ATLAS.ti
QUALITATIVE DATA ANALYSIS

Inter-Coder Agreement Analysis: ATLAS.ti 8 Windows

ATLAS.ti 8 Windows - Inter-Coder Agreement Analysis
Copyright ©2019 by ATLAS.ti Scientific Software Development GmbH, Berlin. All rights reserved.
Manual Version: 652.20190425. Updated for program version: 8.4

Author: Dr. Susanne Friese (QuaRC)
Programming/Design/Layout: Dr. Thomas G. Ringmayr - hypertexxt.com

Contents

Inter-coder Agreement (ICA)

Please Read This First!

Previous versions of ATLAS.ti allowed to calculate inter-coder agreement (ICA) in combination with the online tool CAT. This required a special way of merging projects and the creation of a specific text-only output. Both are no longer available in ATLAS.ti 8. Therefore, CAT can no longer be used in combination with ATLAS.ti.

Instead, we decided to develop our own inter-coder agreement tool that would be both easy to use and produce results of outstanding accuracy. To this end, we worked closely with Prof. Klaus Krippendorff, one of the leading experts in this field and the namesake of important "measurement units" in this area. ATLAS.ti's Inter-coder Agreement tool lets you assess the accuracy of how multiple coders analyze a given body of data, and thus permits you to underscore the validity of your analysis.

The need for such a tool as an integrated element in ATLAS.ti has long been evident and has been frequently requested by users. When we set out to to tackle the task of designing it, it was also clear that would have to be more than the one-dimensional and sometimes crassly simplifying attempts we found elsewhere. Rather, the ATLAS.ti inter-coder Agreement tool was supposed to be equally flexible, accurate, and able to handle the inherent complexity of the subject matter.

This also means that by necessity it requires at least a minimal willingness for the user to delve into some of the basic theoretical foundations of what inter-coder agreement is, what it does and can do, and also what it cannot do. We tried to make the tool as easy to use as possible and require only a small degree of preparation, but by its nature, it could not and cannot be a magic "just click a button and hope for the best" solution.

That said, you are probably aware of the expression, "garbage in, garbage out." If you randomly click on any of the choices that ATLAS.ti offers to calculate an inter-coder agreement coefficient, ATLAS.ti will calculate *something*. Whether the number you receive will be meaningful and useful depends on how you have set up your project and your coding. ATLAS.ti can only provide the technical functionality. You as the user are responsible for applying it in a meaningful and sensible way, especially if you intend to report an inter-coder agreement coefficient in a publication. Hence, we would like to lay out a few of the underlying assumptions and requirements.

If you want to develop a code system as a team, yes, you can start coding independently and then see what you get. But this approach can only be an initial brainstorming at best; a first step towards developing a code system based on which you can test inter-coder agreement. But this will be much later in the process. You first need a structured code system in which each code is clearly defined. The testing also needs to be done by two or more additional coders that are independent of the person who developed the codes. Thus, if you want to measure inter-coder agreement, you need at least three persons: One person developing and defining the codes, and two persons to apply the codes.

Please keep in mind that the inter-coder agreement tool crosses the qualitative-quantitative divide. Establishing inter-coder agreement has its origin in quantitative content analysis (see for instance Krippendorff, 2018; Schreier, 2012). If you want to apply it and want to adhere to scientific standards, you must follow some rules that are much stricter than those for qualitative coding.

Why Call It Inter-coder Agreement Rather Than Inter-coder Reliability?

The coefficients that are available measure the extent of agreement or disagreement of different coders. Based on this measure one can infer reliability, but the coefficients do not measure reliability directly. Therefore, what is measured is inter-coder *agreement* and not inter-coder *reliability*.

Agreement is what we measure; reliability is what we wish to infer from it.

Why Reliability Matters

The purpose of collecting and analyzing data is that researchers find answers to the research questions that motivated the study in the first place. Thus, the data are the trusted ground for any reasoning and discussion of the results. Therefore, the researchers should be confident that their data has been generated taking precaution against distortions and biases, intentional or accidental, and that the mean the same thing to anyone who uses them. Reliability grounds this confidence empirically (Krippendorff, 2004).

There are two ways to ope rationalize reliability, one routed in measurement theory, which is less relevant for the type of data that ATLAS.ti users have. The second one is an interpretivist conception of reliability. When collecting any type of interview data or observations the phenomena of interest usually disappears right after it has been recorded or observed. Therefore, the analyst's ability to examine the phenomena relies heavily on a consensual reading and use of the data that represent the phenomena of interest. Researchers need to presume that their data can be trusted to mean the same to all of their users. This means "*that the reading of textual data as well as of the research results is replicable elsewhere, that* *researchers demonstrably agree on what they are talking about. Here, then, reliability is the degree in which members of a designated community agree on the readings, interpretations, responses to, or uses of given texts or data. [...] Researchers need to demonstrate the trustworthiness of their data by measuring their reliability"* (Krippendorff, 2004, p. 212).

Testing the reliability of the data is a first step. Only after establishing that the reliability is sufficiently high, it makes sense to proceed with the analysis of the data. If there is considerable doubt what the data mean, it will be difficult to justify the further analysis and also the results of this analysis.

Reliability And Validity

Whereas reliability offers the certainty that research findings can be reproduced and that no or only limited external "noise" has contaminated the data or the results, validity assures that the assertions made by the research reflect the reality it claims to represent. Validity concerns truth(s).

Reliability relates to validity in the following ways:

The more unreliable the data, the less likely it is that researchers can draw valid conclusions from the data. In terms of coding data, this means that researchers need to identify valid accounts in the data to a degree better than by chance. If the agreement of two or more coders is not better than the agreement by chance, then reliability is low and you cannot infer that a common understanding of the data exists. Thus: **Unreliability limits the chance of validity.**

On the other hand, **reliability does not necessarily guarantee validity**. Two coders may share the same world view and have the same prejudices may well agree on what they see, but could objectively be wrong. Also, if two researchers may have a unique perspective based on their academic discipline but their reading is not shared by many people outside their own scholarly community, the reliability might be high but the outcome of the research has little chance of being substantiated by evidence of the reality that is inferred. As Krippendorff (2004) states: "Even perfectly dependable mechanical instruments, such as computers, can be wrong – reliably." (p. 213).

A third aspect is the **dilemma between high reliability and validity**. Interesting interpretations might not be reproducible, or interesting data may not occur often enough to establish reliability. Highly reliable data might be boring and oversimplified in order to establish a high reliability in the first place.

At What Phase In Your Project Should ICA Analysis Be Performed?

A good time to have your coding checked by other coders is when you have built a stable code system and all codes are defined. This means, this is somewhere in the middle of the coding process. Once a satisfactory ICA coefficient is achieved, the principal investigator has the assurance that his or her codes can be understood and applied by others and can continue to work with the code system.

Requirements For Coding

Sources for unreliable data are intra-coder inconsistencies and inter-coder disagreements. To detect these, we need to replicate the coding process. Replicability can be assured when several independently working coders agree on the use of the written coding instruction, by highlighting the same textual segments to which the coding instructions apply, identifying the same semantic domains that would describe them, and code them using the same codes for each semantic domain.

Development Of Semantic Domains

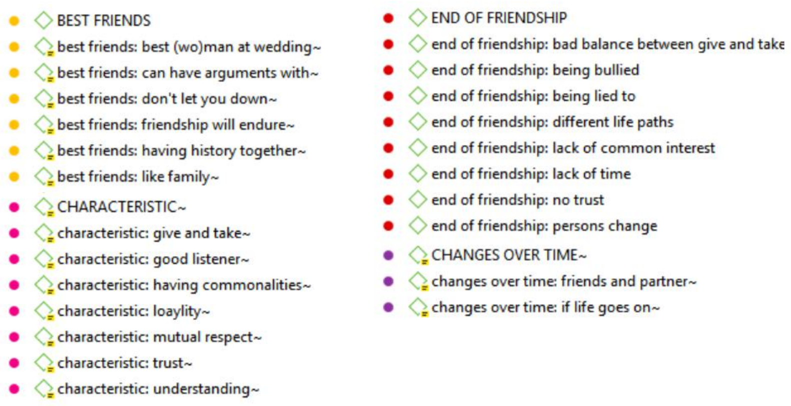
In developing the ICA function, it was decided to use a special name to make it clear that this type of coding is related to an ICA analysis.

A semantic domain is defined as a space of distinct concepts that share common meanings. Examples of semantic domains are emotional states, a set of strategies mentioned to deal with something, motivations for achieving something, gender biases, memberships in different, values, or genres. Each semantic domain embraces **mutually exclusive concepts** indicated by a **code**.

At the development stage of a coding instruction a semantic domain is **open ended**. For example, one may start with the semantic domain "benefits of friendship" by distinguishing "caring for each other" and "supporting each other" but soon discovers that there are also other aspects like "improve health and longevity" and "learning from each other". So a code for these aspects needs to be added to the semantic domain of "benefits of friendship". Once a coding instruction is written and used in testing its reliability, one can no longer expand a semantic domain.

All semantic domains are **context dependent**. The colour of a ball pen has little to do with the state of a drunk, or a political party. Semantic domains may use abstract names, but without acknowledgements of their contexts coding becomes unreliable.

Semantic domains are **logically or conceptually independent** of each other, hence freely combinable as appropriate. In a section of your data, you may find someone describing the characteristics of friendship and changes over time. As these codes come from different semantic domains, they can both be applied.



- ● ◇ BEST FRIENDS
- ● ◇ best friends: best (wo)man at wedding~
- ● ◇ best friends: can have arguments with~
- ● ◇ best friends: don't let you down~
- ● ◇ best friends: friendship will endure~
- ● ◇ best friends: having history together~
- ● ◇ best friends: like family~
- ● ◇ CHARACTERISTIC~
- ● ◇ characteristic: give and take~
- ● ◇ characteristic: good listener~
- ● ◇ characteristic: having commonalities~
- ● ◇ characteristic: loaylity~
- ● ◇ characteristic: mutual respect~
- ● ◇ characteristic: trust~
- ● ◇ characteristic: understanding~
- ● ◇ END OF FRIENDSHIP
- ● ◇ end of friendship: bad balance between give and take
- ● ◇ end of friendship: being bullied
- ● ◇ end of friendship: being lied to
- ● ◇ end of friendship: different life paths
- ● ◇ end of friendship: lack of common interest
- ● ◇ end of friendship: lack of time
- ● ◇ end of friendship: no trust
- ● ◇ end of friendship: persons change
- ● ◇ CHANGES OVER TIME~
- ● ◇ changes over time: friends and partner~
- ● ◇ changes over time: if life goes on~

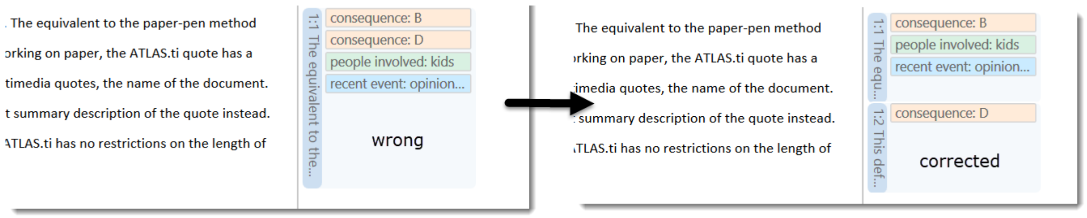
Figure 1: Examples of semantic domains in ATLAS.ti

There are two requirements for developing and working with semantic domains: exhaustiveness and mutual exclusiveness.

Exhaustiveness means that the codes of the code system cover the variability in the data and that no aspect that is relevant for the research question is left-out. On the domain level this means that all main topics are covered. On the sub code level, this means that the codes of a semantic domain cover all aspects of the domain and the data can be sorted in the available codes without forcing them. An easy way out is to include a catch all 'miscellaneous' code for each domain into which coders can add all data that they think does not fit anywhere else. However, keep in mind that such catch all codes will contribute little to answering the research questions.

Mutual exclusiveness affects two areas. One is the meaning of the sub codes: Each of the sub codes within each domain needs to be different and this needs to be clearly specified in the code definitions and coding rules. There should be no doubt when to apply sub code 1, 2 or 3. The second is the application of the sub code: **You can only apply one of the sub codes of a semantic domain to a quotation or to overlapping quotations**. Using the same code colour for all codes of a semantic domain will help you to detect possible errors. See Figure 2.

If you find that you have coded a quotation with two codes from the same semantic domain, you can fix it by splitting the quotation. This means, you change the length of the original quotation and create one new quotation, so you can apply the two codes to two distinct quotations.



. The equivalent to the paper-pen method
orking on paper, the ATLAS.ti quote has a
timedia quotes, the name of the document.
t summary description of the quote instead.
ATLAS.ti has no restrictions on the length of

1:1 The equivalent to the...

consequence: B
consequence: D
people involved: kids
recent event: opinion...

wrong

The equivalent to the paper-pen method
orking on paper, the ATLAS.ti quote has a
imedia quotes, the name of the document.
summary description of the quote instead.
TLAS.ti has no restrictions on the length of

1:1 The equ...

consequence: B
people involved: kids
recent event: opinion...

1:2 This def...

consequence: D

corrected

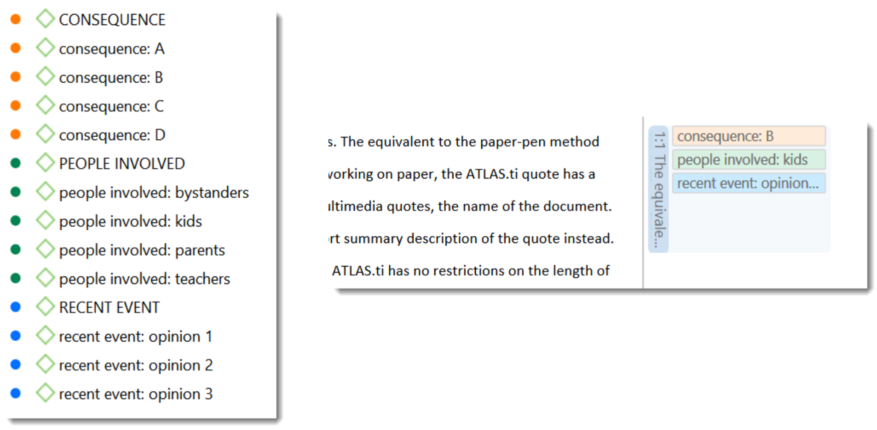
Figure 2: How to correct a violation of mutual exclusiveness

If codes within a semantic domain are not applied in a mutually exclusive manner, the ICA coefficient is inflated – and in the current implementation of the tool cannot be calculated!

In version 2 of the ICA tool released with version 9, ATLAS.ti will calculate the coefficient but will indicate that there is a problem. There will also be a link to the segments in your data that are problematic.

Multi-valued Coding

As also shown in Figure 3, you can apply multiple codes of different semantic domains to the same quotation. This is referred to as **multi-valued coding**. For instance, a respondent expresses an opinion about a recent event. The semantic domain in this example is the RECENT EVENT and the various 'opinions' about the recent event are the codes of his domain. Within the same section or in an overlapping section, the respondent also talks about the people involved in the event and the consequences it had for them. The PEOPLE INVOLVED, and the CONSEQUENCES are two further semantic domains. If your coding should cover all these different aspects that are mentioned in the sentence or paragraph, you need and can apply codes from all three domains.



● CONSEQUENCE
● consequence: A
● consequence: B
● consequence: C
● consequence: D
● PEOPLE INVOLVED
● people involved: bystanders
● people involved: kids
● people involved: parents
● people involved: teachers
● RECENT EVENT
● recent event: opinion 1
● recent event: opinion 2
● recent event: opinion 3

s. The equivalent to the paper-pen method
vorking on paper, the ATLAS.ti quote has a
ıltimedia quotes, the name of the document.
ırt summary description of the quote instead.
ATLAS.ti has no restrictions on the length of

Figure 3: Mulit-valued coding

Coding with codes from multiple semantic domains will allow you to see how the various semantic domains are related. For example, what kind of people are involved in what kinds of events and how they experience the event. For this you can use the code co-occurrence tools at a later stage in the analysis. See full manual.

Common Mistakes

A common belief is that **consensus is better than individual judgment**. Rather than to work independently, coders are encouraged to discuss what they code and to reach their decision by compromise. However, data generated in this way does not ensure reproducibility, nor does it reveal the extent of it. The results often reflect the social structure of the group, the most prestigious member of the group dominating the outcome. Reproducibility requires at least two independent coders.

Another common procedure is **to allow coders to consult each other** if problems arise, e.g. they do not understand a coding instruction, or they have problems applying some of the codes to the data given to them. This also compromises reliability. Ideally, the coding instructions should be clear and easy to understand. If this is not the case, the coders involved in the discussion create their own interpretation of what the codes mean. This is difficult to communicate to others and therefore jeopardizes reproducibility. In addition, the process loses stability as the data coded in the early phases were coded based on a different understanding of the codes. If coders do not work independently and discuss problems during the process of coding, the reliability coefficient might be higher in the end, but this is partly illusory. If the data were to be given to different coders not having the same insights, reliability is likely to be lower again. As it is common that code systems evolve, and code definitions and coding rules need to be refined, you can ask coders to write down all the issues they see with the current instructions and definitions. Based on their notes, the coding system can be refined and tested again, but with different coders.

Another common mistake is **to assume that it is best to ask other experts, colleagues with a long history of involvement in the subject of the research, or close friends and associates** to serve as coders. Those coders are likely to agree, but not because they carefully follow the coding instructions, but because they know each other and the purpose of the research. This also results in higher measures of reliability, but also does not serve reproducibility.

Measuring Inter-coder Agreement

Suppose two coders are given a text of a certain length and do what the coding instructions tell them to do. The simplest agreement coefficient $_{|cu}\alpha$ assesses the degree of agreement among these coders to decide what is relevant to a present research project. The may create so-called reliability data – complying with all the earlier mentioned condition under which they work – so that replicability can be inferred from the observed inter-coder agreement. They may give you the following record of their coding – here only which segments of text are relevant:

Evidently. The first pair of segments agree in length and location on the continuum.

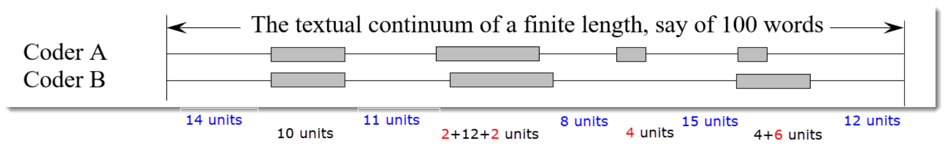


Figure 4: Textual continuum

The second pair agree in length but not in location. They have an intersection in common. In the third case, coder A finds a segment relevant, not recognized by coder B. the largest disagreement is observed in the last pair of segments. Coder A takes a narrower view than coder B.

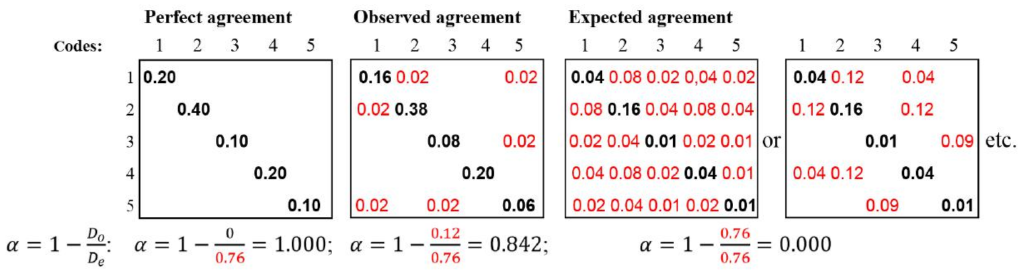
In terms of your ATLAS.ti coding, you need to think of your document as a textual continuum. Each character of your text is a unit of analysis for ICA, starting at character 1 and ending for instance at character 17500. For audio and video documents, the unit of analysis is a second. Images can currently not be used in an ICA analysis. The quotation itself does not go into the analysis, only the characters or seconds that have been coded. If you add multiple documents to the analysis, the continuum is extended like pearls strung on a chain. Thus every character or second where coders agree go into the calculation of the ICA coefficient.

According to Krippendorff (2019), the simplest inter-coder agreement coefficient is:

$$\alpha = 1 - \frac{D_o \quad \text{observed disagreement}}{D_e \quad \text{expected disagreement}}$$

It is a measure of the extent to which data can be trusted to represent the phenomena of analytical interest that one hopes to analyze in place of the raw phenomena.

When coders pay no attention to the text, which means they just apply the codes in an arbitrary manner, their coding has no relationship to what the text is about, then the observed disagreement D_o equals the expected disagreement $D_{e,}$ then α = 1–1 = 0.000. On the other hand, if agreement is perfect, which means disagreement D_o = 0, then α = 1- 0 = 1.000. For further information on how alpha is calcuated, see the appendix.



Perfect agreement

Codes:	1	2	3	4	5
1	0.20				
2		0.40			
3			0.10		
4				0.20	
5					0.10

Observed agreement

Codes:	1	2	3	4	5
1	0.16	0.02			0.02
2	0.02	0.38			
3			0.08		0.02
4				0.20	
5	0.02		0.02		0.06

Expected agreement

Codes:	1	2	3	4	5
1	0.04	0.08	0.02	0,04	0.02
2	0.08	0.16	0.04	0.08	0.04
3	0.02	0.04	0.01	0.02	0.01
4	0.04	0.08	0.02	0.04	0.01
5	0.02	0.04	0.01	0.02	0.01

or

Codes:	1	2	3	4	5
1	0.04	0.12		0.04	
2	0.12	0.16		0.12	
3			0.01		0.09
4	0.04	0.12		0.04	
5			0.09		0.01

etc.

$$\alpha = 1 - \frac{D_o}{D_e}: \quad \alpha = 1 - \frac{0}{0.76} = 1.000; \quad \alpha = 1 - \frac{0.12}{0.76} = 0.842; \quad \alpha = 1 - \frac{0.76}{0.76} = 0.000$$

Figure 5: Contingency table for a more complex example

To better understand the relationship between actual, observed and expected agreement, let's look a the following contingency table:

The matrices with perfect and expected agreement/disagreement serve only as a benchmark. In this simple example you could have worked it out in your head, but in a real example in ATLAS.ti this is no longer possible. The numbers are much higher, and often also odd since the calculation is based on the number of coded / non-coded characters / seconds. Therefore, it is good to see the perfect and expected agreements as a point of reference.

In the example above, the tables represent a semantic domain with five codes. If all coders agreed, they would have applied the semantic domain codes to 10 units each. If they had been applied randomly, they would have been evenly distributed amongst 10 units each (10/5 = 2). In the observed contingency table, we can see that the coders agree in applying code 4. There is some confusion about the application of code 2 and even more for code 5. The coders have applied the latter to three different codes: 1, 3, and 5. For all codes, where there is confusion the code definition needs to be revisited. For code 5 for instance you need to ask why it was confused with code 1 and 3? Are there any overlaps in the definition? Why was it understood in different ways?

Methods For Testing ICA

ATLAS.ti currently offers three methods to test inter-coder agreement: Simple percent agreement, Holsti Index, and two of the Krippendorff's alpha coefficients. For scientific reporting, we recommend the later two. In version 2 of the ICA tool, planned to be released with version 9 of ATLAS.ti, all four of the family of aplpha coefficients will be calculated (see "Krippendorff's Family Of Alpha Coefficients – from the general to the specific". You will also be able to inspect and print the coincidence matrices.

All methods can be used for two or more coders.

Percent Agreement

Percentage Agreement is the simplest measure of inter-coder agreement. It is calculated as the number of times a set of ratings agree, divided by the total number of units of observation that are rated, multiplied by 100.

The benefits of percentage agreement are that it is simple to calculate and it can be used with any type of measurement scale. Let's take a look at the following example: There are ten segments of text and two coders only needed to decide whether a code applies or does not apply:

Segments	1	2	3	4	5	6	7	8	9	10
Coder 1	1	1	0	0	0	0	0	0	0	0
Coder 2	0	1	1	0	0	1	0	1	0	0

Percent Agreement (PA) = number of agreements / total number of segments

PA = 6 / 10

PA =0.6 = 60%

Coder 1 and 2 agree 6 out of 10 times, so percent agreement is 60%. One could argue this is quite good. This calculation, however, does not account for chance agreement between ratings. If the two coders were not to read the data and would just randomly code the 10 segments. we would expect them to agree a certain percentage of the time by chance alone. The question is: How much higher is the 60% agreement over the agreement that would occur by chance? Below only the results are presented if chance agreement is taken into account. If you are interested in the calculation, take a look at Krippendorff (2004, p. 224-226). The agreement that is expected by mere chance is (9.6 +1.6)/20 = 56%. The 60% agreement thus is not impressive at all. Statistically speaking, the performance of the two coders is equivalent to having reliably coded only 1 of the 10 segments, and have arbitrarily assigned 0s and 1s to the other 9 segments.

Holsti Index

Holsti's method (1969) is a variation of percentage agreement, as percent agreement cannot be used if the coders have not all coded the same data segments. When coders were allowed to create their own quotations and did not code pre-defined quotations, the Holsti index needs to be user. P'lease note, that also the Holsti index does not take into account chance agreement.

The formula for the Holsti Index is:

PA (Holsti) = 2A/ (N1+N2)

PA (Holsti) represents percentage of agreement between two coders,

A is the number of the two coders' consensus decisions, and N1 and N2 are numbers of decisions the coders have made respectively .

Percentage agreement and the Holsti Index are equal when all coders code the same units of sample.

Holsti, O. R. (1969). Content analysis for the social sciences and humanities, Reading, MA: Addison-Wesley

Cohens Kappa

Cohen's Kappa (Cohen, 1960) is another coefficient used in the calculation of ICA and offered by other CAQDAS programs. We have been asked by users why ATLAS.ti does not offer it. Kappa is a modification of Scott's pi and according to Krippendorff (2019) and Zwick (1988) a rather unfortunate modification. There is a conceptual flaw in his calculation, as Kappa counts disagreements between observer preferences for available categories as agreements. In addition, both Cohen's kappa and Scott's P assume an infinite sample size. Krippendorff's alpha coefficient is sensitive to different sample sizes and can also be used on small samples.

Zwick, Rebecca (1988). Another look at interrater agreement. Psychological Bulletin, 103, 347-387.

Krippendorff's Family Of Alpha Coefficients – from the general to the specific

Inter-coder agreements can be measured at different levels. At the most general level, you get a value for relevance. That is, have the coders identified the areas relevant to the research question in a consensus manner? With reference to ATLAS.ti, this means agreement or disagreement at the level of the quotation, regardless of what was coded there.

At the next level, you get a coefficient for all semantic domains that are included in the analysis. This value takes into account the multiple-valued coding of codes from different semantic domains and is not just the average of the coefficients for all the semantic domains used in the analysis.

Next, you can see if encoders could identify the same semantic domain and how well they could distinguish between codes within the semantic domain.

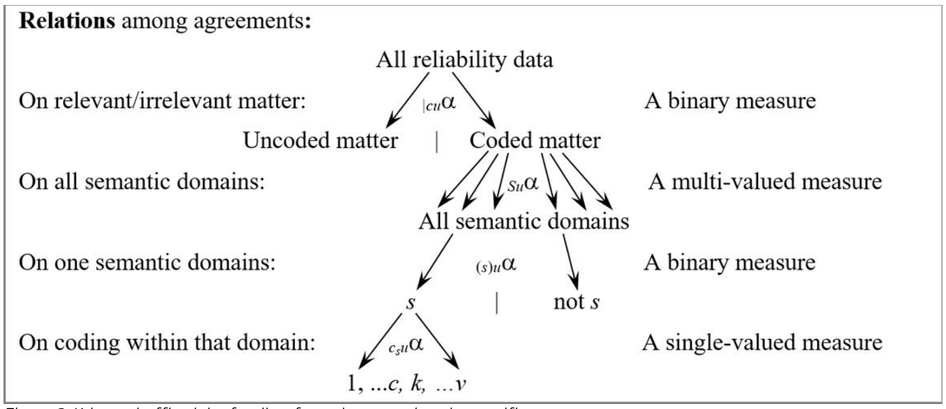


Figure 6: Krippendorff's alpha family – from the general to the specific

$_{|cu}\alpha$ indicates the extent to which coders agree on the relevance of texts for the research project (*currently alpha binary*)

$_{Su}\alpha$ indicates the extent to which coders agree on the presence or absence of semantic domains, (*currently Cu-alpha*)

$_{(s)u}\alpha$ indicates the degree to which coders identify a particular semantic domain (*currently cu-alpha*)

$_{csu}\alpha$ indicates the agreement on coding within a semantic domain (*currently not yet implemented*)

As background reading we recommend: Krippendorff, Klaus (2019). Content Analysis: An Introduction to its Methodology. Thousand Oaks, 4th edition. California: SAGE publications.

Currently implemented are the c-alpha binary and the cu & Cu-Alpha coefficients:

Krippendorff's c-Alpha Binary

The C-Alpha Binary coefficient is a measure for the reliability of distinguishing relevant from irrelevant matter. It is applicable if the coders have created quotations themselves. The coefficient will tell you for each code whether the different coders have identified similar or the same areas in relation to a given meaning (code).

Krippendorff's cu-Alpha / Cu-Alpha

The $_{cu}\alpha$ coefficient omits all references to the gaps between valued units and assesses the reliability of coding relevant matter only. It can be used to infer the reliability o**f each semantic domain**.

The $_{Cu}\alpha$ coefficient with a capital C calculates the reliability **for all selected semantic domains considered together**.

The calculations require that codes of semantic domains have been applied mutually exclusively. This means only one of the sub codes per domain is applied to a given quotation. If this rule is violated cu or Cu-Alpha cannot be calculated.

Decision Rules

A number of decisions have to be made when testing for inter-coder agreement. For instance, how much data material needs to be used, how many instances need to be coded with any given code for the calculation to be possible, and how to evaluate the obtained coefficient.

Sample Size

The data you use for the ICA analysis needs to be representative of the total amount of data you have collected, thus of those data whose reliability is in question. Furthermore, the number of codings per code needs to be sufficiently high. As a rule of thumb, the codings per code should at least yield five agreements by chance. Krippendorff (2019) uses Bloch and Kraemer's formula 3.7 (1989:276) to obtain the required sample size. The table below lists the sample size for the three smallest acceptable reliabilities α_{min}: 0.667 / 0.800 / 0.900; for four levels of statistical significance: 0.100 / 0.050 / 0.010 / 0.005, and for semantic domains up to10 codes (probability $p_{c)}$:

Smallest acceptable α	**.667**				**.800**				**.900**			
Significance level	.100	.050	.010	.005	.100	.050	.010	.005	.100	.050	.010	.005
codes per semantic domain												
10 codes or pc = .100	104*	172	344	421	178	293	587	719	361	595	1190	1459
9 codes or pc = .111	95	156	312	383	162	267	534	654	329	542	1083	1328
8 codes or pc = .125	85	141	281	345	146	241	481	590	297	489	980	1198
7 codes or pc = .143	76	125	251	307	130	214	429	526	265	436	872	1069
6 codes or pc = .167	67	110	220	270	114	189	377	462	233	384	768	941
5 codes or pc = .200	58	95	190	233	99	163	326	400	202	332	667	815
4 codes or pc = .250	49	81	161	198	84	139	277	340	172	283	566	694
3 codes or pc = .333	41	67	135	165	71	116	233	285	144	238	477	584
2 codes or pc = .500	36	60	119	146	62	103	206	252	128	211	422	518

*number of codings for the semantic domain after merging

Table 1: Required sample size (adapted from Krippendorff, 2018)

Example: If you have a semantic domain with 4 codes and each of the codes are equally distributed, you need a minimum of 139 codings for this semantic domain if the minimum alpha should be 0.800 at a 0.05 level of statistical significance. For a lower alpha of 0.667, you need a minimum of 81 codings at the same level of statistical significance. If the frequencies across the sub categories per domain are not equally distributed, you need to increase the sample size. By 4 codes in a semantic domain, the estimated proportion p_c of all values c in the population is .250 (¼).

If the distribution of your codes in a semantic domain is unequal, you need to make a new estimate for the sample size by using a p_c in the formula shown in Figure 7that is correspondingly less than ¼.

If your semantic domain has more than 10 codes, you can calculate the needed sample size using the following equation. It applies if you work with two independent coders.

You can see the corresponding z value from a standard normal distribution table. For a p-value of 0,05, z = 1,65.

$$N_c = 2 * Z_p^2 \left(\frac{(1+\alpha_{min})(3-\alpha_{min})}{4(1-\alpha_{min})P_c(1-P_c)} - \alpha_{min} \right)$$

Figure 7: Calculating sample size if your semantic domains have more than 10 codes and you work with 2 coders

The equation to estimate the needed sample size applies when working with 2 coders. If you work with more coders, you need to increase the sample size accordingly.

Acceptable Levels Of Reliability

The last question to consider is when to accept or reject coded data based on the ICA coefficient. Krippendorff's (2019) recommendations are:

- Strive to reach α = 1,000, even if this is just an ideal.
- Do not accept data with reliability values below α = 0.667.
- In most cases, you can consider a semantic domain to be reliable if α ≥ 0.800.
- Select at least 0.05 as the statistical level of significance to minimize the risk of a Type 1 error, which is to accept data as reliable if this is not the case.

Which cut-off point you choose always depends on the validity requirements that are placed on the research results. If the result of your analysis affects or even jeopardizes someone's life, you should use more stringent criteria. A cut-off point of α = 0.800 means that 80% of the data is coded to a degree better than chance. If you are satisfied with α = 0.500, this means 50% of your data is coded to a degree better than chance. The sounds a bit like tossing a coin and playing heads or tails.

How To Set Up A project For ICA Analysis

You need to start from a Master project, set up by the main analyst or principal investigator. The principal investigator adds all documents and codes that will be subjected to the ICA analysis to the Master project and creates sub projects based on the Master project for each coder.

Summar of the steps

- The principal investigator develops the code system providing clear definitions for each code.
- The principal investigator creates a project bundle file for each coder that either contains only the documents to be coded and the code system, or the documents, the quotations, and the code system.

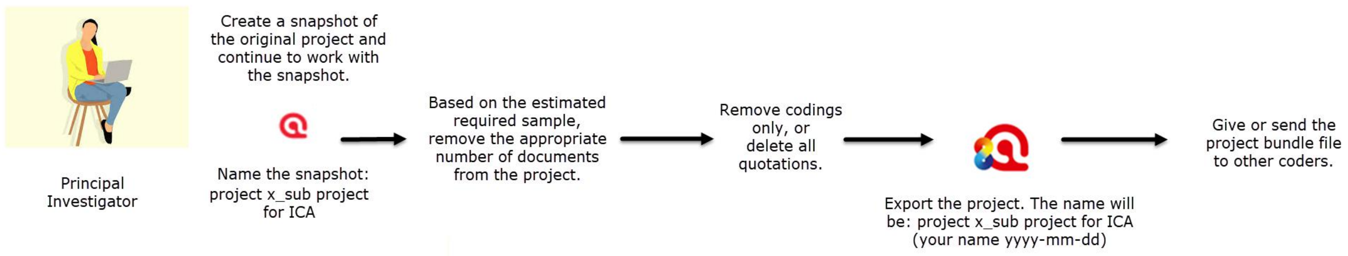

Principal Investigator
Create a snapshot of the original project and continue to work with the snapshot.
Name the snapshot: project x_sub project for ICA
Based on the estimated required sample, remove the appropriate number of documents from the project.
Remove codings only, or delete all quotations.
Export the project. The name will be: project x_sub project for ICA (your name yyyy-mm-dd)
Give or send the project bundle file to other coders.

Figure 8: Project set-up for ICA analysis

- The coders import the project bundle file, rename the project file during the process of import, double-check under which user name they are logged in.
- The coders code the data independently. If they have questions or difficulties applying the codes, they write notes into a memo.
- Once the coders are done, they prepare a project bundle file for the principal investigator.
- The principal investigator imports the project bundle files from each coder and briefly checks the work of each coder and reads the memo they have written (if any).
- The principal investigator starts the inter-coder agreement mode and merges the projects of the coders.
- The principal investigator runs the ICA analysis.

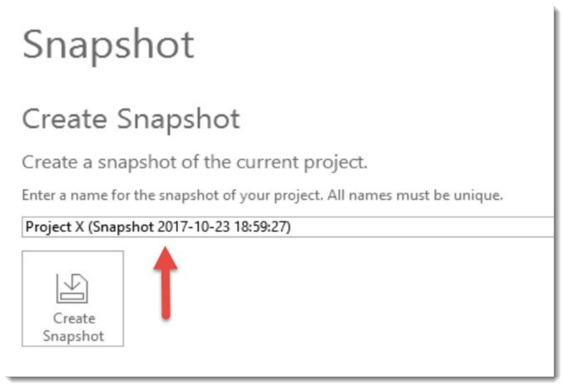
Below you find technial instructions for both the principal investigator and the coders for what they need to know in terms of project management.

Project Management For The Principal Investigator

Creating Snapshots

For an ICA analysis it is important that you create snapshot projects for the coders in order to make sure that all coders work on projects that contain documentsa and codes with identical IDs.

To create a snapshot, select FILE/ SNAPSHOT.


Snapshot

Create Snapshot

Create a snapshot of the current project.

Enter a name for the snapshot of your project. All names must be unique.

Project X (Snapshot 2017-10-23 18:59:27)

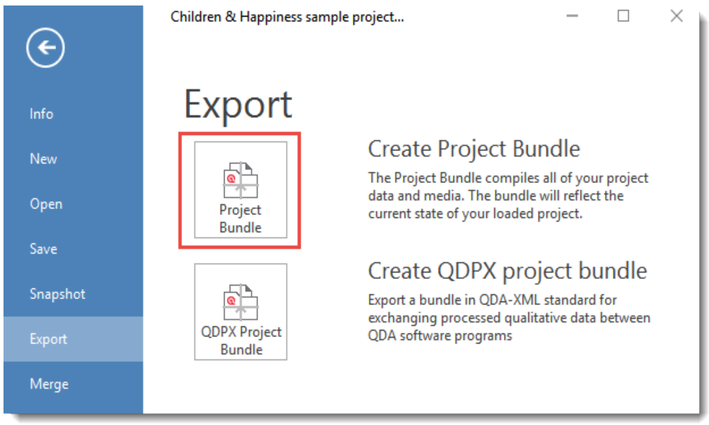
Figure 9: Default name for a project snapshot

Change the default name to indicate that this is a sub project for one of the coders and click CREATE SNAPSHOT.

Just like ATLAS.ti project files, snapshots are saved internally in the ATLAS.ti environment. A snapshot projects also has the same ID as the project from which it is created.

Exporting The Project For Distribution

To export the project for distribution to all team member, select FILE / EXPORT.



Export

Create Project Bundle

The Project Bundle compiles all of your project data and media. The bundle will reflect the current state of your loaded project.

Create QDPX project bundle

Export a bundle in QDA-XML standard for exchanging processed qualitative data between QDA software programs

Figure 10: Creating a project bundle file

Click on the PROJECT BUNDLE button. This opens the Windows File Manager. Select a location for storing the project bundle. The default project bundle name consists of the project name plus the author and date in parentheses behind it, e.g., "**Project X_sub project for ICA (Mary 2017-09-27).**"

Distribute the project bundle file to all coders, e.g. via email, a folder on a server that everyone can access, or also via a cloud link or cloud folder.

The project bundle file has the file extension ".*atlproj*" and can be read by both ATLAS.ti 8.x Windows and Mac.

Project bundle files can be shared via a cloud services like Dropbox, OneDrive, GoogleDrive, etc. - Another option is to upload the project bundle file to a server of your choice and send a link to all team members so that they can download the file.

The default name of the bundle is the project name + author and date. Renaming the bundle does not change the name of the project contained within the bundle. Think of the project bundle file as a box that contains your ATLAS.ti project plus all documents that you want to analyze. Putting a different label on the outside of the box will not change the name of the project file, which is contained inside the box.

Merging Projects

The Merge Tool reunites projects that were originally divided for analytical or economical reasons. It brings together the contributions of the coders. In the following a few general principles about merging are explained:

Master And Imported Projects: The Master project is the project into which another project, called the "Import project" is merged. The Master project has to be loaded first before invoking the **Merge Project** option.

Merge Strategy

The default option is to unify all objects that are **identical** and to add all objects that do not yet exist in the Master project. This is why it is so important to start from a common Master project, create sub projects from it and distribute those to the coders.

Identical Entities

When an entity is created in ATLAS.ti– regardless if it is a document, a code, a quotation, a memo, a network,a group or a comment--this entity receives a unique ID, comparable to a fingerprint. When merging projects, ATLAS.ti compares the IDs of the various entities. If they have the same ID (fingerprint), they are unified. If the fingerprint is different, they are added.

As coders do not add codes when serving as coders for an ICA analysis, there should never be duplicated codes!

Identical entities are unified, all others are added.

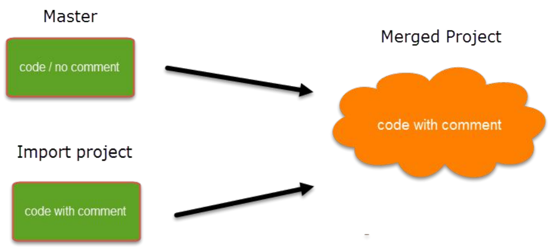


Figure 11: Merge example 2: Entities with comments, no conflict

Entities with and without comments: If there is a code C in the Master project that has no comment, and a code C in the Import project that has a comment, the merged procedure will add this comment to the merged project.

In the case of an unsolvable conflict (e.g., code C in the Master project has a comment, and code C in the Import project also has a comment) the user can define which of the two conflicting entities will win. If the comment of the Master project should be kept, you need to select the option "**Keep**." If the comment of the Import project should be kept, you need to select the option "**Override,**" as shown in Figure 13.

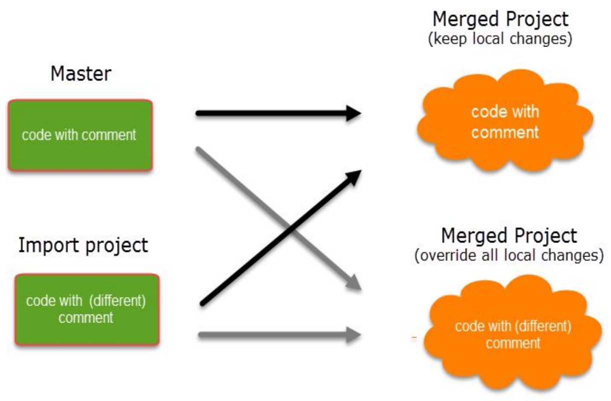


Figure 12: Merge example 3: Entities with conflicting comments

Coders should be instructed not to change code comments. If they want to write notes, they should create a memo and use it as writing space. **If you merge a project for ICA analysis, you should never encounter a merge conflict related to codes!**

How to Merge

Before you merge, the recommendation is that you import all project bundle files first and take a look at the projects. This way you can see whether the coders have done their work as instructed.

- Import the project bundle files of code 1 and 2.
- Open the project of coder 1 and create a snapshot. Do not accept the default name, enter a name that indicates that this is the project for the ICA test for coder 1 and 2.
- Close the project of coder 1 and continue to work with the renamed snapshot.

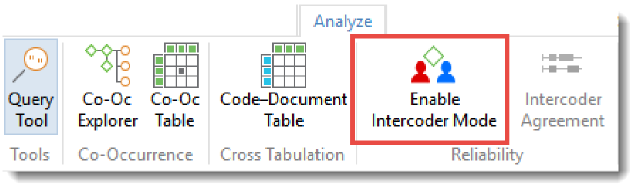
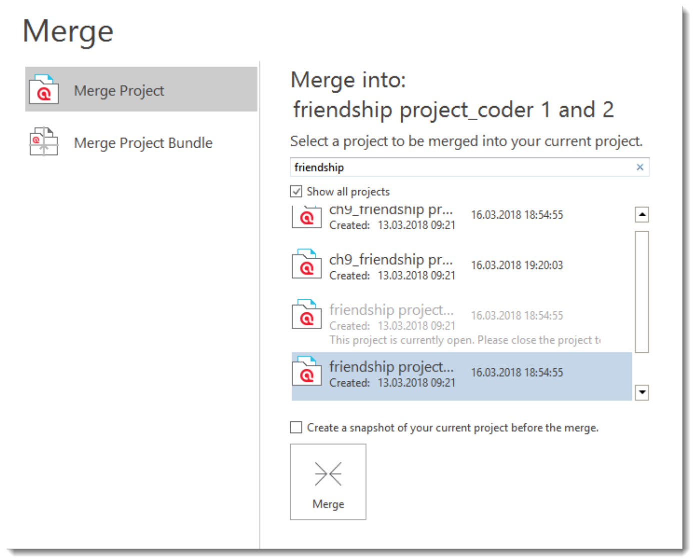


Figure 13: Enabling ICA mode

- Select Analyze / Enable Intercode Mode.

- Select File / Merge.
- Select the Merge Project option and select the project of coder 2.



Merge

Merge Project

Merge Project Bundle

Merge into:
friendship project_coder 1 and 2

Select a project to be merged into your current project.

friendship

☑ Show all projects

ch9_friendship pr... Created: 13.03.2018 09:21	16.03.2018 18:54:55	
ch9_friendship pr... Created: 13.03.2018 09:21	16.03.2018 19:20:03	
friendship project... Created: 13.03.2018 09:21 This project is currently open. Please close the project t	16.03.2018 18:54:55	
friendship project... Created: 13.03.2018 09:21	16.03.2018 18:54:55	

☐ Create a snapshot of your current project before the merge.

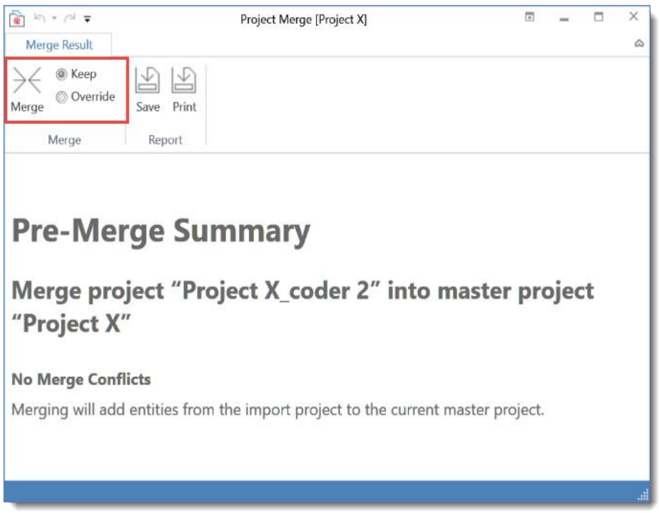
Merge

Figure 14: Merging projects of coder 1 and 2

- Deactivate the option to create a snapshot as you are already working with a snapshot.
- Click on the Merge button.

ATLAS.ti checks the two projects for identical and different items. All identical items will be merged, all different items will be added. See "Merge Strategy“. In ICA mode, the codings of the different coders are added. This way, you can see which coder has applied which code to a quotation and ATLAS.ti can compare the codings in the further analysis.

After this process is completed, you see a **Pre-Merge Summary:**



Pre-Merge Summary

Merge project "Project X_coder 2" into master project "Project X"

No Merge Conflicts

Merging will add entities from the import project to the current master project.

Figure 15: Check the pre-merge summary for conflicts

If the project has been set-up as described above and the coders have followed the instructions, there should be no merge conflict.

Click on MERGE.

After merging, check the final merge report and the merged project for plausibility. If you are satisfied with the results, save the project. If not, you can always select UNDO.

If you work with more than two coders, continue with merging the next sub-project.

Project Management For Coders

User Accounts

ATLAS.ti automatically creates a user account based on the name that you use on your computer.

If you want to check under which name you are working, select the TOOLS & SUPPORT tab and then USER MANAGER.

New User

To create a new user account, select USER MANAGER. Next, click on the button NEW USER at the left-hand side in the ribbon.

After creating a new account, select SWITCH USER to log in using a different user name.

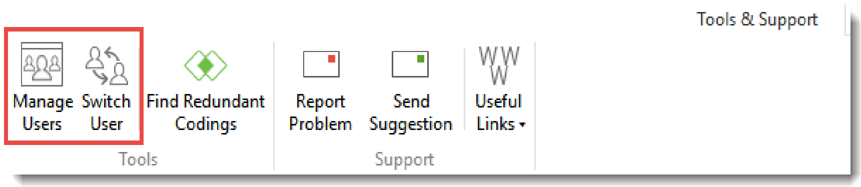


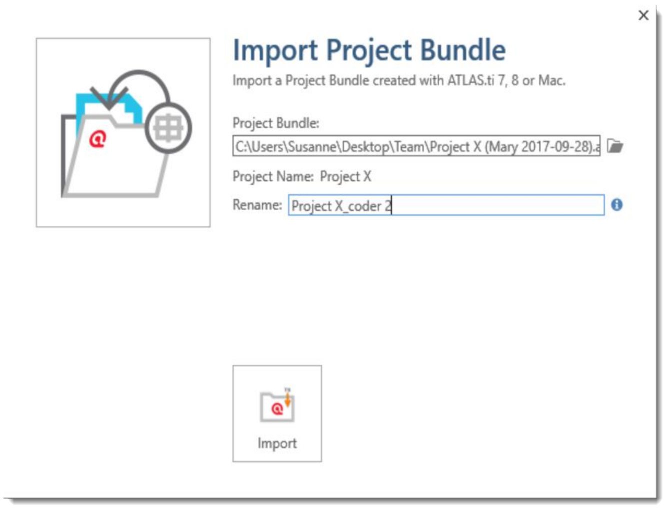
Figure 16: User Management tab in ATLAS.ti 8 Windows

A reason for creating a new user account is if two persons on a team have the same name. When merging projects that come from different computers and that use the same user name, the user name will be duplicated (e.g. Tom and Tom (2). This means you can still distinguish the two users. However, it may be a bit cumbersome to always remember who is who.

Importing The Master Project And Renaming It

Open ATLAS.ti and select the IMPORT PROJECT BUNDLE button. If another project is currently open, select FILE / NEW and then the Import Project Bundle option.

During the process of importing the project, he name of the project can be changed:



Import Project Bundle

Import a Project Bundle created with ATLAS.ti 7, 8 or Mac.

Project Bundle: C:\Users\Susanne\Desktop\Team\Project X (Mary 2017-09-28).a

Project Name: Project X

Rename: Project X_coder 2

Figure 17: Renaming a project file during the process of importing a project bundle

Rename the project by adding your name or initials to the project name. This is important for the project administrator later when he or she merges the projects of all team members (see Figure 11).

Click IMPORT.

Renaming Projects On the Opening Screen

If you want to rename a project after it is imported, you can either do it on the opening screen when starting ATLAS.ti. If the project is already open, you need to close it to return to the opening screen.

On the opening screen, right click on the project that you want to rename and select RENAME PROJECT.

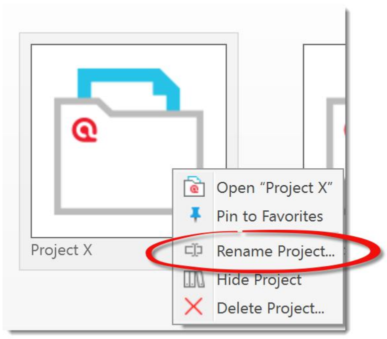


Figure 18: Renaming a project on the opening screen

Exporting Sub-Projects For The Principal Investigator

After the coders are done coding the data, they create their own project bundle file and send it to the principal investigator.

To export the project, select FILE / EXPORT. See "Figure 10". The name of the exported bundle file will include the name of the currently logged in user and the date, e.g. P*roject X_sub project for ICA_coder 2 (Tom 2017-09-30)*.

It is recommend that all team members save a copy of the project bundle file on their computer, or their personal space on a server.

Merging Projects For ICA Analysis

It is recommend to import the project bundles files of the coders first and to check what the coders have done and whether they followed the instructions.

- Import the project bundle files of all coders.
- Open the project of coder 1 and create a snapshot. Do not accept the default name, enter a name that indicates that this is the project for the ICA test for coder 1 and 2.
- Close the project of coder 1 and continue to work with the renamed snapshot.

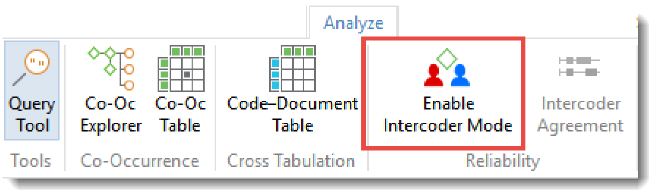
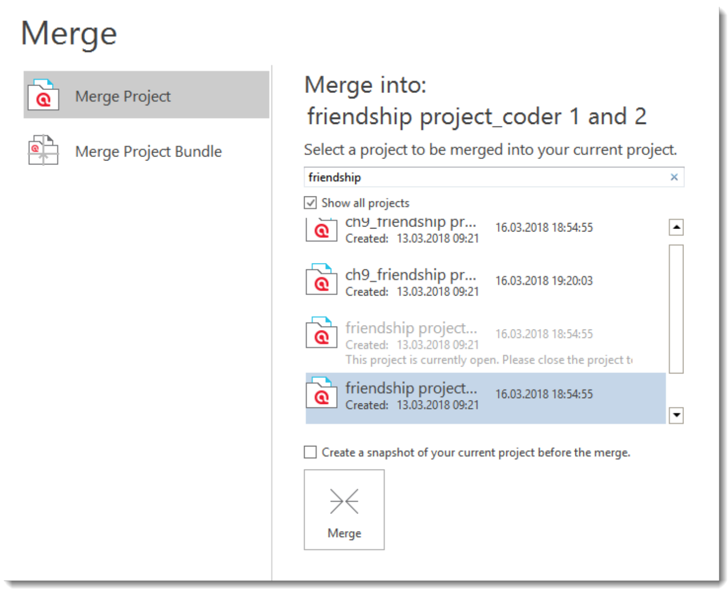


Figure 19: Enable Intercoder Mode

- Select Analyze / Enable Intercode Mode.
- Next, merge the project of coder 2 and save the merged project. Deactivate the option to create a snapshot as you already merged into a snapshot of the project.



Merge into:
friendship project_coder 1 and 2
Select a project to be merged into your current project.
friendship
☑ Show all projects

ch9_friendship pr... Created: 13.03.2018 09:21	16.03.2018 18:54:55
ch9_friendship pr... Created: 13.03.2018 09:21	16.03.2018 19:20:03
friendship project... Created: 13.03.2018 09:21 This project is currently open. Please close the project t	16.03.2018 18:54:55
friendship project... Created: 13.03.2018 09:21	16.03.2018 18:54:55

☐ Create a snapshot of your current project before the merge.

Figure 20: Merging the projects of the coders

In ICA mode, the 'codings' of all coders will become visible.

A coding is the link between a code an a quotation. In ICA mode, you can see which coder has applied which code to a quotation.

- To see the authors of the various codings, change the view settings from icons to user in the **View tab** of the contextual Document ribbon:

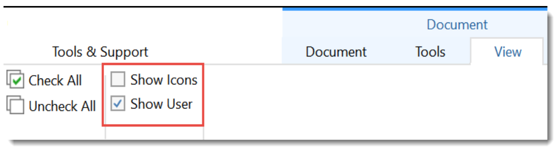


Figure 21: View settings suitable for ICA testing

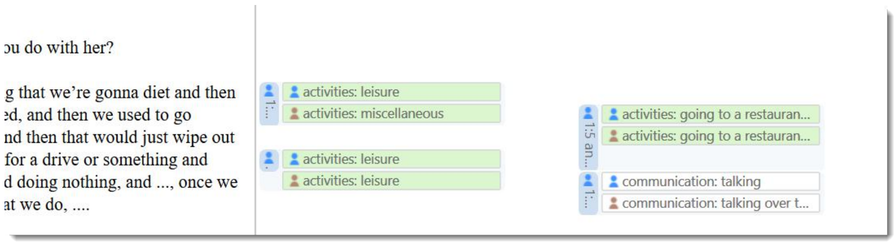

activities: leisure
activities: miscellaneous
activities: leisure
activities: leisure
activities: going to a restauran...
activities: going to a restauran...
communication: talking
communication: talking over t...

Figure 22: Display of the codings of different users

Each coder is indicated by a different color. Hover over the user icon to see the name of the user. In Figure 22, all quotations where created by the blue user Susanne. The brown user was asked to apply the codes of the code system to these predefined quotations.

Preparing For ICA Analysis

- Click on the INTERCODER AGREEMENT button in the ANALYZE tab.
- Click on the ADD CODER button and select two or more coders.
- Click on the ADD DOCUMENTS button and select the documents that should be included in the analysis. The default option is for all selected documents to be regarded as one continuum and all coded quotations of these documents will go into the calculation. You can, however, also view agreements / disagreements per document and get a coefficient for each document.

Only the documents are shown that have been coded by the selected coders.

- Click on the ADD SEMANTIC DOMAIN button and add one or more codes to the domain. Alternatively, you can drag and drop codes from the Project Explorer or the code list into the semantic domain field:

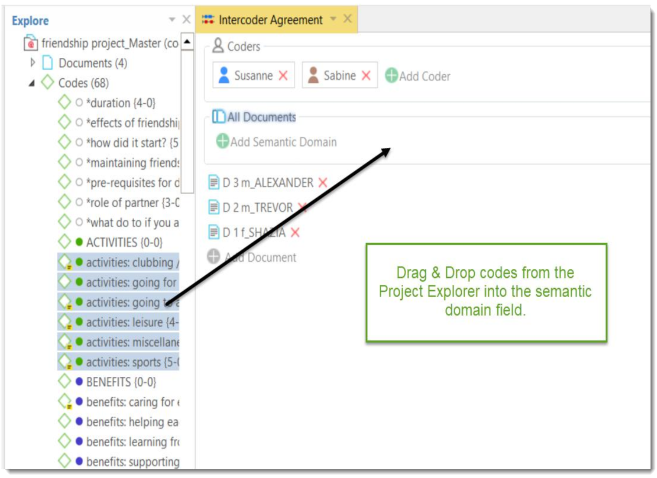


Figure 23: Adding codes to a semantic domain

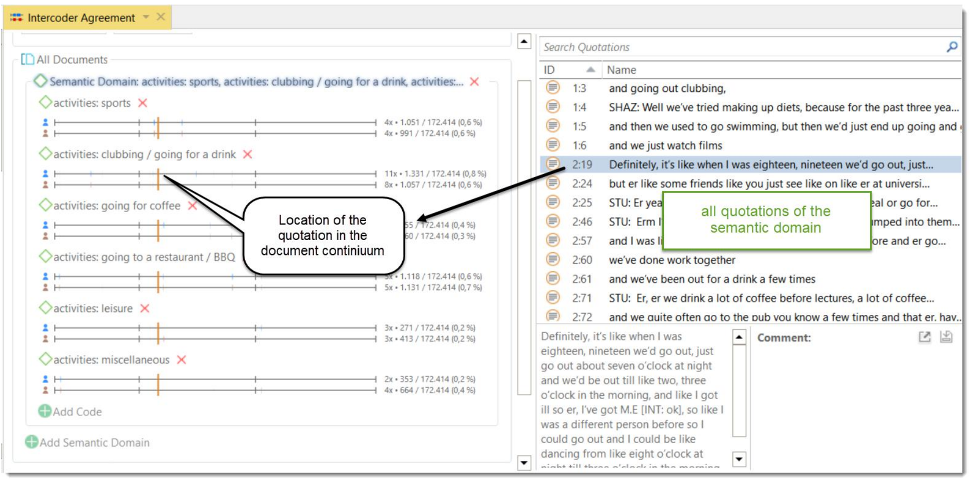
- Repeat this process for each semantic domain.

As there currently is no function to define a semantic domain formerly, all codes that you are adding from the list will be considered to belong to a semantic domain. As you can see in Figure 23, semantic domains were already built indirectly in the process of developing the code system by adding the same prefix to all codes that belong to the same semantic domain like

ACTIVITIES or BENEFITS. Please remember that all sub codes of a semantic domain (i. e., all codes with prefixes) must be applied to quotations in a mutual exclusive manner. See "Requirements For Coding".

A semantic domain can also consist of only one code.

After you have added coders, documents and codes, your screen looks similar to what is shown in Figure 24.


Intercoder Agreement
All Documents
Semantic Domain: activities: sports, activities: clubbing / going for a drink, activities:...
activities: sports
4x • 1.051 / 172.414 (0,6 %)
4x • 991 / 172.414 (0,6 %)
activities: clubbing / going for a drink
11x • 1.331 / 172.414 (0,8 %)
8x • 1.057 / 172.414 (0,6 %)
activities: going for coffee
activities: going to a restaurant / BBQ
5x • 1.131 / 172.414 (0,7 %)
activities: leisure
3x • 271 / 172.414 (0,2 %)
3x • 413 / 172.414 (0,2 %)
activities: miscellaneous
2x • 353 / 172.414 (0,2 %)
4x • 664 / 172.414 (0,4 %)
Add Code
Add Semantic Domain
Location of the quotation in the document continiuum
Search Quotations
all quotations of the semantic domain
Comment:

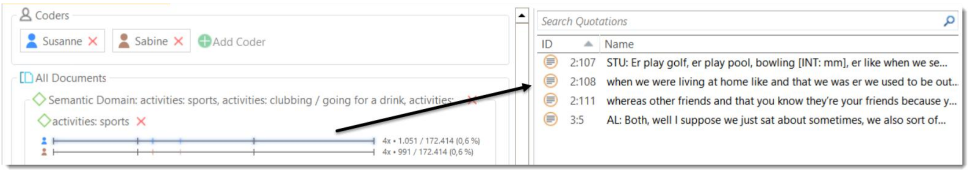
ID	Name
1:3	and going out clubbing,
1:4	SHAZ: Well we've tried making up diets, because for the past three yea...
1:5	and then we used to go swimming, but then we'd just end up going and
1:6	and we just watch films
2:19	Definitely, it's like when I was eighteen, nineteen we'd go out, just...
2:24	but er like some friends like you just see like on like er at universi...
2:25	STU: Er yea ... al or go for...
2:46	STU: Erm I ... mped into them...
2:57	and I was l ... ore and er go...
2:60	we've done work together
2:61	and we've been out for a drink a few times
2:71	STU: Er, er we drink a lot of coffee before lectures, a lot of coffee...
2:72	and we quite often go to the pub you know a few times and that er, hav...

Definitely, it's like when I was eighteen, nineteen we'd go out, just go out about seven o'clock at night and we'd be out till like two, three o'clock in the morning, and like I got ill so er, I've got M.E [INT: ok], so like I was a different person before so I could go out and I could be like dancing from like eight o'clock at

Figure 24: ICA view after adding coders, documents and codes

The quotations of the semantic domain are displayed on the right-hand side. When you select a quotation, the full content is shown in the preview and you can write comments. You also see where the quotation is located in the document continuum. When you scroll to the right in the quotation view window, you see the codes that have been applied by the selected coders.

If you click on one of the lines, only the quotations of the selected code and coder are shown in the quotation window (Figure 25).


Coders
Susanne
Sabine
Add Coder
All Documents
Semantic Domain: activities: sports, activities: clubbing / going for a drink, activities:
activities: sports
4x • 1.051 / 172.414 (0,6 %)
4x • 991 / 172.414 (0,6 %)
Search Quotations

ID	Name
2:107	STU: Er play golf, er play pool, bowling [INT: mm], er like when we se...
2:108	when we were living at home like and that we was er we used to be out...
2:111	whereas other friends and that you know they're your friends because y...
3:5	AL: Both, well I suppose we just sat about sometimes, we also sort of...

Figure 25: Quotations coded by coder Susanne with the code activities: sport

You can remove coders, codes or documents from the analysis by clicking on the red x.

To the right of each coder line you see some descriptive statistics:

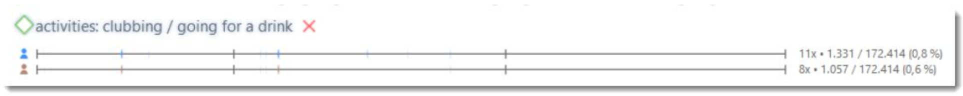

activities: clubbing / going for a drink
11x • 1.331 / 172.414 (0,8 %)
8x • 1.057 / 172.414 (0,6 %)

Figure 26: Descriptive statistics

The blue coder (Susanne) has applied the code "activities clubbing / going for a drink" 11 times. The number of characters that are coded with this code are 1331 out of a total of 172.414 characters in all three selected documents, which is 0,8%.

The brown coder (Sabine) has applied the code "activities clubbing / going for a drink" 8 times. The number of characters that are coded with this code are 1057 out of a total of 172.414 characters in all three selected documents, which is 0,6%.

Calculating An ICA Coefficient

To calculate ICA, select one of the four methods from the ribbon. See "Methods For Testing ICA" for more detail

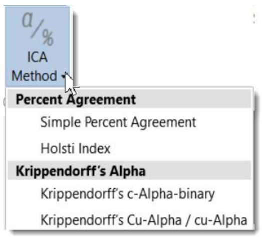
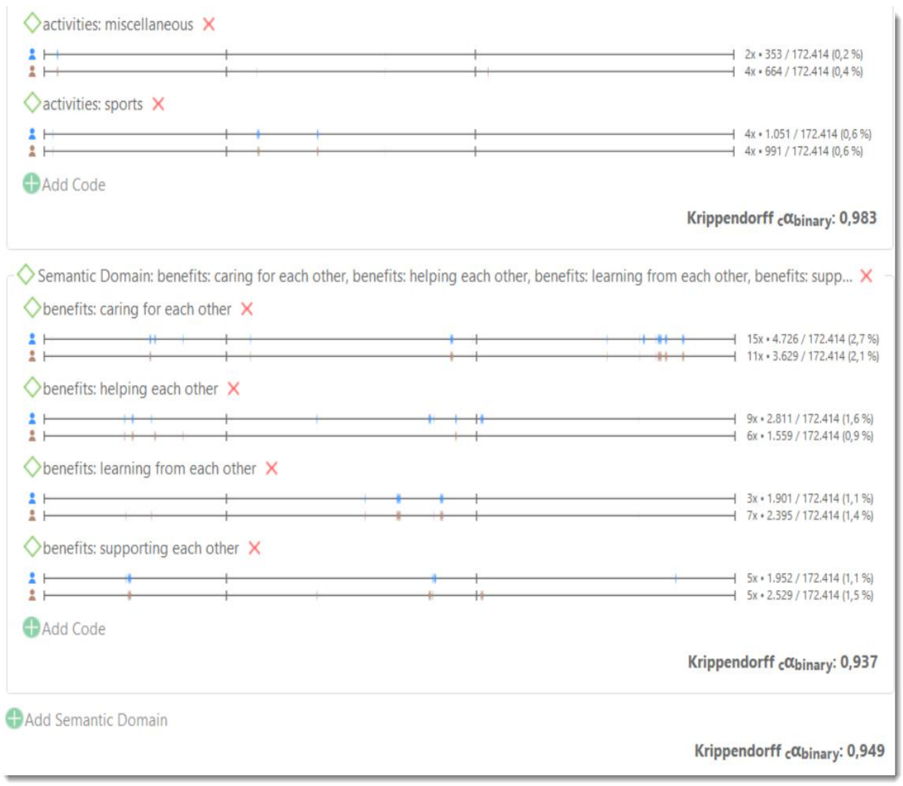


Figure 27: Select an ICA method

In Figure 28, the c-alpha binary for the semantic domain 'activities' is 0, 983 and for the semantic domain 'benefits' is 0,937, For both domains, it is 0,949. This means that the reliability for identifying relevant from irrelevant matter related to the codes of these domain is high.


activities: miscellaneous
2x • 353 / 172.414 (0,2 %)
4x • 664 / 172.414 (0,4 %)

activities: sports
4x • 1.051 / 172.414 (0,6 %)
4x • 991 / 172.414 (0,6 %)

Add Code

Krippendorff $_c\alpha_{binary}$: 0,983

Semantic Domain: benefits: caring for each other, benefits: helping each other, benefits: learning from each other, benefits: supp...

benefits: caring for each other
15x • 4.726 / 172.414 (2,7 %)
11x • 3.629 / 172.414 (2,1 %)

benefits: helping each other
9x • 2.811 / 172.414 (1,6 %)
6x • 1.559 / 172.414 (0,9 %)

benefits: learning from each other
3x • 1.901 / 172.414 (1,1 %)
7x • 2.395 / 172.414 (1,4 %)

benefits: supporting each other
5x • 1.952 / 172.414 (1,1 %)
5x • 2.529 / 172.414 (1,5 %)

Add Code

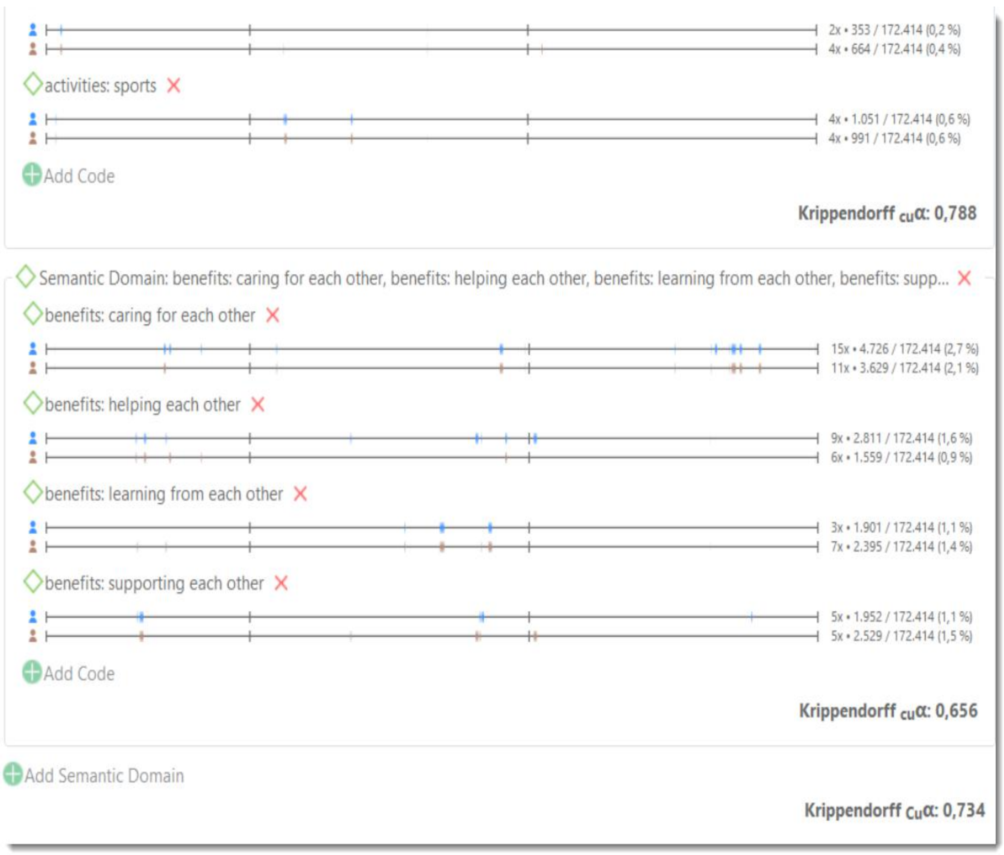
Krippendorff $_c\alpha_{binary}$: 0,937

Add Semantic Domain

Krippendorff $_c\alpha_{binary}$: 0,949

Figure 28: Display of results for c-alpha binary

The results for cu-alpha and Cu-alpha for the two selected domains are shown in Figure 29.



2x • 353 / 172.414 (0,2 %)
4x • 664 / 172.414 (0,4 %)

activities: sports

4x • 1.051 / 172.414 (0,6 %)
4x • 991 / 172.414 (0,6 %)

Add Code

Krippendorff $_{cu}\alpha$: 0,788

Semantic Domain: benefits: caring for each other, benefits: helping each other, benefits: learning from each other, benefits: supp...

benefits: caring for each other

15x • 4.726 / 172.414 (2,7 %)
11x • 3.629 / 172.414 (2,1 %)

benefits: helping each other

9x • 2.811 / 172.414 (1,6 %)
6x • 1.559 / 172.414 (0,9 %)

benefits: learning from each other

3x • 1.901 / 172.414 (1,1 %)
7x • 2.395 / 172.414 (1,4 %)

benefits: supporting each other

5x • 1.952 / 172.414 (1,1 %)
5x • 2.529 / 172.414 (1,5 %)

Add Code

Krippendorff $_{cu}\alpha$: 0,656

Add Semantic Domain

Krippendorff $_{Cu}\alpha$: 0,734

Figure 29: Display of results for cu-alpha and Cu-alpha

The difference between the two measures is that the lower case cu-alpha applies to one semantic domain only, the capital Cu-alpha applies to all selected semantic domains. It is not simply an average of all lower case cu-alphas, as sub codes from different domains can potentially be applied to one quotations. This is referred to as multi-valued coding (see also Krippendorff et al., 2016) and "Multi-valued Coding".

References

Cohen, Jacob (1960). A coefficient of agreement for nomincal scales. E*ducational and Psychological Measurement*, 20, 37–46.

Holsti, O. R. (1969). *Content Analysis for the Social Sciences and Humanities*. Reading, MA: Addison-Wesley

Krippendorff, Klaus (2004/2019). *Content Analysis: An Introduction to Its Methodology (3rd and* 4th edition). Thousand Oaks, CA: Sage.

Zwick, Rebecca (1988). Another look at interrater agreement. *Psychological Bulletin*, 103, 347–87.

Appendix

Default Location For ATLAS.ti Project Data

ATLAS.ti projects and all documents that you add to a project are stored in the application folder on your computer. The Windows application folder is called: AppData and can be found under your user account's home folder. Within the sub folder Roaming, you find folders from a number of different applications, also for ATLAS.ti. The AppData\Roaming folder is a hidden folder and can either be accessed by entering the full path name or by typing `%appdata%` into the Windows 10 search field. Exmaple: If the user name is "Mary." the full path would be `c:\Users\Mary\Appdata\Roaming\Scientific Software\ATLASti.8`

This default location for storing ATLAS.ti data can be changed. See below.

The ATLAS.ti library is a system folder and not meant for user access. Files in this folder cannot be used outside ATLAS.ti.

Changing the Default Location for ATLAS.ti Project Data

As explained above, the default location for ATLAS.ti project files is on the C drive. As it is not possible for all users to work on the C drive, either because there are institutional restrictions, tor the C drive is already full and there is not sufficient space, there is the possibility to create a user-defined location where all of your ATLAS.ti data is stored.

For all users familiar with ATLAS.ti 7: This is NOT the same as in version 7 where you could create project-specific libraries that could be shared.

In ATLAS.ti 8, each user works within her or his own ATLAS.ti environment. What is new in version 8.1 is that you can define where this ATLAS.ti environment is located. This location can be within another folder on the C: drive, any other drive on your computer, a server, or an external drive.

It is not possible, however, to use a cloud sharing service like Dropbox because the specific way in which such systems work can jeopardize the integrity of your ATLAS.ti library.

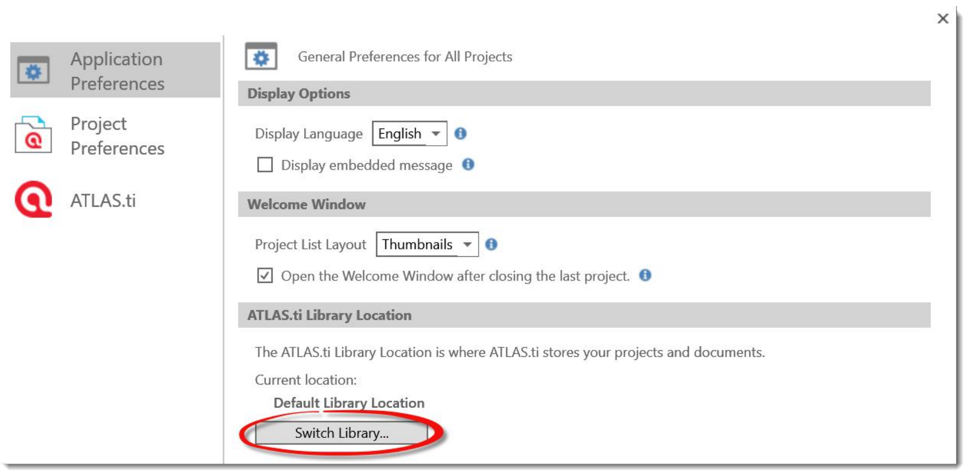
It is possible to work with multiple libraries. Theoretically, you could create a new empty library every time you start a new project. This, however, requires careful data management on your part as you need to keep track on your own where all of these different libraries are located.

We strongly recommend that you create project backups on a regular basis by exporting your projects. To do so select FILE / EXPORT and create a **Project Bundle File.**

How To Change The Default Location

When you start ATLAS.ti, select OPTIONS at the bottom left of your screen. If a project is already open, you need to close it first.

The last option under Application Preferences lets you change the library location or select a different library (SWITCH LIBRARY).



Display Options

Display Language: English

☐ Display embedded message

Welcome Window

Project List Layout: Thumbnails

☑ Open the Welcome Window after closing the last project.

ATLAS.ti Library Location

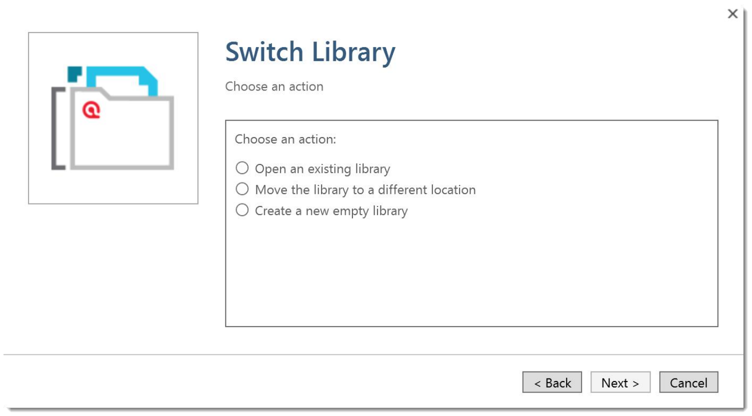
The ATLAS.ti Library Location is where ATLAS.ti stores your projects and documents.

Current location:

Default Library Location

Figure 30: Start the Switch Library process

A wizard opens that guides you through the process. The next choice you need to make is whether you want to open an existing library, move the library to a different location, or create a new empty library.



Choose an action:

○ Open an existing library
○ Move the library to a different location
○ Create a new empty library

Figure 31: Different options for library switching

Open an existing library: Chose this option if you work with multiple libraries and you want to gain access to an already existing library at a different location.

Move the library to a different location: Chose this option if you want or need to move your current library to a different location, e.g. because you do no longer want to work on the C drive. It is possible to keep a copy of the library at the old location.

Note that the ATLAS.ti 8 library contains all of your documents plus your "project file." This is different from version 7, where only the project documents were stored in the library and the user saved the project file (HU) independently of it. In version 8, ATLAS.ti handles and manages ALL of your project-related data in the library.

Create a new empty library: This option allows you to create a new library at a location of your choice. If you already have projects, none of these will be moved over into the new library. To fill this library, you either create new projects from scratch, or you import project bundle files that you previously exported from other libraries.

- After you made your choice, click NEXT.
- If you selected to **open an existing library**, select the location for this library:

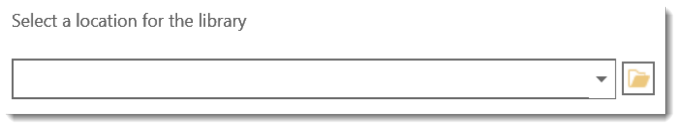
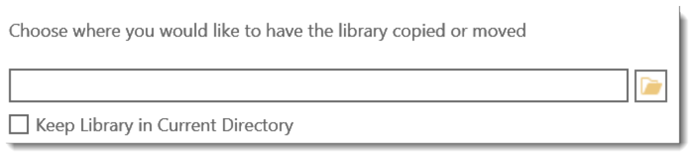


Figure 32: Select the location of the existing library

If you selected to **move the library**, select the location where you want this library to be. You can keep a copy of the actual state of the library at the current location. Just keep in mind that this copy is not updated if you continue your work using the library at the new location.



Choose where you would like to have the library copied or moved
Keep Library in Current Directory

Figure 33: Select where you want this library to be stored

If you selected to **create a new empty library,** select where this new library shall be created. After confirming the new location, your opening screen will be empty and you can begin to fill it with new projects.

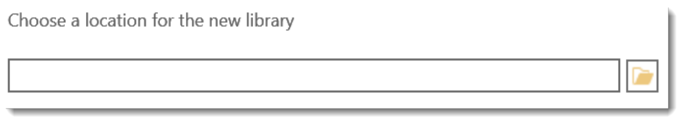


Figure 34: Select a location where you want your new library to be stored

A note from the HelpDesk: We know that your project data is very valuable to you and that you do not want to lose it. Therefore, please select a sensible location for the ATLAS.ti library. For instance: Do not store it on the desktop. You probably have a folder within your document system for the project you are currently working on where you store all project related data. If you cannot or do not want to use the ATLAS.ti default location, we recommend that you create a sub-folder for your ATLAS.ti-related work within your project folder, and within this sub-folder, a dedicated folder for your library. Make sure that your project folder is included in any automated back-up routine of your computer!