



Moving faster

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: 133-156

© 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.

Moving faster

Your project is now taking shape. Having worked through the first few documents in some detail, you are probably gathering analytical insights. If the data are fairly homogeneous, by the time you have coded around six documents, you are likely to find the rate of expansion of your nodes is lessening. Conceptual shifts can occur at any time, but generally speaking it is unlikely that you will undertake another major reorganization of your index system in the way described in Part 7. Now you will want to pursue those ideas through more documents, whilst remaining open to new understanding and insights from any additional data. At this stage you will be helped by tools that automate clerical work, so you can concentrate on interpretive work. In this part, we deal with ways of speeding up document import, coding, attribute import and handling of sets and cases.



“Batching in” multiple documents

It is possible in NVivo to import multiple documents at the same time, so long as their title and description paragraphs are set up on a similar basis. If you now have another four or five documents ready (for most projects, this is much better than 40 or 50!) you might want to import them all at once. Then you can move to a rhythm of coding/ memoing/ annotating/ linking them, with additional techniques to assist in working through.

If “batching” documents in multiples has never occurred to you, please don't feel you should be doing it! This is a good instance of the crucial rule that you should never feel required to do something because your software can. Many methods make qualitative data records slowly and/or process them singly. As you type up your field notes, for example, they are coded and your new understanding reviewed. This new understanding

directs your next excursion into the field. It would make no sense to type a year's field notes in your word processor, and only then batch them into NVivo and start coding, linking and writing memos.

If you are able to control the acquisition of data documents, our strong advice is to do so. Qualitative methods are always enhanced by the researcher's ability to learn from the data, before moving on to test emerging ideas in the next record, with further records created by "theoretical sampling," data-driven.

Some researchers, however, do not have the luxury of making, reflecting on and learning from each document before making more data. If you are a historian, or handling documentary analysis, or doing secondary analysis of past interviews, or discourse analysis of existing texts, the data are ready-to-go. The best advice is to batch them in and do basic coding as soon as possible—broad-brush coding, coding by text search and section coding offer alternative ways of initially grappling with larger volumes of text so that you can rapidly access the documents and assess this wealth of data. Then shape the data and plan your exploration of it thoughtfully, sampling within the available pool of data, coding on and/or working set by set.

Import multiple documents



To import several documents at once, it is advisable to have them set up so that each can acquire their document name and description from the same place.

- From the Project Pad, choose to Make a Project Document, then to Locate and import readable external text file(s). A Select file to read dialogue will open.
- Locate the folder where your files are stored. (If you are importing files in text only format, remember to change the Files of type option.) Use shift-click or control-click to select the documents you wish to import, then click on Open.
- In the Obtaining document name and description dialogue, choose the option which is appropriate for all of your selected documents.

You will now find your new documents listed in the Document Explorer.

Handling data from multiple sources

While multiple documents imported together are typically of one type, projects often also involve multiple sources of data. Multiple sources can bring many benefits, but they do need to be carefully managed in order to realize those benefits. The goal is to be able to gather all data of a particular type together and to distinguish those from data of other types. You can then ask, for example, how were attitudes to this topic expressed in one-to-one interviews, and whether they were expressed differently in focus groups.

In NVivo there are two primary ways to manage multiple sources of data: using document sets and using

document attributes. These are tools which you have begun already to use. In this Part the focus will be on extending their uses as tools for managing and searching data, as well as for working interactively between them. For example, if an attribute for data source is introduced (most efficiently, via a table of attributes, as will be shown below), this can then be used to filter documents to create sets of documents rapidly for the different data sources. Similarly, an imported attribute for cases, used as a filter in the set editor, can code all of a set of documents via the search tool at a case node. This is much less complicated than it sounds, and is a much faster way of placing documents in case nodes than coding them there one by one! The steps will be outlined below, after a few preliminaries. But first we turn attention back to the process of coding, linking and memoing documents, and provide some suggestions for working more efficiently there.

Faster coding

You will find, as you explore the text of further documents, that the rate of creating document memos slows—depending, of course, on how intensively you worked with your first few documents. Gradually your coding also will become more refined. Nodes will be merged, others will be moved, some perhaps to be grouped under an intermediate concept, others may be deleted as no longer being useful. Free Nodes will continue to be sorted and moved into trees.

If the data are fairly homogeneous, then by the time you have coded around six documents, you are likely to find the rate of expansion of your nodes is lessening. You are also very unlikely to be creating whole new trees, unless they come from a reconceptualization of the nodes already created. This means that coding itself (to your probable relief.) will become faster. There are at least two dangers lurking here though:

(a)

The danger that you will not bother to create any more new nodes, but will try to squeeze concepts into existing nodes. How to arrive at a balance between this serious problem—like trying to fit the Ugly Sisters' feet into the fine shoe—and the equally threatening problem of a total proliferation of tiny nodes? If the “squeezing” is a real effort, then don't do it. Remember, NVivo doesn't mind how many nodes you make, and will always merge them later if you discover the shoe did fit. Keep adding to the node memos as you go along as well, especially where you are unsure of the integrity of the coding there.

(b)

The danger that coding will become a task to be completed, a routine that is performed without creative thinking. Lyn calls this “data disposal”—coding in order to get complex data out of your way. When coding becomes data disposal, it is time to stop and consider if anything new is being learned from the additional documents. Or is the problem that you are not giving yourself time to allow your thinking to be stimulated by them? Qualitative coding should almost always be a creative, analytical process, and coding should be regularly “interrupted” by memo writing.

While coding should never be routine, and you will continue to think analytically about your data, growing

familiarity with your nodes and their organization assists in making more rapid decisions about which nodes to use in coding particular text. NVivo complements this process with an alternative very rapid method of assigning codes to text: once your nodes are organized in trees, you can code by typing their numeric addresses, rather than by selecting or typing their titles.

Coding by node address

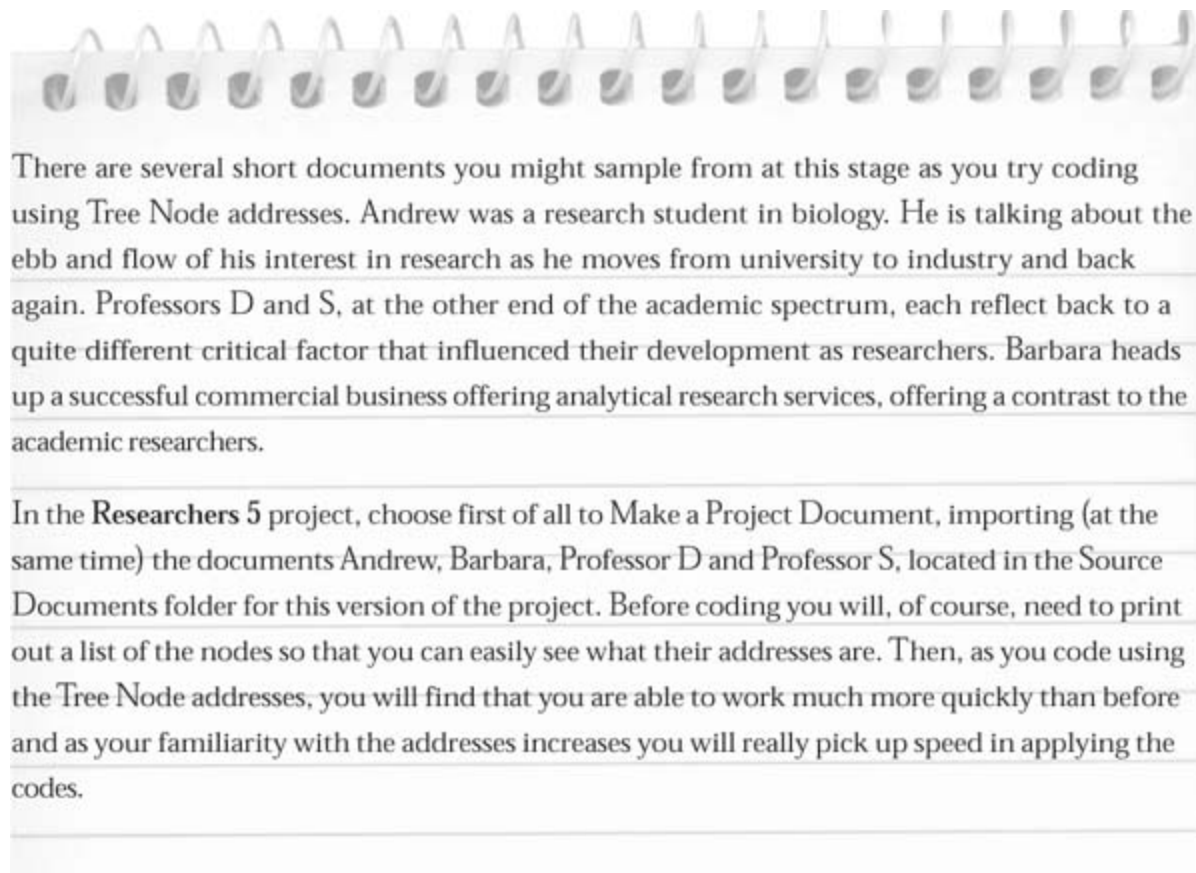


Unless you have a remarkable memory, or a very small number of nodes, work with a list of nodes beside you showing all Tree Nodes with their node addresses and titles (obtained by requesting List all Nodes under the Nodes menu).

- Working in either the Document or Node Browser, highlight the text you wish to code at a Tree Node.
- Type the Tree Node address using the numbers only, with a space between each level. The address will appear in the Speed Coding slot. (No need to put your cursor there.)
- Press Enter on your keyboard to complete the coding process. The full title of your node will be shown in the Speed Coding slot, giving you an opportunity to quickly check it was the right node!



If you are routinely coding at both Tree Nodes and Free Nodes, have your Coder showing the Free Nodes so that they are easily accessible for coding, while at the same time using the node addresses for coding at Tree Nodes.



Coding with Text Search

Like any other search in NVivo, text search finds words or phrases in the text, and codes the resulting finds at a node.

You can specify what context the coding is spread to. This means it is a way of getting coding done fast.

You may simply wish to code at a node every time a person is mentioned, a topic is raised or a question asked. You may come upon a new concept or category for which you wish to make a node, and then wonder whether it really is the first time it has come up or whether it was there before, but you just didn't see it. Text search can be used to check whether, indeed, such things were mentioned elsewhere—providing there are keywords that can be used to “get a handle” on those concepts or categories. And it will do more than just find them.

Text search nodes, like any other node, can be browsed for recoding and coding-on. Having been pointed to potentially relevant sections of your documents, you can check the context of those sections, determine just exactly what is relevant, and code accordingly. What text search is doing, then, is alerting you to where in your data you might access material about the concept or category of interest, so that you can then capture it in a node and continue analysis from there. Note that text search, like any search, can be pointed to exactly the documents or parts of documents you want to search. And if you are using it to check where you might have missed coding, it can be incorporated into a Boolean Difference search so that all those passages you

had already noted and coded are excluded from the results.

Thus text search can be used to locate and/or verify whether an expression has or has not been used, or perhaps to compare usage of, say, different pronouns or expressions in particular contexts, that is, to test ideas from and in the data rather than simply to code.

NVivo can be asked to search more widely, for any of several alternative words or phrases at the same time. This way of searching is often more useful to qualitative studies, where it is likely that there will be more than one word that will point to the text you are interested in. Finds for any of the words listed will be included together in a text search results node. But watch for the distinctiveness of the words you ask NVivo to search for. A search for media could also turn up immediately, for example.

Searching for an exact word or phrase

- From the Project Pad, choose Search Project Database. NVivo's Search Tool will open.
- Click on the icon next to Text Search. A Text Search dialogue will open over the top part of the Search Tool.
- Type, into the slot at the top of the dialogue, the word or phrase you wish to search for in your text. Leave other options as they are for this search. Click OK at the bottom of the Text Search dialogue.
- Specify the Scope for your search: do you want to search All Documents, or just a particular set of documents (or nodes)?
- Click Run Search at the bottom of the Search Tool.

When the search is complete, choose to Browse Node. Oops! All you have in the node is the exact string that was searched for. You can, of course, gain access to the text around each find (using the right mouse button menu), but it might be more useful to see a little more right from the start. At least that will allow you to determine whether finds are totally unwanted.

To view context

- In the Node Browser, use Ctrl A (Select All) to highlight all the finds at that node.
- From your context menu, choose View Enclosing Paragraph.

NVivo will display the enclosing paragraphs for every find, showing the additional text in a different color. This allows you to see more easily where the word or phrase you were searching for is located in the text, without the additional text having been automatically coded there.

Review each of the finds and determine whether you wish to keep them or not. Code additional text to the node for each find you keep (the search results node is already listed in the Speed Coding slot.)

Alternatively:

Re-run the search, but this time, before clicking on Run Search:

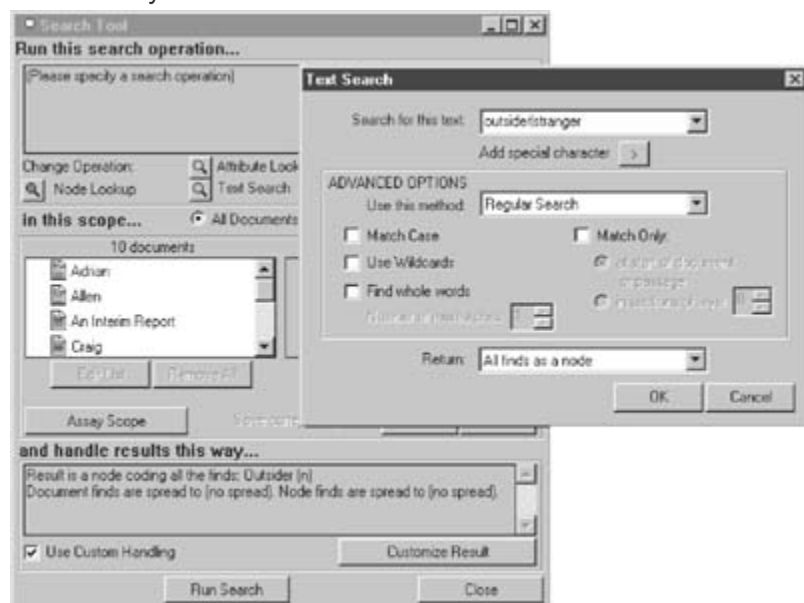
- Click on Customise Result near the base of the Search Tool.
- Choose to Spread finds in a scope document to: Enclosing paragraph. Note another option you might choose is to type in a Number of Characters you want (e.g. 50).
- Click OK. On the Search Tool, Use Custom Handling will now be checked.
- Click on Run Search. Your results will be placed in a new node, as a child of Search Results in the Trees area. Note you could collect the finds into any other node you choose -automatically merging the coding with that node.



In the Node Explorer, note that the new node has been given a description based on how it was created. If the results of the search were useful, rename the node, and perhaps move it elsewhere in your trees.

Searching for multiple words or phrases

- In the Search Tool, click on Text Search.
- Check the Use Wildcards option in the Text Search dialogue. Type the first word or phrase you wish to search for in your text. Click on the > symbol next to Add special character. Choose the alteration character ...|..., the last listed.
Or, type in the | character using your keyboard—it is usually located above the backslash (on the key above Enter).
- Add your alternative word or phrase. You can add further alternation characters and words, if you wish.
- Continue to set up your search by choosing the options you want, as before.
- Browse your results, and use the ability to see context and code further text to maximize the benefit of your text search.



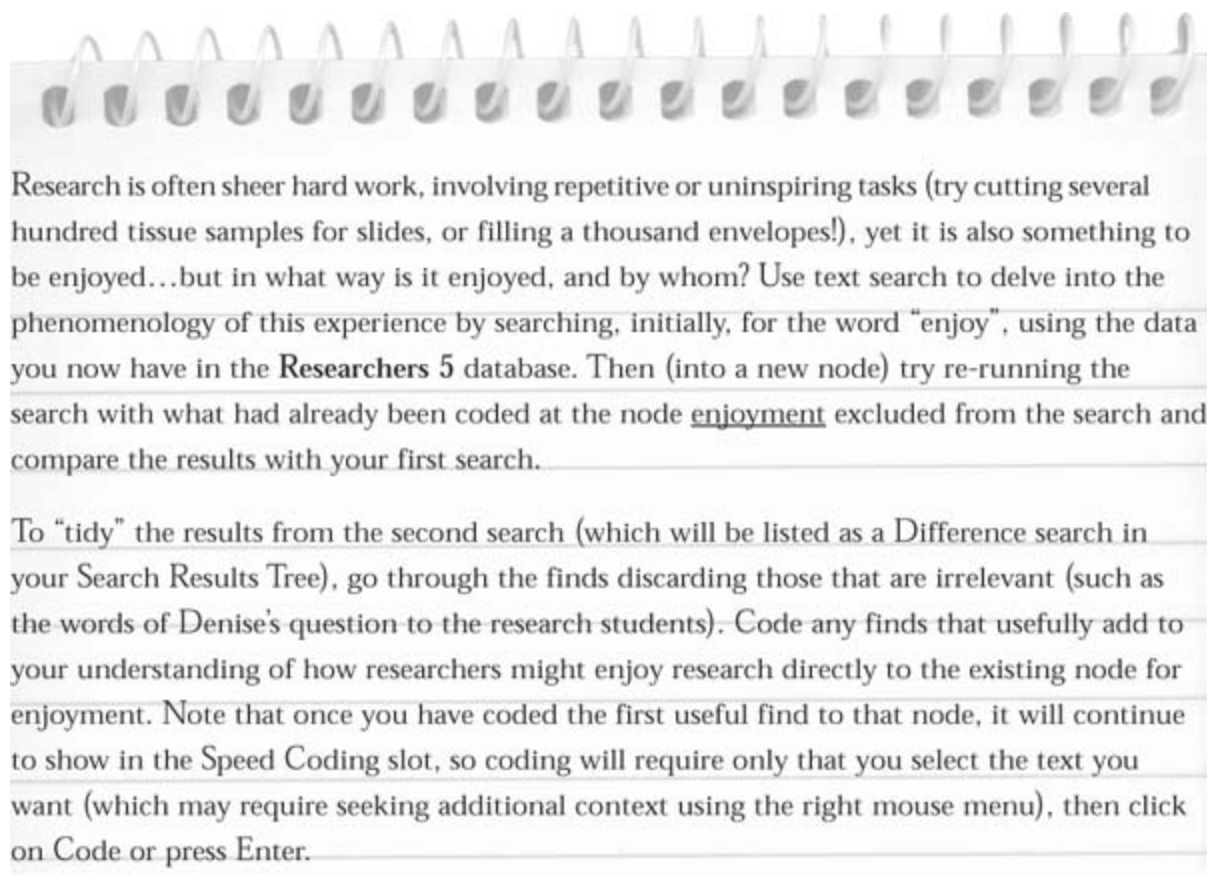
Combining text and other searches

In all of the Boolean and Proximity search operators, you can ask for combinations of searches for text, coding and attributes. This allows much more subtle searching of text, retrieving for example only the words that occur in particular discussions, or in documents from interviews in one social area. Use a Boolean search to find occurrences of a word that have not yet been coded at the node for that category.

Excluding what has been coded already



- In the Search Tool, choose to run a Boolean Search, with Difference (Less) as the Operator.
- Under Find text referenced by any of these items... select Choose Text Patterns....
- In the following dialogue, type in your search expression, with whatever options you wish to use. Click on Add Pattern, so that it shows in the pane to the right of the dialogue (remember to click Use Wildcards if you wish to specify multiple phrases using the alternation character).
- Click OK to return to the Boolean Search dialogue, where you will see the text pattern now showing in the top pane.
- Under ...but by none of these items, select Choose Nodes. Add the node or nodes which already hold coding for the concept you are searching for. Click OK to return them to the Boolean Search dialogue, then OK again to return to the Search Tool.
- Set the scope for your search (you may wish to leave this at All Documents in this case).
- Use Custom Handling to spread the finds to, say, 50 characters either side. This will give you enough to assess whether the find is worth pursuing, without overwhelming you with long passages of text.
- Inspect the results, and code-on as necessary.



Autocoding by section

The quickest coding is achieved with the Section Coder, which will “autocode” sections of the text under nodes made for each of the headings used in the text, thus in one swoop creating the nodes, determining the context (i.e. sections) and doing the coding. You can set it to create the nodes either by the order and level of the sections that are in the documents, or by the titles of the sections (e.g. name of speaker). Each of these forms of section coding achieves a quite different result, so they are appropriate for different tasks or data.

Coding by section number requires that all documents to be coded that way are set up with exactly the same structure of headings and sections. It is therefore ideal for survey data, and benefits from the use of a document template to guarantee uniformity in preparation of the documents. Assuming the content of the questions are being placed in Tree 1, and Heading 1 style was used for main questions with Heading 2 for subquestions, it will result in the answers from all respondents to Question 1 being placed at Node (1 1), those to Question 2 going in to Node (1 2), the answers to Question 2 (b) into Node (1 2 2), and so on.

Coding by section title does not require any uniformity across documents, but expects that there will be repeated use of the same headings, as occurs, for example, when there are multiple speakers in a focus group, or a series of interviews that cover the same topics but not necessarily in the same order. Thus, assuming that speaker identifiers have been used as headings in the focus group transcript (and with no subsections under them), all that was said by each person will be gathered into a single node with their speaker identifier as its title. (If multiple focus groups are conducted, make sure you use unique speaker

identifiers for members of each group.)

In either case, where subsections exist, the heading only (or the heading and just that text appearing before the first subheading) will be coded. If there is a need to code all the text at that level as well, this can easily be achieved. Thus, in a focus group, if you have used headings for both the topic being discussed and (at a lower level of heading style) who is speaking, you will be able to gather the text separately for both topic and speaker.

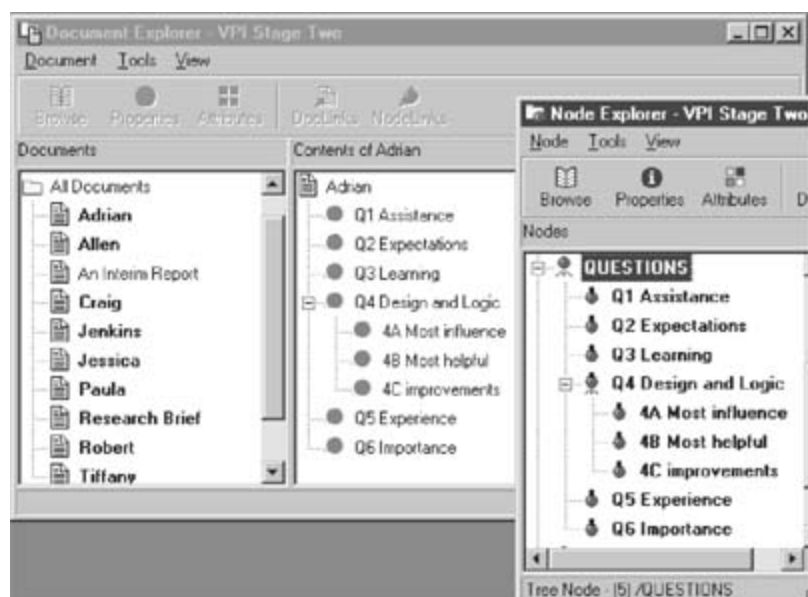
Preparation



- For each of the two examples below, you need to create a new Tree Node (using either the Create Nodes dialogue, or in the Node Explorer using the context menu) below which you can place the nodes that will be generated by the Section Coder. (Note the address of that node.)
- Check the documents' formatting; will the sections properly identify material you wish to code?
- Import (as a batch, if they are surveys) the new documents you wish to code by section.

Coding by section number

- From the list of All Documents in the Document Explorer, select (highlight) one of your survey documents, then in the Document menu, click on Code by Section....
- In the Section Coder dialogue, type the address for the Tree Node you have just created into the Parent tree-node slot (or click on Choose and select it from the list of nodes). Click OK.
- Repeat these two steps for as many surveys as you have imported (note: a multiple select option is being planned for this function).
- Go to your Node Explorer, and expand Trees so that you can see the new Sections (Questions) node. It will now have a + next to it, indicating it has children. Expand it and you will find that it has a child node for every new heading found in your document. They will be arranged in parallel structure to the outline of the sections in your document.



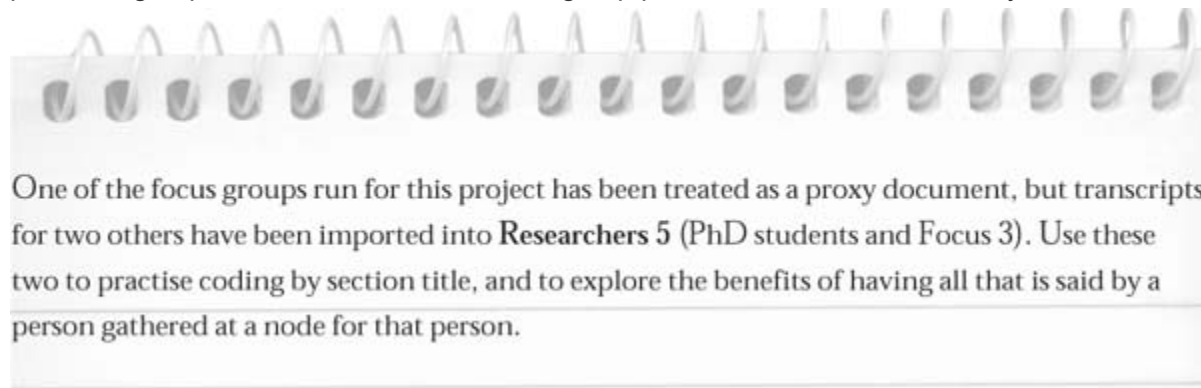
Now you are able to see at a glance what every respondent answered to Q1 (and each other question). You might then simply “split” these by values on an attribute (see below), or code-on as you focus on the topic of that question. Later, you might ask whether something that was said arose in response to that question, or to another.

Coding by section title

- From the list of All Documents in the Document Explorer, select a focus group (or similar) document, then in the Document menu, click on Code by Section....
- In the Section Coder dialogue, change the selection to Code sections by title. Type the address for the Tree Node you have just created into the Parent tree-node slot (or click on Choose and select it from the list of nodes). Click OK.
- Go to your Node Explorer, and expand Trees so that you can see your new node. It will now have a + next to it, indicating it has children. Expand it and you will find that it has a child node for every different heading found in your document. These will include both topics and speakers, if you had both.
- Browse any of the speaker nodes, and you will find there all that person said during the course of the focus group, (so long as the subheadings were the same for each of the speeches).
- Browse a topic node, and you will probably find just the series of headings. Use Ctrl-A to select all the headings at once. From your context menu, choose View Enclosing Section. All of the subsections for that topic for the whole document will come into view, in differently colored text.
- Select all the text again (Ctrl-A), and click Code, to code all the additional text to the node for that topic (this node will automatically be showing in the Speed Coding slot).

If you have more than one focus group covering the same topics, then defer expanding the topic sections until all group documents have been section coded.

Now, by browsing the section nodes, you are able to review everything said in each of your groups about the topics covered. And having coded all that each speaker said, you can now review those nodes and note whether it was the same person who repeatedly used a particular expression, or note the way in which particular group members contributed to the group process, or were influenced by it.



Faster attributes

Earlier, we noted the crucial difference between the conceptual act of coding (creating a topic category and recording there the data that belongs) and the descriptive task of storing information. Attributes were created for documents in Part 3.

You now have quite a collection of documents in your project, and doubtless a lot of coding to do. Putting in the information you have about each document may be a simple matter of a few clicks as you first import or browse each one, or it may be more tedious because of the amount of information you have available about that document. If your study also involved a quantitative data collection, you may want to import salient variables from your statistical database, to use as attribute data in your qualitative analysis. Consider then, setting up your document attribute data in a table and importing it all in one step, rather than having to add the information document by document. (Note that attribute data can be imported in a parallel way for nodes as well—to be discussed in the context of using Cases, below.)

There are two steps to this process. In the first, you need to prepare your data in table-based software, such as Excel. The Appendix provides details of how to do this. In the second step, you will import the data into NVivo. If the table you are importing includes attributes with different types of values (say, most are strings but some are numeric), this will require an additional preparatory step.

Preparing for importing attributes

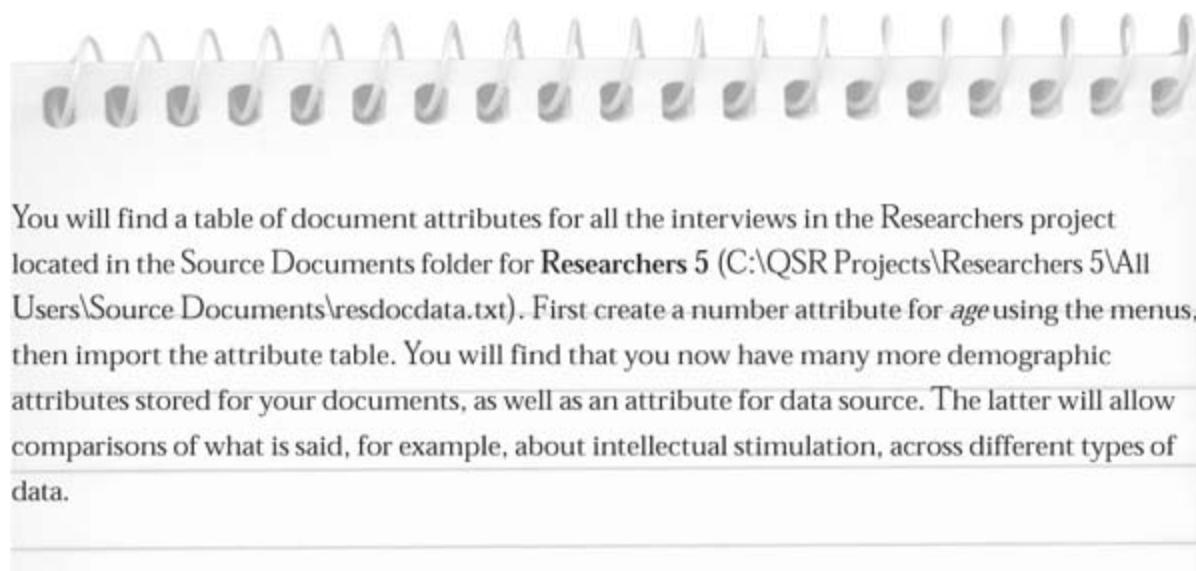


- Prepare your data in Excel (or other table-based software) as per instructions in the Appendix. Save your file in text only format.

- *If your attributes are mostly of one value type, with just one or two of a different type:* Create those attributes which are different from the majority interactively (i.e. using the menus) in NVivo, *before* importing the table (no need to specify values, just the attribute name and value type). Make sure their names are typed in exactly the same way (including capitalization) as they will appear in your table.
- *If you have a larger number of attributes for each type* (e.g. several string attributes, and also several numeric attributes): Set these up in two separate tables, e.g. one for the strings, and another for the numerics, so they can be imported separately.

Import your attribute table into NVivo

- From the Documents menu at the very top of your screen, choose Import Document Attributes...
- At the top of the Document attribute import dialogue, where it says File to import, click on Choose.
- Navigate through your directories to find and select the text file version of the table you have created.
- Check through the options offered for handling attributes in the Document attribute import dialogue.
- If a document title is not recognized: You can choose to allow NVivo to create any documents that do not yet exist (the default option). For those who plan to prepare their documents in NVivo, this provides a rapid method of creating multiple documents. Those who plan to import further documents should select the option Ignore the data for that document to avoid the situation where they set up parallel documents within the program (one renamed and with text, the other with attribute data). If further documents are imported there is no problem in re-importing an attribute table in order to update the new documents with attributes.
- If an attribute name is not recognised: If you are importing other than String attributes, you will need to change the selection for Attribute type.
- Click OK. You will be given a message: Attribute import successful.
- View your imported data by choosing Explore all Document Attributes from the Project Pad. If you had your Attribute Explorer open during the import process, you will need to close and reopen it to see the added attribute data.
- You are now able to view and manipulate your attribute data in tabular format within NVivo through the Attribute Explorer (as described in Part 3).



Faster Sets

We have made sets of documents and nodes so far in the simplest possible way, dragging and dropping in the Explorers. But NVivo has a finer tool for making, changing and filtering items in sets, a tool that has a “hotlink” to the scope of a search. Indeed, you have already met the Document and Node Set Editors, as you have chosen to limit searches to particular (already created) sets of documents or nodes. But now that you have a larger number and range of documents, you will find the Document Set Editor more efficient also for making new sets. As an alternative to dragging documents to a set in the Document Explorer, you can use the values of an attribute (or some other criterion) to filter your documents, saving the result as a new set. This is particularly useful where there are a number of documents to be selected, or where you are not sure which documents meet the criteria you want to establish for belonging to the set.

Use set filtering to make a new set

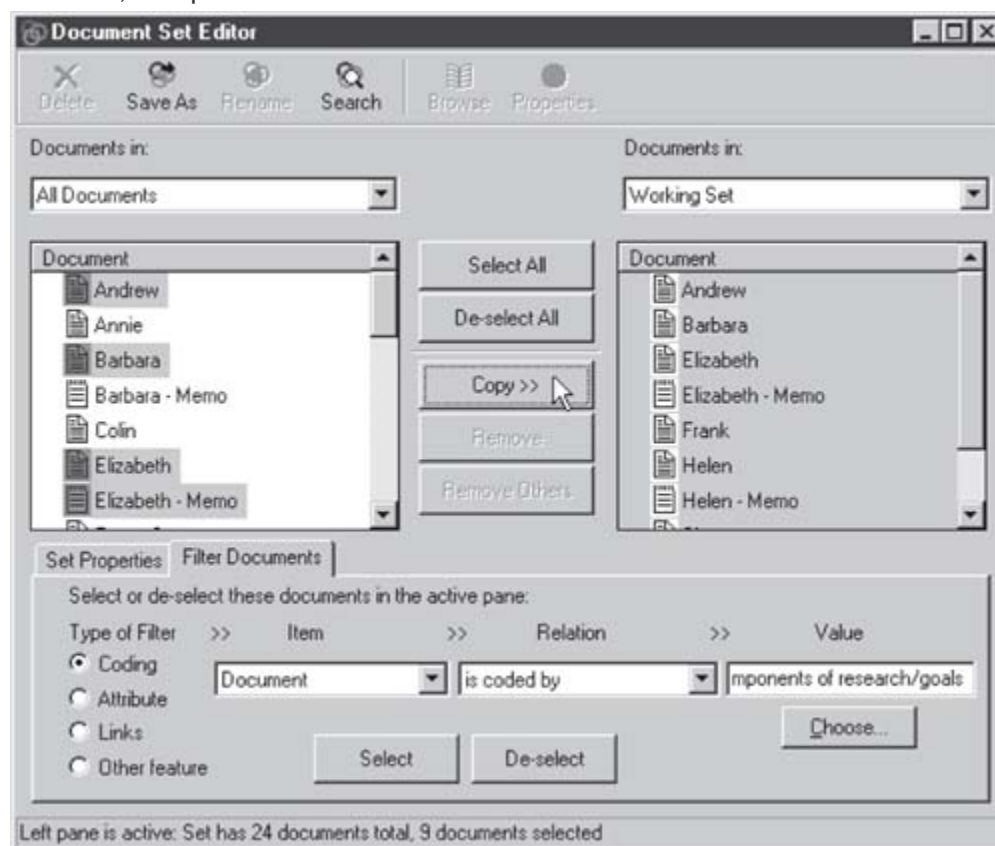


- On the Project Pad, ensure the Documents tab is active. On the right side, click on the Sets tab, and then on Make or Change a Document Set to open the Document Set Editor.
- In the Documents in: slot above the right pane, select Working Set. If the Set contains any documents, then choose Select All, then Remove, to clear the set.
- In the left pane, select the set of All Documents. This pane should now be active.
- Click on Filter Documents. Choose whether you want documents coded by a particular node (they will be selected if coded anywhere with that node), or perhaps documents with a particular attribute, such as their data source.
- Click Select at the base of the Set Editor. The documents which meet this criterion will be selected.

Click Copy ≥ to move them to the right pane.

- Click in the right pane to make it active, then on Save As at the top of the dialogue. Type in a name for the set, and click OK. Close the Document Set Editor.

A parallel process can be used to create Node Sets—perhaps to locate the set of nodes that code your literature, or a particular document.



Now that you have many more documents in **Researchers 5**, and you have imported attributes for them—one of which was an attribute for data source—use that attribute to add new documents to the Set already created for Interviews (watch how you manage that in the Set Editor!) and to create a new Set for Focus Groups.

Faster cases

If you have started to use case nodes, as they were introduced earlier (Part 5), you are probably coding text into them interactively, one segment or document at a time. Now that you have learned to use the section coder, and to use attributes to filter sets, there are faster ways you can create and move text into case nodes. Indeed, the fastest way to create case nodes in the first place may be to begin by importing a table of node attributes for named cases (see below).

Coding text (partial documents) at case nodes

- Revisit coding by section title (above) to work out a strategy to autocode the text of focus groups or other relevant documents.
- Drag nodes created with the Section Coder (e.g. for participants) from their Tree to a Case Type node (where they will become a Case under that Case Type), or use Cut and Paste to move them there. If the case nodes already exist, then use Cut and Merge Node to add the additional text to the case node.

Coding sets of documents at a case node

- When you are creating your document attribute table, add an attribute for cases, with values that define which case node each document is to go into. For example, if you have carried out multiple interviews with Jennifer over time (or with multiple people about Jennifer), then under the cases attribute, give each of these the value Jennifer.
- Use a value for the cases attribute (e.g. Jennifer) as a filter to create a set of documents for that case. Send the set to the search tool (no need to save it first). They will appear as the scope documents in the search tool.
- Under the scope panels, choose the option to Save current scope as: Node. A Choose Node dialogue will open.
- Using the drop-down list and panel in the Choose Nodes dialogue, choose the appropriate Case Node (e.g. Jennifer) for the set of documents showing in the scope. Click OK, and your documents are now coded at the selected case node.
- Repeat for each value of the cases attribute, until all documents are coded at their respective case nodes.

It is best to have the destination nodes already created (you might use a node attribute table to do so—see below) before trying to place the documents, especially if you are doing this for a number of cases. But if you forget to do so, you can create a new case node (but not name it) by clicking on the star (create) icon  in the Choose Node dialogue.

Importing Attributes for Case Nodes

Nodes too can have attributes. To give attributes to a Case Node (e.g. for each participant in a focus group, or each pupil in a school) allows demographic and other variable information to be applied to parts of documents, whereas document attributes apply to whole documents. Then you could ask for example about the contribution made on a topic by participants in different age groups.

As with document attributes, node attributes and their values can be created interactively (the process is exactly symmetrical with the document attribute creation). If you create a node attribute with the same name as a document attribute, NVivo will recognise it and you will be asked if you want to use the same values. But also, as with document attributes, you can create and import node attributes from a table prepared in a spreadsheet or similar program.

Once values of an attribute have been given to a node (such as a Case Node), any text coded to that node (at any time) will acquire those attributes. Additionally, attribute values assigned to a Case Type Node will be acquired by any *new* Cases created thereafter under that Case Type (of value where there are a number of Case Type nodes each with cases under them). With a little advance planning, the Case Type Nodes and Case Nodes can be created and assigned attributes all in one go, by creating and importing a carefully constructed Node Attribute table. They are then ready to have documents (or part documents) coded at them, as outlined above.

Importing attributes for existing case nodes



- Follow the basic instructions for preparing a Node Attribute table, given in the Appendix.
- Note that the format for a case node title uses a full stop in front of the case type, and a colon in front of the case, thus .Schools.Penrose:Julie
- Save your table in the usual way (i.e. first in its native format, then as text only).
- In NVivo, from the Nodes menu at the top of your screen, choose Import Node Attributes...
- At the top of the Node attribute import dialogue, where it says File to import, click on Choose. Navigate through your directories to find and select the text file version of the table you have created.
- Change the first of the default options in the Node attribute import dialogue, so that If a node title is not recognized NVivo will Ignore data for that node.
- Click OK. You will be given a message: Attribute import successful.
- View your imported data by choosing Explore all Node Attributes from the Project Pad. Note that if you had your Attribute Explorer open during the import process, you will need to close and reopen it to see the attribute data.

Using a table to create Case Nodes and assign Case Type values

- In the first column (headed Node) list the names firstly for each of the Case Type nodes. To format a Case Type node, type a full stop immediately in front of its title, so that Penrose, as a Case Type under Schools, would appear as .Schools.Penrose
- After you have listed the Case Type nodes, add the Case Nodes to each line in the first column. Specific titles for the Case Nodes are preceded by a colon, so that Julie at Penrose School will appear as .Schools.Penrose:Julie

- Enter the attribute names, and the specific values for each case type and each case, as for any attribute table.
- Save your table in the usual way (i.e. first in its native format, then as tab separated text).
- In the Project Pad, from the Nodes menu, choose Import Node Attributes...
- At the top of the Node attribute import dialogue, where it says File to import, click on Choose. Navigate through your directories to find and select the text version of the table.
- Retain the default options in the Node attribute import dialogue. The first of these means that the nodes will be created if they do not exist.

Check the Cases area in your Node Explorer, to see all the new case nodes. Check in the Attribute Explorer that attribute values have been assigned as you expected. Now as you code documents to your new Case Nodes, as outlined above, the appropriate values of attributes will be applied.



If you create a case type node, e.g. “Penrose”—and give it values of an attribute, e.g. social status, all cases you create under it will inherit that value.

Faster searches

Well no, not really! But having attributes, sets and where appropriate, cases does make it possible to ask fresh questions of the data, sometimes in an exploratory mode, sometimes more interpretively. So their value extends beyond simply being tools for managing data, to being tools to assist in interrogating the data.

Each provides for a different kind of use when it comes to asking questions of the data: *Sets* are used for scoping searches and reports and are therefore ideal for asking questions about a subset of the data; *Attributes* can be included as items in searches, and are therefore ideal for asking comparative questions about the data; *Cases* bring together multiple documents or part documents and data from multiple sources, so that by undertaking within-case or cross-case analyses you can ask questions about processes within a “bounded” entity, and compare that entity with others of the same type.

Using attributes to make comparisons

Attribute values invite comparisons, for example: “Was there a gendered response to this question?” “What do my different sources of data say about this issue?” “Does a person's level of [scaled variable] make a difference to their experience of that situation?” Group membership (defined by attribute values) in these situations is categorical information, but what members of each group might have said on an issue is qualitative information. NVivo can take the qualitative text on a particular subject, and “split” it into new nodes based on the categories of the grouping variable, thus allowing comparison of the text across groups.

Because attribute data applies to whole documents, the text at any node in a document must intersect with the attribute value(s) that apply to that document. Searches involving attributes therefore typically use the

Boolean Matrix Intersection operator. You can elect to take one node and intersect it with the values of an attribute, making a table of results with multiple cells across a single row, or you could examine a whole set of nodes (e.g. all the subcategories, or children, in one tree) in relation to all the values of an attribute, making for a full cross tabulation with many rows and columns. While the latter is useful to provide an overview of patterns (e.g. by examining the number of documents coded in each cell), it can be more confusing when you wish to examine the text in the cells—simply because of the number of cells and amount of data available. In such situations, you are often better off to take one issue, or one situation, and examine that in relation to the values of the attribute of interest before moving on to look in detail at another.

A matrix with attributes



- From the Project Pad, choose to Search Project Database.
Run this search operation...
- In the Search Tool, click on the search icon next to Boolean Search.
- For your Operator: choose Matrix Intersection. The dialogue will expand to provide options for two sets of search items.
- Under Find text referenced by these items... click on Choose Nodes. In the Choose Nodes dialogue, find, highlight and add the nodes you want to examine across groups.
- Under ...pairwise with each of these items, click on Choose Attribute Values.
- In the Choose Attribute Values dialogue, double-click on Document Attributes, then on the attribute you are interested in using to split your data. The values it can take will be expanded below it. Select the values you wish to examine and move them to the right pane.



A node's children or subtree can be added. A range of an attribute's values can be added using "greater than or less than" (for text as well as numbers).

- Click OK to return to the Search Tool.
in this scope.
- Set the scope for your search to the document set you wish to search (probably, All nonmemo documents. Run Search!
The search results are stored in children of the node: (x x) /Search results/Matrix Intersection.
- Click OK. You will be taken to a display of the results of the search, set out in table format to show the cross-tabulation of items selected. Items from the first pool (the node/s) define the rows of the cross-tabulation, and items from the second pool (values of an attribute) define the columns.

- From the pull-down list in the slot next to Display, select Number of documents coded (or an alternative option). Numbers will appear in the cells in your table, and they will become shaded to indicate the density of coding.
- Click on a cell of the table and choose Browse/Code Node to open the Node Browser for that cell. Compare what is there with the text at another cell in the same row.

Examining the interaction of attributes

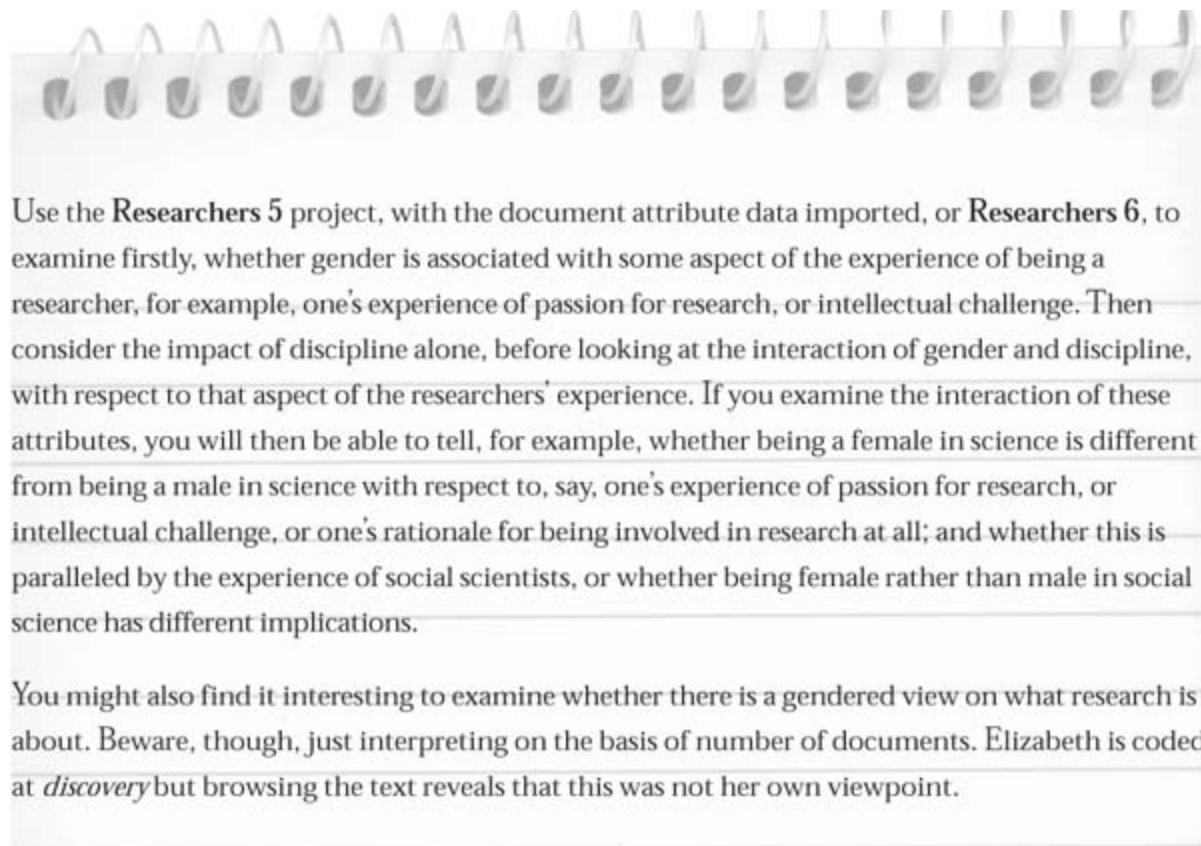
Often attributes are interrelated. It may be difficult to tell, for example, whether apparent differences in an academic's experience of research is more to do with their gender or their discipline, or whether it is the combination of these interrelated variables that matters, rather than one *or* the other. Use of a Matrix Intersection search scoped to the node of interest can provide an answer to this much more complex question.

A matrix of attributes



- Choose to intersect the values of one attribute with the values of another (Boolean Search: Matrix Intersection).
- Choose Custom Scope then Remove All documents and Edit List for Nodes.
- In the Node Set Editor, select the node of interest, Copy ≥ to the empty Working Set. Click on the Search icon to return to the Search Tool.
- Run the search.

Consider the text in each cell of the resulting matrix to determine whether reports of experience (or whatever the text coded at the scope node was about) vary for different combinations of the values of these attributes. Note that it is possible (though less likely) that an interactive effect could be revealed even where there is no apparent effect from either attribute on its own.



Taking Stock

A much bigger project, very fast! But if the tools have been well used, it is a less unruly project than earlier versions when there was virtually no data!

Critical processes in qualitative research defy speed; they require reflection and review. While some data processing tasks are not intrinsically qualitative, having them done may enhance enquiry processes and, if the shaping tools are familiar, it is worth exploring which processes can be speeded up. Coding by text search gives you automatic categorizing that is no replacement for reading and thinking about your data (none of NVivo's tools offer that!) but certainly a swift means of getting access to the data and providing for coding-on later, when meanings are clearer, and you have a gap in the pressure and excitement of data buildup. Batching of document import, section coding and table import of attributes automate clerical processes so you can get on with the interpretive ones.

This Part has shown how rapidly these processes can give access to your data, in ways that can be immediately productive and immediately show it to you anew, so you can review, and review all or some data, revisit and revise the theories that you are tentatively sketching. Now, back to interpretation.

<http://dx.doi.org/10.4135/9780857020079.n8>