

NVivo 7 Tutorial 8:

Asking questions

This tutorial covers a range of techniques for finding and querying your project items and content – and using or saving the results. You may wish to work through it in two parts – first Find, then Query. But they are presented together here to help you see their differences and the ways they complement each other. Any **Find** or **Query** in NVivo 7 is a way of shaping or coding data. The result of a **Find** or **Query** can be saved as a set or a node. But they are ways of asking different questions.

Chapter 8 of *Handling Qualitative Data* is about the ways of searching text and coding, how different these are from manual methods and cautions in interpreting them.

To review what you need to know about **Find** and **Query**, go to the online Help.

Find and Query – what's the difference?

The **Find** tool is for simple or advanced questions about *your project items*. Use it for locating lost items, asking what you have in your project, checking sample characteristics.

The **Query** tool is for simple or advanced questions about the *content of those items*.

Three differences matter.

1. Find and Query answer different sorts of questions:

Find will locate any project *item* in specified folders. **Advanced Find** looks for project items with specified features. **Grouped Find** returns items grouped as you request, by project item.

Queries look for source *content* with specified features, such as words in the text, coding or attributes of a case. And you can build up questions using a range of logical and location operators.

2. Find and Query offer different sorts of answers:

Both **Find** and **Query** *list* what is retrieved, and give the option to *save* those items as a set or a node.

Find gives you only a *list of project items* as answer. (For **Grouped Find**, it will be a sorted list.)

Queries find the actual *content* requested. Results can be previewed and then optionally saved at a node. Then content can be coded, or items saved, for further querying.

3. Queries can be saved as project items

You can optionally **add** a Query to your project: it is saved in the Queries folder. These queries can be used as a record of your work. They can be altered and rerun through other parts of the project data or later data stages.

1. Finding items

By now you have probably noticed that (like MS Outlook) NVivo has a **Find** bar at the top of your **List View** window. If so, you have probably used it – this is a very simple way of finding any project item or items. For example, you have a document and a case node for Mary – or was it Maria...? – and did you make a memo...?

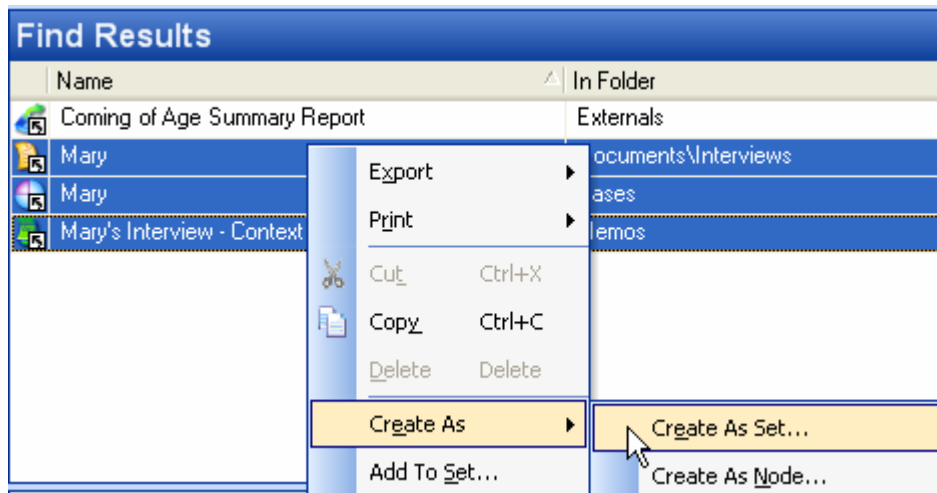


To find a project item

1. Enter the item name (or any part of the name) in the **Look for** field. You can include the * and wildcards (see online Help for information on these).
2. From the **Search In** drop-down list, select the folder(s) you want to search.
3. Click **Find Now**. The items that fitted your request appear in the **List View**.

Using the results of a Find

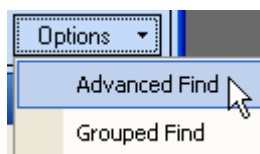
1. As in any **List View**, you can select any item to see content in **Detail View**.
2. If you wish to keep these finds, one way is to make a set of them. Select some or all items. From the **Context Menu**, select **Create As>Create as Set**.



3. In the **New Set** dialog, name the set. You now have made a Set that contains e.g. everything with Mary's name in it. (Note the Summary report was excluded from the set!) Now you will be able to focus on just those items in future analysis.)

Using Advanced Find

1. In the **Find** tool click on **Options** at the far right side. ("Options" are the best kept secret of Find toolbars!)



2. Select **Advanced Find**.

3. On the **Intermediate** tab, select the items you wish to **Look for** (this will set the characteristics you can ask about). For example, if you wish to find all cases where age group is over 60, **Look for Cases**, and click **Cases where** and select the attribute and value.
4. Add any other specification that will get what you want. For example, I don't want Case nodes if there is no content coded yet for that case. So here I have also specified "**That code Any Item**".)

The screenshot shows the 'Advanced Find' window with the following settings:

- Look for: Cases
- Tab: Intermediate
- Created: Anytime
- Modified: Anytime
- In: Any Set
- With See Also Link: From
- With a: Relationship with
- That code: Any Item
- That are: Ancestors of
- Cases where: Age Group, equals value, 30-39

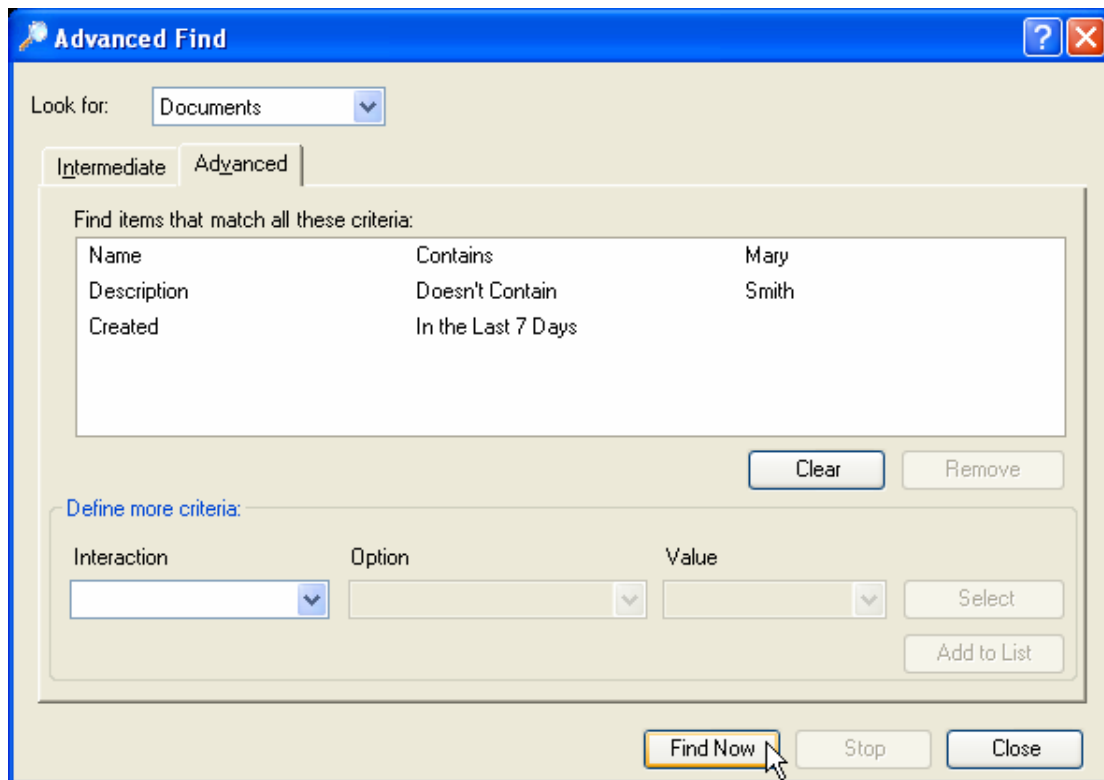
Buttons at the bottom: Find Now, Stop, Close.



Use **Advanced** options to save work later. **Advanced Find** is also a way of asking questions of your sample (for example, let's check if we have anyone aged 60+). Or use it for project management: (which documents are not yet coded at any node?)

And now, really Advanced!

The second tab on the **Advanced Find** box is – yes – **Advanced**! Don't be put off: this window allows you to build up plain language requirements for just what you want. It can be as simple as a "this but not that" request: were there any documents whose name contained Mary other than those about Mary Smith?



1. Select the items you wish to **Look for**.
2. In the bottom panel, **Define more criteria** by choosing interactions of those items with others. For each, click **Add to List**. Note the range of criteria you can specify.
3. Click **Find Now**. The items that fitted your requests – only items that fit *all* your requests – will appear in the **List View**. Read or save them as a set or a node as before.



If you have used NVivo or N6 previously, you will recognize this as a (much easier) way of doing one form of a task previously done through the Search Tool: Boolean intersection. You have found *all and only* the with characteristics you specify. But if you want to know about the *content* of those documents, you need another tool.

And now for Query!

The next step is to explore Query.

2. Text Search Query

Any query in NVivo has a format that will become familiar.

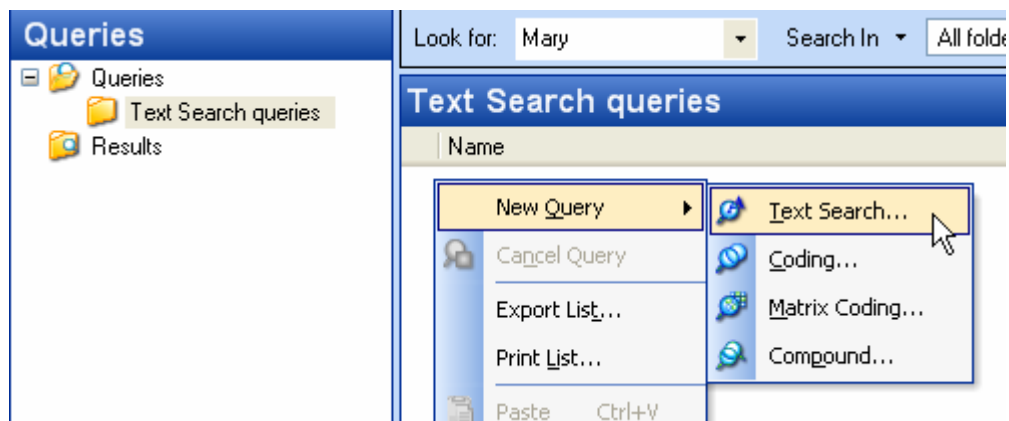
In three steps, you work through three tabs, (see the three areas above) to:

1. specify the **criteria** for your search
 2. select from the **options** for results
 3. (optionally) **Add** the query to your project for reuse.
- And then, if you wish, you can run the query.

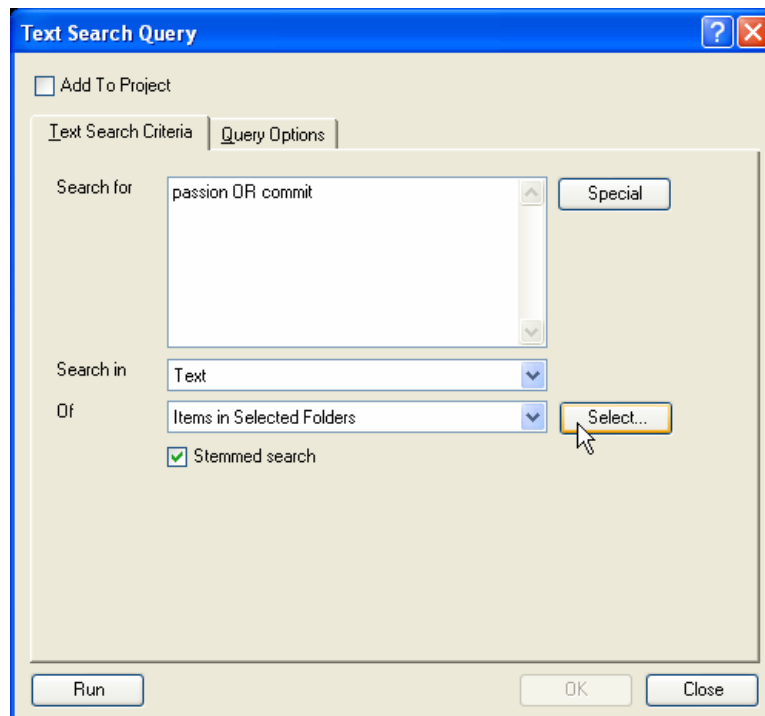
Here, we start with a Text Search Query.

To set Text Search Query Criteria

1. In the Navigation window, click on **Queries**. That folder contains folders for **Queries** and **Results**. You can make your own sub-folders for different sorts of Queries (but not for Results). Create a sub-folder for **Text search queries**.
2. Click on the folder you want to place a query in, and click in the **List View**. Using any of the ways to make a new item (**Project** or **Context** menu or **New** button) create a new **Text Search** query.



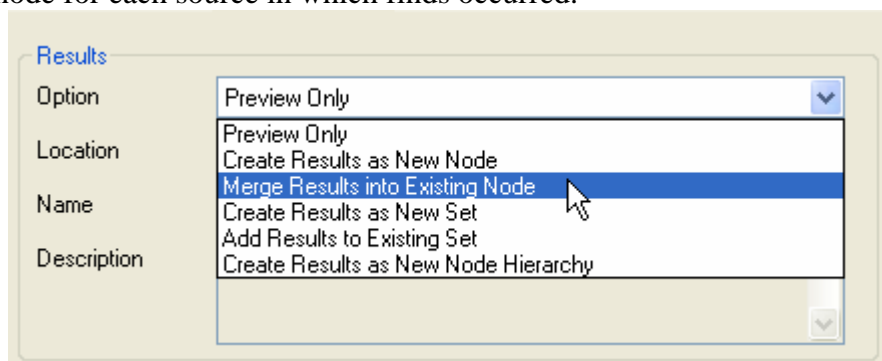
3. The **Text Search Query** window opens. Type in what you want to **Search For**. (Here I am searching for the words “passion” or “commit” and any words with the same stem e.g. “passionate” and “committed”).



4. Set the *scope* for this search, i.e. what you will **Search in**. Select text or annotations or both, and then **Of** which folders or items. (See section 6. **Scoping a Query** below.)

To set how the query results will appear and be used

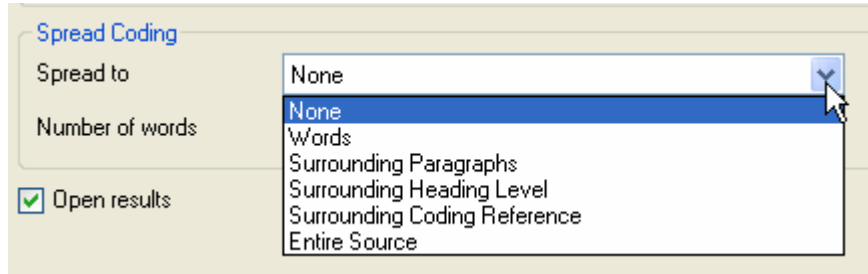
1. Click on the **Query Options** tab, to specify how you get the results.
2. First, ask yourself, what do you want to do with the Results? The options are to:
 - preview results (they'll appear in a list of items with finds);
 - save the finds by coding them, *either* at a new node *or* an existing one;
 - add the items with finds *either* to a new set *or* to an existing set;
 - create a new node hierarchy of nodes coding the results – that is a tree, with a node for each source in which finds occurred.



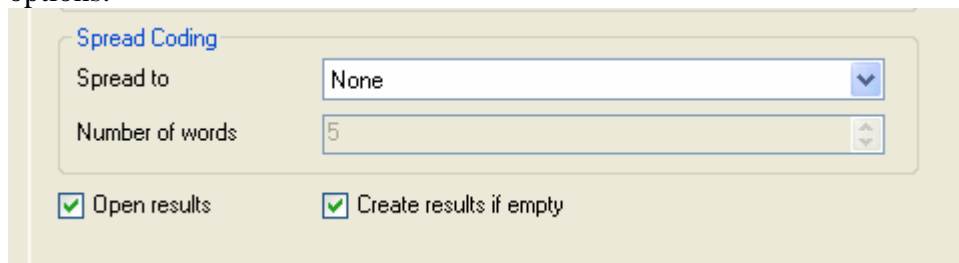
This step deserves your careful attention. As you learn the Query process, experiment with the uses of the different options. The ability to preview only gives you a fast way of finding whether there are any finds, or which sources they are coming from. Remember you can always save the listed items as a set.

3. Now, do you want to **Spread Coding** of your finds, to see more content around each find? And if so, what context do you want included?

Note, you can spread to see content later – so long as you have saved the results as a node – and this may be preferable since it will show finds and context in different color font. (See below on using the results.)



4. And finally, don't overlook the two tick boxes below the Spread Coding options.



Do you want to open the results in the Detail View, to work with the content immediately? And do you want a results node created even if no finds were made?

To save the query

Before you run the query, ask do you wish to save it? This will allow you to rerun it through later data or alter it to be more relevant.

1. Click the top left corner box **Add to Project**.
2. The **General** tab appears so you can name and give a description to your query. (Use the description to record what it was designed to do, in case you forget.)



Whilst you are learning the tool, it's a neat idea to use the option to Add to Project so if you find you didn't ask for quite what you wanted you can very quickly go back and change the specification.

Note that from the list view of results you can also return to last run query (that's an option on the context menu.) But if you add queries to the project often, regular housekeeping is useful!

To run the query

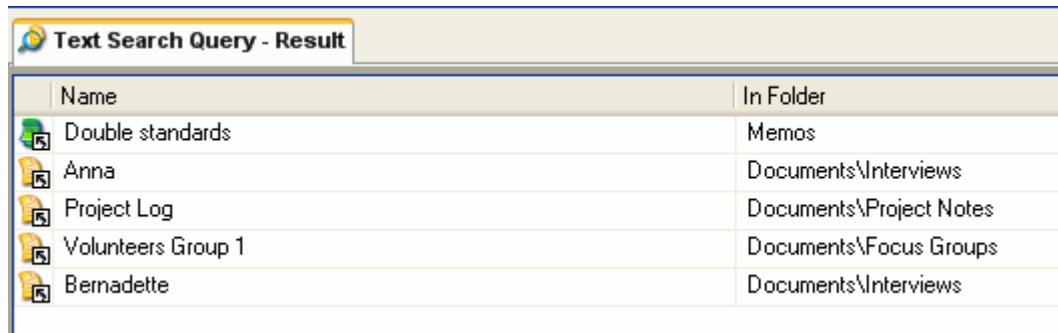
1. Click **Run!**
2. The results will appear as you specified. Results show in the Detail View Always go to the results to check that the query did what you meant it to do.

3. Using the results of your query

Using Preview

When you run any query, the first question may be: did you get any results and if so, what? Preview answers this question, and allows you to explore what you did get.

1. The **List View** lists the items which contained finds.
2. Click on any item to see it in the **Detail View** with the passage(s) found highlighted.



Name	In Folder
Double standards	Memos
Anna	Documents\Interviews
Project Log	Documents\Project Notes
Volunteers Group 1	Documents\Focus Groups
Bernadette	Documents\Interviews

Saving the results as coding at nodes

The two options to save as a new node or merge with an existing node both *code* the results. All content that satisfies your specifications will be coded at a node you specify. As with any other node, its content can be read in **Detail View**, the context of coded excerpts can be expanded, you can jump to the source or *code on* to other nodes.

(For example you might discover that there are two meanings of “commitment” in these accounts of volunteering – and all these finds refer to how committed people’s time is in other activities! Code on to a node for “too little time”?)

Using text search for automating coding

The ability to code the results of your search has very considerable implications for your data management. Consider using text search for accessing material to code if it is reliably indicated by the presence of a word or words you can accurately identify with this text search query tool.

1. Create a Text Search query as above. Specify the word or words as accurately as possible, using wildcards, stemmed search to ensure that you minimize false finds.
2. In **Options**, specify to **Create Results as New Node** (or if appropriate, add to an existing node). Specify the **Spread** of finds that will be useful for your purposes.
3. Click **Run**. The node is created and opened in Detail View.



A node in a Results area cannot be altered. If you wish to alter a Results node, and keep working with it, move it into the Free or Trees area. But you can **code on** from a Results node to another Free or Tree node - this doesn't alter the results node. If you want to work with query results, this is often the best method.

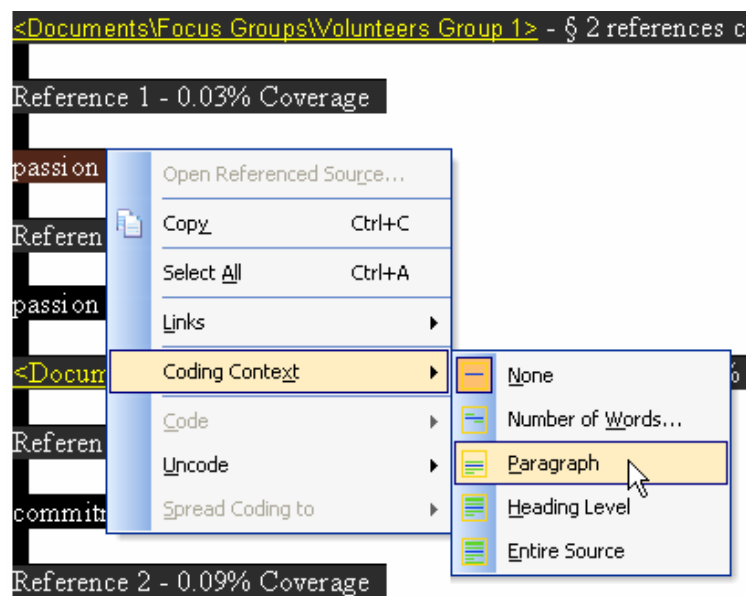
To see and code context of your finds

Text searches always require context, but often you don't want the context to hide where the actual "hit" was. To examine the "hits" in context, and separately code the appropriate surrounding text:

1. Select at **Options** to **Preview Only** results but don't **Spread** finds.
2. From the List View of **Results Preview**, visit each find in the source to see content. Select the appropriate wider content, and code it at the Free or Tree node you are creating for the outcomes of your search.

OR

1. Select at **Options** to save results as a new Results node and don't **Spread** the finds;
2. Open the Results node, **Select All** the content (Ctrl+A), and from the right mouse Context menu or the toolbar icon, choose to **Coding Context** and the amount of context you want to see.



The required context will appear in the Detail View, in pale text, so the original find (which is all that is currently coded) is easily identified.

They could have a personal motivation. Often volunteers are connected in some way to their cause – through a family connection. Or the skills they donate are a personal **passion** – such as cooking, or craft.

3. Read each passage, select just the context you want to code – and **code on** to a new Free or Tree node that will contain only the finds and context you consider relevant for your purposes.

Tree Nodes		
Name	Sources	References
elderly, retired	6	8
energetic	2	3
financially secure	3	6
free of commitments	2	3
have spare time	5	7
helping the poor	2	4
lonely	4	4
outgoing	2	2
passionate	4	8
politically involved	1	1

passion or commit search

Reference 1 - 0.18% Coverage

Volunteering is where you do it. Volunteers identify with the organizations and communities that they serve. These associations were especially strong for those whose volunteer work involved a significant commitment of time. Volunteers expressed pride in

Help with text search

Are you a user who refuses to use online **Help**?

This is one area of NVivo where you must seek Help, as the NVivo search engine, like web-based search engines, allows you to do far more subtle searching than merely look for a word or words. Go to **Using Special Characters and Operators** for many ways of making your searches powerful.



Using the Software

Text Search: Special Characters and Operators

When conducting a text search you can:

- Use [wildcards](#) in place of characters
- Combine multiple words or phrases using [boolean operators](#)
- Find words with similar or identical spelling using a [fuzzy search](#)
- Specify proximity using [near](#) search characters
- Specify [relevance weightings for words and phrases](#)

Some [metacharacters and stop words](#) and are reserved for special use in searches and you cannot search for them explicitly.

Related Topics

[Using the Software](#)
[Working With Your Data](#)

You can short-cut coding considerably by building up a node coding all likely words, with context, and then working from it, deleting false finds and coding on to more finely defined nodes.



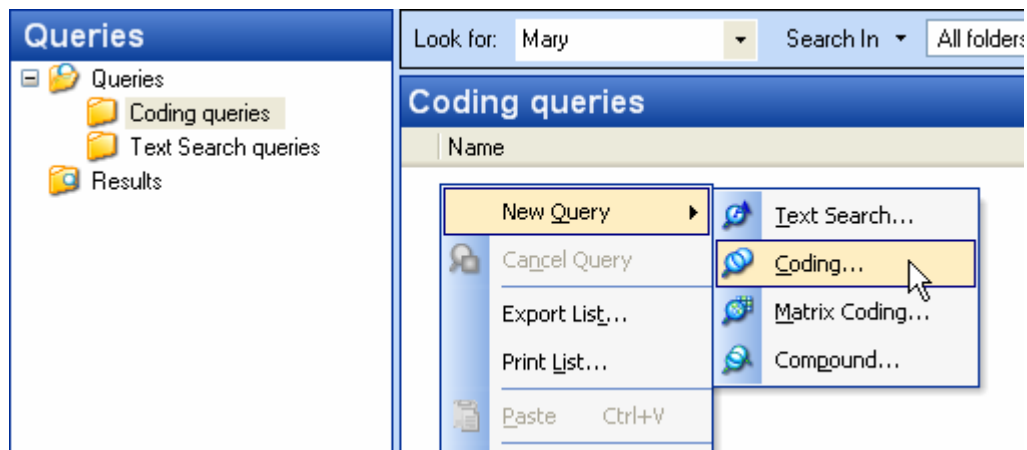
Always **be careful** to use and interpret this rapid coding appropriately: go to *Handling Qualitative Data* pp 155-8 for advice and warnings on using mechanical text search processes for coding.

4. Coding Query and Compound Query

Now to try asking questions about your coding. You will find the steps very like those for text search.

There are three further query types available. (As above, it's recommended to make folders for the different types.) Here we start simple.

Matrix coding queries are the topic for the next tutorial. In your own time, play with Compound queries – they combine Text Search and Coding queries.

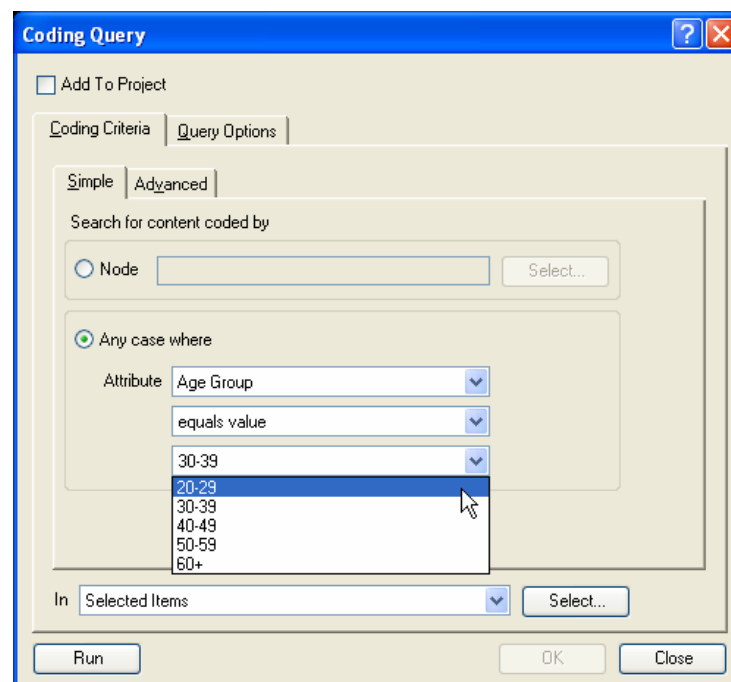


To make a simple coding query

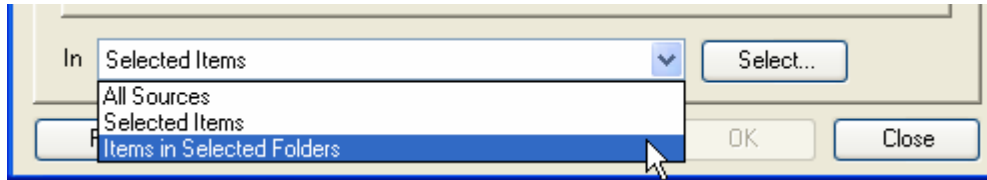
The **Simple** tab on the **Coding Query** window allows you to ask for everything coded at a node or everything coded at a case with a particular attribute value.

Why do this through a Query? Because just as for Text Search, you can *scope* the query to just the items you want to ask about.

1. Select **New Query>Coding**. The Coding Query dialog opens. Note that its tabs are very similar to those for Text Search queries.



2. Now click **Coding Criteria**, and specify the simple search. You can search for either **content coded by** a node or (as below) **Any case where** a specified attribute has a specified value.
3. **In** what data? Set the scope, just as for Text Search, by choosing to search **All sources** or just the items (which can be sets) or folders you want to focus on.

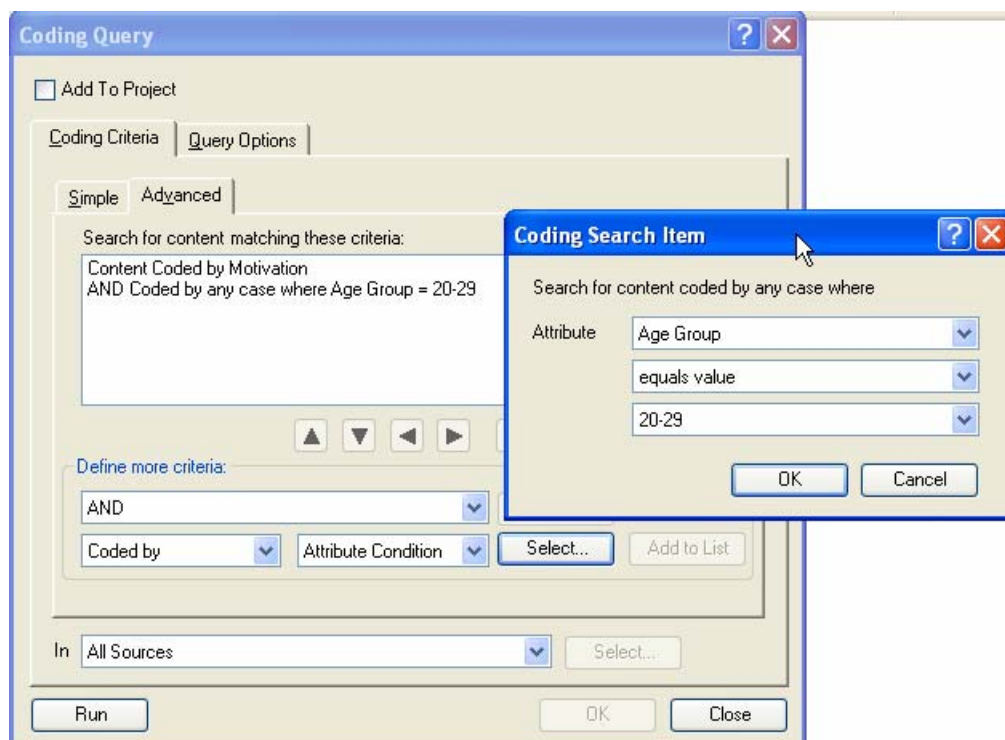


4. Just as for Text Search, click **Query Options** to specify whether you get a **Preview Only** or save as coding at nodes or save items in sets
5. Now decide whether you wish to **Spread** the finds to a wider context. (Revisit **Using the results of your search** for revision on **Query Options**.)
6. In the **Coding Query** window, just as for Text Search, you can specify if you want to keep this Query. To do so, click **Add to Project**. The **General** tab opens: name the Query.
7. Click **Run**. And check what happened.

To make an advanced coding query

Now to the more subtle ways of questioning your data.

1. Make a new Query as before, click **Add to Project** and name the query.
2. Click **Coding Criteria**, and click the **Advanced** tab.
3. As with **Advanced Find**, build up requirements for what content is coded by. Specify the requirement, select the project item, and click **Add to List**.



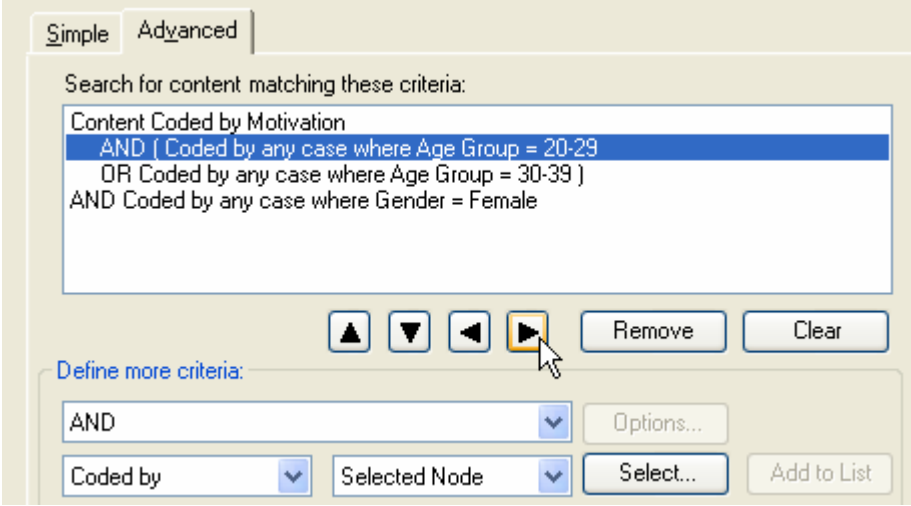
Note that in coding query you can combine requirements about coding at nodes AND requirements about the values of attributes of cases – for both, you specify that content must have the required coding. Think through the question you are asking, such as: *I want everything about motivation if speaker is in their 20s – that is, if the content is coded at motivation and at any case node where Age group=20-29.*

4. When you have added the required specifications for your query, check that the *scope* is as you wish. The default is **In All Sources**. Narrow the search by selecting items, or sets or folders.
5. Click the **Query Options** tab to specify, as before, how the results will be delivered – preview only or coding at a node or nodes or saving in a set.
6. Click **Run**.

To build an advanced coding query

Return to that Coding Query and build more requirements into it.

Note that as you specify conditions and **Add to List**, you can move conditions up or down, and use the sideways arrows to bracket conditions that you wish to operate together.



The screenshot shows the NVivo Query Tool interface. At the top, there are two tabs: 'Simple' and 'Advanced'. Below the tabs, the text reads 'Search for content matching these criteria:'. A list of criteria is displayed in a text area: 'Content Coded by Motivation', 'AND (Coded by any case where Age Group = 20-29', 'OR Coded by any case where Age Group = 30-39)', and 'AND Coded by any case where Gender = Female'. Below the list are four arrow buttons (up, down, left, right) and two buttons labeled 'Remove' and 'Clear'. Below these is a section titled 'Define more criteria:'. It contains a dropdown menu with 'AND' selected, an 'Options...' button, a 'Coded by' dropdown menu, a 'Selected Node' dropdown menu, a 'Select...' button, and an 'Add to List' button.

As your project progresses, you will return to the Query Tool for very many different purposes. This is where you find occurrences of words, check who is using them, look for patterns in your coding, test hunches and seek new ideas.

Interpreting your coding query

NVivo can conduct searches of your coding that would not be possible by manual methods. These are very powerful, but must be properly interpreted. Note that you can't claim from the search above that you now know everything that motivates women in under 40. The node codes only what *you* saw and *coded* in those documents. You are the weak link in any such search!



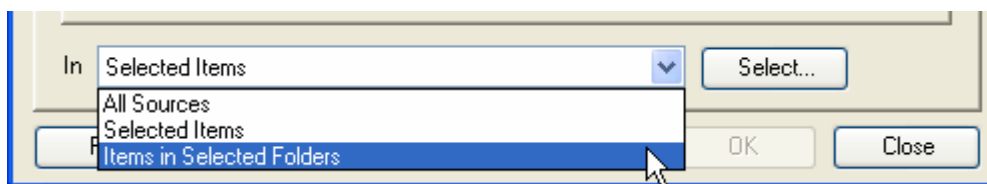
I know this is sounding repetitious, but please, **be careful!** It is essential to interpret these mechanical searches and use their results with great care. Check the warnings in *Handling Qualitative Data* Chapter 8.

5. Scoping a query

An important feature of the Query Tool is that you can always ask where you would like this search to “go” – what data records do you want to focus on? Using the ability to *scope* is critical for qualitative research, so it’s worth taking time to learn the tool. Query without scope is a blunt instrument.

You can set the scope for any query to any item(s) or any folder – or to any existing set of documents and/or nodes. In two searches, scoped differently, you might compare two parts of your project. (If you have different folders for interviews and focus groups, run a search through documents in each. Now you can see if the issues were discussed differently in the focus groups.)

For text search query you can also scope the search to text and or annotations. (This option is not in coding queries, since annotations can’t be coded.)



To narrow the scope of a query

1. Return to the Query Tool, and **Text Search Query**. Open the query you saved earlier.
2. Check the **Query Options** tab: if you left the defaults, you asked to **Search In Text Of All Sources**.
3. Select to **Search In Text and Annotations**. You have broadened the scope of your query to look for the words specified in annotations as well as main text.
4. Now to get Text and Annotations “**Of**” just some particular data. In the dropdown list, select **Selected Items**.
5. In the **Select Project Items** box, select a set (e.g. your set of sources and nodes containing the name Mary).
6. Run the query again and check the results. You have found all occurrences of that text in any project items whose name contained “Mary”.

Using Sets in Query work

The scoping of query processes is a very important role for Sets. Particularly if you are doing a series of searches for text or queries about your coding across part of your data, you will be assisted by making a set of the items to be searched. The set can be selected immediately from the Query Tool.

Note, sets can be the **input or output** of a query.

- You can scope your text search or coding query to a set;
- You can save any or all of the items in any list view to a set.
- Results of any Find or Query can be saved as a set. This means you can scope a new search to what you found in the previous search.

Remember, Sets can contain sources and/or nodes (which code parts of sources.) Simply create a set to identify the records you want to search.

6. Using Query for your project

No tutorial can lead you to the searches you need for *your own* project. Now that you know the basic tools for searching text or coding, you need to set out in plain language the searches you would like to do at this stage in your project.

For advice on framing and conducting searches in plain language, and logging what they find, see Chapter 8 of *Handling Qualitative Data*.

As a final exercise for this tutorial:

1. Make a table whose first column is of plain language questions to ask using Find or Query.
2. In a second column, record the Find or Query to be used. Use the online Help to locate the operators that will get at the answer to your questions.
3. In a third column, record the scope to be set for the search.
4. Run these searches, and record in a fourth column what you learned.



Of course you must back up your project, but this time, before you do so, tidy it up. You probably have many search results nodes from playing with the Query Tool.

Remember that Results nodes cannot be altered, so the only reason for keeping them is to return to examine the result of that query, or compare with a subsequent query. If you don't expect to need them, write in a memo about what you learned and delete these records of searches you don't need to keep. If any results nodes are to be kept and developed, by adding more coding, they must be moved into the Trees or Free areas. Move them (using cut and paste, as in Tutorial 6,) and rename them sensibly. Add a description to record how each was made - it's all part of your project's log trail.

This concludes Tutorial 8. For considerations of using these specialist tools, and interpreting them, go to Chapter 8 of *Handling Qualitative Data*.

You now have data, linked to other material and to ideas via coding, and know how to make catalogs of nodes and models to show logical and theoretical relations, and how to search the records and their coding.

The next task is to use the ability to search your documents and your coding for different purposes, to show and interrogate patterns. So the next tutorial is about matrices.