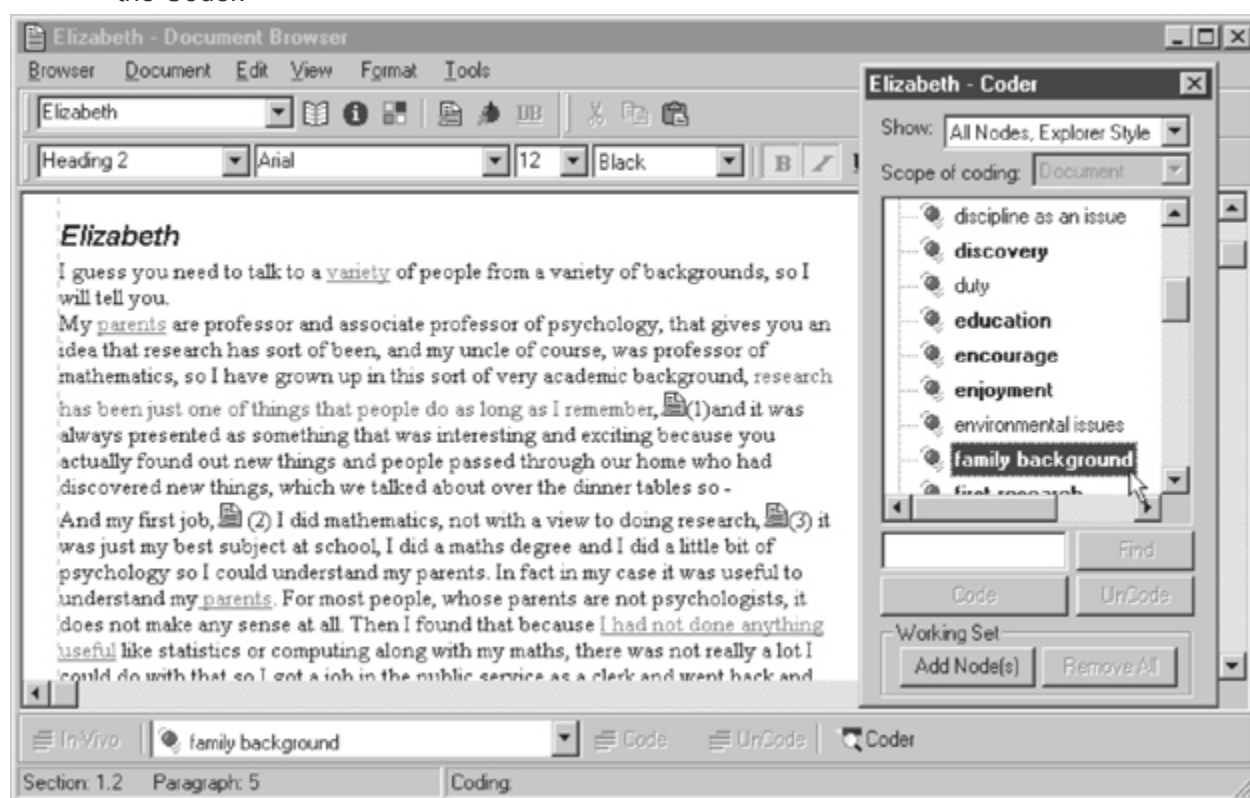
Recently Used Nodes



- Select a recently used node from the Speed Coding slot and click Code—as described in Part 2.

Use the Coder to work with nodes

- Click on Coder in the Speed Coding bar. The Coder will open to the right of your document, showing Free Nodes already created. (“Jiggle” this around the screen, to get it in a position allowing access to the Browser's scroll bar.)
- Select a passage and click on the Free Node you wish to code it at. Click on Code at the bottom of the Coder.



Drag and drop coding

- Having highlighted a passage, select a node in either the coder or Node Explorer. “Drag” it to the passage (or drag the passage to the node). The cursor will show a “no go” symbol  until it is on a highlighted passage, or a node, then a “drop”  sign to indicate it has reached “home.”
- The node title becomes bold showing there is coding at that node. If that node is selected in the Coder, the text coded at it is highlighted in the Browser.



Selecting the text to be coded can be speeded if the selection is to be a whole sentence, paragraph or section. From the Edit menu of the Document Browser, choose to Select Word, Sentence, Paragraph or Section. *Alternatively*, a double-click will select a word, and four rapid clicks in a passage will select the whole paragraph.

Checking Coding

Qualitative coding often requires reflection on previous coding, not merely processing of more data. An overview of coding in either Document or Node Browser can be obtained by viewing coding stripes, but you can also check what has been coded at particular nodes in more exact detail.

Use the Coder to review coding



- Open the Coder.
- Click on any bold node title (bold indicates that there is coding there for the open document). The exact text coded at that node will be highlighted with a teal color, or with red if the text is also selected.
- Interested in what's coded at that node from another document? Select any Recently Used Document from the slot at the top left-hand corner of the Document Browser. Click once on it and it will open in the Browser. Any text already coded at the node selected in the Coder will be highlighted.

If you have so far resisted the temptation to go to the node, now is the time!

Coding-on from a node

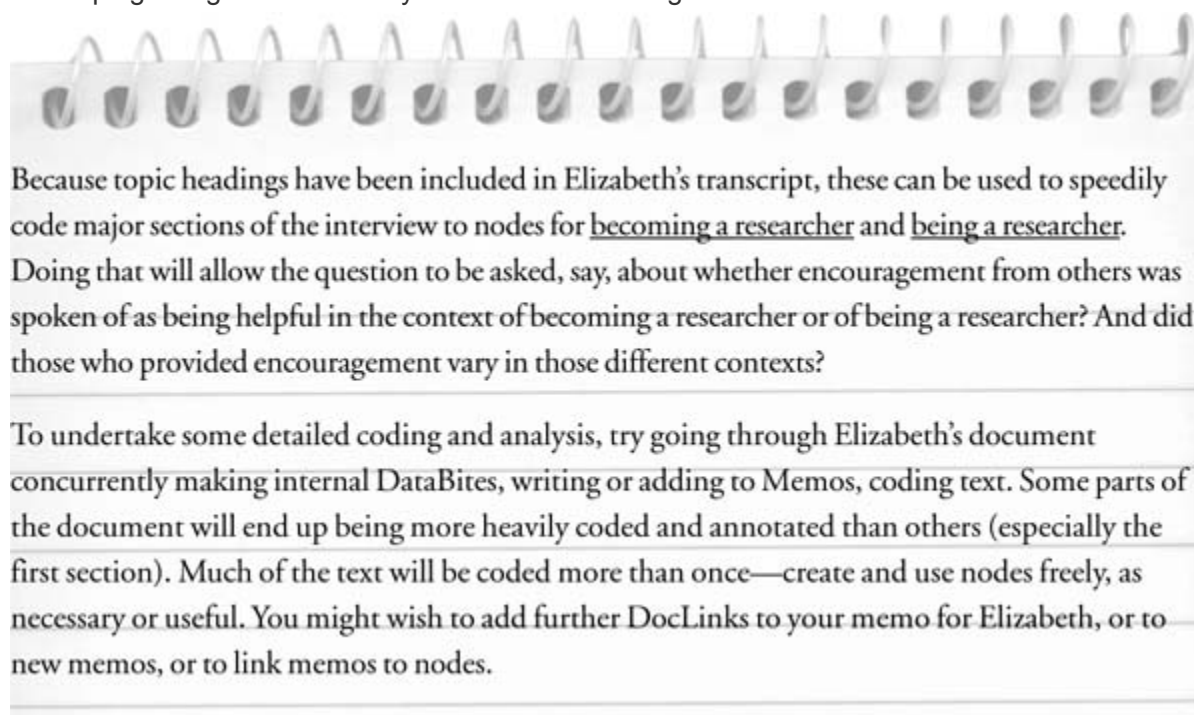
At a node are the references to all the text you have coded there from any document. So browsing a node shows you the data in a quite different context—that of the category, not the original data source. In Part 2 we saw that the context of the coded segment is always available. But so too is going *on* from the first coding, to rethink and develop your coding. All the coding facilities available for documents are also available in the Node Browser. So at any stage when you are considering the material you have coded at a category, you can code on to new or existing categories.

Browse and code a node



- Browse the node of interest (select it in the Node Explorer and click Browse) 
- Reflect on the material coded at this category. It's likely to be an odd collection, since you have not yet refined the meanings of your nodes.
- Use the normal methods of coding, exactly as in the Document Browser, to code this material on to further existing categories.
- Use the methods for making new nodes to create finer categories to express the subtler interpretations you are now making.

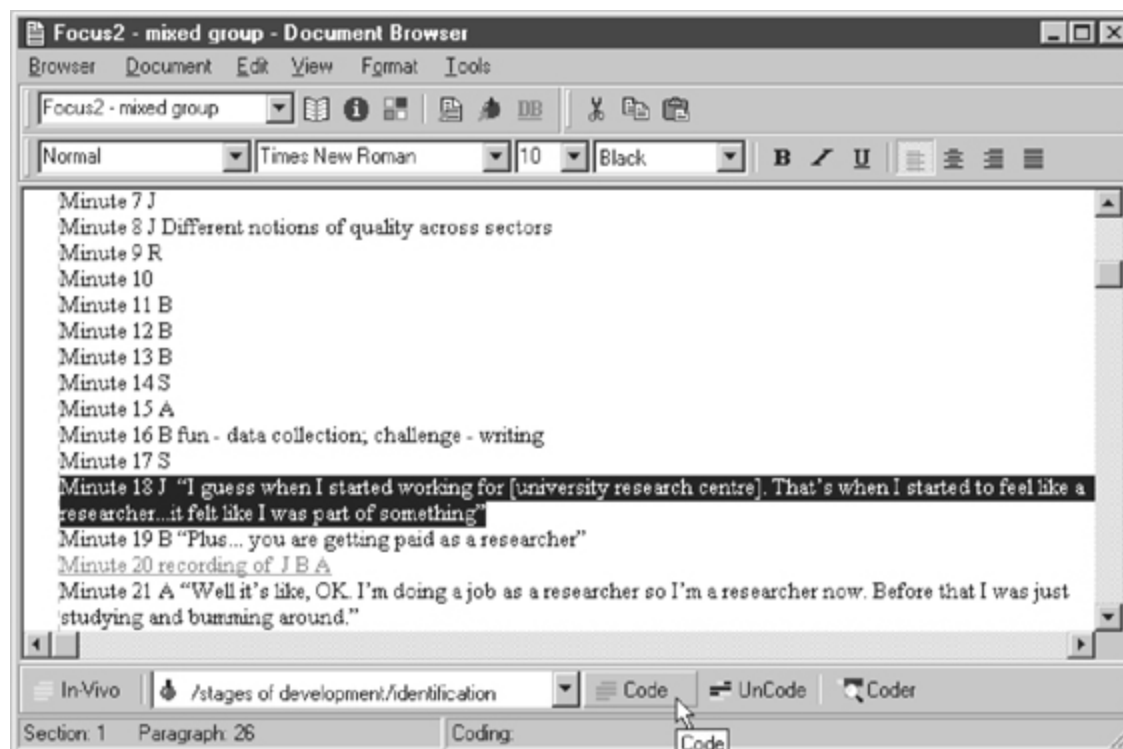
You just did something quite extraordinary. Before they had a live Node Browser, qualitative researchers could not code-on and so reshaping coding involved a much more complicated process of making notes about desired changes and then returning to the documents. Later we will use the ability to code-on as a way of developing categories from early “broad-brush” coding or from the results of a text search.



Coding a proxy document

The processes outlined above for coding and reviewing coding apply equally to coding of a proxy document. The only difference is that when coding the proxy document's text, you can code references to units within the original source. The proxy document can include additional notes, memos and DataBites as well as pre-formatted references to e.g. tape count or page number.

Retrieval of coding of a node will show all these details, including the unit of the original source coded there, and will offer the usual access to linked documents and annotations.



Avoiding pitfalls in coding

The multiple tools provided by NVivo encourage the analyst to think about each document in detail whilst also making links, making comparisons and identifying commonalities. The results of that thinking are recorded in codes, memos, journals and models. These tools help you to move beyond “disposing” of data into coded segments, and to begin to see patterns and process. The contexts of coded segments are always available, and thinking about coding is always ongoing. The danger is that coding becomes a seductive task, a way of moving through data and gaining a sense of achievement, but also, to some extent, a way of deferring thinking about the data. Maintain the discipline of memoing ideas generated by the data, and of journaling thoughts related to the project as a whole. It is useful always to read quickly through a new document, to get a sense of what this document is about and to record that, before thinking codes. And then to say at the end, “Well, what did I learn from that?” Because you're thinking about the participants and the concepts being generated, other experiences, other conversations, other snippets from previous reading are recalled. Consider these, and if noteworthy, record that in the Project Journal.

Reflections on a document

Having imported and coded a document, you might wish to reflect on what you've learned so far in answer to your questions. There are several ways to do this. By reporting on documents and nodes you can review progress. You can add further comments to the memo for your document, or in your project journal. Or you can sketch new insights as a first component of a new model.

Making a text report of your document

For many reasons you may want a report of the document *out* of NVivo. If the text was edited in NVivo, you may wish to save the new version as a word processor file. You might wish to make a report (perhaps showing coding stripes) to include in a project discussion. The text of your DataBites and details of other links can be shown as endnotes to the text report. Or you might want a report of your document with paragraphs and sections numbered. After coding, this will allow you to quickly see the overall structure and where, within that, text retrieved at nodes is coming from. Before engaging in on screen coding, you might want to take this type of report and sit under a tree, away from your computer, and scribble on it as you read, deciding on coding and memos.

Making and printing a report



- From the Document menu at the top of the Document Browser, choose Make Text Report. *Or:* In the Document Explorer, highlight the document name, and from the Document menu choose Make Text Report.
- Choose to Show DataLink details as end-notes.

The document will open in a new Text Editor. Any internal DataBite annotations will be listed at the end of the text (in full), along with any references to external DataBites or to memos (without the actual text). If the default options were unchanged, the sections and paragraphs of the document are numbered.

- From the File menu choose Print, or click on the printer icon in the toolbar of the Text Editor.

Saving the file so that it can be opened in your word processor

- Rather than print, from the File menu in the Text Editor, choose Save As.
- The Save As dialogue will open with Text Files (*.TXT) showing in the Save as type slot at the base of the dialogue. Click on the arrow to reveal and select Rich Text Files (*.RTF) as the format for your report, if you wish to preserve its formatting.
- Provide a name for the report file. Note that NVivo defaults to its own Reports folder (C:\QSR Projects\[your-project]\All Users\Reports) when you choose to save a report. It's useful to save reports there in the first instance to avoid losing them!
- The report can now be opened in your word processor.

Printing a report with coding stripes

- In the Document Browser, from the View menu choose Coding Stripes.
- From the Browser menu, choose Print. Click OK. (Note: there is no option for limiting the size of print or the number of pages to be printed.)

- To change the paper size or orientation, click on Properties in the Print dialogue before clicking OK. Adjust the options on the Setup and Paper tabs in the Properties dialogue as needed.

Warning: Page settings have been set to default to US letter: if you use A4 you will need to change these.

Reflecting in a model

By reporting documents and reviewing nodes you can reflect on the shape of your project. By editing memos or a journal you can capture ideas developed through working with the document/s. But it helps also to be able to draw, and sometimes re-draw developing or changing ideas about the processes that are occurring. In Part 1 the Modeler was first used to draw preliminary ideas about your project, using items created within the model. In Part 2 nodes from your journal were built into the model. The tools discovered there can now be used to sketch what has been learned from your first document.

As you work through your documents and ideas develop, change the model, add to it, add it as an item in a later model, or simply store it as a record of your thinking at this stage.

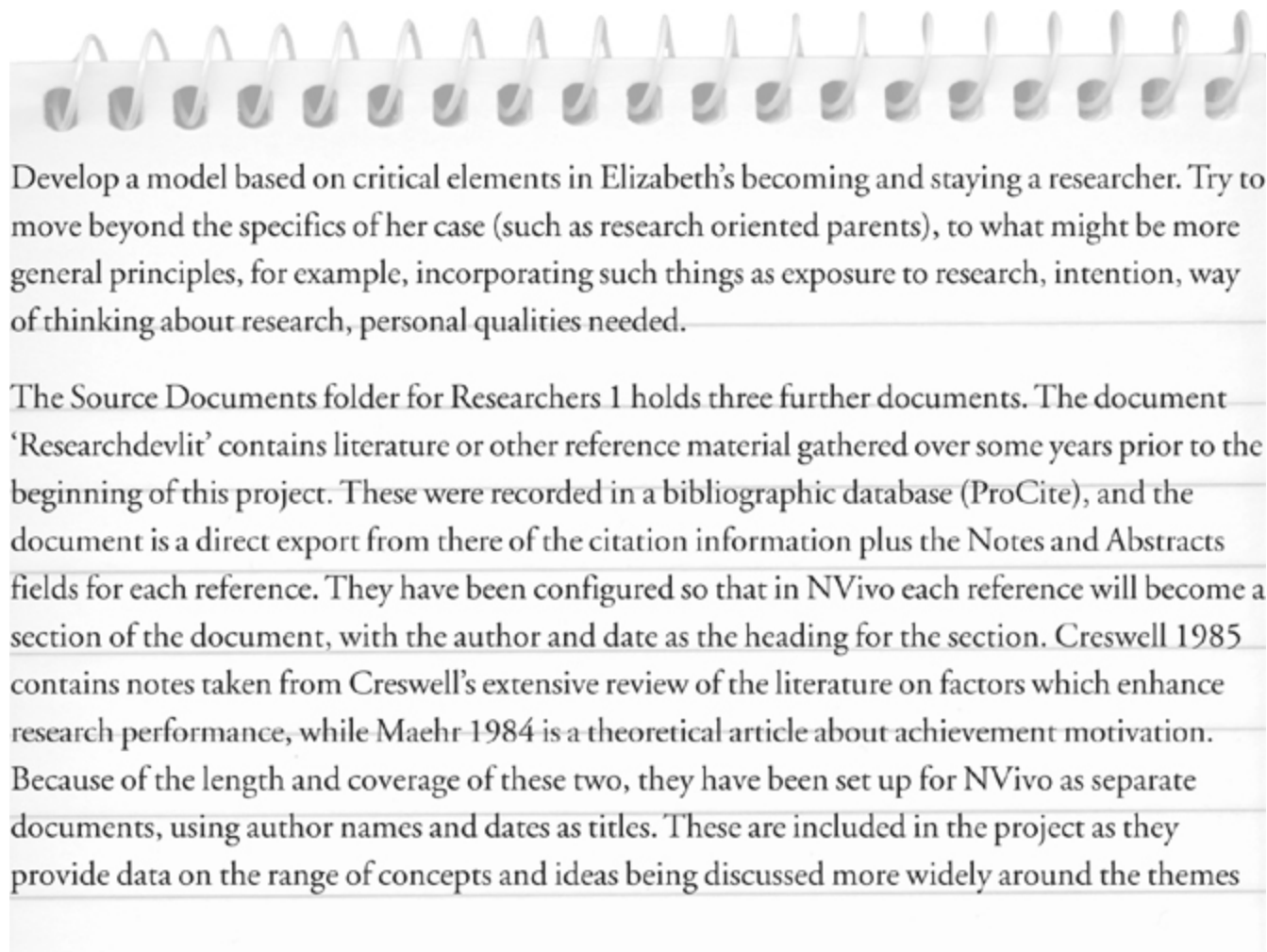
To create a new model



- Open the Model Explorer and expand the window.
- From the Model menu, choose New. A New Model will be listed under Models on the left side of the Explorer. Click on New Model (or highlight it and click on the Properties icon) and type in a name for your new model.

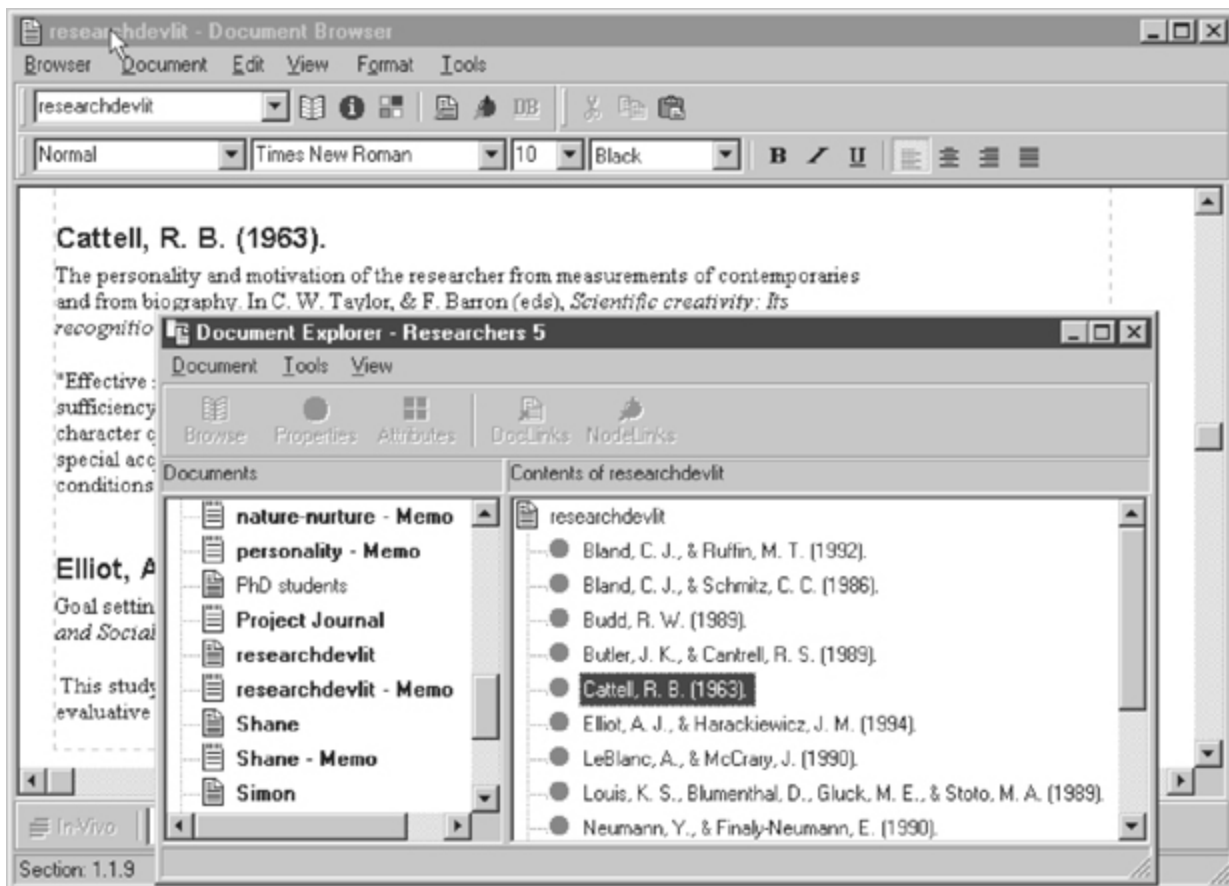
Adding items to your model—a reminder

- Click on the Nodes icon in the toolbar or choose Add Node from the context menu. From the dialogue, select the nodes you want, one at a time. Drag them around the screen, thinking about how you would want them to relate to each other.
- To create an additional concept for the purposes of the model, click on the New Item icon in the toolbar (or access from the context menu) and double-click (or chose Properties) to name the new item.
- Add in a document—linking in memos about particular ideas can be useful.
- Add links between the nodes/concepts/document on your screen, to show relationships between them. Choose to see a link's Properties to add labels or change its appearance.



Develop a model based on critical elements in Elizabeth's becoming and staying a researcher. Try to move beyond the specifics of her case (such as research oriented parents), to what might be more general principles, for example, incorporating such things as exposure to research, intention, way of thinking about research, personal qualities needed.

The Source Documents folder for Researchers 1 holds three further documents. The document 'Researchdevlit' contains literature or other reference material gathered over some years prior to the beginning of this project. These were recorded in a bibliographic database (ProCite), and the document is a direct export from there of the citation information plus the Notes and Abstracts fields for each reference. They have been configured so that in NVivo each reference will become a section of the document, with the author and date as the heading for the section. Creswell 1985 contains notes taken from Creswell's extensive review of the literature on factors which enhance research performance, while Maehr 1984 is a theoretical article about achievement motivation. Because of the length and coverage of these two, they have been set up for NVivo as separate documents, using author names and dates as titles. These are included in the project as they provide data on the range of concepts and ideas being discussed more widely around the themes



of developing academic researchers and building up (or predicting) faculty research performance. Note how headings have been used so the vital author and date information needed for writing up literature will be easily accessible from the Document Explorer. These documents provide further sources of data for the project.

Taking stock

At this stage in a project, you can seek a sense of momentum. The project is under way. You should be able to reach for the tool you need to record or locate the idea, the insight, the links, the connections that are starting to emerge from the data you are now handling in detail. As you get more surefooted, these will feel less like techniques used in sequence and more like thinking aloud. The ability to code in different ways, and to code on from the results of your coding, opens significant new paths to analysis. Having worked through one or two early data documents, you may find you need now to change the pace of your processing of data, as data gathering often gains momentum. A snowball sample behaves like a snowball, growing alarmingly as it rolls on. Entry into the field is suddenly facilitated. You need tools for moving data into the project fast, without damaging it. Broad-brush coding allows you to code now, rethink later.

Just as the project is now underway, so too your system of thinking about the data is integrating. At this stage, the research process should seem driven by the data and the ideas generated by the data, not by predetermined sequences. And the project should start to seem like or develop as a web of interconnected data and ideas, not discrete parts. Now you can start to shape the project as a whole, so you can manage

the data more skillfully, and do greater justice to the ideas you are crafting into embryonic theories.

<http://dx.doi.org/10.4135/9780857020079.n4>