ATLAS.ti for Mac 1.0.x
User Manual

ATLAS.ti for Mac –User Manual

Manual Version: 78.20150121. Updated for program version: 1.0.14

Author: Dr. Susanne Friese (QuaRC)
Programming/Design/Layout: hypertexxt.com

Contents

Introduction....5
Mac-Specific Features....5
Main Concepts and Features....6
The ATLAS.ti Project File....6
Primary Documents....6
Document Groups....6
Quotations....7
Main Steps in Working with ATLAS.ti....12
The Process....12
General Steps when Working with ATLAS.ti....13
The Interface....15
Project Management....17
Creating A New Project....17
Adding Documents To A Project....18
The Edit Menu....21
Importing Survey Data....22
How survey data is treated in ATLAS.ti....22
Preparing an Excel™ Table for Import....23
How to import survey data....24
Analyzing Survey Data....25
Working With Quotations....26
Starting Your Analysis By Identifying Quotations....26
Useful Options When Working With Quotations....27
Working With Codes....29
Auto Coding....32
Suggestions For Structuring The List Of Codes....36
Creating (Code) Groups....37
Working With Memos....40
Difference Between Memos And Codes....40
How Memos And Comments Differ....40
Creating A New Memo....41
Working with Groups....43
Purpose of Creating and Working with Groups....43
Common Procedures....44
Creating Groups In the Navigator....45
Working With Smart Groups....47
Working with Hyperlinks....51

What Codes Cannot Do....51
Creating Hyperlinks....53

Margin Drag & Drop....58

Move Linked Objects....58

Working with Network Views....60

Further Analysis Tools....71

Simple Retrieval In The Code Or Quotation Manager....71
Working With Smart Codes....73
Working With The Code Cooccurence Table....83
The Word Cruncher....87

Creating Reports....89

Reports Based On The Entire Project....89
Print With Margin....90

Introduction

This User Guide is written for users with no prior experience of working with ATLAS.ti, but also for users new to the Mac version.

The Mac release version (1.0.x) is suitable for you if you…

- …start a new project from scratch
- …have no team project exchange requirements
- …are currently engaged in the code & retrieve and model building stages of your project
- …use Mac compatible multimedia documents in your projects exclusively
- …currently have no need for geo documents
- …feel more comfortable with having virtually unlimited undo/redo levels
- …can wait for more in-depth analytic features
- …can wait for reports to become available
- …work in different languages simultaneously – full Unicode support!
- …don't need associated documents yet (e. g., transcript and media)

Please check the "Feature Matrix" from http://mac.atlasti.com on every once in a while over the next few months to see if your "favorite" tool has already been implemented.

Remember: This version does not yet support team projects !

Mac-Specific Features

There are quite a number of neat new features in ATLAS.ti for Mac that are not available under Windows. These include the inspectors, the quotation display in the Quotation Manager, the way of handling audio and video data, the Undo function, full Unicode support, and several more.

For a full comparison of features see the Feature Matrix table at http://mac.atlasti.com/

The table (and this guide) will be updated on a regular basis to reflect the functionality and features that will be added to the Mac version in the coming months.

In developing ATLAS.ti for Mac, the aim was to create a software in which Mac users they feel at home, not just a copy the Windows version. Therefore the two versions will never be completely identical. Still, some of the features that you already find in the Mac version will also be implemented in ATLAS.ti 8. The Windows version and the Mac version will continue to "learn" from each other and will continue to mutually inspire future development.

Main Concepts And Features

The concepts of primary documents, quotations, codes, and memos are the overall foundation you need to be familiar with when working with ATLAS.ti, complemented by a variety of special aspects such as groups, network views (=the main visualization tool), and analytical/data querying tools. All of these come together in the overall project container.

The ATLAS.ti Project File

The most basic level of an an ATLAS.ti project consist of the Primary Documents, followed closely by the "quotations" (= segments/selections from the Primary Documents).

On the next level, codes refer to quotations. Memos - you meet them everywhere.

A project can become a highly connected entity, a dense web of primary data, associated memos and codes, and interrelations between the codes and the data. To find your way through this web, ATLAS.ti provides powerful browsing, retrieval and editing tools.

Primary Documents

Primary Documents (commonly abbreviated as PD or PDoc) represent the data you have added to an ATLAS.ti project. These can be text, image, audio, video or geographic materials that you wish to interpret. ATLAS.ti creates *copies* of the documents you assign to the project, i. e., your original documents do not become part of the project.

Document Groups

Documents can be grouped by any criteria that you need for your analysis. E,g. You might want to group them by gender: male and female, by age groups, education, family status, geographical region, document type, time aspects, etc. Such groups can later be used to restrict code-based searches like: "Show me all data segments coded with 'attitude towards the environment' but only for females who live in London as compared to females who live in the country side."

You can also use PD groups as a filter, for example to reduce other types of output, like a frequency count for codes across a particular group of documents.

The more advanced functionality of document groups, i.e. using them as global filters, is not yet implemented in the current Mac version!

Quotations

A quotation is a segment/portion of a PD that is deemed interesting or important by the user.

In textual documents, a quotation is an arbitrary sequence of characters ranging in length from a single character, to a word, a sentence, or a paragraph, even up to the entire data file. In an image, it can be any portion of the image, in an audio or video clip a segment of a certain length, etc.; more on that below.

Free quotations resemble passages "scribbled" on the margin of a book.

Usually, quotations are created manually by the researcher. However, if repetitive words or phrases are contained in the text, the Auto-Coding feature can be used to automatically segment these quotations and assign a code to them (see "Auto Coding").

Although the creation of quotations is almost always part of a broader task like coding or writing memos, "free" quotations can be created that indicate interesting parts in the primary data for which a meaningful classification has not yet been found.

Quotations As Layers

Quotations can be regarded like a transparent layer on top of a document. Technically speaking, a quotation consists of the identifier (a number) and a pair of coordinates that specify the beginning and end of the quotation. The content of a PD file (the data source) is therefore not altered by the creation, deletion, or modification of quotations.

Quotations are stored inside the HU, independent of the document to which they belong.

Additional reading: http://downloads.atlasti.com/library/Maietta_2009-05_6.pdf

Types Of Quotations

There are six different types of quotations (corresponding with the six different types of formats ATLAS.ti accepts) :

Text Quotations

A textual quotation originates from an arbitrary sequence of selected characters.

Textual quotations represent (for the computer) a sequence of characters ("strings") and can be of arbitrary size. Sentences, speech turns, or paragraphs are often the basis for the length of textual quotations. Only text offers enough "syntactical clues" to allow for searches for the occurrence of specific evidence that may support a concept. Text also offers the option for automatic segmentation as used by the Auto-Coding procedure (see "Auto Coding").

Graphic Quotations

The creation, activation, and display of graphical quotations has similarities with, but also differs from, their textual counterparts.

A graphical quotation is a rectangular region inside a graphical PD. From its data structure, it is identical to textual quotations, since their main attributes are also the PD identifier and two coordinates that mark the beginning and end, defining a rectangle through its upper left and lower right corner.

Handling graphical quotations is largely analogous to marking text passages in a textual document.

PDF Quotations

PDF quotations can be of a textual or of a graphical nature. The quotation references for textual quotations indicate the page number and the start and end position on the basis of character counts. For example: (31:1537-31:1745) means that this quotation is from page 31, starting at character 1537 and ending at character 1745. The reference for coded images indicates the position of the quotation within the PDF file, like (@422-@618).

Audio And Video Quotations

Audio and video quotations can be as short as a few milliseconds. The length of a quotation is selected on a time line. Segment starting points and length are displayed in the following formats:

`milliseconds / HH:MM:SS:ms`

Google Earth (GE) Quotations

Not yet implemented in the Mac version.

Codes

The term "code" is used in many different ways. First, we would like to define what that term means in qualitative research, and then in ATLAS.ti.

Coding Objectives

From a methodological standpoint, codes serve a variety of purposes. They capture meaning in the data. They also serve as handles for specific occurrences in the data that cannot be found by simple text-based search techniques.

Codes are used as classification devices at different levels of abstraction in order to create sets of related information units for the purpose of comparison (e. g., a concept like "Coping Strategy").

Keep code names brief and succinct. Use the comment pane for longer elaborations.

From a "low level" tool perspective, codes are typically short pieces of text referencing other pieces of text, graphical, audio, or video data. Their purpose is to classify an often large number of textual or other data units.

In the realm of information retrieval systems, the terms "index," "indexing," or "keyword" are often used for what we call "code" or "coding."

The length of a code should be restricted and should not be too verbose. If textual annotations are what you want, you should use quotation *comments* instead.

Memos

Memos capture your thoughts regarding the text and are an important device for creating theory. A memo may "stand alone," or it may refer to quotations, codes, and other memos. They can be grouped according to types (method, theoretical, descriptive, etc.), which is helpful in organizing and sorting them. Memos may also be included as the objects of analysis by assigning them as PDs.

http://downloads.atlasti.com/library/Friese_2009-09_1.pdf
http://downloads.atlasti.com/library/Friese_2008-12_8.pdf

Groups

Groups are a way to form clusters of PDs, codes, and memos for easier handling (see "Working with Groups." Primary Document groups can be regarded as attributes or variables. Groups can be combined using logical operators. These are called Smart Groups (see page 47).

Network Views

Network Views are a bit more sophisticated than groups. They allow you to conceptualize the structure by connecting sets of similar elements together in a visual diagram. With the aid of Network Views you can express relationships between codes, quotations, and memos. PDs, groups and even Network Views can also be "nodes" in a network view. See "Working with Network Views."

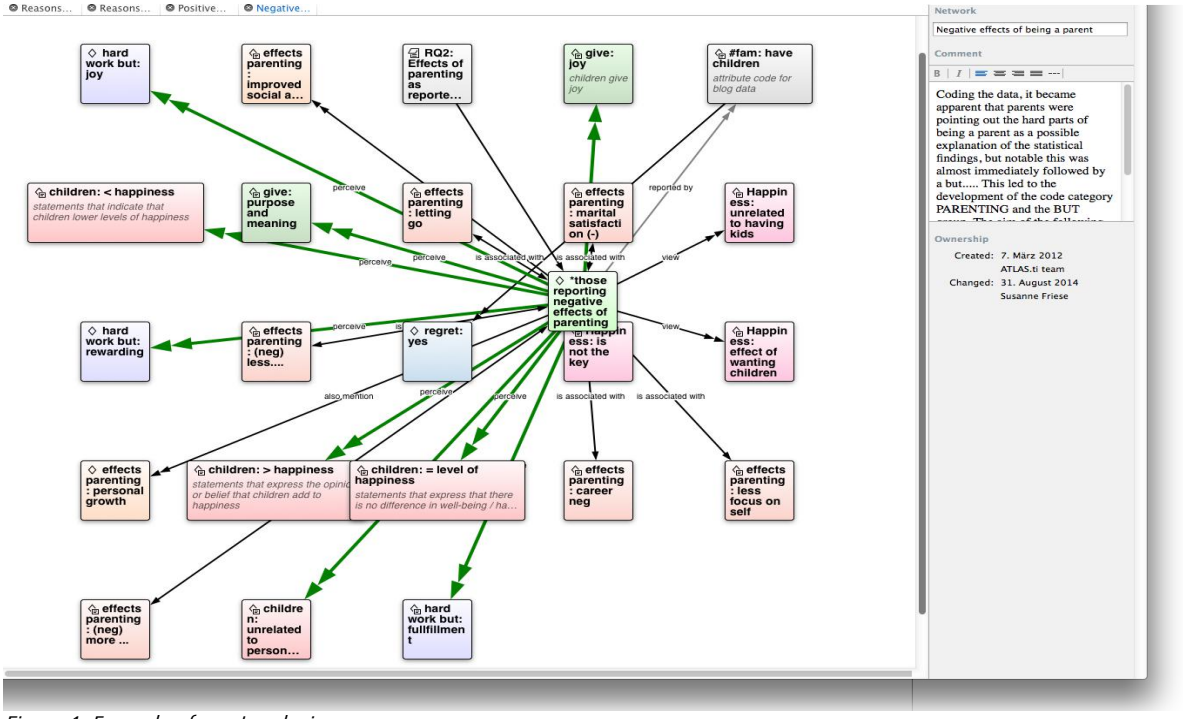


Figure 1: Example of a network view

Nodes, Links And Relations

A node is any object that is displayed in a Network View. You can change their look and move them around in the Network Editor.

Relations are link prototypes used to create a link between two codes or between two quotations. An example is the "is a" (ISA) relation, which is frequently used to link concepts of different abstraction level (e. g., DOG <isa> MAMMAL).

Network View Manager

The Network View Manager contains a list of all saved Network Views previously constructed by the user. It can be used to create new Network Views, to access or delete existing ones, or to write and edit comments.

Network Editor

The Network Editor displays and offers all editing capability to construct and refine semantic networks. In addition, it allows the visual creation and traversal of hypertext structures.

Relation Editor

Should the already built-in relations that are used to connect objects in Network Views prove not sufficient, you can edit them or create new ones using the Relation Editor.

Link Managers

The Link Managers provide an overview of all code-code links and of all quotation-quotation links you have created.

You find more information on the network function under "Working with Network Views."

Analysis

ATLAS.ti contains multiple powerful, dedicated analytical tool to help to make sense of your data once it is coded. See "Further Analysis Tools."

Query Tool (Not Yet Implemented)

For more complex search requests, the Query Tool will be at your disposal in the near future. The query tool will allow you to formulate search requests that are based on combinations of codes using one or a combination of 14 different operators, Boolean, semantic, and proximity operators.

Smart Codes

A Smart Code differs from a standard code. A standard code is directly linked with the quotations to which it is associated, while a Smart Code is a stored query, thus provides an answer to a question (in the best case) that typically consists of several combined codes. For further information, see "Working With Smart Codes."

Code Cooccurence Table

The Code Cooccurence Table is more exploratory than the Query Tool. Rather than determining the codes yourself, you can ask ATLAS.ti which codes happen to co-occur in the margin area. The code-cooccurence table provides frequency of co-occurrence and a coefficient measuring the strength of the relation is calculated. Since a coefficient is only appropriate for some type of data, its display can be activated or deactivated. It is always possible to directly access the data behind the frequency counts or coefficient.

You find this function under the **Analysis** menu. For further information, see "Working With The Code Cooccurence Table."

Codes-Primary Documents-Table (Not Yet Implemented)

The Codes-Primary Documents-Table contains the frequency of codes across documents. Aggregated counts based on code and primary document groups are also available. The table can be exported as Excel compatible table. Optionally, the table cells can also contain the word counts for the quotations per code across documents.

Export

Reports

Currently reports can only be generated from the entire project. The output options available under each menu option are based on the XML style sheets that you can also find in the Windows version.

Excel Export

Currently the results of the Code Cooccurence Table and the results of the Word Cruncher can be exported to Excel.

Main Steps In Working With ATLAS.ti

The figure below illustrates the main steps of working with ATLAS.ti, starting with the creation of a project, adding documents, identifying interesting things in the data and coding them. Memos and comments can be written at any stage of the process, whereas there is possibly a shift from writing comments like adding meta information to your documents, first code nodes that later turn in to code definitions, initial thoughts about specific data segments (the ATLAS.ti quotations) to more extensive memo writing during the later stages of the analysis. Once your data is coded, it is ready to be queried using the various analysis tools provided. The insights gained can then be visualized using the ATLAS.ti network view function.

Some steps must be taken in sequence. For instance, logic dictates that you cannot query anything or look for co-occurences if your data has not yet been coded. But other than that there are no strict rules. Network views, in addition to presenting findings, also have an exploratory component and as such can help you to see your data from a different perspective. This may provide further ideas for coding, querying, or even further data collection.

The Process

There are two principal modes of working with ATLAS.ti, the data level and the conceptual level. The Data Level includes activities like segmentation of data files; coding text, image, audio, and video passages; and writing comments and memos. The Conceptual Level focuses on querying data and model-building activities such as linking codes to networks, in addition to writing some more comments and memos.

Data Level Work

Data-level research activities include segmenting the data that you have assigned to a project into quotations, adding comments to respective passages (note-making/annotating), and coding selected text passages or data segments, secondary materials, annotations, and memos to facilitate their later retrieval. The act of comparing noteworthy segments leads to a creative conceptualization phase that involves higher-level interpretive work and theory-building.

ATLAS.ti assists you in all of these tasks and provides a comprehensive overview of your work as well as rapid search, retrieval, and browsing functions.

Within ATLAS.ti, initial ideas often find expression through their assignment to a code or memo, to which similar ideas or text selections also become assigned. ATLAS.ti provides the researcher with a highly effective means for quickly retrieving all data selections and notes relevant to one idea.

Conceptual Level Work

Beyond coding and simple data retrieval:

ATLAS.ti allows you to query your data in lots of different ways, combining complex code queries with variables, exploring relationships between codes and to visualize your findings using the network tool.

ATLAS.ti allows you to visually "connect" selected passages, memos, and codes into diagrams that graphically outline complex relations. This feature virtually transforms your text-based workspace into a graphical "playground" where you can construct concepts and theories based on relationships between codes, data segments, or memos.

This process sometimes uncovers other relations in the data that were not obvious before and still allows you the ability to instantly revert to your notes or primary data selection.

General Steps When Working With ATLAS.ti

The following sequence of steps is, of course, not mandatory, but describes a common "script:"

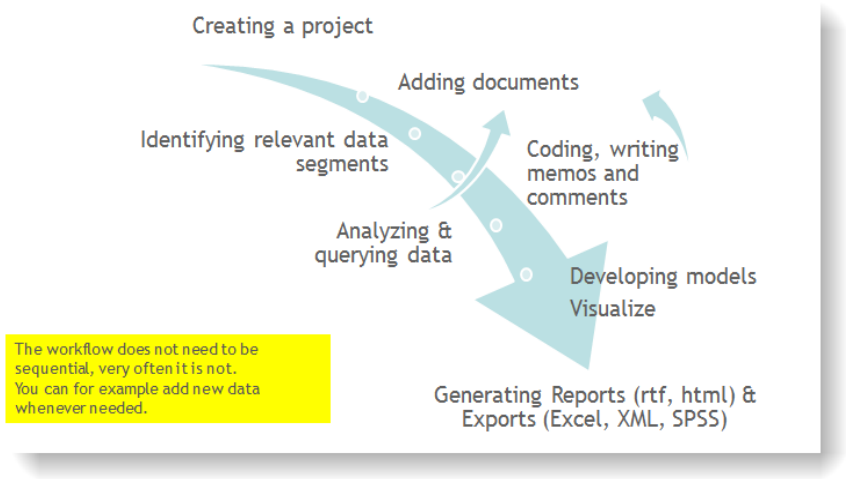


Figure 2: Main work flow

- Create a project (think of it as your "idea container"), meant to enclose your data, all your findings, codes, memos, and structures under a single name.
- Next, add documents, text, graphic, audio and video files, and Google Earth as so called Primary Documents (PDs) to your ATLAS.ti project.
- Organize PDs, codes, and memos using "groups" (see "Working with Groups")
- Read and select text passages or identify areas in an image or select segments on the time line of an audio or video file that are of further interest, assign key words (codes), and write comments and memos that contain your thinking about the data. The Word Cruncher and auto coding tool can help you to find interesting sections in text documents. We call this the data level working phase.
- Compare data segments based on the codes you have assigned; possibly add more data files to the project.

Query the data based on your research questions utilizing the different tools ATLAS.ti provides. The key words to look for are: simple retrieval, complex code retrievals using the query tool, simple or complex retrievals in combination with variables, smart codes, code cooccurences and the codes-primary documents-table.

Build semantic, prepositional or terminological networks from the codes you have created. These networks, together with your codes and memos, form the framework for emerging theory. Look for the chapter "Working with Network Views."

Finally, compile a written report based on the memos you have written throughout the various phases of your project and the networks you have created and even publish your project.

The Interface

Below you see the main features of the ATLAS.ti for Mac interface:

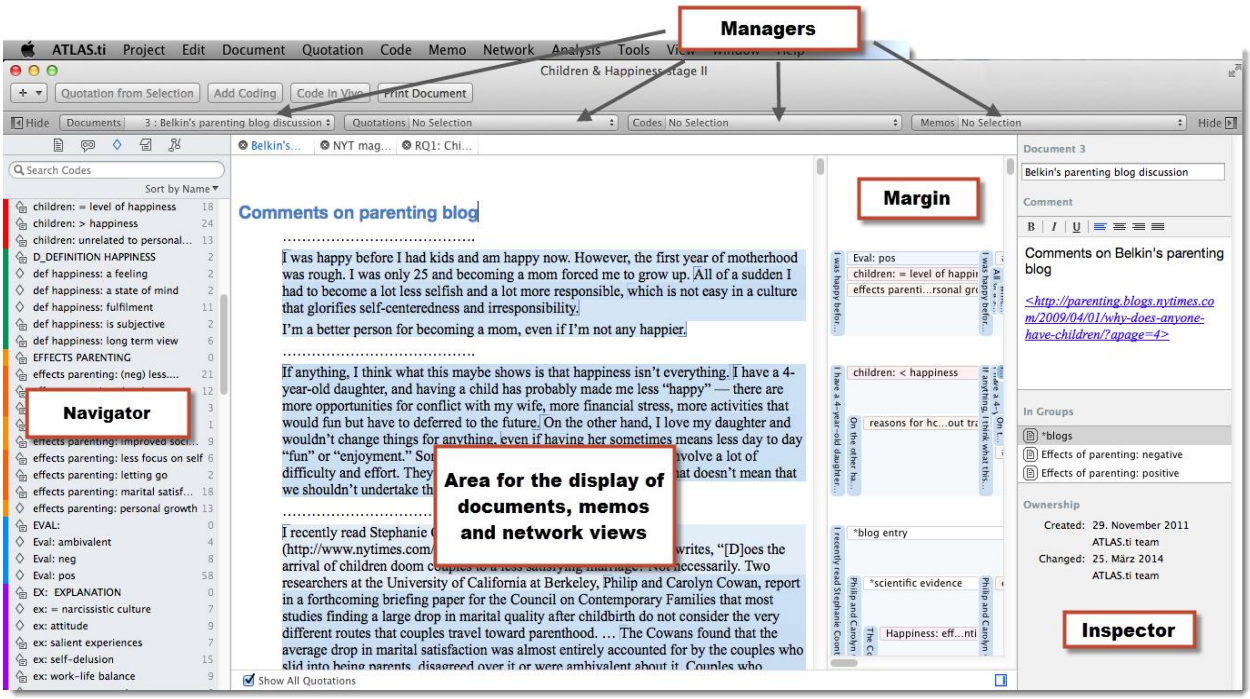


Figure 3: The ATLAS.ti for Mac user interface

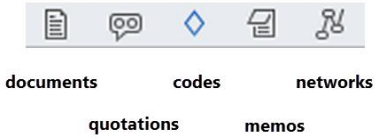


Figure 4: Entity representation

Personalize Your Screen

A number of panes that you see in the default interface view can be hidden (e.g., the navigator or the inspector). If hidingis possible, you see a little icon or a text "Hide" / "Show" that you can click.

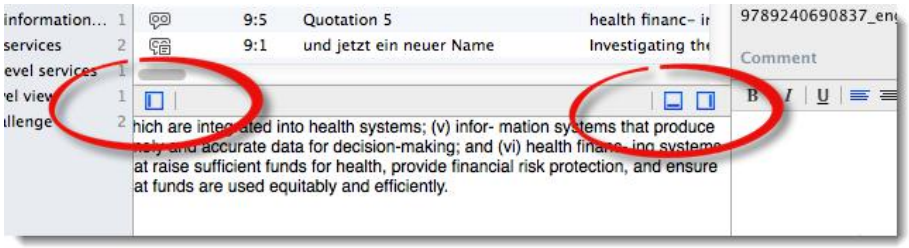


Figure 5: Hide or show optional panes

Finding Your Way Around The Software

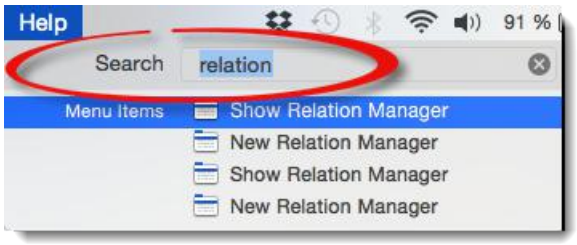


Figure 6: Finding menu options you are not sure about where they are

If you are not sure where to find a function, just search for it under **HELP**. For instance you want to know where the Relation Manager is. Typing in the first letters of a search term, shows a list of hits.

When selecting an entry that seems to be the one that you are looking for, it will show you where you can find it. Click on the selected hit in the list, and ATLAS.ti for Mac will open up the function for you.

Project Management

Creating A New Project

When you open ATLAS.ti for Mac for the first time, you are presented with two options: Create a new project, or import an existing one from other ATLAS.ti applications.

- Select **Create a new ATLAS.ti Project** and enter a project name:

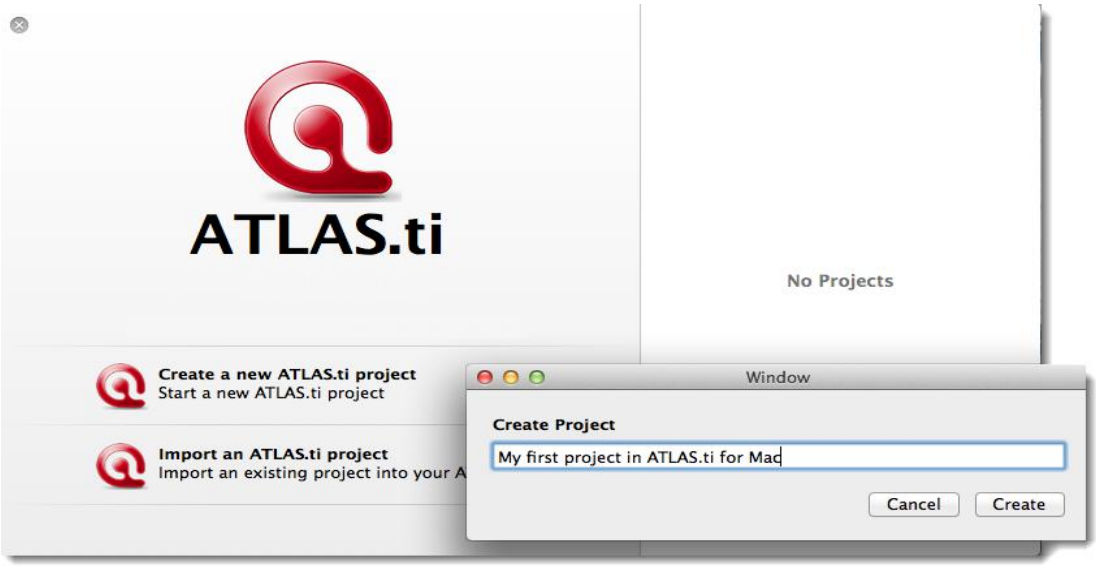


Figure 7: Creating a new project

Importing Projects From ATLAS.ti For Windows

- Update your ATLAS.ti for Windows to version 7.5 or higher.
- Export your project(s) using the via **Project / Export / Mac Transfer Bundle.**
- Move the Transfer Bundle file to your Mac or to a location where it can be accessed from your Mac
- On the Mac, simply double-click on the exported file, or use the import option shown in Error: Reference source not found above.

Importing Projects From IPad And Android

[This functionality becomes available in January 2015.]

Exporting Projects On The IPad

Select a project and then tap the **EXPORT** button.

Exporting Projects On Android Devices

To export your project, tap the overflow menu in the action bar and select the option **EXPORT PROJECT**.

Exporting Projects From Mac To Windows

This functionality will become available with ATLAS.ti 8 for Windows (expected in late summer 2015).

Adding Documents To A Project

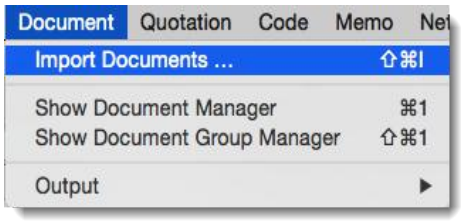


Figure 8: Adding documents to a project

Supported Formats

ATLAS.ti for Mac supports text (Word (doc and docx), RTF, OpenOffice (odt), PDF, image, audio and video files (all file formats that QuickTime supports, currently no Windows Media files).

How To Add Documents

Documents are imported and your project is saved silently.

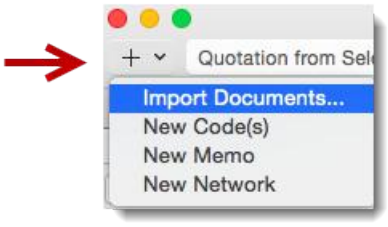


Figure 9: An alternative to the main menu option

To add documents, select **DOCUMENT / IMPORT DOCUMENTS...**.

An alternative to using the main menu option to add documents is to click the **+** button on the top left hand side of your screen (see left). More options like drag & drop will follow.

Loading Documents

All added documents can be accessed via the navigator that you find on the left-hand side of the screen, or via tabs. Once you have memos and network views later on, they can also be accessed this way.

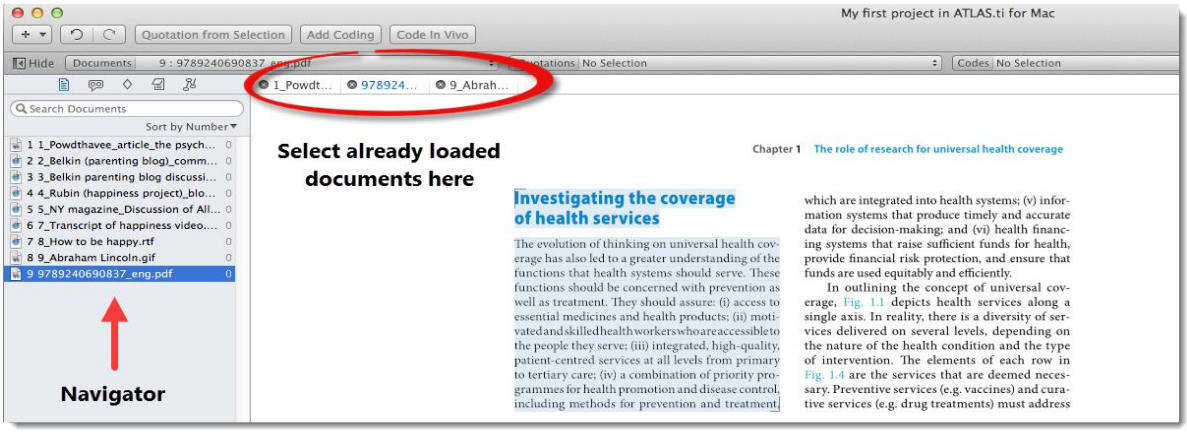


Figure 10: Loading documents

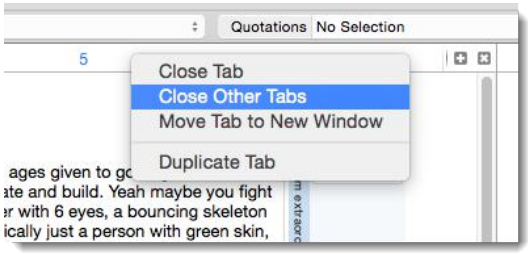


Figure 11: Close Tabs

Load one or more documents with a double-click on the document in the navigator (see below).

To close the selected document tab or all other loaded documents within a region, right click on the header area and select the appropriate option from the context menu: **Close Tab** or ***Close Other Tabs*** (see above).

Multiple Document View

You can load multiple documents side-by-side or split your screen horizontally, or both. And this does not only apply to documents, you can also open a document and a network view or a document and a memo side-by-side, or two documents side-by- side and a memo horizontally above or below (see for instance Figure 33). This is how it works:

Click on the plus sign to add a new region.

Right-click on the document tab and select **Duplicate Tab** (see left) to load an already opened document into a new region. You also find the "Duplicate Tab" option under the main menu **Windows / Duplicate Tab**.

TO ADD A NEW REGION VIA DRAG-AND-DROP

First load a document, memo or network into the display area.

Grab the document / memo / network view tab that you want to move, left-click and drag it to the right, left, top or bottom border of the main screen. When a small purple bar appears, you can drop it (see Figure 12).

If you want to replace another document / memo / network view that is currently loaded, drag and drop the object tab on top of document / memo / network view area (see below). Instead of just the small bar, the entire region for this object is highlighted in purple.

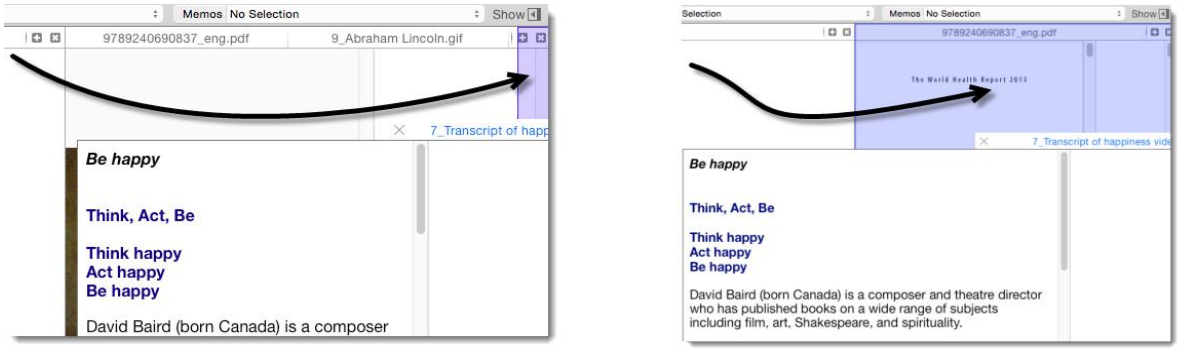


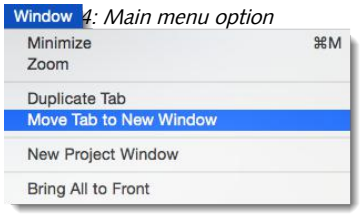
Figure 12: Loading documents side-by-side

Opening Multiple Windows

To open a document, network or memo in a window of its own:

Drag and drop the object tab outside the ATLAS.ti window and drop it there

Or select the option **Move Tab to New Window** from the context menu or the main menu Windows.



4: Main menu option

Figure 13: Opening multiple windows via drag-and-drop

Different from the multiple-document view, the new window contains its own inspector and navigator.

The Edit Menu

You can currently use the the options of the edit menu for comments and memos only. The full functionality of this menu becomes available once the following functions are added to the program: (a) Creation of new text documents in ATLAS.ti, e.g. for transcribing your data, and (b) capability to edit and modify imported text documents.

This functionality will become available in the first quarter of 2015.

Importing Survey Data

These days a lot of surveys are conducted online. A positive side effect is that (a) all data is immediately available in digital format and (b) respondents are often willing to provide lengthy answers to open-ended questions. This was rarely the case with paper-and-pencil surveys . Statistical programs like SPSS™ offer some options to analyze open-ended questions, but basically require you to encode each answer with a number. A proper qualitative analysis of the answers is not possible with these tools. ATLAS.ti, however, does permit comprehensive analysis with great ease and in great depth.

A typical work flow for working with survey data looks like this:

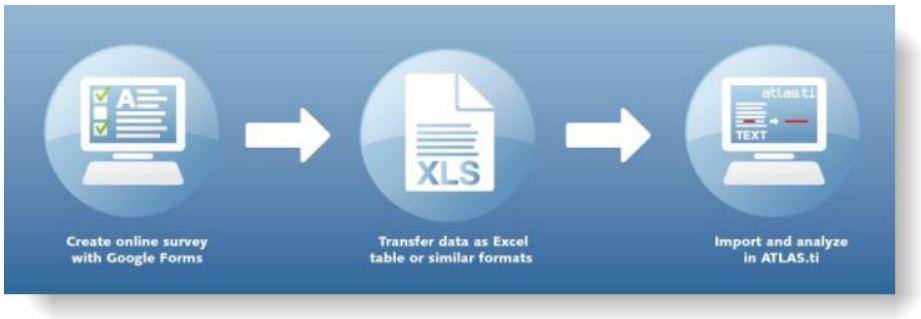


Figure 15: Work flow when importing survey data

Online surveys can be created using a variety of tools. What most of them have in common is that they let you export data as Excel™ file. And this is what you need to prepare for import in ATLAS.ti (see below). Based on specific prefixes that you add to your variable names, ATLAS.ti interprets the column headers and cells of the Excel™ table in various ways and turns them into primary documents, the contents of the primary documents, primary document families, quotations, codes, comments, and code families.

Data are imported case-based. This means each row of the Excel™ table that is imported from the online survey tool is transformed into a primary document.

How Survey Data Is Treated In ATLAS.ti

To represent the results of an online survey within the framework of ATLAS.ti in the most accurate and complete manner, a few basic requirements need to be considered.

A survey broadly consists of the name of the survey, the questions, and the answers from each respondent. Questions can be of different types, such as single-choice questions (yes/no, or offering more than two options), multiple-choice questions, or open ended questions. Within the framework of ATLAS.ti these concepts are mapped as follows:

Open-ended question: question answer	Code (and code comment) Content of a quotation
Single Choice 0/1	PD group
Single Choice > 2 options	PD group from question plus value
Multiple Choice	PD group from question plus value

This does not mean that you cannot turn a multiple choice question into a code, it all depends on how you define the columns in the Excel table. If you turn single or multiple choice questions into codes, you later need to do some automatic coding based on the response choices available in the questionnaire.

Always remember that ATLAS.ti is not a software for analyzing statistical data, although it is possible to add information from traditional quantitative questions. A general recommendation is to include the following variables from the online survey:

- Variables describing the respondents, i. e. the classical demographic variables like age group, gender, profession, educational level, income groups, etc.. Turn those variables into PD families.
- Responses to open-ended questions. Turn those into coded segments.
- Selected other variables like answers to single or multiple choice questions that are important in relation to analyzing the open-ended questions.

Preparing An Excel™ Table For Import

The following table shows the list of all available prefixes that you can to add to the column headers in Excel™ in order for ATLAS.ti to interpret the cells in various ways.

Prefixes and their interpretation	
Prefix	Result
!	Defines the column to list the names of the primary documents. This column can be omitted. If you do no want to provide a specific name for each document, ATLAS.ti automatically creates this column. The default name for each document is case 1, case 2, case 3 and so on.
^	Defines the column to list the primary document's author
&	Defines the primary document's date, expects ISO8601 format, see: http://www.cl.cam.ac.uk/~mgk25/iso-time.html Example: 4th of February 1995 or 1995-02-04 in ISO8601 format = YYYYMMDD = 19950204
<	Ignore this column, use to exclude stuff inserted by the survey tool
.	Creates a PD family from the field name. Currently the cell needs to contain the number 1, or the words Yes or Ja to be applied. Use for single choice questions where respondents can answer yes or no (or encoded as 1 and 0). Only the responses encoded with yes/ja or 1 become part of the PD family.
:	Creates a PD Family from the field name plus cell value. Use for single choice question like gender.
#	Creates a PD Family from the field name plus the actual cell value. Use for single choice questions with more than two answer options or for multiple choice questions.

	Add no prefix to all open ended questions. The text in the column header is used as code. To avoid very long code name, use a short form to indicate the question and add the full question as comment (see next row)
: :	All text entered after two colons is added to the object's comment field. This can be applied to cells resulting in PD families or codes.

!caseID	&dateTime	^Name	.Academic	.Army:Did you serve?	:Sex	#Favorites::What bands do you listen do regularly?	Tell us a dream	Open End::Now how do I make a question be interpreted as open-ended?
Case 1	20100414T1 20000	Michelle	0	1	female	Beatles, Stones	I was walking on the roof of a 12-story building....	This is a lengthy response to a question that may elicit looooong answers.
Case 2	20100415T1 71000	Hans	1	0	male	Stones, Scorpions	I am sorry, I don't remember any dreams at all	Quite simple: every field name (question) that has no prefix is an open-ended question be default.

Figure 16: Sample table ready for import

By importing the above sample table, we get:

- The name, date and author of the resulting PD.
- Two dichotomous families **Academic** and **Army.** The following comment is added to the PD Family "Army": Did you serve?
- Two families **Sex::male** and **Sex::female**
- Four families **Favorites::Beatles**, **Favorites::Stones**, etc. For the PD families the following comment is added: What bands do you listen to regularly.
- Two codes **Tell us a dream** and **Open End.** The following comment is added to the comment field of the code "Open End": Now how do I make a question be interpreted as open-ended?

How To Import Survey Data

Prepare an Excel table as explained above.

You can add the prefixes already when you prepare the online survey, or afterward by editing the the resulting Excel table. The drawback of adding the prefixes to the survey is that such early applied mark-up also displays in the online survey interface and your respondents might wonder about the funny characters at the beginning of questions.

Editing the downloaded Excel table prior to importing it into ATLAS.ti has the advantage of not displaying unusual formatting characters to the user. A disadvantage is that you need to do this every time you download the table, for instance, because you are checking incrementally and not all respondents have filled out the questionnaire yet.

Save the table (as xls or xlsx file).

Close Excel (or at least the document that you want to import).

Open ATLAS.ti and select PROJECT / IMPORT SURVEY.

The import procedure starts and ATLAS.ti informs you when the data is fully imported.

During the import, one primary document is created from every row. Unlike "normal" PDs that get their contents from files, these primary documents need to have their content (data source) created as well.

The creation of the textual contents is done as follows: Each row is scanned from left to right, column by column. Each cell's content belonging to an open-ended question field is appended to the currently built document (case). In addition, each piece of appended text is also referenced as a *quotation* automatically coded with the current question.

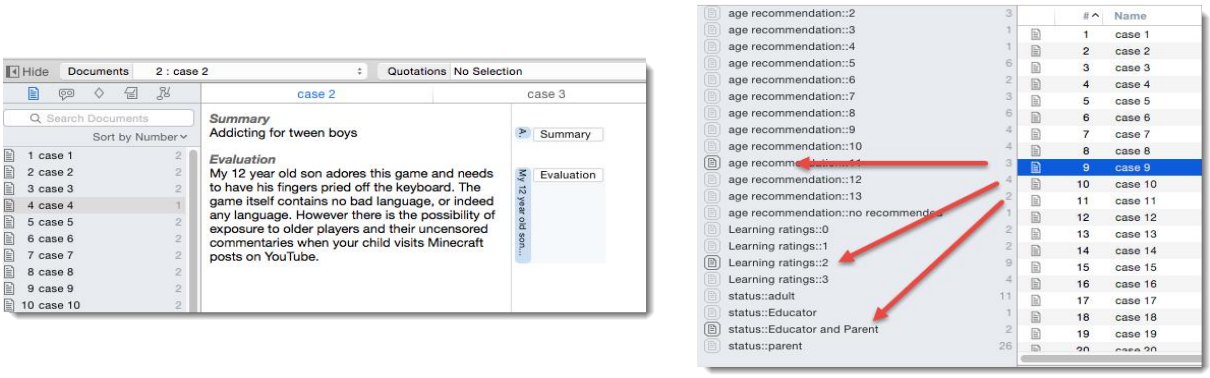


Figure 17: Results of a survey import

Those columns defined as variables are turned into PD groups. See "Working with Groups"for further information. Special families are created for all imported PDs and codes to make filtering of the survey material easy (see Figure 17 above).

Save the project.

If you import the same table repeatedly, rows with already existing PDs are ignored. This way, you do not have to wait until the last respondent has filled out the questionnaire.

Analyzing Survey Data

In order to analyze survey data, you should get acquainted with the following features and functions of ATLAS.ti:

- Primary Document Groups, especially their use as data attributes (see "Purpose of Creating and Working with Groups").
- Basic coding techniques to add some additional codes to the pre-coded responses (see "Working With Codes").
- Code Groups (see page 37).
- The Codes-Primary-Documents-Table (not yet implemented)
- The query tool and the scope function (not yet implemented)
- Creating and working with smart groups (see page 47).
- The Code Cooccurence Table (see page 83).

Working With Quotations

Starting Your Analysis By Identifying Quotations

It is not always desired to start setting codes immediately. You can begin by identifying interesting segments in your data, mark them and comment on them. This is exemplified below based on a video document. If you prefer to start coding your data directly, continue reading here: "Working With Codes."

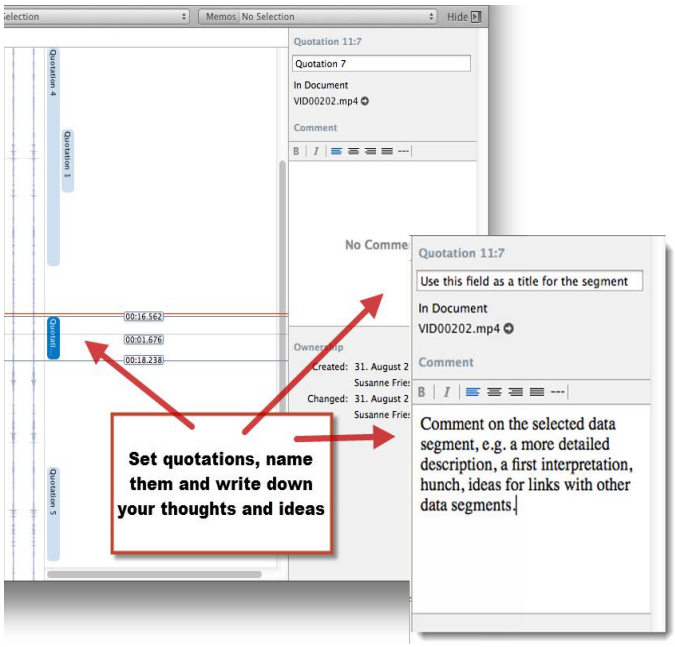


Figure 18: Working with quotations

In a video or audio document, you create a quotation simply by highlighting an area on the audio-wave form. For all other document types, you need to click on the button **Quotation from Selection**. Soon you will also be able to select this option from the context menu.

After creating a quotation, you can modify the default name and write a comment in the inspector (see Figure 18).

To keep track of your data and your thoughts open the Quotation Manager by clicking on the *Quotations* button, or select **Quotation / Show Quotation Manager** from the main menu:

Each quotation automatically receives an ID and a name. The default name is Quotation 1, Quotation 2, Quotation 3 and so on and can be renamed. If a quotation contains text, the first 100 characters are included in the column 'text content'.

Quotation ID: The quotation ID consists of the document number and a number indicating the chronological sequence when a quotation was created in the document. Thus, quotation 9:5 comes from document 9 and was the fifths quotation that was created in this document.

Useful Options When Working With Quotations

Rename Quotations And Write Comments

In the inspector on the right-hand side you can modify the quotation name and review or edit quotation comments.

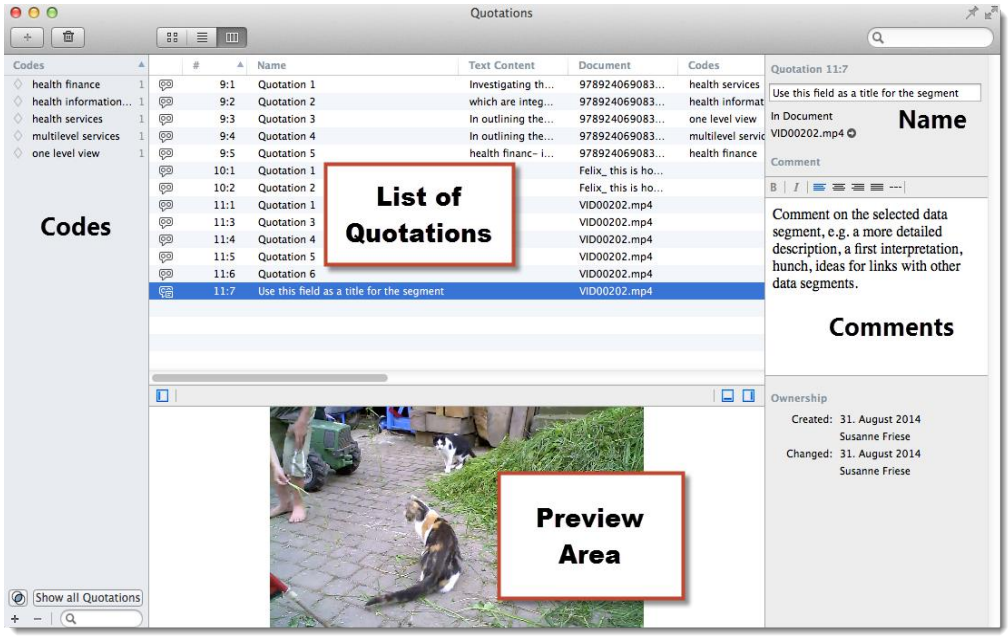


Figure 19: The Quotation Manager

Reviewing Your Quotations

When you select a quotation in the list, its content is displayed in the preview area. This also to all media types. You can read text quotations, listen to audio quotes, view image and video quotations. This is a convenient way to browse through your quotations (see Figure 19 above).

Reviewing Quotations By Code

When selecting a code on the left-hand side in the Quotation Manager, a filter is set and only the quotation(s) coded with the selected code are shown in the list.

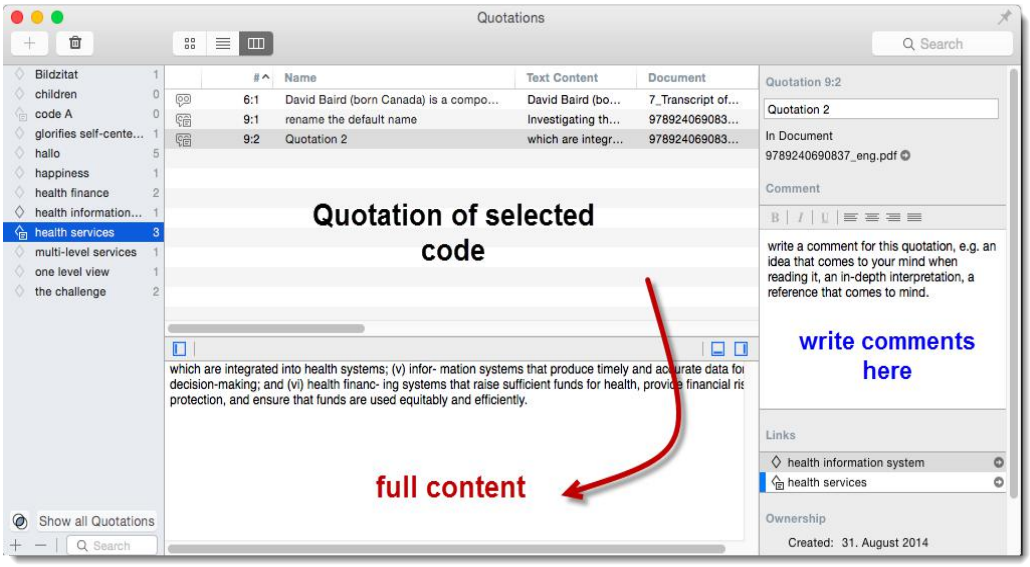


Figure 20: Reviewing quotations by code

Modifying The Length Of A Quotation

Just drag the start or end points to a different position. The quotation bar in the margin area automatically follows. This applies to all media types (see Figure 21).

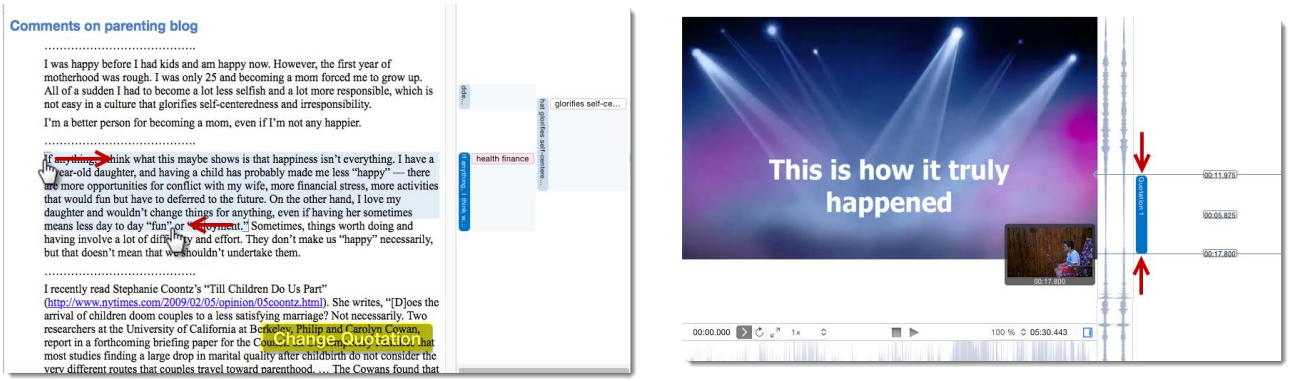


Figure 21: Modifying the length of a quotation

Working With Codes

After adding documents, you can begin to code your data. If you prefer to start by identifying interesting data segments first, see "Working With Quotations."

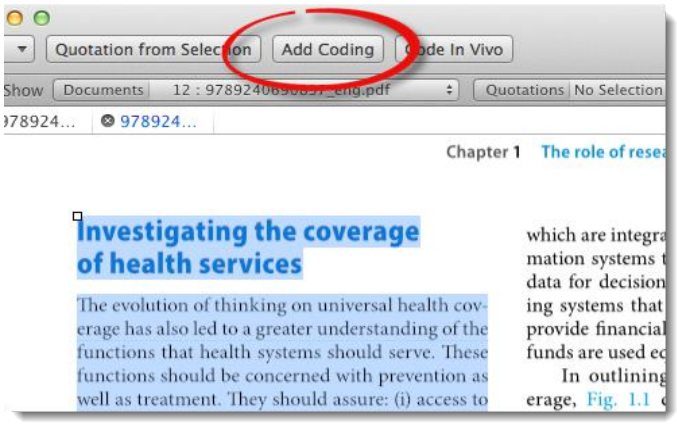
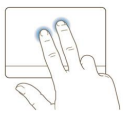


Figure 22: Coding a selected data segment

To code a data segment, highlight it with your mouse and select the **Add Coding** button **(short-cut: ⇧⌘O)**



An alternative is to right-click on the highlighted segment and select **Add Codes** from the secondary menu.

After coding, the quotation name and the code label are displayed in the margin area (see left).

Coding image, audio or video data works in the same way: You highlight a segment with your mouse, click on the Add coding button (or right-click and select **Add Codes**), and enter a code. When coding audio or video data, you highlight a segment on the audio-wave form (see Figure 18 below).

Quick Coding

Quick Coding assigns the last used code to the current data segment. This is an efficient method for the consecutive coding

of segments using the most recently used code.

Select a data segment and click on the Quick Coding button **(short-cut: ⇧⌘K)**

The Code Manager

Open the Code Manager by clicking on the button **Codes** above the document area, or select **Code / Show Code Manager** from the main menu.

If you want the Code Manager to stay on top, click on the pin icon in the top right-hand corner.

The Code Manager offers the following options: you can add write code comments, add code colors, retrieve coded segments, create code groups or use it for drag & drop coding. Further, you can review code frequency and density.

Frequency (indicated by the quotation icon): number of quotations coded with the code

Density (Indicated by the code icon) is the number of linkages to other codes that the user has created.

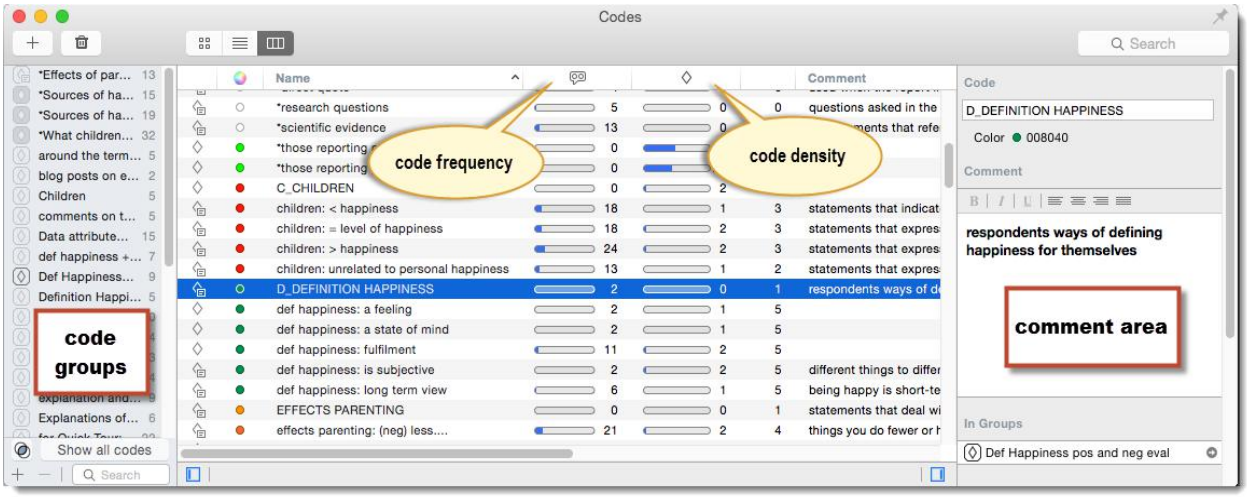


Figure 23: The Code Manager

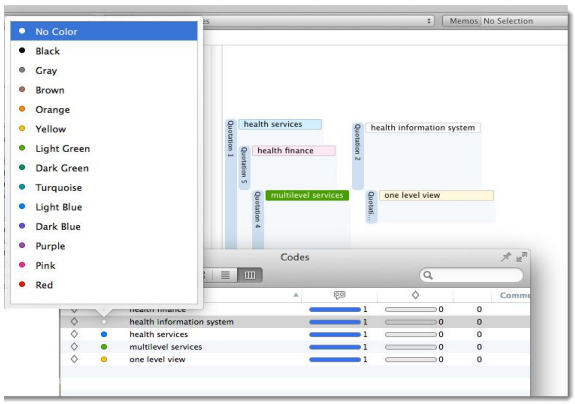


Figure 24: Adding code colors and display in margin area

Adding Code Color

Click on the circle in the "color column" just before the code name and select a color for each code. In Figure 24 below you see how code colors are displayed in the margin.

Coding With Existing Codes

You can either drag & drop a code from the Code Manager onto a selected data segment, or as shown in the Figure 25 below, from the navigator. See also "Margin Drag & Drop."

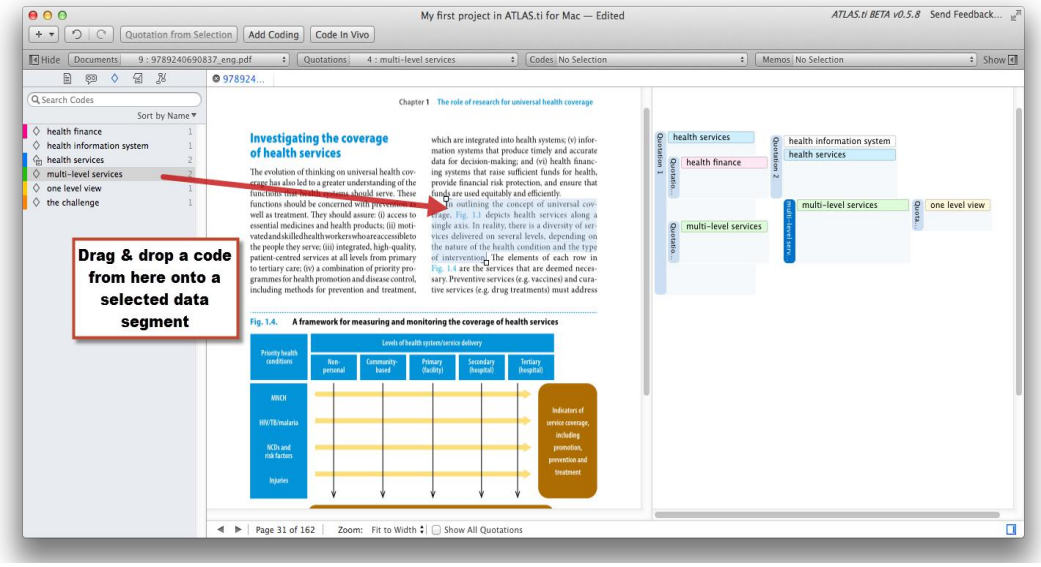


Figure 25: Coding via drag & drop

Unlinking Codes

Right-click on a code in the margin area and select the option **REMOVE FROM QUOTATION**.

Merging Codes

Clean up synonymous codes by merging them.

When developing a coding schema, it may happen in the course of the analysis that two or more codes essentially mean the same thing. One cause could be the import of code lists with different names but similar meanings.

ATLAS.ti offers a procedure to merge synonymous codes into one resulting "target" code. This target code replaces the merged codes and "inherits" all of their references, i. e., quotations, links to other codes or memos, and their comments.

There are two ways to merge codes: a list-based method, and one that works from within the Network Editor.

In the Code Manager select the code that you want to merge with another code.

Drag-and-drop this code to the "target" code that you want to keep. A window opens. Select the last available option: ***MERGE CODE A INTO CODE B.*** The quotations are added to the target code and the merged code is removed from the list of codes.

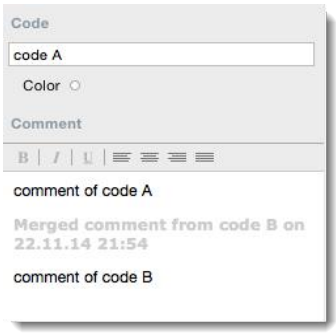


Figure 26: audit trail after merging

Figure 27: Merging two codes

If the merged code has a comment, the comment is added to the target code. If both codes have a comment, an audit trail is provided (see Figure 26)

Exchanging / Swapping A Code

To replace one code with another in the margin area, drag and drop a code (either from the Code Manager or the navigation pane) on top of another code in the margin area. Select the **Swap** option from the context menu.

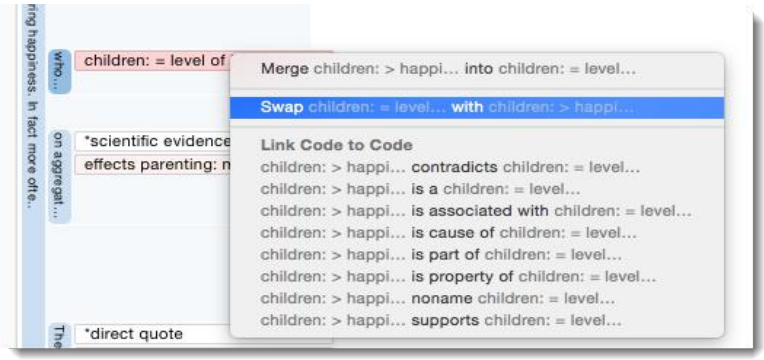


Figure 28: Swapping or exchanging a code in the margin area

If you drag-and-drop a code from the margin area on top of another code in the margin area, then these two codes are **exchanged**..

Auto Coding

Auto-Coding = Text Search + Automatic Segmentation + Coding

The Auto-Coding tool finds text passages, selects a specified amount of text (e. g., the exact match, or spread to the surrounding word, sentence, or paragraph), and then codes the passages with a previously selected code.

Auto-coding is useful when coding structural information like speaker turns in group interviews, or if you are dealing with a lot of text material. Auto-coding allows you to quickly collect ideas that belong to a certain concept on the basis of words or patterns found in the text.

To Start Auto Coding

To open the Auto-Coding Dialog, select **Codes / Auto Coding** from the main menu.

Figure 29: The auto-coding dialog

Find: A search expressions can be single words or phrases, or you can create more complex search expressions using regular expressions (GREP). See "GREP Search."

Code: Click on the down arrow to chose an existing code from the list. If the code you need is not available yet, create one on the Code Manager.

Extending: When a matched string is found, the size of the segment to be extended to the following lengths:

- The *Exact Match* only
- The *Word* surrounding the matched string
- The *Line* surrounding the matched string
- The Sentence surrounding the matched string
- The matched string expanded to *paragraph* boundaries
- Or the entire document

Scope of Search can be all textual PDs, the selected PD only, or it may be restricted to a set of texts from a PD group.

Example: Based on the GREP search expression shown in Figure 29, all occurrences of words that contain child or toddler or baby or babies in the currently selected PD are found and the surrounding sentence is coded with the code "children." The GREP OR expression | can be entered using the short-cut ***alt+7***.

You can select the option **Code All** without checking each hit, or you can review each hit be clicking the **Next** or **Previous** button. Then select *Code* for each instance that you want to code.

Preparing Documents Optimized For Auto-Coding

The following instructions are useful for preparing transcriptions of focus group sessions, questionnaires, or interviews. Such data usually contains different speakers' sections. The hints given here also apply for other documents that include sections you wish to identify for auto-coding.

When transcribing interview or focus group data, enter a blank line between speaker turns if you are going to to use the auto-coding tool for coding speaker units.

It would be tedious to code speaker or section turns manually. Two things are needed: A good "marker" for which to search and, once the marker is found, a reliable identification of the unit (sentence, paragraph) to be coded.

Insert easily identifiable markup in the text to let the auto-coding pattern matcher do this for you. In combination with a few formatting rules, documents can be created that can readily support auto-coding quite a bit.

Simple examples are:

P: or *<Peter>*for a respondent with the name Peter

07-01-11letter for a letter written on the first of July 2011.

The identifier should be used exclusively to mark passages in the text that indeed relate to the person or object identified. The plain word *Peter* will likely also occur elsewhere in the text (for instance, when another person is referring to Peter). The markup "*P:*" or *<Peter>* however, is unlikely to occur elsewhere.

In order for the Auto-Coding tool to select a complete speaker section, a *section delimiter* is needed. As discussed above, a speaker or section turn will start with the speaker's identifier markup. The end of a section is best marked by an empty line, i.e. two paragraph markers (see Figure 30). The text of one speaker needs to be contained within one paragraph. If you brake a speech into multiple paragraphs, only the first paragraph will be coded.

When auto-coding such a document, you would choose **PARAGRAPH** for

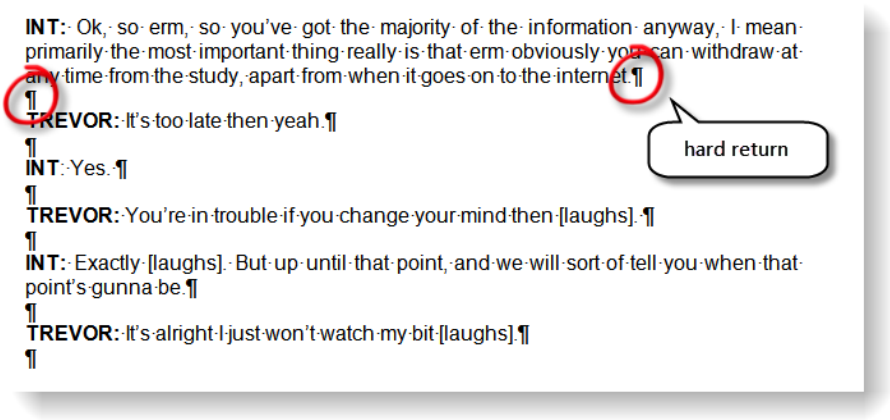


Figure 30: Example transcript

extending the matched text.

GREP Search

GREP searches are very flexible.

The core of a GREP search is the inclusion of special characters in the search string that control the matching process. GREP finds instances in your data that match certain patterns.

The ATLAS.ti GREP search offers a subset of the Regular Expression language used in sophisticated text search systems. For detailed information about regular expressions see http://en.wikipedia.org/wiki/Regular_expression

Regular Expressions efficiently exploit the syntactical properties of the technical device called "text."

Supported GREP Expressions

GREP Expression	Description
^	Matches an empty string at the beginning of a line.
$	Matches an empty string at the end of a line.
.	Matches any character except a new line.
+	Matches at least one occurrence of the preceding expression or character.
*	Matches the preceding element zero or more times. For example, `ab*c` matches "ac," "abc," "abbbc," etc.
?	Matches the preceding element zero or one time. For example, `ba?` matches "b" or "ba." []
[]	Matches a range or set of characters: `[a-z]` or `[0-9]` or `[aeiou]`. For example: `[0-9]` finds all numeric characters, while `[^0-9]` finds all non-numeric character
\b	Matches an empty string at a word boundary
\B	Matches an empty string not a word boundary
\<	Matches an empty string at the beginning of a word
\>	Matches an empty string at the end of a word
\	The escape character disables the special GREP functionality of the following character. For example: `\[` matches an opening bracket.
|	OR. Enclose ORed expressions with parentheses if OR should be restricted to certain sequences of characters or expression

Examples Of GREP Searches

The expression `man|woman` matches "man" and "woman."

You could also use `(|wo)man` to the same effect. `H(a|e)llo` matches "Hello" and "Hallo." `H(a|e)+llo` matches "Haaaaaallo" as well as "Heeeeeaaaaeaeaeaeaello."

And how about the `(angry|lazy|stupid) (man|woman) (walk|run|play|fight)ing with the gr(a|e)y dog` – get the idea?

GREP Expression	Description
\d	Matches any digit (equivalent to [0-9])
\D	Matches anything but a digit
\s	Matches a white-space character
\S	Matches anything but a white-space character
\w	Matches any word constituent character
\W	any character but a word constituent

Note that these elements are components of the character classes, i. e. they have to be enclosed in an extra set of square brackets to form a valid regular expression. A non-empty string of digits or arbitrary length would be represented as [[:digit:]]+

These escapes are also allowed in character classes: `[\w+-]` means "any character that is either a word constituent, or a plus, or a minus."

`\<\w+\>` matches any whole word

`\<[[:alpha:]]+\>` matches whole words containing only alphanumeric characters.

Character classes can also include the following elements:

GREP Expression	Description
[:alnum:]	Any alphanumeric, i.e., a word constituent, character
[:alpha:]	Any alphabetic character
[:cntrl:]	Any control character. In this version, it means any character whose ASCII code is < 32.
[:digit:]	Any decimal digit
[:graph:]	Any graphical character. In this version, this mean any character with the code >= 32.
[:lower:]	Any lowercase character
[:punct:]	any punctuation character
[:space:]	Any white-space character
[:upper:]	Any uppercase character
[:xdigit:]	Any hexadecimal character

In the following, a few search examples are presented showing the matching GREP expression in the column on the right.

GREP expression	Examples
\[.*\]	Find text (of arbitrary length) enclosed within brackets. Note that the brackets have to be escaped with "\" as they are themselves control characters
200[1-4]	Find all years between 2001 and 2004:
\d\d$	Find all numbers with 2 digits at the end of a line or paragraph:
M[ae][iy]er	Find all "Meyer"s - spelled in four different ways:
^.[:]	Find all lines (paragraphs) starting with one arbitrary letter followed by a colon. - If you search for a "d" following the colon you will have to use the Escape character: ^.\:d Otherwise, the letter "d" would be interpreted as a digit.

Suggestions For Structuring The List Of Codes

A frequent question is how to add a structure to the otherwise flat code list in ATLAS.ti. The easiest way to work with higher and lower order codes is to structure your codes alphabetically in the Code Manager, e. g., according to the basic pattern below:

```
category A_sub 1
category A_sub 2
category A_sub 3
category B_sub 1
category B_sub 2
category B_sub 3
```
etc.

As a means to visualize the beginning of a category, you can enter a free code that is not linked to any quotation, as shown below. Such a main category code might initially be empty, but may proof to be quite handy during further coding work. You may come across some data that fits the category but there is no fitting sub code yet, or you are unsure where to put it. Then you can use the main category code to collect these instances. Once a number of instances are collected, you can review them and think some more about them. By reading through or viewing/listening to a couple of examples it becomes often easier to decide how to code it. You may decide to create a new sub code or decide that an existing sub code fits after all. Adding colors will also help you to distinguish between different types of level of codes.

```
CATEGORY A
category A_sub 1
category A_sub 2
category A_sub 3
CATEGORY B
category B_sub 1
category B_sub 2
category B_sub 3
```

A further possibility is to sort codes by numbers (or numbers and letters):

```
1 CATEGORY A
11_A_sub 1
12_A_sub 2
13_A_sub 3
2 CATEGORY B
21_B_sub 1
22_B_sub 2
23_B_sub 3
```
etc.

The sorting order is: (1) special characters (* + ' # - _ : . ; , etc.), (2) numbers, and (3) letters. Therefore, it is NOT a good idea to use prefixes like 1, 1.1. , 1.1.1 , 2, 2.1, 2.1.2 etc.. Such a numbering scheme will necessarily wreak havoc with your intended sorting order ("10" will be sorted before "2"). When using numerals , always use "01," "02," "03," or and so on for sorting to be on the safe side.

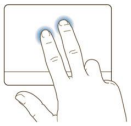
All terms preceding a colon (:) indicate the main category name; the terms following the underscore or colon constitute sub codes. Other projects may require additional sub levels. But don't overdo it!

As main category code and sub code names may contain more than one word, an empty space is not sufficient to separate the two levels of coding. Therefore it is best to use one the of special characters that you find on your keyboard to visually separate levels of coding.

Creating (Code) Groups

Codes, documents and memos can be grouped. Groups are the equivalent to "groups" in ATLAS.ti for Windows. Groups can be used for sorting and organizing purposes, but also as filter. Currently groups can be used to set local filters, which means the filter only applies to the currently open window and not to the entire project. Global filter options will be added in the future.

This is how you create groups:



Open one of the Managers. Click on the plus sign at the bottom of the Groups pane or right-click inside the Groups pane and select **New Group**. Enter a group name.

Add members to the group via drag and drop from the list of documents / codes / or memos in the respective manager.

If you click on a group, the items in the manager are filtered and only the members of the selected group are listed. Click on **SHOW ALL CODES** to reset the filter.

To remove an item from a group:

Open the Group Manager, e.g. via the main menu (Show Documents- / Codes- / Memos Group Manager), or double-click on a group; or right-click on a group in a manager. Then select: **OPEN IN GROUP MANAGER**.

The Group Manager

All group managers currently offer the following options:

You can search for groups, rename groups, write descriptive comments, review creation and modification dates, add new code groups by clicking on the + button, and add and remove group members

To Add New Group Members

Select one or more objects from the list "*Not in Group*" and click on the button with the double arrows pointing to the left. Or double-click each item.

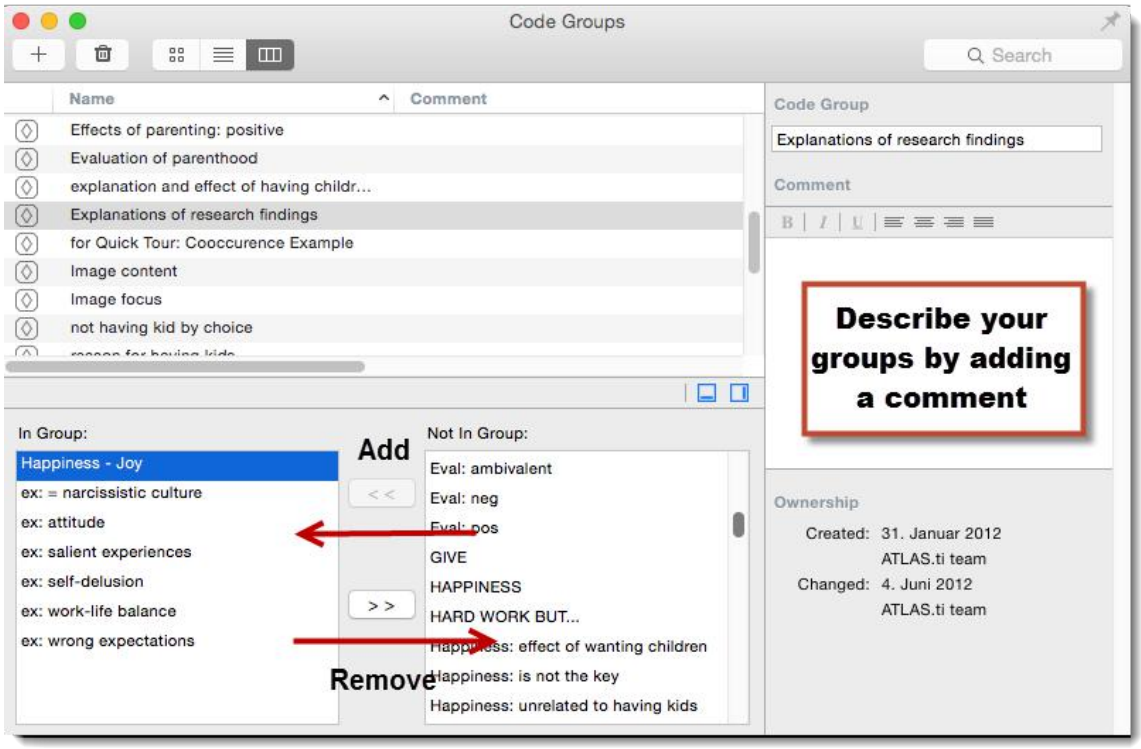
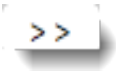


Figure 31: The Code Group Manager

To Remove Group Members



Select one or more objects from the list "*In Group*" and click on the button with the double arrows pointing to the right. Or double-click each item.

Working With Memos

Possible uses for memos:

- To take methodological notes
- As a "bulletin board" in team projects
- As a project planning device.
- As a primary document
- To create code lists

Writing memos is an important task in every phase of the qualitative analysis process. The **ideas** captured in memos are often the "pieces of a puzzle" that are later put together in the phase of report writing. **Theory-building**, often associated with building networks, also can involve the use of memos.

Memos are explanatory and descriptive texts that may be **associated with other "objects"** like quotations, codes, or other memos. Memos can also **"stand alone"** – simply as part of an HU. They can contain **methodological notes**; they can be used as a **bulletin board** to exchange information between team members; you can use them to write **notes about the analytical process**, keeping a journal of to-dos. Memos may also serve as a **repository** for symbols, text templates, and embedded objects that you may want to insert into PDs or other memos.

Difference Between Memos And Codes

Code names are (or should be) succinct, dense descriptors for concepts emerging during the stage of closely studying the data. They often reduce complex findings to crisp placeholders and/or theoretically relevant concepts.

Beginners often stuff lengthy treatises into a code name, blurring the distinction between codes, comments, and memos and thereby mistaking codes for their more appropriate siblings.

If you find yourself using more than a few words as code word, consider using quotations or the code comment instead.

Like codes, memos have names. These names, or titles, are used for displaying memos in browsers, and help to find specific memos. Just like code names, a memo's title should be short and concise. Don't confuse the name with its content!

How Memos And Comments Differ

A *comment* is always a **part** of another object;. A *memo* is an object by itself.

Memos are very similar to comments in that both are intended to hold lengthy texts, as opposed to codes that are simply naming a concept. Comments exclusively belong to one entity. For example, the PD comment is part of the primary document.

Comments are not displayed in browsers separately from the object to which they are attached. Memos can be associated with more than one object and have an additional **type** attribute, e. g., theoretical, methodological, commentary, etc. They can also be free-standing, unlike comments.

Creating A New Memo

Open the Memo Manager by clicking on the Memos button or select **Memo / Show Memo Manager** from the main menu.

Click on the + sign to add a new memo. Enter a name.

Double-click to open it. It will open as new tab.

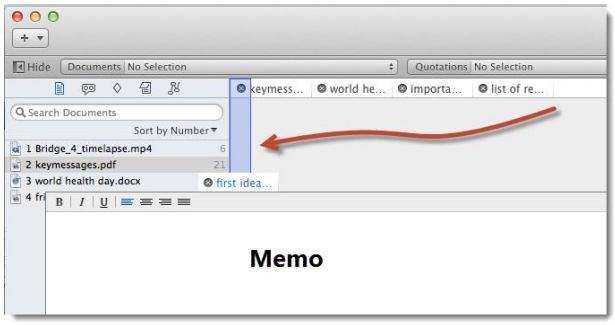


Figure 32: Open a memo, document or network in a new region

Loading Memos

You can also open a memo into its own region, so that you can see both the memo and your data. To do so, drag the memo tab to the right, left, top or bottom edge of the document area. If you see a blue band, you can drop it (see figure below).

You are free to split your screen vertically or horizontally (see also "Multiple Document View").

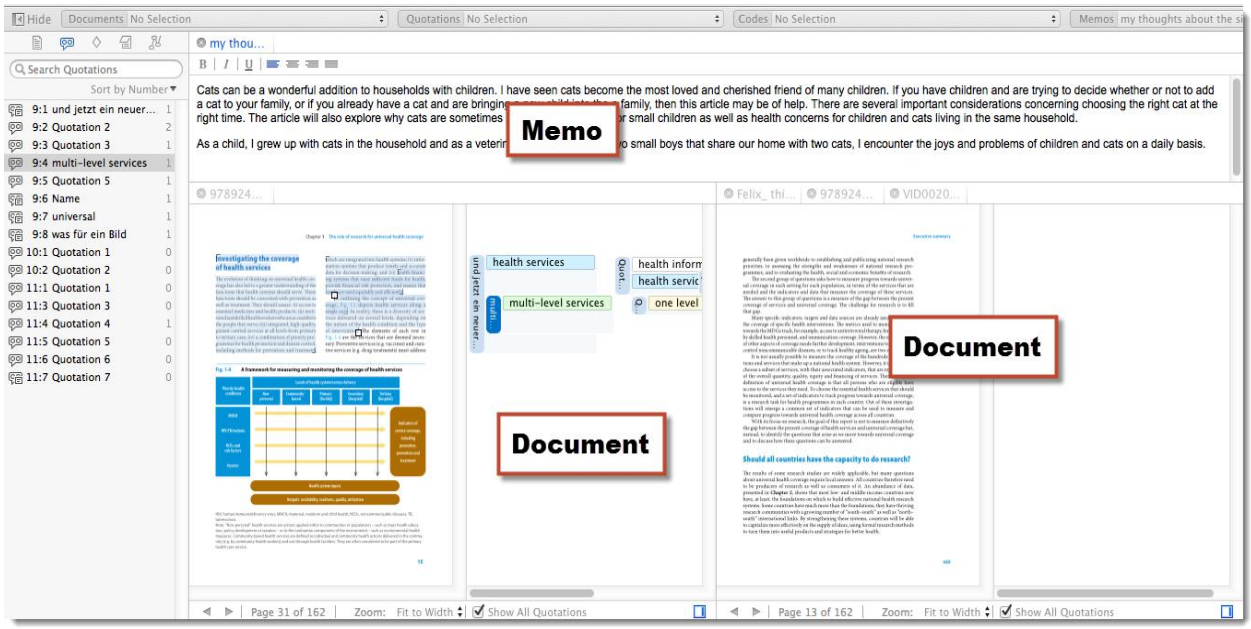


Figure 33: Alternative arrangements of documents and memo

To link a memo to a data segment, drag & drop it from the Memo Manager onto a quotation, or select a memo in navigator and drag & drop it from there.

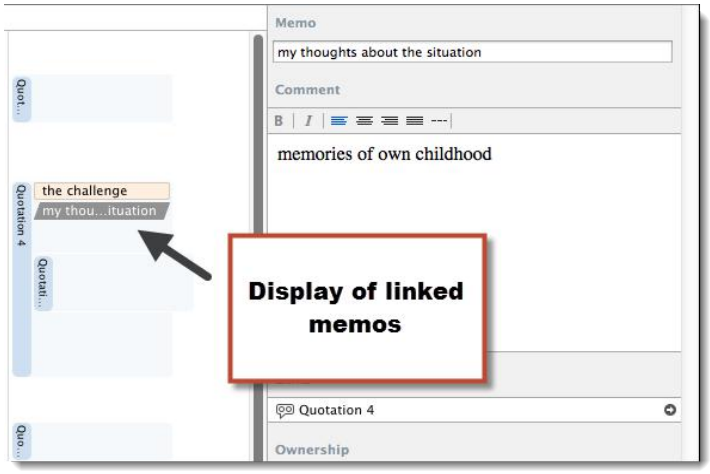


Figure 34: Working with memos

When selecting a memo in the margin area, the inspector on the right-hand side is displaying the detailed information for this memo (see Figure 34).

Working With Groups

Just as codes describe sets of quotations, groups cluster Primary Documents, Codes, and Memos. One important objective is to manage large amounts of objects by classifying them into subsets, e. g., all *theoretical* codes, all PDs from *respondents of a certain age group* or *location*, all memos related to a *theme*, etc. Objects of the same type can be a member of multiple groups. For example, the interview transcript of Eva Smith, can be a member of document group Gender: female and a member of document group: Location: city.

Unlike Network Views, which can contain objects of different types as nodes, groups can only contain one type of object. For example, a code can never be a member of a memo group.

Purpose Of Creating And Working With Groups

Partitioning objects into groups reduces the number of "chunks" requiring the researcher's attention. Groups are often used for filtering and when formulating queries in the Query Tool. The navigators in the managers offer a convenient way to create them and to set them as local and global filters.

In the current version, groups can only be used as local filter, i.e. when selecting a group in a manager, the list of objects in the manager is filtered and only the objects of the selected group is shown. A global filter in comparison has an effect on the entire project. This latter functionality will be implemented in the spring of 2015.

Example: When conducting an interview study with respondents from various backgrounds and locations, **PD groups** can be created to classify the respondents into:

- Female / Male
- Marital status
- Age Group 1 (20-30), Age Group 2 (31-40), Age Group 3 (41-50)
- Educational level, etc.

Once implemented, you can use **PD groups** to compare and contrast answers of different groups of respondents, or check whether they are differences between locations, across time or type of documents. In the Query Tool, you will be able to use PD groups to restrict the scope of the search, for example to ask for all quotations coded by Code_A and Code_B that occur in documents of white-collar female respondents from location B. Thus, PD groups in effect can be used as variables. Figure 35 below shows how PD groups can be used as local filters: The manager only shows documents of married female respondents.

Currently groups can only be used as local filters, i.e. you can filter the objects within managers.

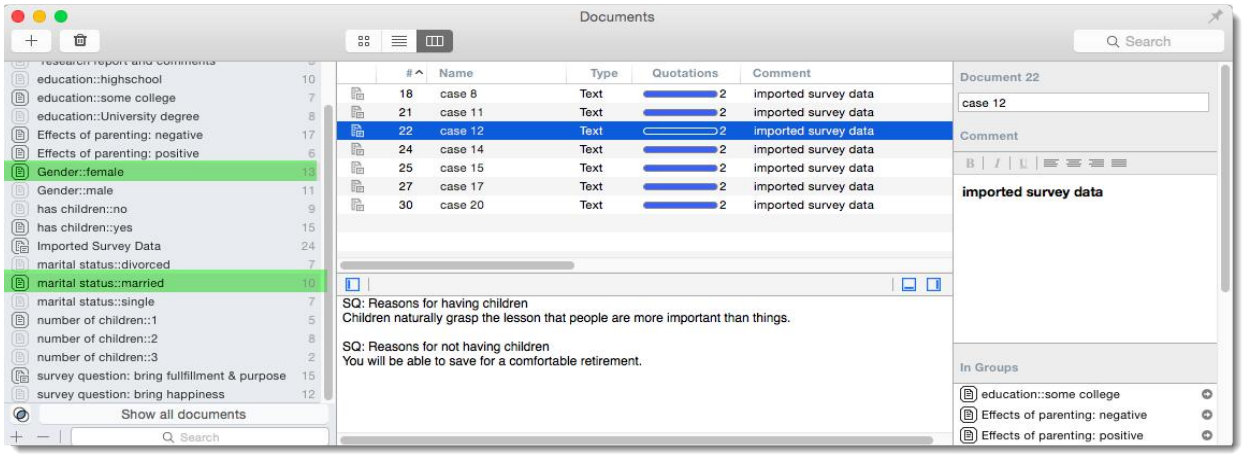


Figure 35: Use PD groups as local filter in the PD Manager

The **PD-Group table** function provides a convenient way to import and export variables (feature not yet implemented).

Code groups can be used to loosely group codes that belong together. You can use them to sort, filter and organize your codes in the Code Manager in the navigator. Other than in Network Views where specific relationships between codes need to be defined, in code groups it is not necessary to specify the ways in which codes relate to each other (feature available)

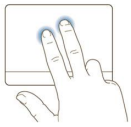
Use **memo groups** to sort, filter, and organize your memos (feature available)

Common Procedures

In the following, the general procedures for working with groups are described.

Groups can be created in both the Group Manager and the navigator of the respective managers. The navigators are better integrated into the regular work-flow. Thus, for daily regular activities it is easier to use the navigators. Currently you need to use the Group Manager to remove items from a family and to describe your families. Later you will also be able to create smart groups in the manager using AND, OR, XOR and NOT as operators. The navigators offer AND and OR combinations only.

Creating Groups In The Navigator



In case you are familiar with the Windows version of ATLAS.ti, groups are the equivalent to "families." Groups can be used for sorting and organizing purposes, and currently to set local filters. This means the filter only applies to the currently open window and not to the entire project. Global filter options will be added in the future.

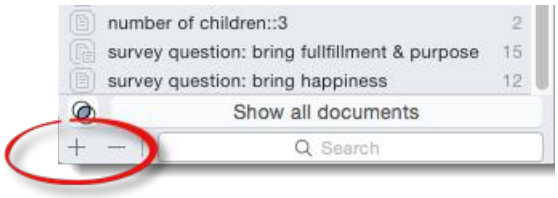


Figure 36: Adding a group in the navigator

- Open one of the Managers. Click on the plus sign at the bottom of the Groups pane or right-click inside the Groups pane and select **New Group**. Enter a group name.
- Add members to the group via drag and drop from the list of documents / codes / or memos in the respective manager.

Filtering By Group In A Manager

- If you click on a group, the items in the manager are filtered and only the members of the selected group are listed (see Figure 35).
- Click on **Show all (Object Name)** to reset the filter.

To Remove An Item From A Group

- Open the Group Manager (see below).

The Group Manager

All group managers currently offers the following options: You can create new groups, remove and add members, search for groups, rename groups, write descriptive comments, review creation and modification dates-

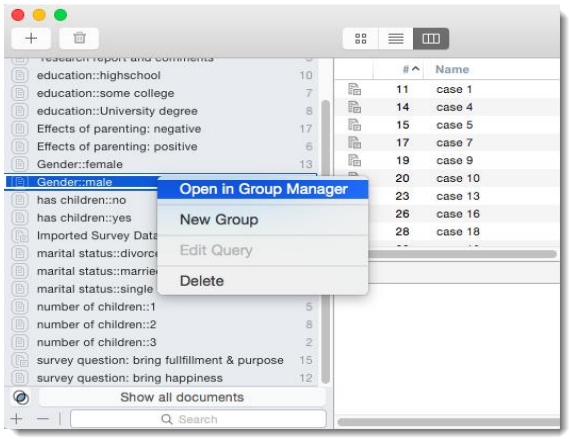


Figure 37: Group Manager

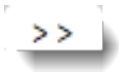
Opening A Group Manager

In the main menu, under Documents, Codes, and Memos you find an option to open the respective group manager. You can also open a Group Manager via a context menu in any of the object managers, or simply double-click on a group to open the Group Manager.

Adding And Removing Group Members

Select one or more objects from the list "*Not in Group*" and click on the button with the double arrows pointing to the left "*"In Group." Or double-click each item.*

Select one or more objects from the list "*In Group*" and click on the button with the double arrows pointing to the right "*Not in Group."* Or double-click each item.



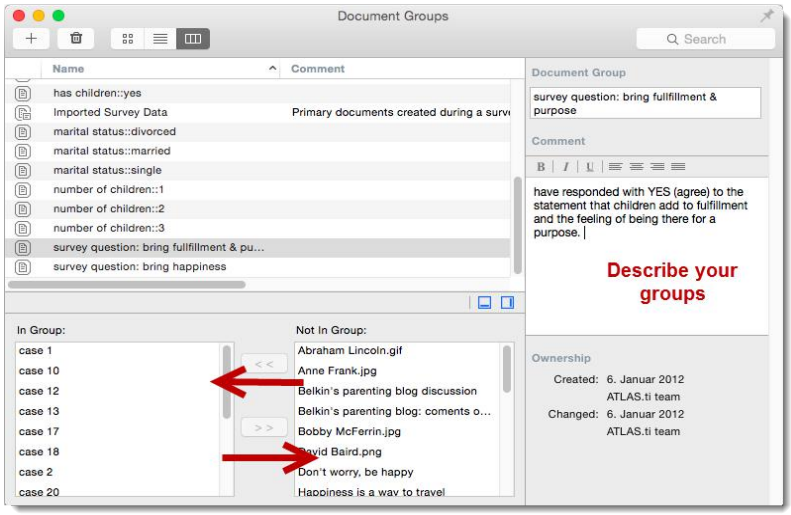
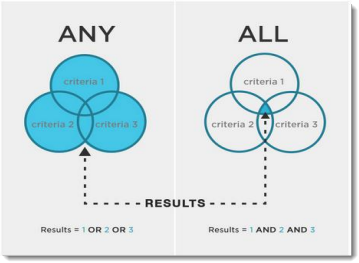


Figure 38: Adding and removing an item to / from a group

Working With Smart Groups



Smart groups are combination of other groups. They can also contain other smart groups. You can build complex groups incrementally from existing groups using a set of powerful Boolean operators. You can either generate temporary groups using the Venn diagram in the navigation panel, or you can create and store smart groups in the Smart Group Editor.

Creating Temporary Smart Groups

You can create new groups based on AND and OR (ALL or ANY) operations using the Venn diagram in the navigation pane for groups. For instance, as shown in Figure 39 you can select two code groups by holding down the ***cmd*** key. If you want to display all codes from both groups, set the Venn diagram to OR (ANY).

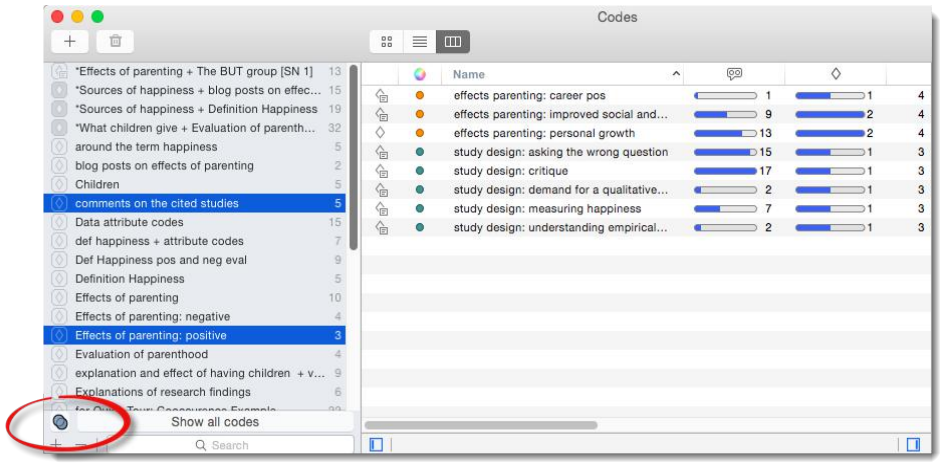


Figure 39: Displaying two code groups using OR (All)

If you are interested in the intersection of two groups, use AND (ALL). You can switch between ANY and ALL by clicking on the Venn diagram.

Another way to create smart groups is via the smart group editor.

The Smart Group Editor

Open the smart group tool via a right-click inside the group navigation panel. Select the option **New Smart Group**.

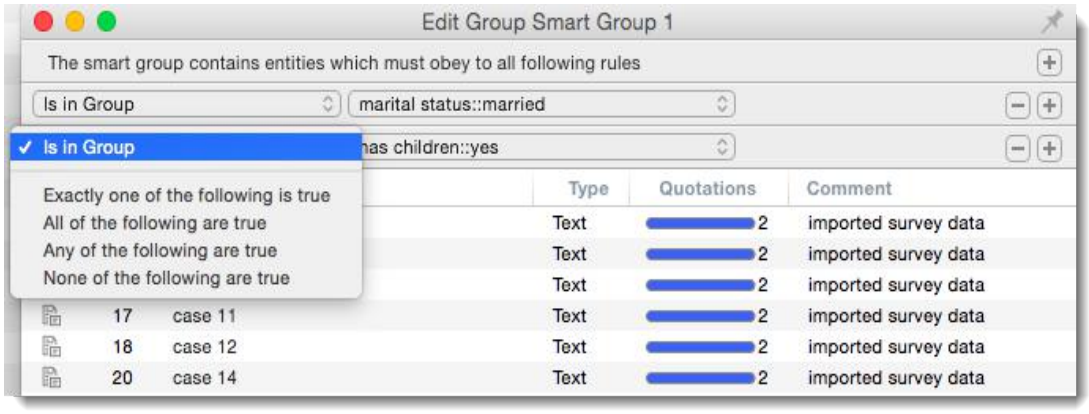


Figure 40: Smart Group Editor

In Figure 40, the smart PD group contains all married respondents with children. In order to get this combination, the option "All of the following are true" was selected. This is the equivalent of the AND operator. In the following all four operators are explained and an example query is shown.

Exactly One Of The Following Is True (XOR)

Figure 41: XOR query: Exactly one of the following is true

At least one of…," including the case where ALL conditions match. The XOR operator asks that "EXACTLY one of…" the conditions must meet. It translates into everyday "either-or." Example: you want to group all respondents that have either an engineering degree or an MBA (excluding those respondents who have both an engineering degree and an MBA).

All Of The Following Are True (AND)

The AND operator finds quotations that match ALL the conditions specified in the query. Use the AND operator to find all respondents who have both an engineering degree and an MBA. Or find codes that are both in code group A and B.

Figure 42: AND query: All of the following are true

Any Of The Following Are True (OR)

The OR operator retrieves all objects (i. e., documents) that are added to any of the groups used in the expression. Staying with the above example,

if you combine the document group "engineering degree" and "MBA" using the OR operator, you get all respondents with an engineering degree, all respondents with an MBA, and all respondents who have both an engineering degree and an MBA.

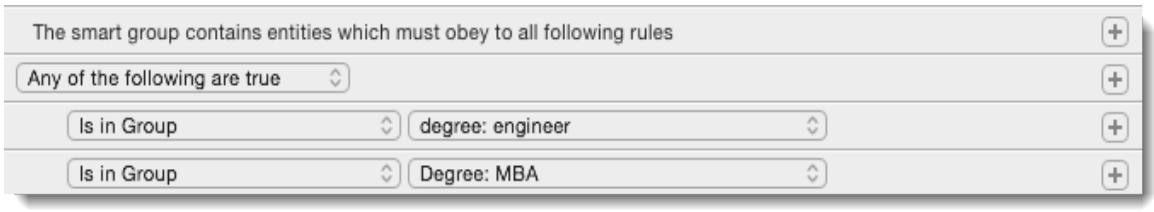


Figure 43: OR query: Any of the following are true

None Of The Following Are True (NOT)

The NOT operator tests for the absence of a condition. Technically, it subtracts the findings of the non-negated term from all other available objects of the same type. If you select "none of the following are true" and the group "engineering degree," you retrieve all respondents that have an MBA or other degrees.

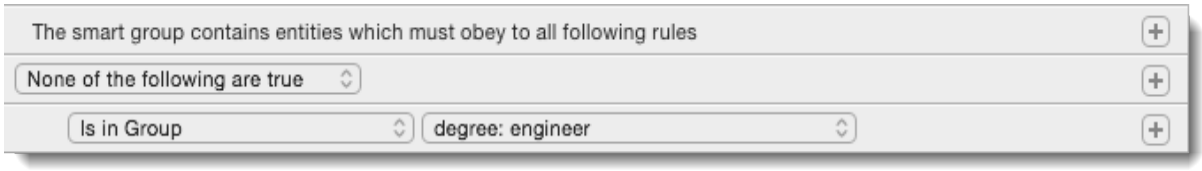


Figure 44: NOT query: None of the following are true

If you want to retrieve those with an MBA that do not have an engineering degree, but possibly others degrees, select:

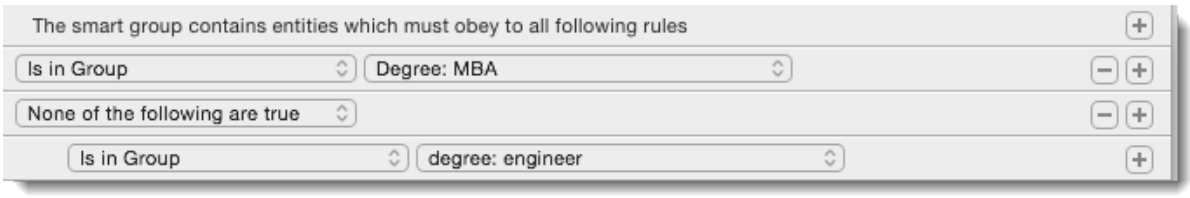


Figure 45: Example query: Those with an MBA, but no engineering degree

If you only want to find respondents with an MBA that do not have other degrees, select:

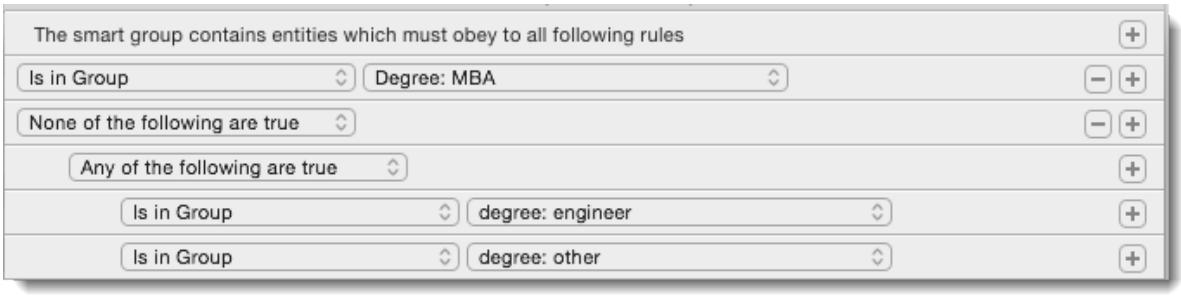


Figure 46: Example query: Respondents with an MBA but no other degree

All smart groups are listed in the navigator together with the other groups. They are automatically named by ATLAS.ti: Smart Group1, Smart Group 2, Smart Group 3 and so on. The group icon is filled in (see left). It is recommended to rename the smart groups after creation in order to remember what they contain.

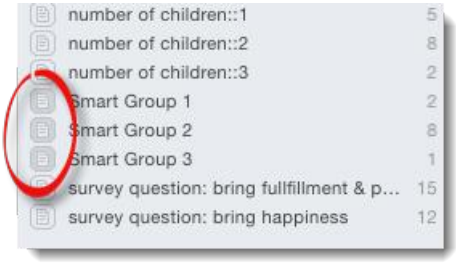


Figure 47: Display of smart groups in navigator

A smart group can be edited at any time. In order to do so, right-click on a smart group and select **EDIT SMART GROUP**. This opens the Smart Group Editor again.

Purpose Of Smart Groups

The purpose of smart groups is to use them as filter in your ongoing analysis. For instance you may want to examine the difference in attitude among respondents who have an engineering degree only, an MBA only and both an engineering degree and and MBA. Via smart groups you prepare the filter you need and combine them with a code query, or use them to prepare a Codes-Primary Documents-Table.

Currently you can only filter the list of objects in the manager using smart groups. However, more advanced analysis options will be implemented in the coming months (spring 2015).

Working With Hyperlinks

Pure classification is not always sufficient.

A network with text (or other media) as nodes is often referred to as a *hypertext*. The original sequential text is de-linearized, broken down into pieces that are then reconnected, making it possible to traverse from one piece of data to another piece of data regardless of their original positions.

The procedures described so far have focused on working with codes. Direct linking of data segments (quotations) to other data segments offers similar flexibility in choosing and defining relations.

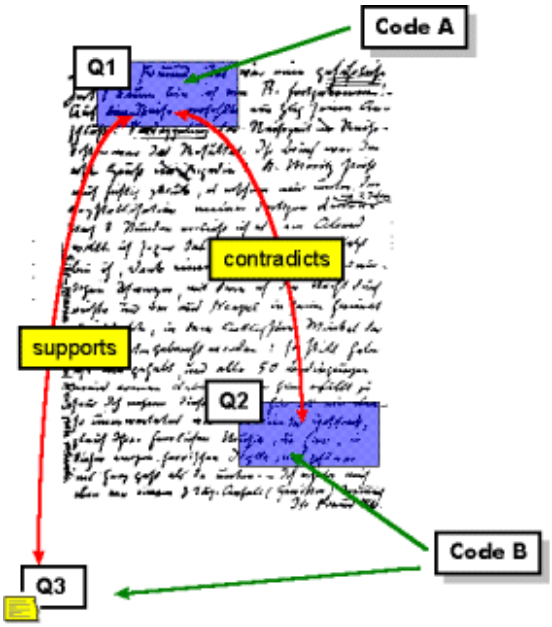


Figure 48: Hypertext captures the macro-structure of a text

What Codes Cannot Do

Maybe we should ask a different question first: How can you express that statement X in text A *contradicts* statement Y in text B, or how can you retrieve all contradictory statements of a specific utterance if all you have is codes and their associations with the data?

The "code & retrieve" paradigm, which is so prevalent for many systems supporting the qualitative researcher, is not adequate for certain types of analysis. In formal terms, attaching codes to chunks of data creates named sets of segments with almost no internal structure. This is not to say that partitioning lots of text segments into sets is not useful. On the contrary, classification leads to manageable amounts of segments that later can be retrieved with the help of the attached code words. But this may not be the only way you want to look at your data.

The concept of a hypertext introduces explicit relations between passages. These links have to be built manually and result from an intellectual effort. The system cannot decide for you that segment x is in contradiction to segment y. But after the work of establishing the links, you can make

semantically richer retrievals: "Show all statements that are contradictory of statement x."

Hypertext allows you to create different paths through the data you are analyzing. For example, you may create a time line different from the strict sequence of the original text. The concept of hypertext introduces explicit relations between passages. While a code offers fast access to sets of data segments, it defines only a simple relation between them, namely **equivalence**.

Hyperlinks, which directly relate data segments, express more differentiated relationships between quotations: contradiction, support, illustration, etc.

No code is needed to connect quotation Q1 with one that it contradicts (Q2).

Today's largest hypertext system: the World Wide Web

Cross-references between text passages are very common even in conventional media like books - just think of religious and juridical texts, literature, journals etc. Footnotes and end notes are another common deviation from the pure linearity of sequential text. However, in conventional media, not much navigational support is provided for "traversing" between the pieces of data that reference each other.

Computer-related hypertext applications include, for example, online help systems that display operational information in suitable small chunks (compared to lengthy printed information), but with a considerable amount of linkage to other pieces of information. A well-known hypermedia structure is the **World Wide Web** with its textual, graphical and other multimedia information distributed world-wide.

Graphical Hyperlink Maps

Hypertext makes text-text relations explicit

ATLAS.ti incorporates procedures for creating and browsing hypertext structures. It allows for two or more quotations being connected using *named* relations. Further, you can create graphical maps (using Network Views) to make parts of your hyperspace accessible in a comfortable way. Hyperlinks may connect quotations (textual, graphical, multimedia) across documents (inter-textual links) or may link segments within the same primary document (inter-textual links). The natural boundary for hyperlinks, like all structures in ATLAS.ti, is your project. The figure below shows a network views based on a number of hyperlinks and a few codes. Other node types can also be included in the Network View, like the memo in the upper left corner.

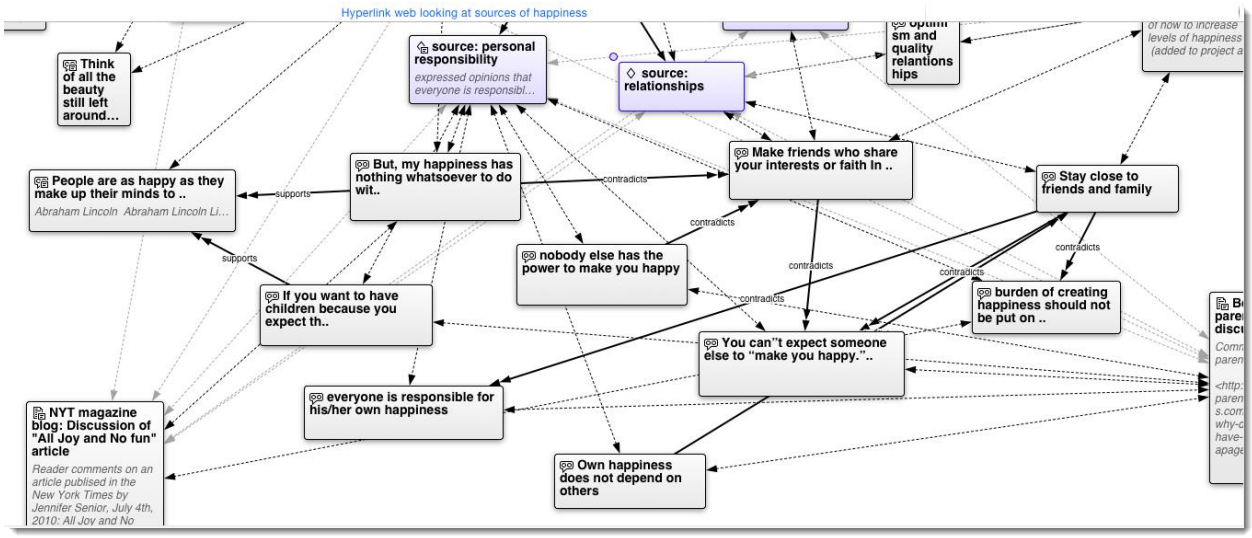


Figure 49: Example of a hyperlink network including some code and document nodes

Creating Hyperlinks

Creating Hyperlinks In The Quotation Manager

To link two quotations to each other, simply drag one quotation (srouce) and drop it on top of another quotation (target) – either in the Quotation Manager or in the margin area.

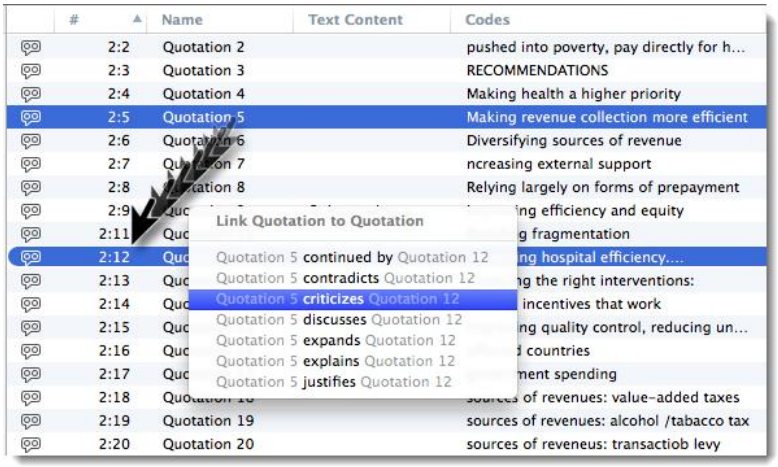


Figure 50: Linking two quotations to each other in the Quotation Manager

Next, select one of the offered relations. If the offered relations do not suit your needs, you can create new relations in the Relations Manager: From the main menu select: **Quotations / Show Relation Manager**. (see also "Defining New Hyperlink Relations."

Creating Hyperlinks In The Margin Area

The margin is an optimal work space for navigating hyperlinks

Like the method described above, creating hyperlinks in the margin area is best suitable for connecting two quotations that are in close proximity to each other. If they are not, utilize the multi-document view for linking (see "Utilizing The Multi-Document View For Creating Hyperlinks."

- Select a quotation bar in the margin area (the source).
- Hold down the left mouse button and drag the bar onto another quotation bar (the target).
- Release the left mouse button. The Relation menu opens. Select a relation.

Display Of Hyperlinks In The Margin Area

Hyperlínks in the margin area are colored in gray and if a transitive link was selected like "supports," the source link points to the right. And vice versa, the target link points to the left (see Figure 51 below). Symmetric links point in both direction. Whether a link is directed or non-directed can be specified in the Link Manager, see "Defining New Hyperlink Relations."

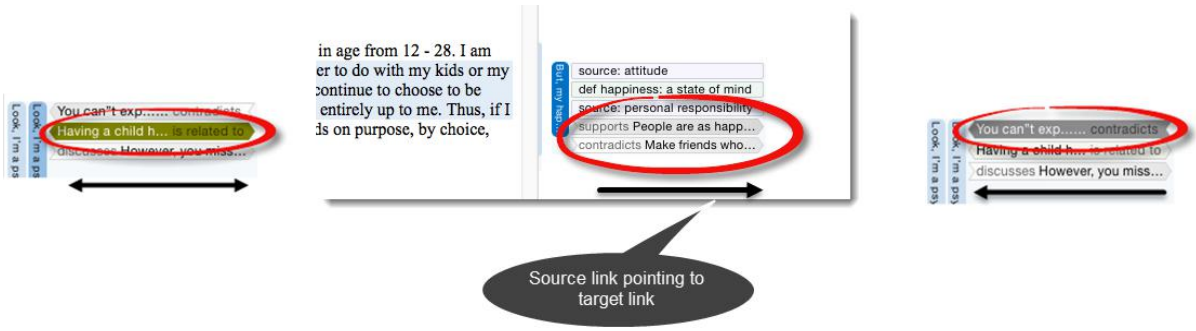


Figure 51: Display of hyperlinks in the margin area

Utilizing The Multi-Document View For Creating Hyperlinks

- Open two documents side-by-side (see "Multiple Document View").
- Grab a quotation bar and drag it across to the document in the second tab and drop it onto another quotation bar.
- Select a relation.

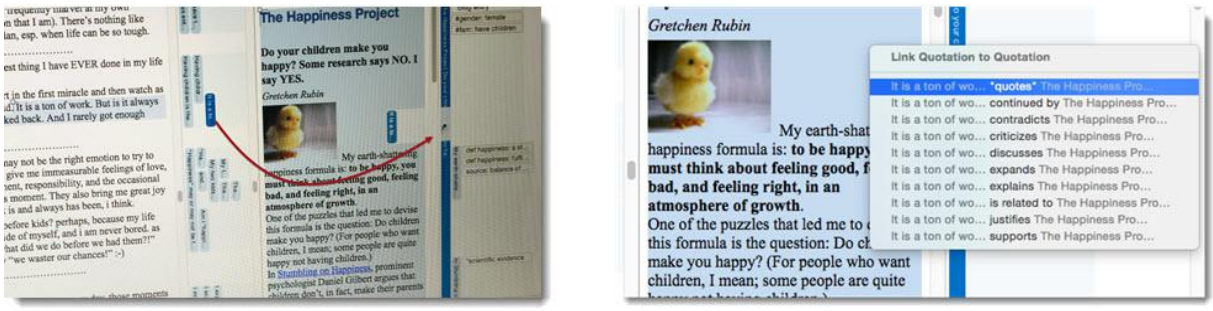


Figure 52: Creating hyperlinks utilizing the multiple-document view

Modifying Hyperlink Relations

Currently, you can only modify the relation type in the Link Manager:

Open the Link Manager via **Quotation / Show Link Manager.**

Figure 53: Modifying hyperlink relations in the Link Manager

In the Relation column, click on a relation and select a the desired relation.

Defining New Hyperlink Relations

The procedure for defining or editing hypertext relations is equivalent to the methods described for editing code-code relations (see also "Working with Network Views").

Open the Relation Manager from the Quotation's main menu: **Quotation / Show Relation Manager**.

Click on the **+** button to add a new relation.

Define the properties for the new relation and write a comment (optional). The various fields are explained in more detail below.

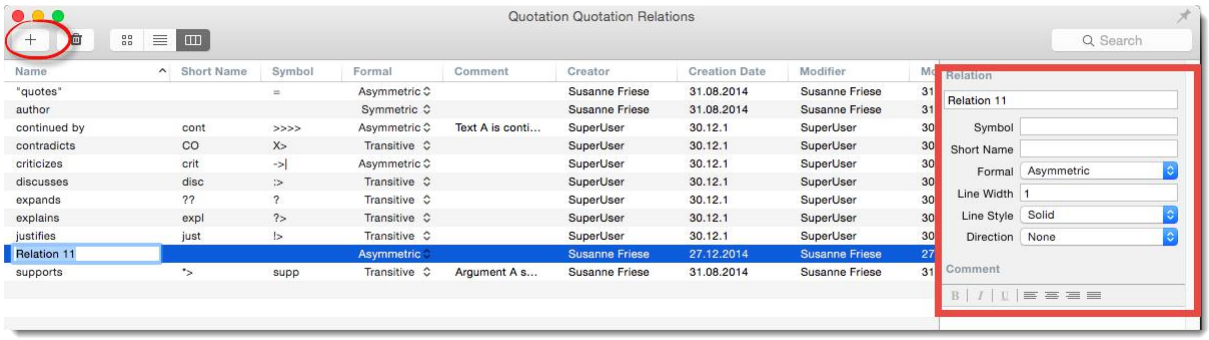


Figure 54: Creating a new relation in the Relation Manager

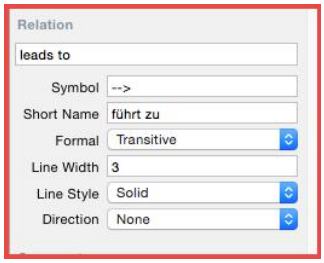


Figure 55: Relation details

Relation Properties

The following can be defined: Cosmetic and descriptive as well as structural aspects of relations.

COSMETICS

Cosmetic aspects include the name and the line. The relation name is shown when creating relations and in a Network View Editor. Once implemented, the symbol and short name can be used as an alternative display option in network views. Instead of a short name, you can for instance also use a different language as shown on the left. Further you can define the width and solidity of the line. Color will be added later.

FORMAL PROPERTY

The formal property associated with a relation has a cosmetic effect and it controls the "procedural semantics" of the semantic operators in the Query Tool. When you want to utilize the semantic operators (children / parent), transitive relations need to be used.

PREFERRED LAYOUT DIRECTION

A more sophisticated "cosmetic" property is the preferred layout direction. By using this relation characteristic, the user can assert some control on the automatic layout algorithm. Indeed, this option justifies the name "semantic layout."

COMMENT

As with all entities in ATLAS.ti, a comment can be attached. The text entered as a relation comment is displayed within a Network Editor after opening a context menu on a link, or when selecting **Display Relation**. (not yet implemented)

Bear in mind that a comment written for a *relation* is different from a comment written for a *link*. The comment for a relation is of a global nature and defines the relation type: e. g., what is meant by the relation "is associated with." A *link* using this relation connects two specific quotations. When writing a comment for this link, the meaning is local and explains why two quotations were connected using this relation.

Traversing Hyperlinks

The margin area permits a very intuitive traversal of hyperlinks.

For a better view, open up the context menu in the margin area and only select **SHOW HYPERLINKS**. Deactivate all other options.

Double-click a hyperlink in the margin to jump to the linked quotation.

Margin Drag & Drop

All objects populating the margin area (i. e., "margin objects") support drag & drop. The bar visualize quotations segmenting the document, and - depending on the context - the objects attached to the bars represent themselves or the link with the quotation.

The effect of a drag & drop operation depends on the objects that are involved as drag sources (those that are dragged) and targets (those onto which objects are dropped). A large variety of objects from the margin area can be dropped into the margin area. Furthermore, objects can also be dragged from other object managers and browsers.

Move Linked Objects

When a code, memo, or a quotation is dropped on a quotation bar, a new link is created between the object and the quotation represented by the bar. You can for instance drag-and-drop quotations, codes and memos from the navigator or any of the managers.

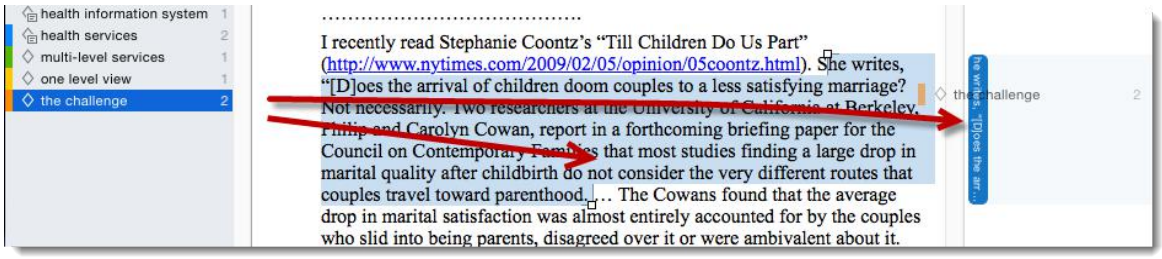


Figure 56: Coding via drag and drop from the navigator

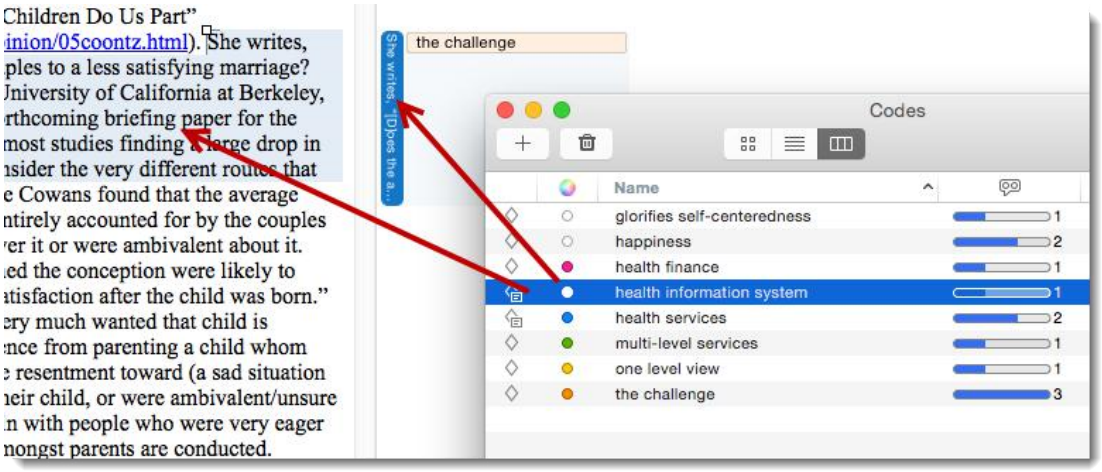


Figure 57: Coding via drag & drop from the Code Manager

Copy Linked Objects

An object in the margin is unlinked from its original quotation when it is dropped on another quotation bar. To keep the object from unlinking from its original place, hold down the **alt**-key when dropping. This resembles dragging objects from managers and browsers into the margin area, which does not change existing links.

You will see a green button with a plus sign if an object is copied and not moved.

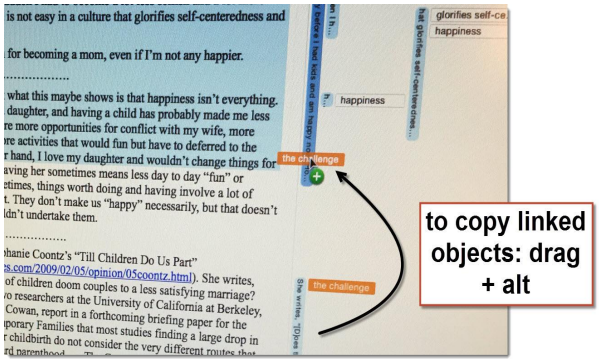


Figure 58: When copying linked objects, hold down the alt key

Linking Quotations

Dragging a quotation bar onto another quotation bar creates a hyperlink between the two quotations.

Open two documents side-by-side (see "Multiple Document View").

Drag the quotation bar from one document onto the quotation bar in another document. A list of relations opens to chose from. Select one of the offered relations. If none of the existing relations adequately reflects the relation between the two quotations, you can create new relations in the relation manager (see "Defining New Hyperlink Relations"). After selecting a relation the two quotations are linked and the hyperlink is displayed in the margin area (see Figure 51).

Figure 59: Creating a hyperlink across documents when opening two documents side-by-side

Working With Network Views

ATLAS.ti allows you to establish *named links* to more clearly express the nature of the relationships between concepts. With named links, you may express a sentence like "a broken leg causes pain" by two nodes (the source node "broken leg" and the target node "pain") connected with a named link ("causes" or "is-cause-of").

A default set of relations is provided. You may define your own.

The name of a link is displayed in the Network Editor as a label attached to the link midway between the two connected nodes. Six pre-set relations - or link types - are available in ATLAS.ti. These standard relations can be substituted, modified, or supplemented by user-defined relations. The default relations are listed in the table below. C1 and C2 are source and target nodes, respectively.

Relation	Label 1	Label 2	Width	Color	Formal Attribute	Layout Direction
C1 is-associated-with C2	==	R	1	Black	Symmetric	✳
C1 is-part-of C2	[]	G	1	Black	Transitive	✳
C1 is-cause-of C2	=>	N	1	Black	Transitive	✳
C1 contradicts C2	<>	A	1	Black	Symmetric	✳
C1 is-a C2	Isa	0	2	Black	Transitive	✳
C1 noname C2			1	Black	Symmetric	✳
C1 is-property-of C2	*}	P	1	Black	Asymmetric	✳

Some of these characteristics directly affect the display of links, while others affect processing (e. g., search routines, automatic layout). A link between concepts is displayed in a Network Editor by a line with the relation's label. You can choose from three different labels: relation name, symbol and a short name.

The "formal attribute" affects both the display and processing capabilities of a relation. For example: All asymmetric relations are symbolized in the Network Editor with an arrow pointing toward the target code. Symmetric relations are displayed with an arrow at both ends.

A typical transitive relation is the is-cause-of relation: if C1 is-cause-of C2 and C2 is-cause-of C3, it follows that C1 is-cause-of C3. Transitive relations also enable the "semantic retrieval" based on parent and child nodes (see "Semantic Operators."

The following properties are user-definable: the three labels, which can be used as alternative display options, the width and color of the line linking two nodes (not yet implemented), whether a link is directed or non-directed, and the preferred layout direction. The preferred layout direction affects the layout of a network when ATLAS.ti automatically arranges the nodes.

Relations are like "styles" in a word processor. Changing the style will change all occurrences of its usage, in this case: the links

Link Vs. Relation

It is important to understand the difference between a relation (or a link type) and the link itself: There is only one "is part of" relation, but potentially many links *using* it. In the Network View below, the relation "consequence" is used only once, while the relation "strategy" is used four times.

Another way to think of links and relations is to view links as *instances* of relations. Links are well informed about the characteristics of relations, which define their styles. If a characteristic of a relation is changed (e. g., line width, color, symbol), these changes are propagated to all links using it.

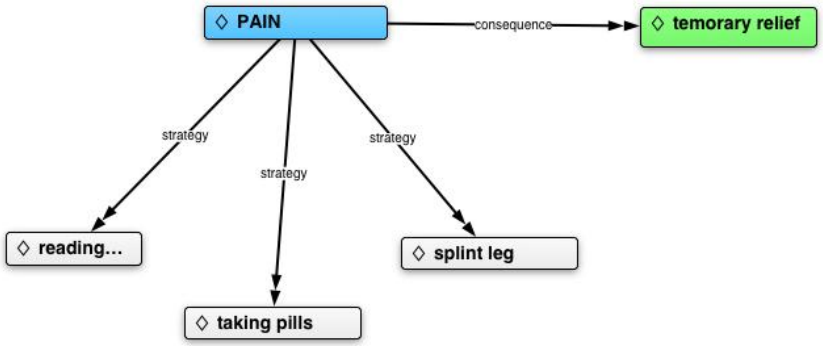


Figure 60: Four links using two relations, "strategy" and "consequence"

The Role Of Relations

It is useful to understand the role that relations play in the construction of a theory. The concepts (codes) that are linked using relations represent aspects of the problem domain under investigation. On the other hand, the relations used to link these domain concepts are part of the methodology used to analyze the phenomena. As important epistemological tools they constitute the main questions that guide the development of a model or a theory.

Define your own "epistemological primitives."

The "Grounded Theory" method of Glaser & Strauss uses relations like "is-phenomenon, "is-context-of," "is-consequence-of," "is-condition-for," "is-strategy-for," etc., to relate concepts found during the data-oriented *open coding* phase.

In the analysis of argumentation structures, other relations are more suitable: e. g., "is evidence of," "is contradictory to," "warrants," etc. A medical expert attempting to capture diagnostic knowledge would use, e. g., "is-symptom-of," and "is medication for."

The Network Editor

The Network Editor lets you visually connect codes and other objects to create semantic networks or hypertext webs.

The Network Editor offers an intuitive and powerful method to create and manipulate network structures. It favors a direct manipulation technique: You can literally "grab" codes, quotations, memos, or other objects using your cursor and move them around the screen as well as draw and cut links between them.

The following describes various methods available for creating and editing Network Views.

Network View Characteristics

Network Views have certain important characteristics:

- Several different Network Views on the same network are possible.
- Network Views can be given names under which they are stored and accessed inside the HU.

- Network Views can be commented.
- Network Views are displayed and edited in the Network Editor.
- Network Views allow individual layout of the nodes.
- As a node, a single object can be a member of any number of Network Views, just like a code can be an element of more than one code group.
- An object, e. g., a specific code, can only appear once in any Network View.

Network Views allow for a flexible but logically consistent display of the network of objects, so there are a few constraints to keep in mind:

If code A is linked to code B using the relation "is associated with," then every Network View that contains code A and code B will necessarily include the relation"is associated with" between the two. Furthermore, as only one link can exist between any two nodes at any given time, no Network View will display any other relation between those two nodes.

If however, you want to link code A and B differently in a different network view, then you need to work with "dummy or modifier codes." These are empty codes, i. e. you have not used them for coding, but you need them to modify a relation.

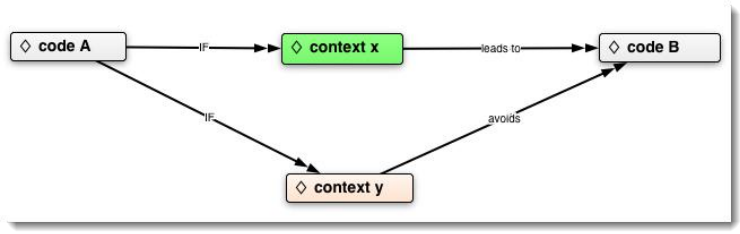


Figure 61: Using modifier codes for case-based network views

Node Types

The following object classes can be displayed and edited as nodes within the Network Editor. The display characteristics of the nodes can be altered in a variety of ways.

Codes As Nodes

Codes are probably the most prominent objects in ATLAS.ti networks. They provide the main ingredients for models and theories.

Memos As Nodes

Memos in networks are often an important supplement to code networks.Several theoretical memos can be imported into a network to map out their relationship. The visual layout provides comfortable territory for moving from memo to memo to read and contemplate each individually and the relationship(s) between them.

Primary Documents As Nodes

PDs as nodes are useful sometimes, but in the presence of quotations may clutter the view by myriad links. However, PDs as nodes make a nice graphical content table for graphical primary documents. When selecting the option **Display/Full Image for PDs**, "thumbnail" images of the PDs are displayed. (This feature is not yet implemented.)

Quotations As Nodes

Quotations and codes have one thing in common that is not true for the other objects. They can link to each other (quotations to quotations and code to codes) with fully qualified "first class" links using relations.

The inclusion of quotations in a Network View supports the construction and inspection of hyperlink structures.

Network Views As Nodes

Network Views as nodes allow the inclusion of Network Views in other Network Views. To open the network view in a new tab, right-click and select **Open Network**.

Groups As Nodes

Groups are a useful device to group codes, memos, or PDs. Instead of displaying all of the codes belonging to the concept, the Code group may be displayed. Or you display the group with all of its members.

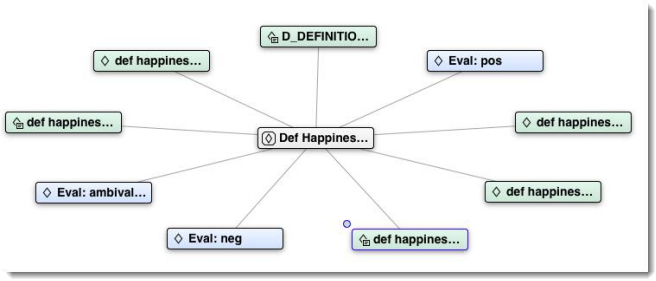


Figure 62: Display of a code group with its members in a network view

Creating A New Network View

To create a new network view, select **Network / New Network**. This opens a tab with now content. The name network 1 / network 2 / network 3, etc is generated automatically. If you decide that you want to keep a particular network view, you can rename it later in the Network View Manager (**Network / Show Network Manager**).

In the network view manager you can enter a comment for each network to describe it.

Drag and drop items from the Managers, margin area, or the navigator into the view.

The node types (document, quotation, code, memo, network view) can be recognized by their entity specific icon (compare Figure 4). If you right-click on a node, you see further options that are available (Figure 63).

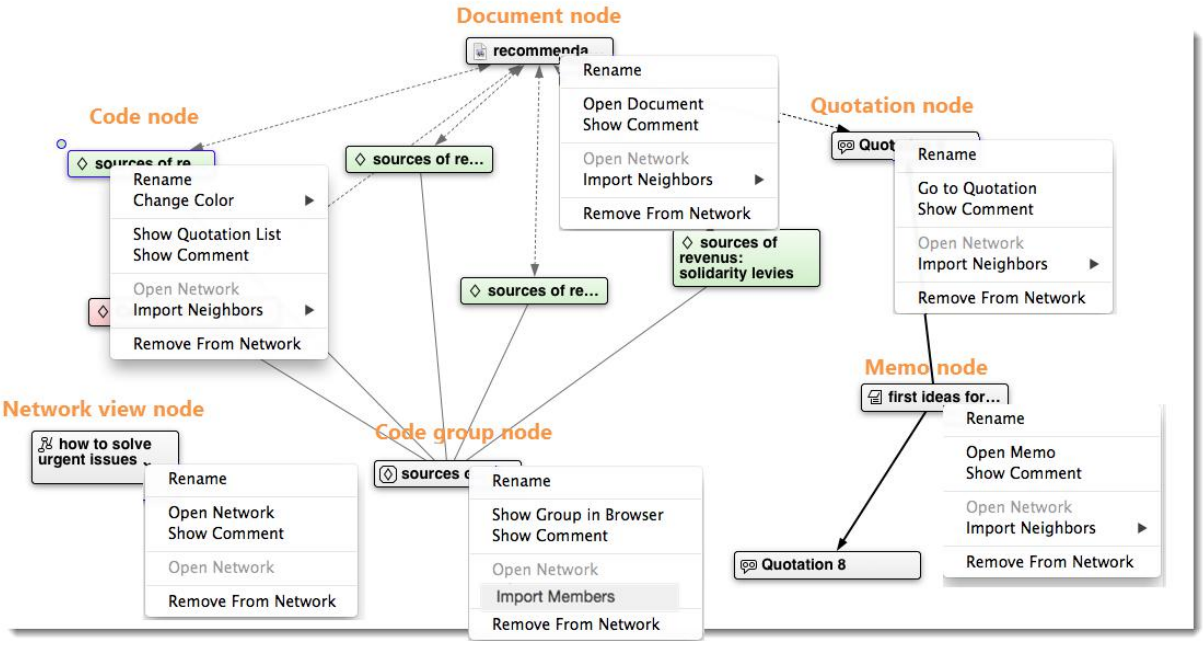


Figure 63: Node Types and available options

Linking Nodes In Network Views

The links between nodes in a network are real connections between the objects. Therefore, creating and removing links should not be regarded as solely "cosmetic" operations. Links make permanent changes to your entire project.

Strong Links (First Class)

You can link almost all ATLAS.ti objects to each other. Qualified named relations can however only be created between two codes and between two quotations, the so called hyperlinks (see also "Working with Hyperlinks").

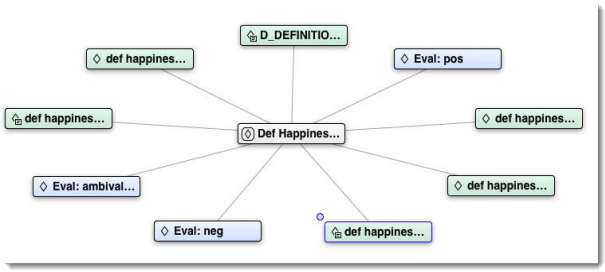


Figure 65: Network View displaying weak links

Weak Links (Second Class)

Other objects can be linked, but you cannot specify the relation between them. Therefore they are called "weak links" or second class links in ATLAS.ti These are:

- Code – memo links

- Code – quotation links (this is the same as coding)
- Memo – memo links
- Memo – quotation links
- Groups and their members (see Figure 62 above and left).
-
- Documents and contained quotations.

Document-quotation links cannot directly be created by the user, they can only be displayed. It is possible to display the quotations that are contained within a document, via the importing neighbor option (see below), but you cannot link a quotation to a document in a network view. It would not make much sense. e.g. to link a quotation from document 5 to document 3, as it does not belong to it.

Creating Strong Links

To link for instance two codes to each other, select a code node. A circle appears at the top left corner of the node. If you begin to drag starting from the circle, a line appears. Drop the line on top of another code node. A list of relation opens. Select the one that best fits the relation between the two codes.

Another way to link either two quotations to each other or two codes is via drag and drop in the respective managers.

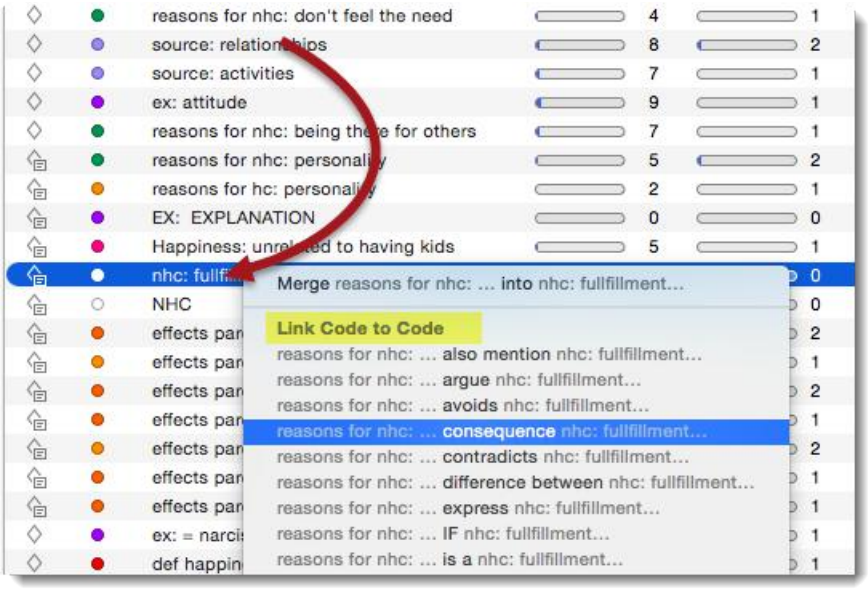


Figure 66: Code to code link

To create new relations, open the Relation Manager (see "Defining New Hyperlink Relations" and "Defining New Code-Code Relations."

Creating Weak Links

To link for instance a codes to a memo, select a code node. A circle appears at the top left corner of the node. As you begin to drag, an

arrow appears. Drop the arrow on top of the memo node an drop the line.

Linking Two Nodes Using Mouse And Keyboard

A quick way to link two nodes.

- Place the mouse pointer over the source node.
- Hold down the ALT key on your keyboard.
- Hold down the left mouse button and drag the mouse pointer to the target node.
- Release the left mouse button and the ALT key.
- In case you create a strong link (code-code or quote-quote link), select a relation.

Selecting Links

To select a link or a relation, simply click on the link. It turns blue. To open the context menu to either delete or flip the link, right-click.

Modifying Links

The type of a link (e. g., its relation) can currently only be changed in the Link Manager. Further options will follow.

- Open the appropriate Link Manager via QUOTATION / SHOW LINK MANAGER or CODE / SHOW LINK MANAGER
- In the Relation column, click on a relation and select a different relation (compare Figure 53).

Selecting Multiple Or All Nodes

To select all or a selected group of nodes, you can either draw a frame around all nodes with your mouse or hold down the *cmd* key and click on each node that you want to select.

Moving Nodes

By moving nodes to different positions, you can modify an initial layout created by the automatic layout procedure. To move a single node, just drag it with the mouse to the desired postion.

To Move Multiple Nodes

- Create a multiple selection of nodes as described above. Then grab one of the selected nodes with the mouse and move the entire group of selected nodes to a different position.

Removing Nodes From Network Views

Removing nodes from the view simply takes the nodes out of a Network View. The nodes remain in your project. Removed nodes can be "re-imported" at any time.

Select the nodes to be excluded from the view. Right-click and select the option **Remove From Network.**

Defining New Code-Code Relations

The procedure for defining or editing code-code relations is equivalent to the methods described above for defining hyperlink relations (see Defining New Hyperlink Relations).

Open the Relation Manager from the Code's main menu: **Code / Show Relation Manager.**

Click on the **+** button to add a new relation. Define the properties for the new relation and write a comment (optional). The various fields are explained in more detail below.

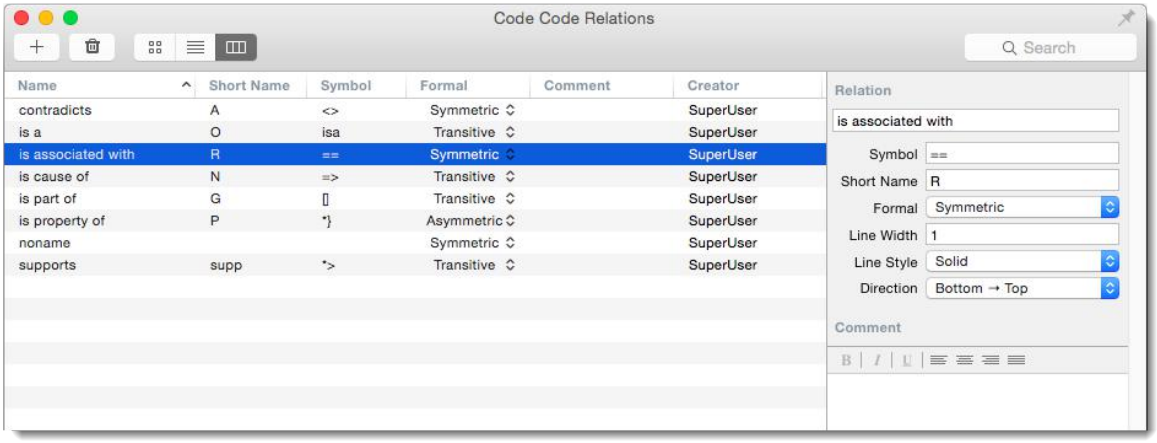


Figure 67: Available default code-code relations

Relation Properties

The following can be defined: Cosmetic and descriptive as well as structural aspects of relations.

COSMETICS

Cosmetic aspects include the name and the line. The relation name is shown when creating relations and in a Network View Editor. Once implemented, the symbol and short name can be used as an alternative display option in network views. Instead of a short name, you can for instance also use a different language as shown on the left. Further you can define the width and solidity of the line. Color will be added later.

FORMAL PROPERTY

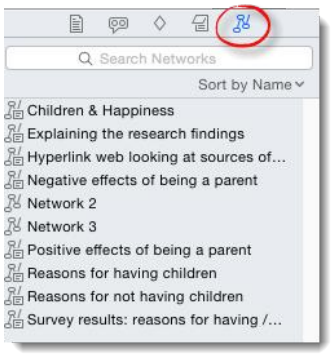
The formal property associated with a relation has a cosmetic effect and it controls the "procedural semantics" of the semantic operators in the Query Tool. When you want to utilize the semantic operators (children / parent), transitive relations need to be used.

PREFERRED LAYOUT DIRECTION

A more sophisticated "cosmetic" property is the preferred layout direction. By using this relation characteristic, the user can assert some control on the

automatic layout algorithm. Indeed, this option justifies the name "semantic layout."

COMMENT



As with all entities in ATLAS.ti, a comment can be attached. The text entered as a relation comment is displayed within a Network Editor after opening a context menu on a link, or when selecting **Display Relation**. (not yet implemented)

Bear in mind that a comment written for a relation is different from a comment written for a *link*. The comment for a relation is of a global nature and defines the relation type: e. g., what is meant by the relation "is associated with." A *link* using this relation connects two specific quotations. When writing a comment for this link, the meaning is local and explains why two quotations were connected using this relation.

Accessing And Opening Existing Network Views

You can access network views via the Network View Manager and via the navigator on the left hand side of your screen.

- Select **Network / Show Network Manager.**
- Open the navigation pane and select the network view tab (see left).

Analytic Functions In Network Views

Import Node Neighbors

This method imports all direct neighbors of the selected nodes into the Network View. This option is currently available from the node's context menu. Importing direct neighbors allows you to construct a *connected* Network View step-by-step. (In a connected graph, there is always a direct or indirect path between any two nodes.)

Import common Neighbors imports all directly linked objects. All other options let you specifically select which objects you want to import. Below the options for the various objects types are listed:

Quotations	Import Common Neighbors Import Codes Import Memos Import Hyperlinks
Documents / Codes / Memos	Import Common Neighbors Import Codes Import Memos Import Quotations Import Groups
Groups	Import Members

- Select the node(s) whose neighbors are to be included in the Network View.

Right-click and select **Import Neighbors** from the context menu and make your choice.

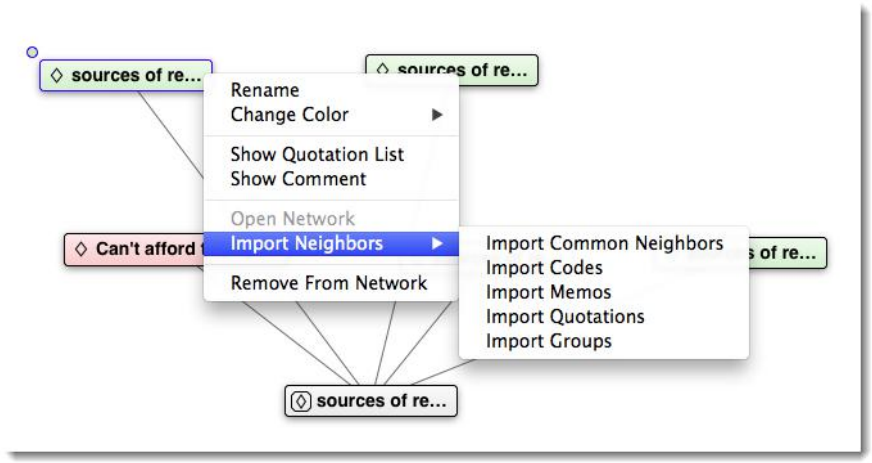


Figure 68: Importing code neighbors

Creating Case-based Network Views

Importing a document's codes neighbors allows to create case-based network views. You can ask questions like: Which of the codes have been applied in which document (and where do they not occur). ATLAS.ti automatically draws light gray dashed lines between codes and the primary documents. This is how you do it:

Create a new network view: **Network / New Network.**

Drag-and-drop two or more primary document nodes into it, e.g. from the navigator.

Right-click on each PD node and select the option **Import Neighbors / Import Codes**. Don't forget to do it for all PD nodes, even if results pop up immediately.

Usually a lot of different codes have been applied to a document. Therefore this option works best, if you apply a global filter for codes. However, global filters have not yet been implemented. The example below compares two respondents from a survey (case 6 and 9). The survey only included two open-ended questions and therefore the number of codes is still comprehensible.

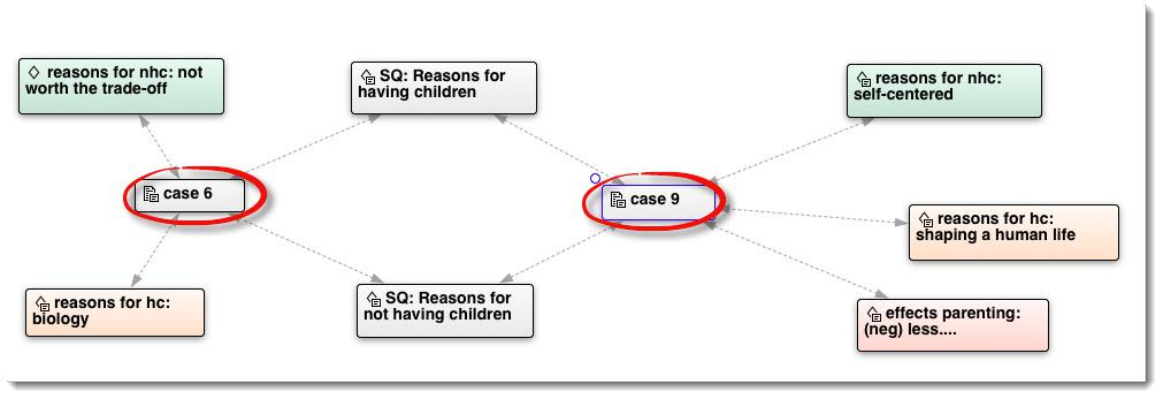


Figure 69: Case-based comparison using the import codes option for Pds.

Import Co-occurring Codes (Not Yet Implemented)

For code nodes, there is a special import feature that exploits the spatial relations of different codings. A code co-occurs with another if it has been used to code quotations that are in close proximity: embedded, overlapping, or if two or more codes are applied to the same quotation.

The proximity of coding applied to a text can also be exploited via the Query Tool's "co-occurrence" proximity operator. However, while the Query Tool yields quotations for explicitly specified codes, the import function brings in only the codes.

Further Analysis Tools

The analysis options in version 1.0.x of ATLAS.ti for Mac are yet not complete, however will be extended continuously. The following options are currently available:

- Simple retrieval by code in the Quotation and Code Manager
- AND and OR queries in the Quotation Manager
- The creation of smart codes offering the full range of operators (Boolean, semantic and proximity operators)
- Code-Cooccurence Table

Not available yet but still to come:

- Reports based on queries
- Full query tool, including the scope option, i. e. code queries in combination with variables (= PD groups)
- Codes-Primary Documents-Table

Simple Retrieval In The Code Or Quotation Manager

Select a code in the Code Manager. If not opened, open the inspector on the right-hand side of the Manager. All linked quotations are listed in the bottom pane of the inspector.

Click on the right-arrow to view each quotation in context.

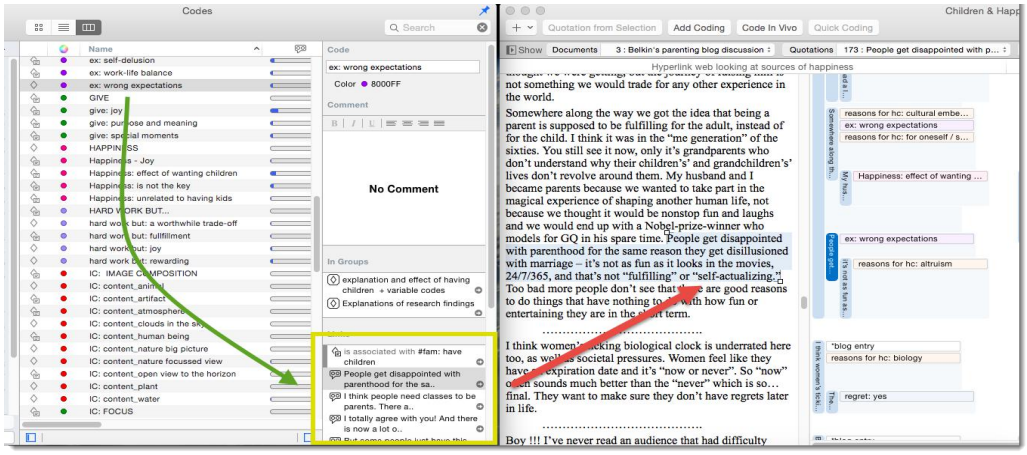


Figure 70: Simple Retrieval, access of quotations via Link section in inspector

OR

Open the Code Manager and double-click on a code. The list of quotations coded by this code opens. Click on a quotation to view it full length and in context.

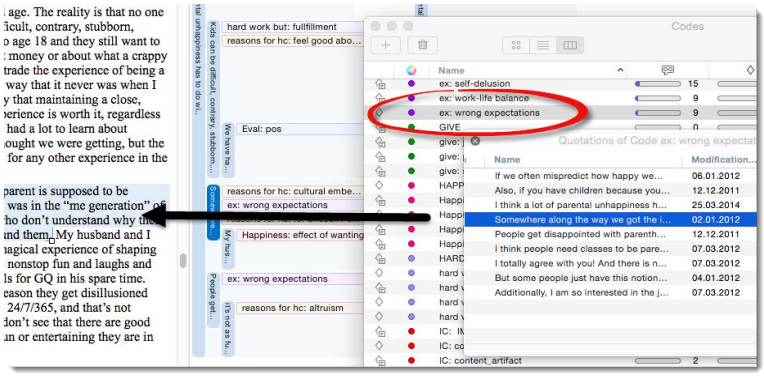


Figure 71: Simple Retrieval via double click on a code

OR

Open the Quotation Manager

Select a code in the navigator on the left-hand side. This filters the list of quotations and only those quotations linked to the selected code are shown. Click on each quotation to view its content in the preview pane at the bottom.

Do view the quotations in context, make sure that the Quotation Manager stays on top by selecting the pin (top right). Double-click on each quotation to view it in context.

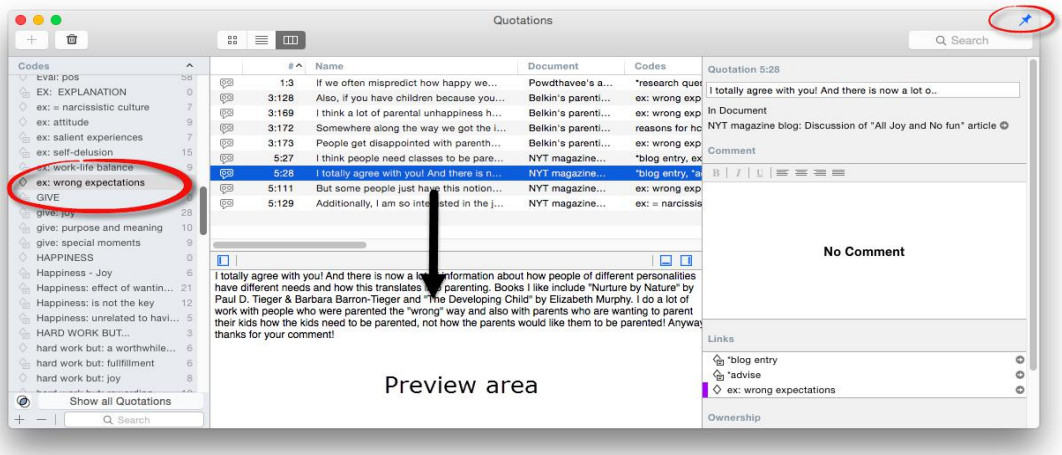


Figure 72: Simple retrieval in the Quotation Manager making use of the preview pane

Simple AND (All) / OR (Any) Queries In The Quotation Manager

The simple retrieval option in the Quotation Manager can be extended to include also more than one code:

Open the Quotation Manager. At the bottom of the screen you see a VENN diagram. You can set it to AND (= All must apply) or OR (any selected apply).

Select two or more codes holding down the ***cmd*** key.

Figure 73 shows the result of an example AND query. The two codes '#fam: have children' and '#fam:t-teenagers' have been selected holding down the ***cmd*** key. The quotation list displays the resulting 6 quotations that contain all comments written by parents of teenage kids.

The quotation content is displayed in the bottom pane.

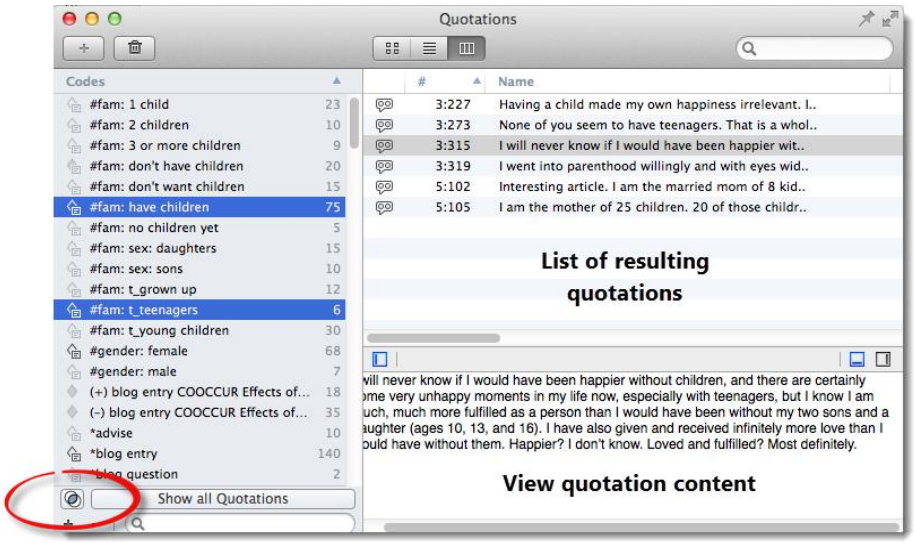


Figure 73: Running an AND query in the Quotation Manager

Working With Smart Codes

"Intelligent" Smart Codes compute their quotations "on demand."

Smart codes are a convenient way to store queries. They are very similar in look and feel to normal codes, with one important difference: Instead of "hardwired" connections to quotations, smart codes store a query to compute their virtual references whenever needed. They "automatically" change their behavior during the course of theory building. If you have a smart code based on a query like

```
(Code A | Code B) COOCCUR Code C
```

and you add or delete quotations linked to either Code A, B or C, then the quotations linked to the smart code will automatically be adjusted.

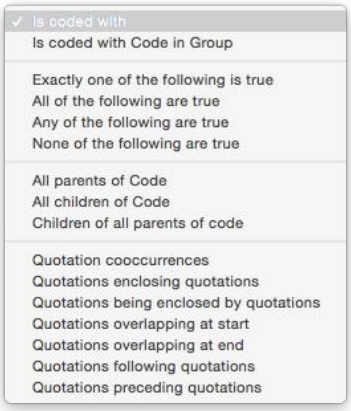


Figure 74: List of available options to build a smart code

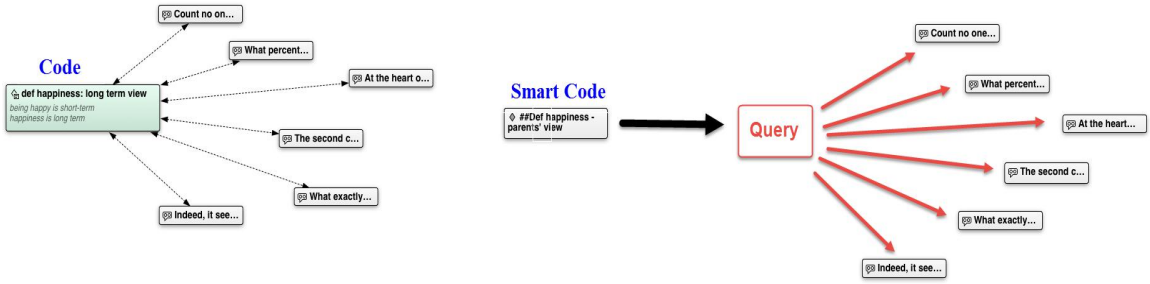


Figure 75: Difference between a code and smart code: Codes are directly linked to quotations; smart codes are a stored query

Smart Codes can be selected in a code list (either in the Code Manager or the navigators) like any other code and they will display their quotations in an identical way.

Smart codes are displayed in the Code Manager and in the Navigator just like regular codes and can be recognized by the filled icon (see left). The list of quotations associated with a smart code can be displayed with a double-click, just as for any other code.

Smart Codes are *not* displayed in the margin area and as they are stored queries they can *not* be used for coding.

Smart Codes can however be added to code groups, to Network Views, and, last but not least, as powerful operands in other queries, allowing you to incrementally build complex queries.

Creating Smart Codes

To create a new smart code, select **Code / New Smart** Code from the main menu. Enter a name for your smart code and click **Create**.

This opens the smart code editor where you can define the conditions for your smart codes. Below, the available operators are explained in detail. This is followed by a description of a few example queries. To the left you see the menu that lists the various options you have to build a smart code. You can build a smart code based on codes or code groups, or on a combination of both (first section)

Operators

You find three sets of operators in the following sections:

Boolean operators allow combinations of keywords according to set operations. They are the most common operators used in information retrieval systems.

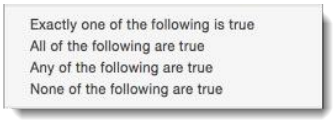
Semantic operators exploit the network structures that were built from the codes.

Proximity operators are used to analyze the spatial relations (e. g., distance, embeddedness, overlapping, co-occurrence) between coded data segments.

BOOLEAN OPERATORS

Four Boolean operators are available: XOR (exactly one), AND (All), OR (any) and NOT (none).

OR, XOR, and AND are *binary operators* which need exactly two operands as input. NOT needs only one operand. However, the operands themselves may be of arbitrary complexity. Codes, code groups, or other smart codes can be used as operands: "(A OR B) AND (NOT C AND D)."



XOR (Exactly one of the following is true): The XOR operator asks that "EXACTLY one of…" the conditions must meet. It translates into everyday "either-or." Example: "All quotations coded with EITHER 'Earth' OR 'Fire' (but not with both)."

AND (All of the following are true): The AND operator finds quotations that match ALL the conditions specified in the query. This means you have applied two or more codes to the same quotation. Example: "All quotations coded with 'Earth' AND 'Fire'." The AND operator is very selective and often produces an empty result set. "Precision" of this operator is high, but the "recall" is rather low. It produces best results when combined with less restrictive operators or when the overall number of the available text segments is large.

OR (ANY of the following are true): The OR operator does not really match the everyday usage of "OR." Its meaning is "At least one of…," including the case where ALL conditions match. The OR operator retrieves all data segments (i. e., quotations) that are coded with any of the codes used in the expression. Example: "All quotations coded with 'Earth' OR 'Fire'."

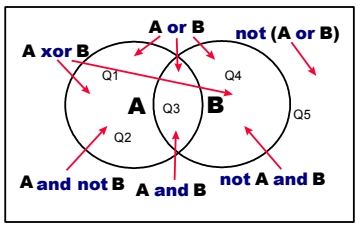


Figure 76: Boolean queries depicted as Venn diagrams

NOT (None of the following are true): The NOT operator tests for the absence of a condition. Technically, it subtracts the findings of the non-negated term from all data segments available. Given 120 quotations in the HU and 12 quotations assigned to code "Fire," the query "NOT Fire" retrieves 108 quotations - those which are *not* coded with "Fire." Of course, the operator can be used with an arbitrary expression as in the argument "NOT (Earth OR Fire)" which is the equivalent of "neither Earth nor Fire."

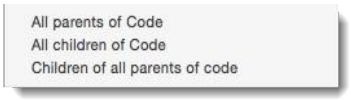
The OR operator has the potential to generate a HUGE number of hits. It has high "recall" (a lot is retrieved), but low "precision" (many of the retrieved quotations may not necessarily fit).

Venn diagrams are descriptive schemes for illustrating the different set operations associated with Boolean operators.

The rectangle encloses the set of all retrievable quotations, e. g. the "document universe." The two circles represent two codes A and B. Q1 to Q5 are quotations coded with A, B, or none (Q5).

SEMANTIC OPERATORS

The operators in this section exploit connected codes resulting from previous theory-building work. While Boolean-based queries are *extensional* and simply enumerate the elements of combined sets (e. g., LOVE or KINDNESS), semantic operators are *intentional*, as they already capture some meaning expressed in appropriately linked concepts (e. g., All Children of code `(POSITIVE ATTITUDES)`).



The **All parents of Code** operator looks at all directly linked codes and their quotations at higher levels.

The **All children of Code** operator traverses the network from higher to lower concepts, collecting all quotations from any of the sub codes. Only transitive" relations between the codes are processed (see "Relation Properties"); all others are types ignored. When building a terminology from your codes, use the ISA relation for sub-term links.

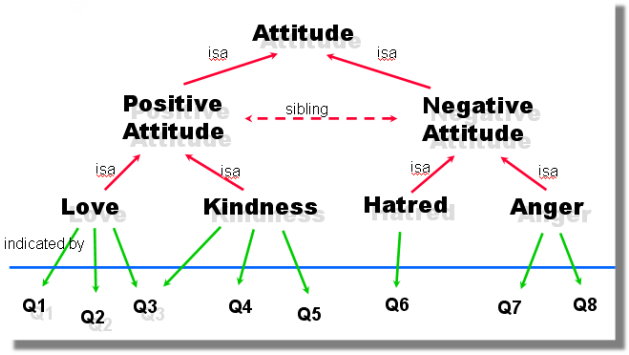


Figure 77: A hierarchy of concepts suitable for semantic retrieval

Like the OR (any) operator in the set of Boolean operators, the 'All children of Code' operator may produce large result sets. However, because you make use of a theory, the "precision" is likely better as compared to using OR (i. e., you get only what you expect). Of course, if your network contains dubious connections ("computer ISA intelligent entity"), the quality of your retrieval will decline.

The **Children of all parents of Code** operator finds all quotations that are connected to the selected code or any other descendants of the same parent code. Example: "All quotations coded with Love or any other Positive Attitude, here: kindness." See Figure 77 above.

With such a network of codes the following queries would make sense (Q1 to Q8 = quotations):

All children of Code (Positive Attitude) => {Q1, Q2, Q3, Q4, Q5}

All children of Code (Negative Attitude} => {Q6, Q7, Q8}

All children of Code (Attitude) => {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8}.

PROXIMITY OPERATORS

Because of non-commutativity, every proximity operator comes in two versions.

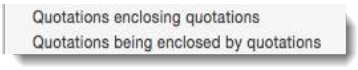
Proximity describes the spatial relation between quotations. Quotations can be embedded in one another, one may follow another, etc. The operators in this section exploit these relationships. They require two operands as their arguments. They differ from the other operators in one important aspect: proximity operators are *non-commutative*. This property makes their usage a little more difficult to learn.

Non-commutativity requires a certain input *sequence* for the operands. While "A OR B" is equal to "B OR A," this does not hold for any of the proximity operators: "A FOLLOWS B" is not equal to "B FOLLOWS A." When building a query, always enter the expressions in the order in which they appear in their natural language manifestation.

Another important characteristic for these operators is the specification of the operand for which you want the quotations retrieved. "A WITHIN B" specifies the constraint, but you must also specify if you want the quotations for the As or the Bs. This is done implicitly by the sequence. The code (or term) that is entered first is the one in which you are interested. If B's quotations are requested, you have to enter "B ENCLOSES A" using the query language described below.

EMBEDDING OPERATORS

The *embedding* operators describe quotations that are contained in one another and that are coded with certain codes.



Quotations enclosing quotations: A ENCLOSES B retrieves all quotations coded with A that contain quotations coded with B.

Quotations being enclosed by quotations: A being enclosed by B (WITHIN) retrieves all quotations coded with A that are contained within data segments coded with B.

For example, if you want to retrieve all segments for "give: joy" related to the code "#fam: have children" (see figure 78), you would need to enter the query as follows:

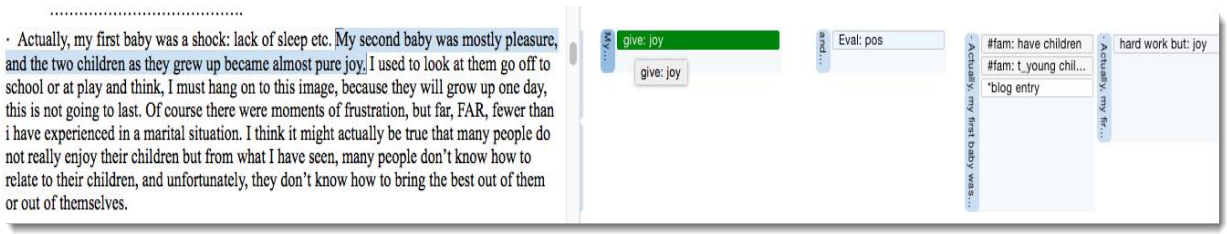


Figure 78: Visualizing the spatial relations between segments

Figure 79: Being enclosed operators

If you were to enter the query the other way around (i.e., the code 'fam: have children' first), the query tool would not deliver any results.

If you enter: Quotations enclosing quotations: "#fam: have children," "give joy, the the larger segments coded with "#fam: have children"are retrieved:

Figure 80: Enclosing operators

Quotations overlapping at start
Quotations overlapping at end

From the above example we learn that you begin with the code whose content you are most interested in. **Overlap Operators**

The *overlap* operators describe quotations that overlap one another:

Quotations overlapping at start: A OVERLAPS B retrieves all quotations coded with A that overlap quotations coded with B

Quotations overlapping at end: A OVERLAPPED BY B retrieves all quotations coded with A that are overlapped by quotations coded with B.

If you are interested in data segments coded with "Happiness: effect of wanting children" that co-occur with "hard work but," you click: "Happiness: effect of wanting children," "hard work but" COCCUR. If you want to read the "hard work but" segments, you enter the query the other way around, i. e. starting with the "hard work but" code.

THE CO-OCCURRENCE OPERATOR

Quotation cooccurrences

Often when interested in the relation between two or more codes, you don't really care whether something overlaps or is overlapped by, or is within or encloses. It this is the case, you simply use the **Quotation cooccurence** operator. Co-occurrence is essentially a short-cut for a combination of the proximity operators discussed above, plus the operator AND (all). AND is a Boolean operator, but also finds cooccurrence, namely if coded segments overlap 100%.

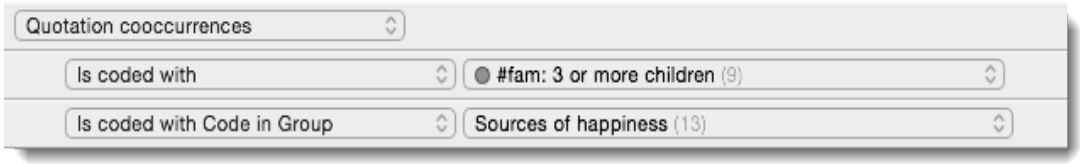


Figure 81: Example of a quotation cooccurence query

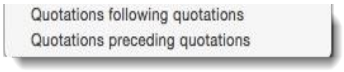
….retrieves all quotations coded with '#fam: 3 or more children' that cooccur with quotations coded with codes in the code group 'Sources of happiness' in whatever way.

The procedures used for calculating co-occurrence for two codes is also used in the Network Editor when importing co-occurring codes into a network view. See "Import Co-occurring Codes" and Working With The Code Cooccurence Table).

The more general cooccurence operator is quite useful when working with transcripts. In interviews people often jump back and forth in time or between contexts, and therefore it often does not make much sense to use the very specific operators embedding or overlap operators. With other types of data they are however quite useful. Think of video data where it might be important whether action A was already going on before action B started or vice versa. Or if you have coded longer section in your data like biographical time periods in a persons life and then did some more fine-grained coding within these time periods. Then the WITHIN operator comes in handy. The same applies when working with pre-coded survey data. ATLAS.ti pre-codes your questions, then you do some further coding. This enables you to ask for instance for all quotations coded with "topic x" WITHIN "question 5."

ADJACENCY OPERATORS

The *distance* operators describe a sequence of disjoint quotations.



Quotations following quotations: A FOLLOWS B retrieves all quotations coded with A that follow quotations coded with B.

Quotations preceding quotations: A PRECEDES B retrieves all quotations coded with A followed by quotations coded with B.

When selecting any of the two operators, you can specify a maximum distance. Possible base units are characters and paragraphs for text, milliseconds for audio files, frames for video data and pixels for images.

Example Queries

The examples presented are based on the children and happiness project.

You can download the project file here:
http://downloads.atlasti.com/samples/Children+Happiness_analysis.AtlPac

If you are not yet familiar with this sample project, here is a short description: The idea for this project was sparked by a journal article that summarizes relevant literature on the relationship between happiness and having children. The results based on statistical analysis show a negative correlation between measures of happiness and having children. This was picked up by a person writing a parenting blog and she posted it on her blog. The New York Times Magazine also published a long article about it. In turn, numerous people commented both on the blog and on the NYT article. These comments comprise one major part of the example project, and we can examine the reaction of readers being confronted with the statement: Children make you unhappy.

In terms of project setup, a document has been created for the comments posted on the parenting blog and for the comments posted on the NYT articles. Thus, respondent characteristics, e.g whether they have children or not, whether they are male or female have been coded. Creating document groups for male/female respondents, or parents/non-parents would not have worked with this type of data as each document contains responses from many different people.

EXAMPLE 1

Do parents who report positive aspects of parenting also talk about negative aspects? If so, which aspects are those?

Codes needed: #fam: have children / code group: effects parenting: negative / code group: effects parenting: positive

At first you need to find all statements of parents (#fam: have children) about positive effects of parenting and store the results as smart code. As we later want to find within the same response also statements about negative aspects of parenting, we need to start the query with the code '#fam: have children'. This quotation covers the full comment that the person has written.

Next we look for statements about negative effects of parenting that occur within the comment that also includes a positive effect.

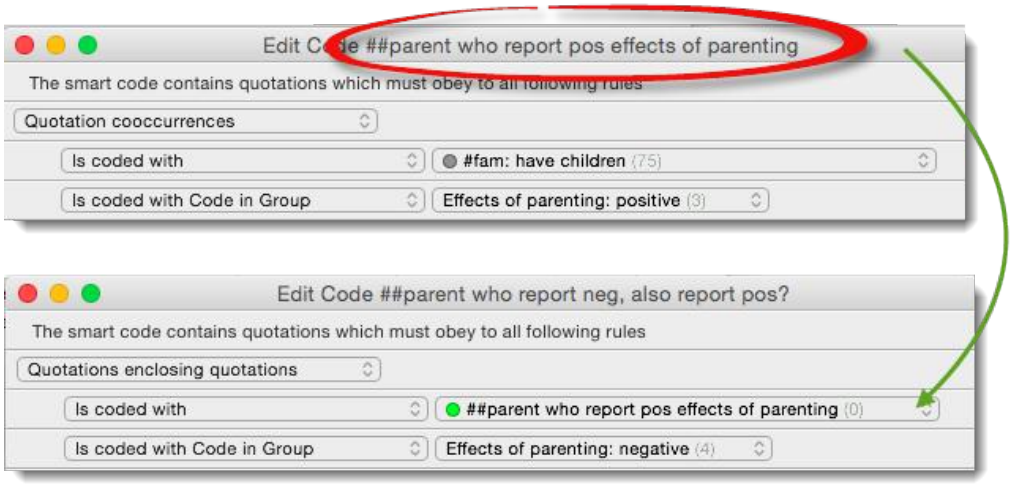


Figure 82: Smart codes needed to answer question 1

To answer the second part of the question – Which aspects are those?, you can either look through the quotations in context and check in the margin area which of the codes of the "negative effects" code group have been applied. Or you can use the Code Cooccurence Table (see page 83 and "Example Queries") and check for cooccurences between the smart code '##parent who report pos effects of parenting' and the single codes that express negative effects of parenting.

EXAMPLE 2

What is the attitude toward the relationship between children and happiness of those respondents who question the study design?

Let's do this step-by-step again before we show you how to put it all in one query:

Codes needed: children: < happiness / children: = level of happiness / children: > happiness / study design: asking the wrong question / study design: critique / *blog entry

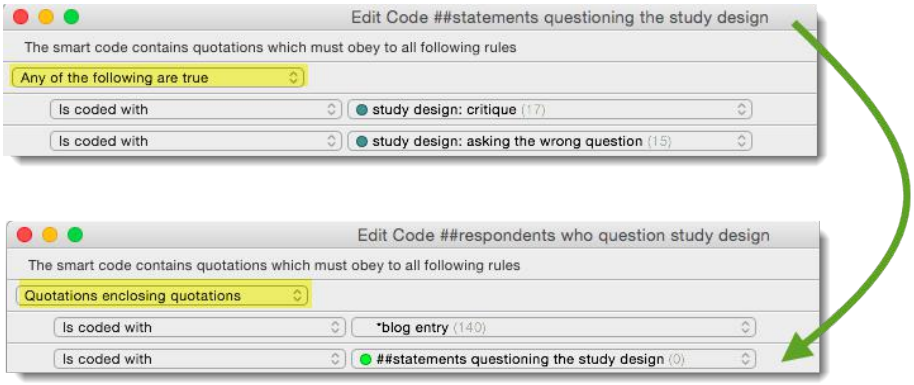


Figure 83: First steps in finding an answer to question 2

We first need to find all statements questioning the study design and in a second step we need to look for those statements within the blog comments. This results in a list of blog comments that can then be related to the attitude of these respondents have with regard to the question whether children make happier or not. The second part of the question is best answered again using the Code Cooccurence Table (see below "Example Queries").

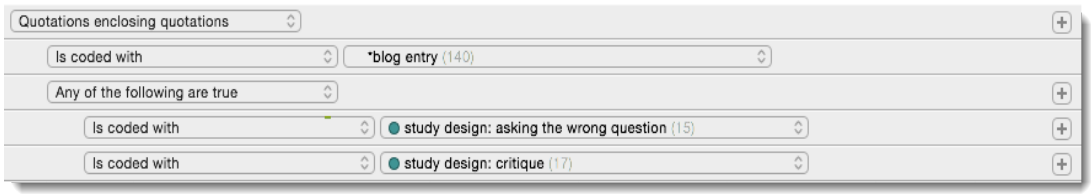


Figure 84: Looking for statements within the blog comments that either question or critique the study design

BUILDING MORE COMPLEX QUERIES

Without creating the in-between step and creating the smart code, ##statements questioning the study design, the query looks as follows:

If you want to relate the resulting quotations to the attitude code group (whether children make you happier, less happy or the level of happiness remains the same), extend the query on the top level by clicking on the **+** sign above your first entry:

Figure 85: Extending the query further

Editing Queries

If you want to modify a query, open the Code Manager and right-click on a smart code. Select the option **Edit Query**.

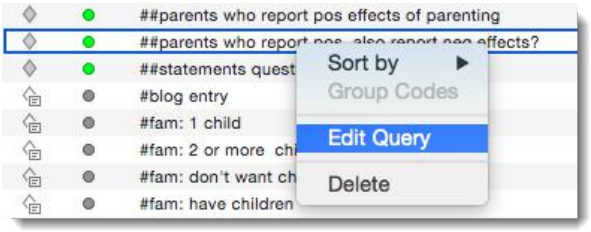


Figure 86: Editing smart codes

You may have noticed that I started the smart code name with two hash signs (##). As the codes are sorted in alphabetic order, this pushes the codes on top of the code list. Otherwise they "disappear" in the list of all other codes. They are recognizable by the filled-in code icon, but still I prefer to add a distinguishing color (here: bright green). And to add them to a code group (here: *smart codes) for quick access. The effect of the asterisk * in the name is that the code group 'smart codes' is on top of the list.

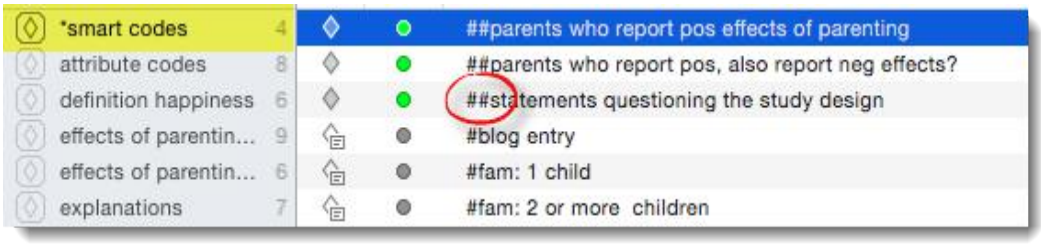


Figure 87: Quick access to smart codes

In future versions of the program, smart codes will be displayed in a separate part of the Code Manager. Then this workaround will no longer be necessary.

Working With The Code Cooccurence Table

You find the Code Cooccurence Table under the **Analysis** menu. In Figure 88 below you see an example query comparing the answers provided by respondents with and without children (column codes) with regard to a number of different issues (row codes):

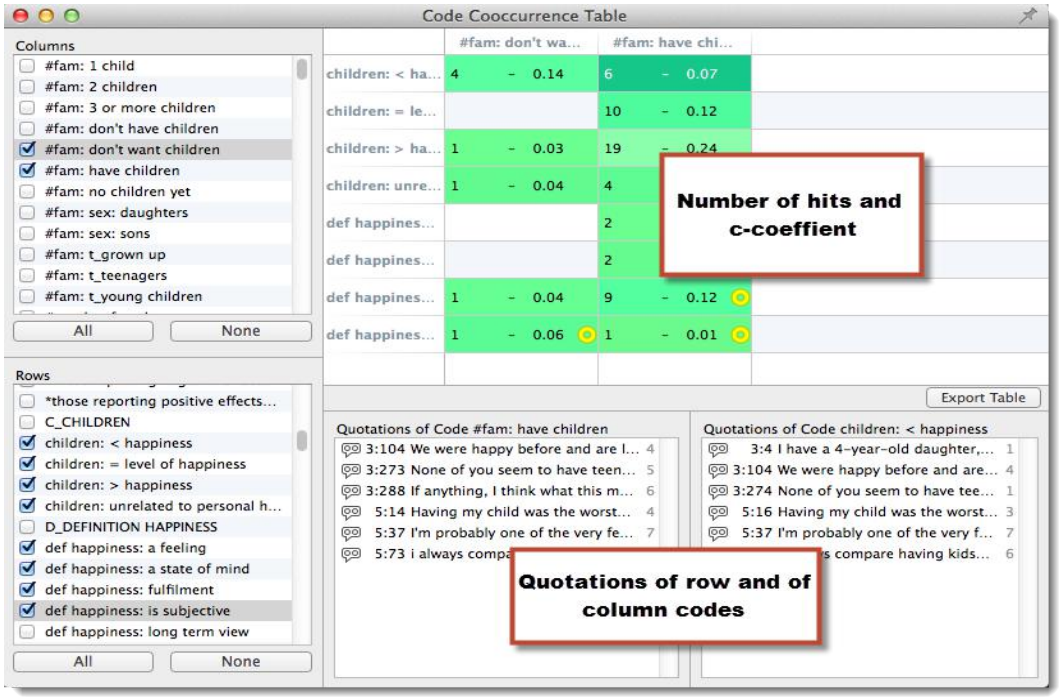


Figure 88: Code Cooccurence Table

The operators that are used to calculate code cooccurences are enclosing / being enclosed / overlapping at start / overlapping at end / All of the following are true. For further information on these operators see "Operators."

To create a table as shown in Figure 88 select column and row codes on the left-hand side by ticking the boxes in front of the codes. The table fills in automatically when selecting row codes. The first number is the number of hits – how often do the code and column codes co-occur; the second number is the c-coefficient (see below for further information).

Currently both, number of hits and c-coefficient are displayed. In the future, you will be able to chose which number(s) you want to see. The c-coefficient only makes sense and is interpretable with specific type of data and higher case numbers.

Click on a cell in the table to view the resulting quotations in the pane at the bottom. Double-click to access a quotation in context.

EXPORT

Currently the output option consists of exporting the table as Excel-compatible file in csv format; only the c-coeffient is exported. Other options like a full report of all cooccuring quotations will follow.

The C-Coefficient

The calculation of the c-coefficient is based on approaches borrowed from quantitative content analysis. Thus, interpreting such a coefficient is only meaningful with a sizable data set and not for an interview study with 10 respondents.

The c-coefficient should vary between 0: codes do not co-occur, and 1: these two codes co-occur wherever they are used. It is calculated as follows:

```
c := n12/(n1 + n2) – n12
```

n12 = co-occurrence frequency of two codes c1 and c2, whereby n1 and n2 are their occurrence frequency

What you may experience is the following:

Out of range. The C-index exceeds the 0 - 1 range it is supposed to stay with.

Colored circles. Cells can have additional visual cues, e. g., a red, yellow or orange circle.

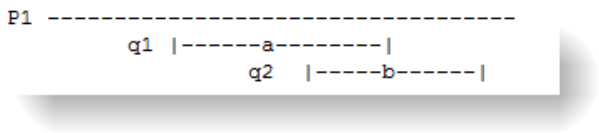


Figure 89: Out of range example 1

Out Of Range

The c-index (structurally resembling the Tanimoto and Jaquard Coefficient, which are similarity measures) assumes separate non-overlapping text entities. Only then can we expect a correct range of values. However, ATLAS.ti's quotations may overlap to any degree.

Case 1: Two differently coded quotations overlap, we assume no more quotations available. Let P1 be a textual document, q1 and q2 be quotations and a,b be codes. q1 is coded with a, q2 is coded with b.

Using the formula: c := n_ab/(n_a + n_b) – n_ab, we get:

n_ab = 1 one co-occurrence of a and b
n_a = 1, n_b = 1 a and b each code exactly one quotation.
c = 1/(1 + 1) – 1 = 1

Such a scenario results in the maximum co-occurrence of 1 !

Case 2: q1 is coded with both codes a and b, the overlapping quotation q2 is coded with b.

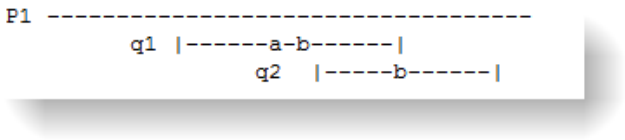


Figure 90: Out of range example 2

n_ab = 2. q1 alone counts for a co-occurrence event and the overlapping q1*q2 for another.

n_a = 1, n_b = 2
c = 2/(1 + 2) – 2 = 2!!

This results in a value of twice the allowed maximum. Thus, the C index is not appropriate to correctly represent co-occurrence in redundantly overlapping texts. If the c-coefficient exceeds 1, you need to do some cleaning up and eliminate the redundant codes.

Color Indicators

Red circle: When the c-index exceeds 1 (see "Out of range").

Yellow circle: An inherent issue with the C-index and similar measures is that it is distorted by code frequencies that differ too much. In such cases the coefficient tends to be much smaller than the potential significance of the cooccurrence. For instance, if you had coded 100 quotations with code "depression" and 10 with "mother" and you had 5 co-occurrences:

n_dep = 100, n_mother = 10, n_dep-mother = 5
c = 5/(100 + 10) - 5 = 5/105 = 0.048

A c index of only 0.048 may slip your eye easily, although code "mother" appears in 50% of all its applications with code "depression." Looking from code "depression" only 5% cooccurr with code "mother."

If the ratio between the codes frequencies exceeds a certain threshold (currently 5) the yellow light goes on in the cell. So whenever a cell shows the yellow marker it should invite you to look into the co-occurrences of this cell despite a low c-index.

Orange Circle: The orange circle is simply a mixture of the red and yellow conditions.

Example Queries

Returning to the examples used in the section on smart codes, let's now take a look at the negative aspects of parenting that those respondents report that also write about positive aspects.

Open the Code Cooccurrence Table.

Select the smart code ##parents who report positive effect of parenting as column code, and all effects neg codes as row codes. This results in the following table:

	##parents who report pos effects of parenting
effects neg: less fun/ satisfaction	4
effects neg: loss of freedom	2
effects neg: more worries/stress	4
effects neg: on career	0
effects neg: on financial issues	1
effects neg: on relationships	3
effects neg: on self	1
effects neg: takes lots of energy	0

Figure 91: Results of a cooccurence table and exported version showing number of hits only

To see the full code names, move with your mouse of the column or row codes. In the near future, you will be able to resize the columns, so that the full code names can be displayed. On the right-hand side of Figure 91, you see the exported Excel / Numbers table. As the c-coefficient does not make any sense with this small data set, the number of hits are shown. What you can see from the table is that those reporting positive effects of parenting also write about issues like having less fun, loss of freedom, more worries, more stress, financial issues,negative effects on relationships and on self. In order to interpret these findings, you also need to go back to the data and read what exactly these respondents have been writing. This can be done by clicking on the quotations for the row codes in the Code Coccurence table.

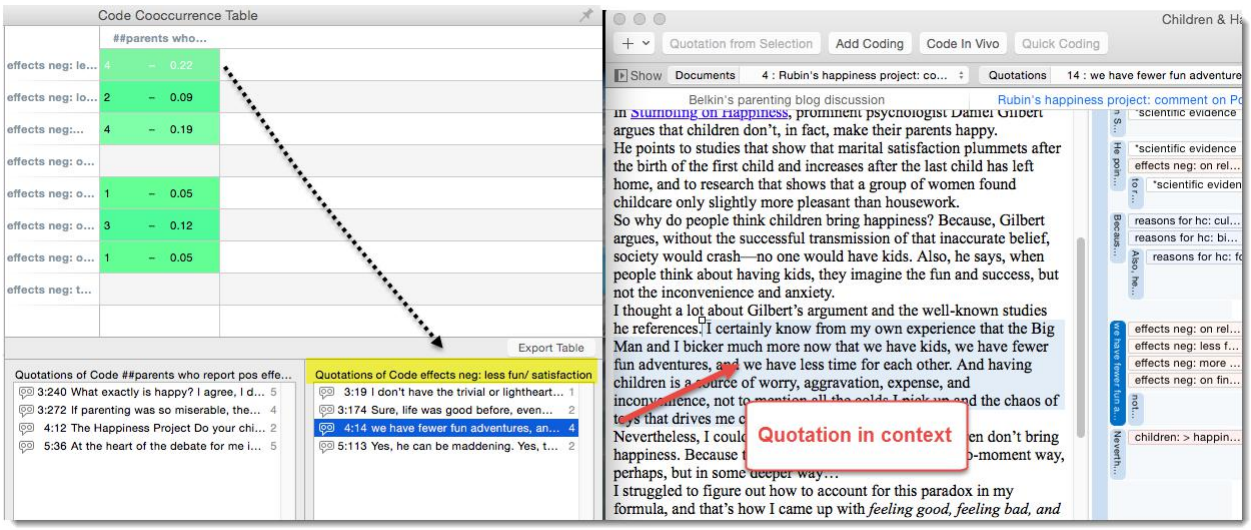


Figure 92: View resulting quotations in context

The Word Cruncher

This feature offers word "crunching" capabilities for a simple quantitative content analysis. It creates a list of word frequency counts for the selected or all (currently filtered) textual PDs. An **exception list** can be used to control the analysis. Further you can create a **word cloud** for the selected documents, or for all documents (not yet implemented).

Word	Length	Count	%	Belkin's parenting blog...	NYT magazin blog:...
i	1	556	3,766	352	204
have	4	261	1,768	185	76
my	2	257	1,741	174	83
children	8	215	1,456	168	47
you	3	201	1,362	104	97
not	3	189	1,280	124	65
with	4	166	1,124	107	59
we	2	140	0,948	80	60
but	3	137	0,928	91	46
they	4	131	0,887	93	38
kids	4	126	0,853	63	63
who	3	110	0,745	81	29
life	4	109	0,738	66	43
happiness	9	108	0,732	90	18
happy	5	104	0,704	74	30
people	6	102	0,691	62	40
more	4	97	0,657	71	26
this	4	96	0,650	60	36
having	6	92	0,623	66	26
was	3	92	0,623	62	30
me	2	91	0,616	55	36

Figure 93: Example of a Word Cruncher output - comparing the word count of two documents

Creating A Word Count Table

From the main menu, select **Analysis / Word Cruncher.** The Word Cruncher opens.

Select one or more documents in the Documents column on the left. The table immediately fills with content. At the bottom of the table you have a few further options:

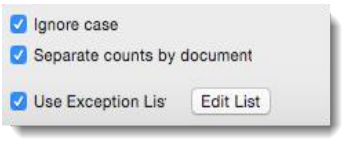


Figure 94: Word Cruncher options

If you select "Separate counts by document" you get the total and relative counts for all selected documents, and separate frequency counts for each document.

Results can be improved by preparing adequate exceptions lists. You can add words to the exception list by right-clicking on a word in the Word Cruncher, or by editing the list.

Editing The Exception List

Select the option **Edit List.**

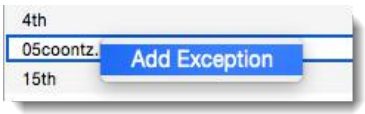


Figure 95: Adding words to the exception list

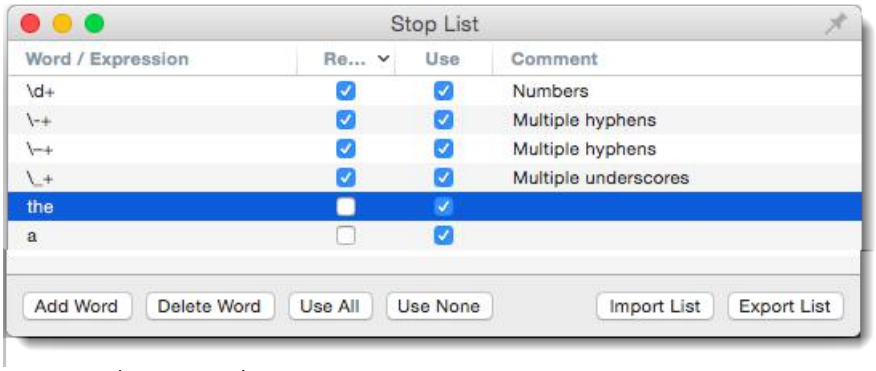


Figure 96: The exception list

The first four expressions excluding the count of numbers, multiples hyphens and multiple underscores are already entered. You can de- / activate words or expressions by clicking on the **Use** column. The first four entries, in addition to being useful, also serve as an example for the kinds of regular expressions you can use. For more detail see table below.

To add or delete words click the appropriate button at the bottom of the window.

The following table lists the regular expressions that can be used:

Expression	Effect
\<.\>	excludes words consisting of a single character
/d+	excludes numbers of any length
–+	excludes strings of hyphens of arbitrary size
_+	excludes strings of underscores
0+	exclude sequences of zeros

Word Cruncher Output

Click on the button **EXPORT TABLE** to export the table to be opened in Excel or Numbers. Use the Numbers or MS Excel™ functionality such as sorting by highest to lowest frequency for further exploration.

Creating Reports

Reports Based On The Entire Project

You find an output option under the main menu of each entity type: Document, Quotation, Code, Memo and Network. Currently, all output options refer to the entire project. You cannot for instance export quotations of just one code. Such reports will, however, be implemented soon.

Below a few output options are shown as examples:

Document / Output / List of Document Groups and their Members

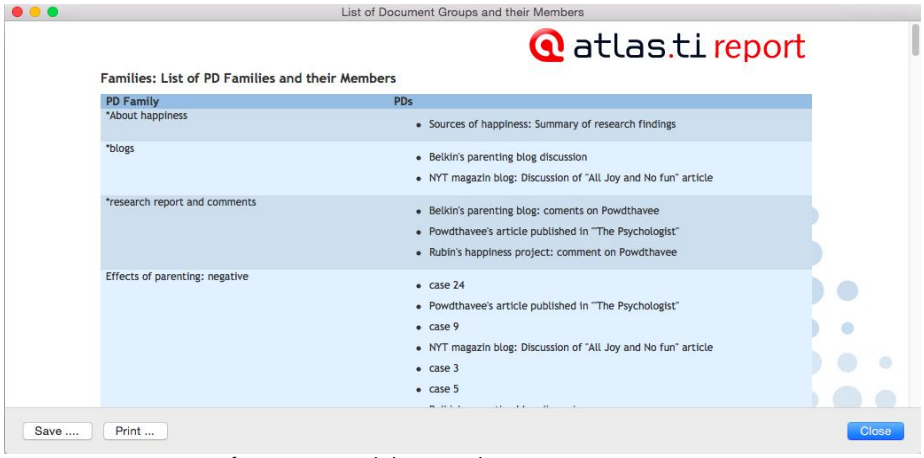


Figure 97: Report: List of PD groups and their members

Quotation / Output / Quotations by Code

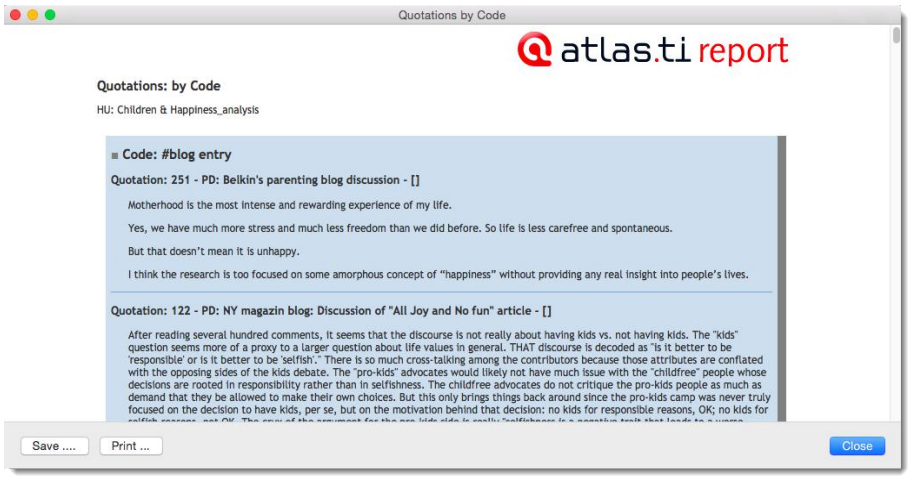


Figure 98: Report: Quotations by Code

Also feel free to try other output options, such as

- Codes / Output / Codebook
- Codes / Output / List of Codes by Document
- Memos / Output / Memos with Content and Linked Quotations
- Network / Output / List of Code-Code Links with Comments

etc.

Print With Margin

The Print with Margin option creates a WYSIWYG printout of coded documents (What You See Is What You Get). Instead of printing a document, you can also save it as PDF file, mail the PDF file or add it to iBooks, etc. See Figure 7 that shows the available options.

Print with Margin is available for textual primary documents, PDF and image PDs.

The printout resembles the screen display at the time of creating the output. Only those margin objects are included that are currently displayed in the margin. If not everything that you want to be visible is included, adjust the margin area. To change the type of objects that are displayed, right click on a white space in the margin area to open the context menu and select the objects to be included in the output.

To Print A Primary Document With Margin

Load the primary document that you want to print.

Select **Project / Print Document** from the main menu.

The printer dialogue window opens. Make your choices. Select landscape view if your codes spread over several columns. If you change the settings the preview adjusts automatically. If you do not want to print the entire document, select start and end page. The pages are shown in the preview.

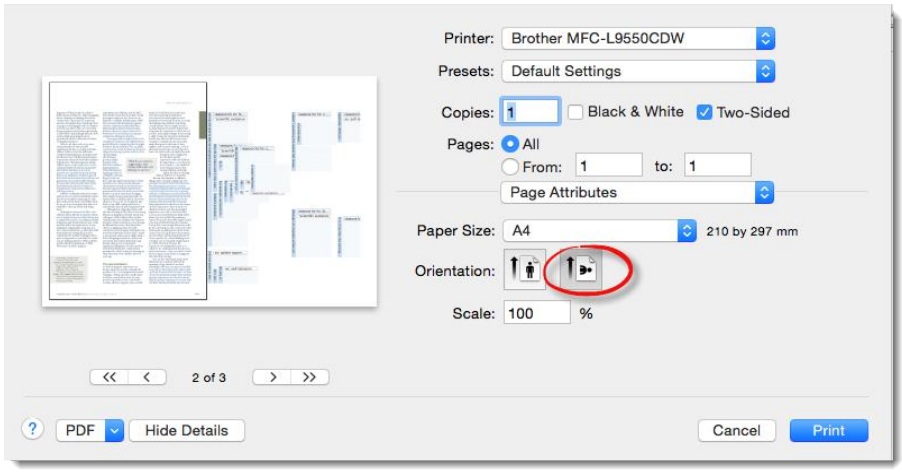


Figure 99: Print with margin settings

Prior to printing the document, you can also open a PDF preview or select other options. See Figure 100).

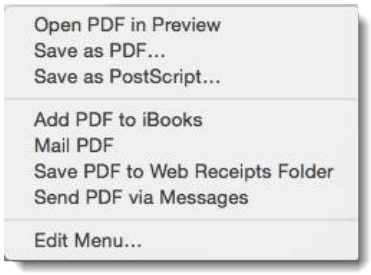


Figure 100: Print with margin options