

## NVivo 7 Tutorial 6:

### Relationships and Other Nodes: Handling Ideas

The previous tutorial was about discovering or creating new ideas and the nodes that store them in NVivo. This tutorial is about managing the relations between ideas.

NVivo provides four folders for nodes, representing four different sorts of categories at which you may want to code data. Three have been encountered in earlier tutorials – Free Nodes, Tree Nodes and Cases.

The first section of this tutorial is on the fourth, Relationships. It shows how you can store relationships you see in your data, and code the evidence for those relationships.

The next sections are about managing categories. Ideas isolated are the researcher's enemy. And ideas that are not organized and managed are often very problematic. You may lose them, or worse, you will lose them sometimes. The next sections are about how to catalog ideas, by reviewing, merging and rearranging nodes and their coding in trees. Tree organization also facilitates many query operations.

As you work in your project, you will find that coding data or exploring relations of ideas always requires finding nodes, and seeing their associations. In order to bring data and ideas together, and create new understandings, data management is necessary. As in most areas of qualitative method, efficiency is necessary for creativity.

Revisit Tutorial 5 for ways of creating, naming and describing nodes and coding at them.



If you are setting up your own "real" project, you will be greatly assisted by early attention to the management of your ideas. Software enables far more categories to be created, and far more coding to be done at them, than manual methods. This is a great advantage, unless the categories become a problem to the researcher. From the start of your project, work *with* your node system, visiting and reviewing, revising and improving it constantly. This is a major way of "talking to" your project.

The goal is to have the node system as parsimonious and efficient as possible, without losing any insights or hunches that should be contained in it. (Rather like a shrink-wrapped package, a useful node system is as small as it can be whilst being as big as it needs to be!)

Go to Chapter 6 of *Handling Qualitative Data* for advice on cataloging the categories that are part of the research design or emerge from the data.

To review what you need to know about creating nodes and managing them in NVivo, go to [Help](#).

# 1. Relationships

**Relationships** in NVivo 7 define relations you see between sources (documents, externals or memos), nodes (free, tree, case and query results) or sets. If two of these items seem to you to be related, you can record this in your project, creating a Relationship that links them.

And that Relationship is itself a node. Like any other node, it can have coding. If you wish, you can code at the Relationship the data that you see as pertinent to it.

Relationships can be added between any sources and any nodes (except relationships) or any sets of sources and/or nodes. Like any other node, they can have any amount of coding, and the coded content can be browsed and recoded (see previous Tutorial).

In the **Nodes** folder of your project is a separate folder for **Relationships**. As you make relationships they will be shown in the **List View**.

A relationship has a “from” and a “to” item, linked by an arrow whose “type” you define.

In the **Classifications** folder is a folder for **Relationship Types**. You name and describe *types* of relationships, and specify how they are shown - a nondirectional “association”, or a one-way arrow or symmetrical (double headed) arrow.

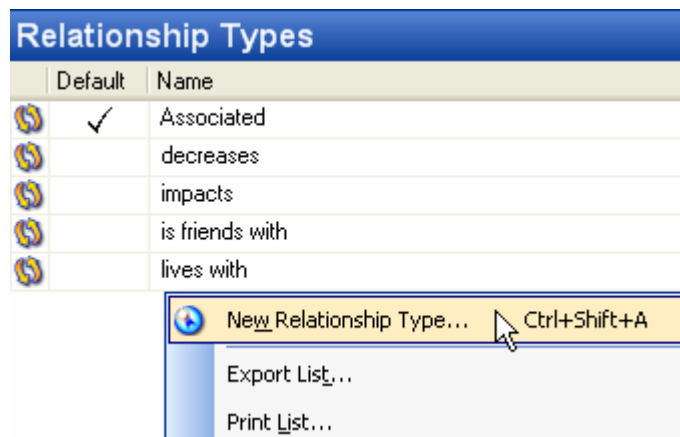
Family values seem to encourage (though not directly cause) motivations to volunteering? You need a relationship *type* “encourage”. And a *relationship* between two project items – e.g. the memo about family values and the node for motivation.



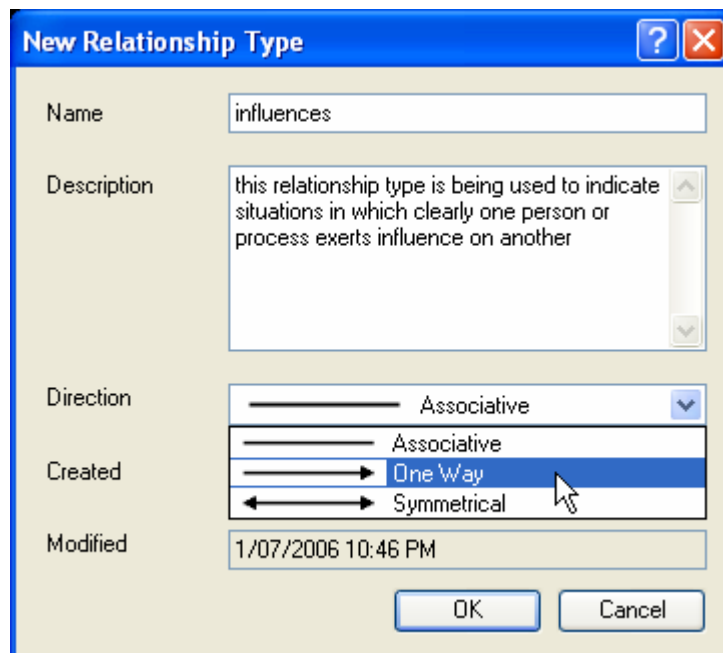
If you don't have nodes or sources to represent the relationship's “to” and “from” items, create them. NVivo has no problem with nodes or sources with no content. In the example above, “family values” could be created as a memo, named and described in the project. Later as you explore the relationship you may wish to write your changing ideas about family values in the memo, then code some material at the relationship.

## To make a Relationship Type

1. Go to the **Classifications** folder, and click on **Relationship Types**.
2. As for any other new item, click in the **List View** and then from the **Project** or **Context** menu or the **New** button select to create a New Relationship Type.

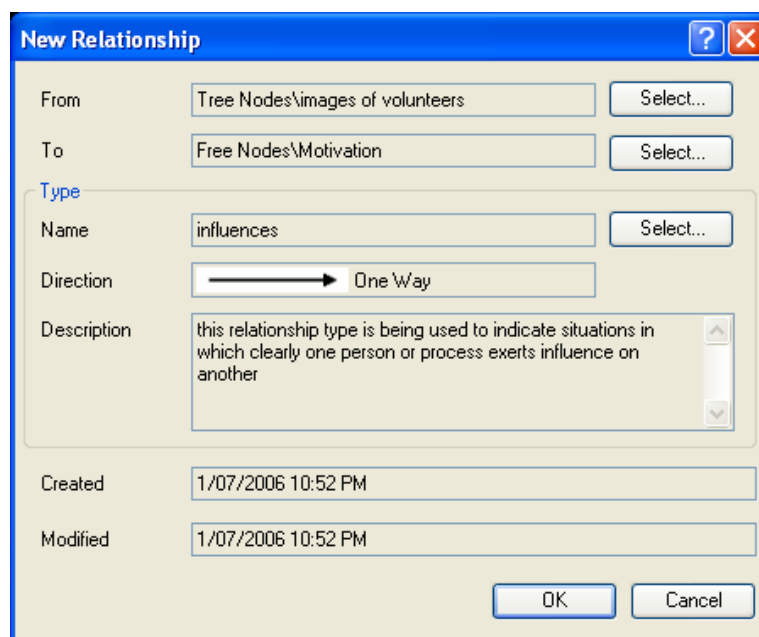


3. In the dialog, name the Relationship Type, and add a description. (This is usually very important as you may be creating many Relationships of this type and need to check consistency.)
4. Now select the direction of the Relationship



### To make a Relationship

1. Select the **Nodes** folder, and in it the **Relationship** folder.
2. As for any other new item, click in the **List View** and then from the **Project** or **Context** menu or the **New** button create your new item.
3. Select a “from” and a “to” item, and then the “type” for this relationship. Note that the description of your Relationship Type appears, so you can check that this is appropriate. Click **OK**.



- The new relationship appears in the **List View**. Note that as you record more relationships, you can sort them in **List View** according to each column – for example, get all the relationships of type “decreases” listed together.

Relationships					
From Name	From Folder	Type	To Name	To Folder	Direction
Anna	Cases	lives with	Sunil	Cases	←
Bernadette	Cases	is friends with	Ken	Cases	→
contexts	Tree Nodes	impacts	images of volunteer	Tree Nodes	→
contexts	Tree Nodes	impacts	assumptions	Tree Nodes	→
Ken	Cases	is friends with	Bernadette	Cases	→
time\lack of time	Tree Nodes	decreases	Motivation	Free Nodes	→
personal goals\fa	Tree Nodes	Associated	Motivation	Free Nodes	←

- Each Relationship is a node, and you can select it, just like any node, to code at it any source content that you see as pertinent to this relationship.

### Q.6 Rewards for volunteer work

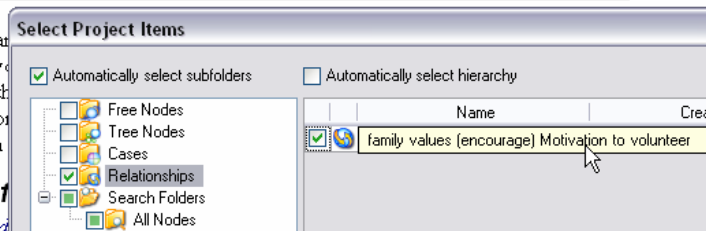
*What do you think motivates volunteers? (What do they get out of it?)*

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.

Recognition is important in community groups I work with. It's interesting news for other people. Perhaps the most important thing is making a difference, and it's making a difference.

### Q.6a Incentives for volunteer work

*What incentives should be provided?*



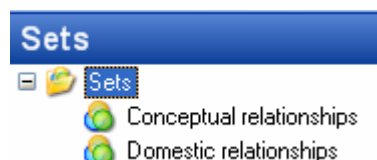
## Using Relationships in your project

Relationships that can code the relevant data are an entirely new tool with NVivo 7. They may be used for very specific statements of fact (Anna lives with Sunil) or very abstract reflections (lack of time decreases motivation to volunteer),

For your project, ask early what relationships you are anticipating, then as you work with the data, create and code at Relationships that express what you are discovering.



If you create two or more different sorts of relationships, note that while they cannot be in subfolders, they can, like any nodes, be managed in Sets. Here the researcher has created different sets because some relationships record conceptual relations, and others simply domestic relations like living together.



## 2. Reviewing your nodes

If you have been coding data, you may already have begun assessing the nodes you have made, and rather than placing them all in Free Nodes, sorting them into “Trees”.

As explained in Chapter 6 of *Handling Qualitative Data*, tree structures allow you to catalog nodes so they are easily found and consistently used.

NVivo does not require that you keep any nodes in Trees, but you will be greatly assisted by doing so. Cataloguing nodes does not require “top down” thinking. Rather, as you see that some of the free nodes created are all a “sort of” a more general category, you may create that category and move the nodes under that “parent”.

### To do a review of nodes

1. On a whiteboard or very large sheet of paper, or in a model in NVivo, review the nodes created so far. Do you see the categories needed for thinking about your project there? If not, sketch in the other categories you think you will need.
2. Now in your NVivo project, open **Nodes>Free Nodes**. In the **List View** review the free nodes created so far. Click on relevant columns to sort them – e.g. according to amount of coding, or date of modification.
  - Are they named accurately and described adequately? Go to a node’s **Properties** to change any name or description.
  - Are there more categories needed to ask the questions or explore the issues you are tackling? If so, create the nodes.
3. Do they belong together in any logical order? Don’t force it! In NVivo (in a memo or a model) or if you prefer, on paper or whiteboard, sketch the shape of a starter catalog that looks logically sensible. If a category doesn’t belong with others, leave it free.
  - Looking “top down”, from your project design; what are the main areas of enquiry? Revisit the categories you started the project with
  - Now consider the categories you created from the data. Do they hang together in logical groups? Ask each, what is it a “sort of”. If some nodes group logically, note this. (For detailed advice on cataloging nodes and evaluating the node systems you create, go to Chapter 6 of *Handling Qualitative Data*).



Consider carefully how the growing node system relates to your design or intentions for this project. A good rule of thumb is that if you want to ask questions about your data in NVivo, you will need nodes in terms of which the questions can be stated. If you have done a project design, or a literature review, try importing that into NVivo and coding it: the project needs the nodes that result.

### 3. Rearranging nodes

At this stage, you may make major changes to your project. You did back it up at the end of the last tutorial, didn't you?

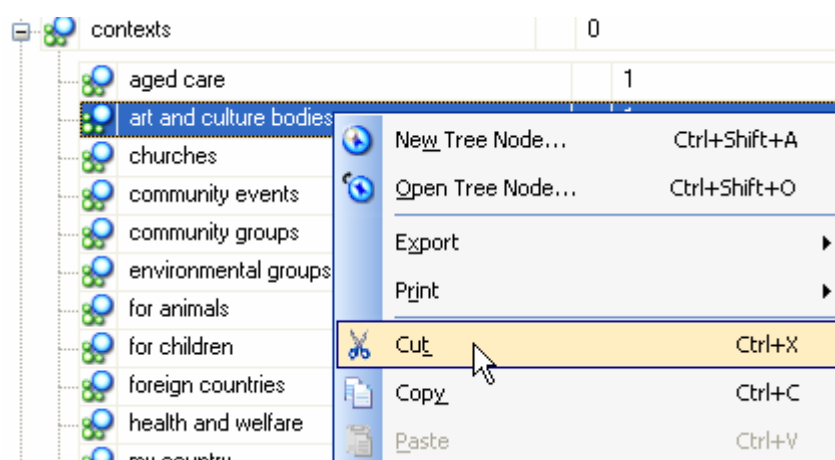
You can *cut* or *copy*, and *paste* or *merge*, any node in NVivo, without losing any of the data it codes, or its links to a memo or, if it's a case, its attributes. So if your nodes now should be catalogued, the steps are simple.

#### To cut and paste a node

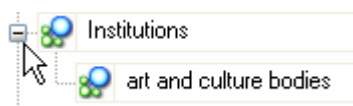
1. Back in your NVivo project, create nodes in the Trees area for the main categories you have sorted your ideas into. These new tree nodes will be “parents” for the subcategory “children” that belong there. (The child nodes are a “sort of” the parent.) Thus my node for “self worth” might go with other categories about perceived benefits under the parent node, “benefits of volunteering”). Create any parent nodes indicated by your review.
2. Go to the Free Nodes you created by coding in the previous tutorial. Select one that properly “fits” as a “sort of” one of your new parent nodes.
3. Right mouse click and **Cut** the node from the Free Nodes area



(or use the Cut button).



4. Go to the parent tree node you want to place the cut node under. Right mouse and select to **Paste** the node. Click the “+” at the parent node, and you will see your pasted node, which still holds all the coding you had done.



If you prefer caution, **Copy**, and then when you're sure it's pasted correctly, **Delete** the free node.)

## To shape your node system

1. Continue cutting and pasting nodes as indicated by your review, until you are confident the arrangement is logical and that it does not in any way force your thinking about the categories or pre-empt things you might later find. (Chapter 6 of *Handling Qualitative Data* advises on design of tree catalogs avoiding these traps.) Remember, if a node won't logically "fit" in a tree, leave it in Free Nodes.
2. Now code another document, working with the categories as they are now arranged, so you can find existing ones and place new ones. You will find that as you work with the nodes, now, it is very easy to cut or copy and paste or merge as the system of nodes starts making sense.
3. Visit the categories you have left "free" of the catalog and review them.

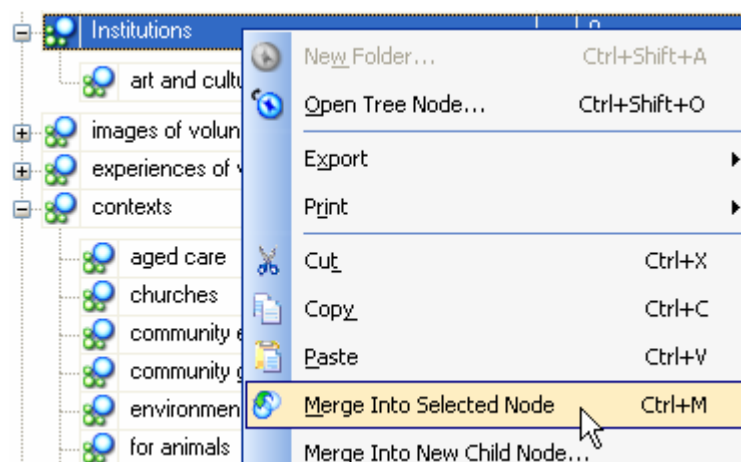


A good habit is to review nodes left "free" often - as the catalog forms, the remaining Free Nodes are special. They are the categories that don't fit anywhere. Maybe they are just irrelevant? But maybe they are the still unconnected ideas that will become crucial in making sense of the more organized thinking about your project.

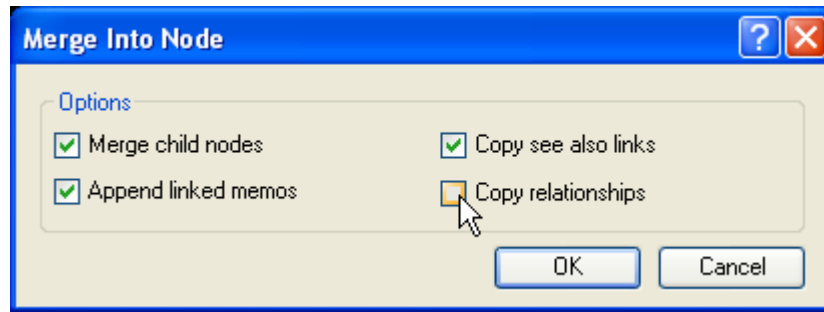
## To merge nodes

Often at this stage of a project you find that you have more than one node for the same idea or concept. This is quite common, not to be regarded as a problem. Indeed it's good practice to keep nodes separate if there's a chance that the meanings are subtly different, and bring them together only when you get surer of their common ground. At that stage, you need to be able to merge their coding - and any other contents or links of the node.

1. Select a node you want to merge into another. Cut or Copy it (depending on your purposes).
2. Now select the node you want to merge this coding into.
3. From the right mouse context menu, select to Merge into Selected Node.



4. select what you wish to merge, from options presented.



5. Click **OK**. The nodes are merged.

## Evaluating your node system

As the nodes are moved and merged, you will find a stronger sense of how they relate to each other. Treat the tree catalog like the catalog in the library, which assists you to look for an item rapidly, check if it's there, browse the items near it and note what's missing.

Like any catalog, it is only as useful as it is logical. You will find that revisiting your node system is richly rewarded with increased efficiency and a stronger picture of how the project is shaping.

Regularly, make time to check for and remove or merge duplicated nodes, examine and delete nodes that have proved unnecessary or redundant, and ensure that the logical structure of the node system remains strong and clear to you.



The catalog of nodes will change, as your understanding of what's going on in these data grows. So the node system at any time will be a record of where you are "at" in the project. A highly useful technique for logging progress is to save and archive a list of your nodes at regular intervals, or at significant milestones in the project.

For advice on constructing a tree structure that will work for your project, and for ways of checking it remains logical and useful, go to Chapter 6 of *Handling Qualitative Data*.

## 4. Listing and Reporting on Nodes

Now to get a report on the node catalog you have created. You can print the nodes that are displayed in **List View**. This is a what-you-see-is-what-you-get printout, so you will need to expand the hierarchies that you want to include. Alternatively, you can make a very detailed report on all or some of your nodes.

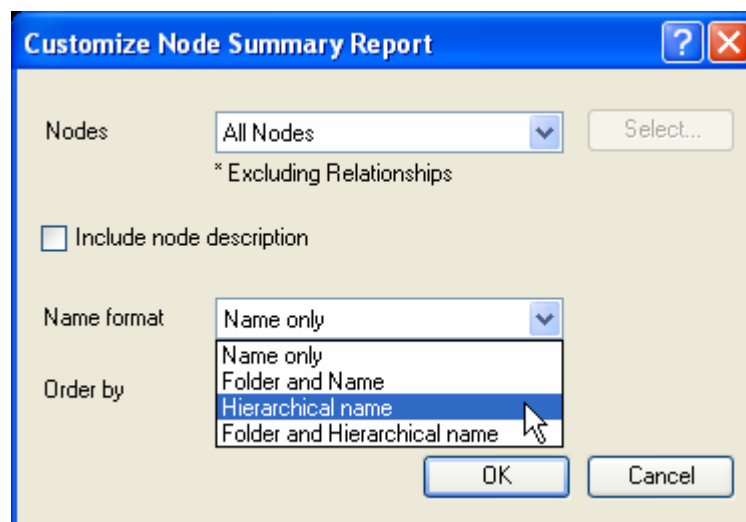
### To print the list view:

1. In **Navigation View**, click the **Nodes** button.
2. Select the folder containing the nodes you want to print. To print a list of all nodes, select the **All Nodes** folder under the **Search Nodes** folder.
3. On the **File** menu, click **Print List**.

### To make a report on your nodes

There is a range of reports available with detail and formal layout, from the **Tools** menu. The **Node Summary report** will give you everything you could want to know about each of your nodes and its coding.

1. On the **Tools** menu, select **Reports>Node Summary Report**.
2. In the **Customize Node Summary Report** window, select the nodes you want to report on, and specify your requirements. Note: hierarchical name will give you the tree structure if you have nodes in trees.



6. Click **OK**. The report appears on the screen.

Read the report, and if it is useful, print it.



Note that the report (like all reports generated this way) is not immediately editable. If you wish to save it as an editable report, from the toolbar, choose to **Export** the report.

## To log your development of nodes

1. Write a **memo** about this first stage of catalog construction.
2. If you are working in a team, and wish to merge your projects later, this is a good time to design a common catalog of basic categories and discuss how you will use it to ensure consistency but allow innovation. If you are planning on merging projects by Project Import, please go to the Online Help for advice.

## To create a coding comparison report

Amongst many other reports of your nodes and coding, you can make a special report on the similarities and differences in coding between two sources.

This report can be used for exploring different coding styles or checking differences in coding by different team members, as well as comparing coding across time as your ideas develop.

1. From the **Tools** menu select **Reports>Coding Comparison**.
2. In the **Customize Coding Comparison Report** dialog box, select the required options to specify the two sources whose coding you are comparing and the nodes for which it will be compared. You can also specify what information on the node content you want. Go to Help for details.
3. Click OK.



Do be careful to interpret the results appropriately: coding reliability is a challenging goal in qualitative research, and claims to reliability must be carefully established and discussed (see *Handling Qualitative Data* pp 98-100.)

This concludes the NVivo 7 Tutorial 6. Go to [Chapter 6 of \*Handling Qualitative Data\*](#) for advice on these processes.

You now have not only data and, ideas about your data, but also an evolving management system for those ideas. Time to back up your project!

As you continue relating data and ideas via links and coding, the ideas will be accessible for review and reflection.

If the reorganization of your nodes is satisfactory, you will now be able to “see” the relationships you are discovering and the logical groups of concepts and categories that are emerging in the project via the catalogs of nodes. But the catalog does not show the theories you are growing about how these categories relate. The next task is to represent the project visually in a model.