Visual Qualitative Data Analysis Management Model Building in Education Research & Business

Short User’s Manual

Version 4.1 for Windows 95 and Windows NT
Copyright © 1997 Thomas Muhr, Scientific Software Development, Berlin. All rights reserved.

Copying or duplicating this manual or any part thereof is a violation of the law. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including, but not limited to photocopying, without written permission from Scientific Software Development, Berlin.

ATLAS/ti - Visual Qualitative Data Analysis - Management - Model Building- Release 4.1

LICENSE AGREEMENT

1. LICENSE RIGHTS AND RESTRICTIONS. Scientific Software Development grants to you the non-exclusive, non-assignable right to use the enclosed software in object code form (the "ATLAS/ti Code") on as many computer systems as are granted by the type of license. You may not reverse engineer, decompile, or disassemble the ATLAS/ti Code, except to the extent that the foregoing restriction is expressly prohibited by applicable law. You may not rent or lease the ATLAS/ti Code, or otherwise transfer the ATLAS/ti Code and accompanying written materials. All rights not expressly granted are reserved by Scientific Software Development.

2. NO WARRANTIES. The ATLAS/ti Code and accompanying written materials are provided "as is", without warranty of any kind. To the maximum extent permitted by law, Scientific Software Development disclaims all warranties, either express or implied, including but not limited to implied warranties of merchantability, fitness for a particular purpose and noninfringement. The entire risk arising out of the use or performance of the ATLAS/ti Code and any accompanying written materials remains with you.

3. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law: in no event shall Scientific Software Development, its suppliers, subsidiaries or representatives be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the ATLAS/ti Code, even if Scientific Software Development has been advised of the possibility of such damages.

4. MISCELLANEOUS. This Agreement is governed by international laws. If either party employs attorneys to enforce any rights arising out of or relating to this Agreement, the prevailing party shall be entitled to recover reasonable attorneys fees, costs and expenses.

Trademarks

ATLAS.ti is a registered trademark of Scientific Software Development. Adobe Acrobat is a trademark of Adobe Inc.; IBM and OS/2 are registered trademarks of IBM Corporation; Microsoft Windows, Word for Windows are trademarks of Microsoft Corporation; SPSS is a trademark of SPSS Inc.; Netscape is a trademark of Netscape Inc.. Other product names mentioned herein are used for identification purposes only, and may be trademarks of their respective companies.
Contents

Introduction ......................................................... 1
Foreword .................................................................. 1
Objectives ............................................................ 1
A VISE Principle .................................................... 2
Visualization .......................................................... 2
Integration ............................................................. 2
Serendipity ............................................................. 2
Exploration ............................................................ 2
Design Objectives .................................................. 3
About this Manual ................................................ 3
Viewing and Printing the "real" Thing ......................... 3
Viewing the Manual from CD-ROM .......................... 4
Installing the Manual on your Computer ................... 4
Manual on the Web ................................................ 4
Online Help ............................................................ 4

Installing ATLAS/ti ................................................. 5
System requirements ............................................. 5
Previous Installations .......................................... 5
Upgrading from ATLAS/ti for Windows 4.0 .............. 5
Starting the Installation ........................................ 5
Windows 3.1 and Windows for Workgroups 3.11 ........ 6
Network Installations ............................................ 6
Uninstalling ATLAS/ti ............................................ 6

Main Concepts ..................................................... 7
The Process .......................................................... 7
Textual Level Work ............................................... 7
Conceptual Level Work ......................................... 7
Organizational Level ............................................. 7
General Steps when Working with ATLAS/ti .......... 8
The Hermeneutic Unit ............................................. 8
Primary Documents .............................................. 9
Primary Document Media Types ............................ 9
Quotations .......................................................... 10
Quotations as Layers ........................................... 10
Types of Quotations ............................................. 10
Codes .................................................................... 11
Super Codes ....................................................... 11
Memos .................................................................... 11
Families .............................................................. 12
Networks .............................................................. 12
General Properties of Objects ................................. 12
Comments .......................................................... 12
Authorship ............................................................ 12

Main Workspace - The HU Editor ........................... 13
Overview ............................................................. 13
Components of the HU Editor ......................................................... 13
Title Bar .................................................................................. 13
Main Menu .............................................................................. 14
Tool Bars ................................................................................ 14
Object Drop-Down Lists ......................................................... 15
Primary Document Pane ......................................................... 15
Line Number Area .................................................................... 15
Margin Area ............................................................................ 15
The Pop-up Menus ................................................................... 16
The Object Lists ........................................................................ 18
Opening an Object List ......................................................... 18
Selecting Items in Object Lists .............................................. 18
Primary Document List .......................................................... 19
Quotation List .......................................................................... 19
Code List ................................................................................ 19
Memo List ............................................................................... 19
Lists in Margin Area ............................................................... 20

A Quick Tour for Beginners ..................................................... 21
Getting Started ........................................................................ 21
Starting the System ............................................................... 21
Select a Primary Document ................................................... 22
Display a Quotation ............................................................... 22
Display Codes for a Quotation ............................................... 22
Display quotations for codes .................................................. 22
Memos .................................................................................... 23
The Object Explorer ............................................................... 23
Creating a New Hermeneutic Unit .......................................... 23
Assign Primary Documents .................................................... 24
Creating a Quotation ............................................................. 24
Coding ..................................................................................... 25
A Little Drag & Drop .............................................................. 26
A Short Excursion into Networking .......................................... 27
Don’t be Afraid of the QueryTool ........................................... 27
Create a Super Code ............................................................. 28

The Textual Level-Basic Functions ........................................... 29
Overview ................................................................................ 29
Starting and Closing an ATLAS/ti Session ............................... 29
Starting an ATLAS/ti Session .................................................. 29
Logging In ............................................................................... 29
Leaving ATLAS/ti ................................................................. 29
Creating, Opening and Saving a Hermeneutic Unit ................. 29
Creating a Hermeneutic Unit .................................................. 29
Preparing Primary Documents .............................................. 30
Preparing Textual PDs ........................................................... 30
Preparing Graphical and Audio PDs ...................................... 31
Primary Document Management ........................................... 32
Assigning Primary Documents .............................................. 32
Primary Documents are Accessed by Reference ..................... 32
Disconnecting Primary Documents ....................................... 32
Changing Path or Filename ................................................... 32
ANSI or ASCII? Changing the Encoding Type ....................... 33
Rearranging PDs ................................................................. 33
Renumbering all Primary Documents ................................... 33
Working on Primary Documents .............................................. 33
  Loading a Primary Document .............................................. 33
  Navigating within Primary Documents .................................. 34
  Marking Text Segments ...................................................... 35
  Creating Quotations .......................................................... 36
  Modifying Quotations ....................................................... 37
  Deleting Quotations .......................................................... 37
Special Handling of Non-Textual Primary Documents .................. 37
  Handling of Graphical Primary Documents ......................... 37
  Special Handling of Audio PDs ............................................ 38
Editing Primary Documents ..................................................... 38
  Primary Document are Read-Only by Nature .......................... 38
Coding Techniques ............................................................... 39
  Introduction ........................................................................ 39
  General Coding Procedure .................................................. 39
  How Codes are Displayed .................................................... 40
  Open Coding ....................................................................... 41
  In-Vivo Coding ................................................................... 41
  Code-by-List ....................................................................... 42
  Code-by-List - Inverse .......................................................... 43
  Quick Coding ...................................................................... 43
  Writing Code Comments ...................................................... 43
More Coding Related Functions ................................................ 44
  Free Codes ......................................................................... 44
  Importing a List of Codes ..................................................... 44
  Renaming a Code ............................................................... 44
  Deleting Codes ................................................................... 45
  Unlinking Codes .................................................................. 45
Decoding: Retrieving Coded Quotations ...................................... 46
  Click-Retrieval in Context .................................................... 47
  Print Report of Coded Quotations .......................................... 47
  Complex Queries .................................................................. 47
  Other Places to Activate Quotations ....................................... 47
Writing Memos ........................................................................ 48
  Difference between Memos and Codes .................................. 48
  Difference between Memos and Comments ............................ 48
  Memoing Primary Document Selections ................................. 48
  Memo Types ....................................................................... 49
Creating Reports ..................................................................... 49
  Output Targets ..................................................................... 49
  History of All Objects .......................................................... 50
  Frequencies: The Codes - Primary Document Matrix ............... 50
  Generating Code Lists ........................................................ 50
  Code Output Dialog ............................................................. 50

Text Search ............................................................................. 51
  The Text Search Tool .......................................................... 51
  Search Options .................................................................... 52
Simple Search ......................................................................... 52
Category Search ...................................................................... 52
  Defining Categories ............................................................ 53
  Hierarchies of Search Categories ......................................... 53
  How To Use Category Search ............................................... 54
  Managing Categories ........................................................... 54
GREP Search ......................................................................... 55
  Supported GREP Expressions ............................................... 55
The Object Explorer

How to? ...................................................................................... 77
Renaming Objects ....................................................................... 78
Opening Object Menus ................................................................. 78
Displaying and Editing Comments ................................................ 78
Code Trees & Forests .................................................................. 78

The QueryTool

The Screen .................................................................................. 79
Operands ...................................................................................... 80
Basic Operands .......................................................................... 80
Complex Operands ..................................................................... 80
Operators .................................................................................. 80
Overview .................................................................................. 80
Boolean Operators .................................................................... 81
Semantic Operators .................................................................. 82
Proximity Operators .................................................................. 83
The Query Language .................................................................. 84
Operands First, Operators Next ................................................... 84
Creating a Query with the QueryTool .......................................... 84
Query Examples ......................................................................... 85
Super Codes ............................................................................... 86
Creating Super Codes from Queries ............................................. 87
Auto-Optimization of Supercode Queries .................................... 87
What You Cant do with Super Codes .......................................... 87
Textbase Selection ..................................................................... 88
Output ........................................................................................ 88
Cleaning up the Hitlist ................................................................. 88
Creating a Report ....................................................................... 88
Miscellaneous Functions ............................................................. 88

Hypertext

Advantages of Hypertext ............................................................ 90
Creating Hypertext Links ............................................................ 90
Traversing Hypertext Links ......................................................... 91

WWW, Merging HUs, SPSS Export and Collaboration

World Wide Web Publishing ......................................................... 92
Merging Hermeneutic Units ........................................................ 92
Introduction

Foreword

"In my graduate student days, there were no tape recorders for making an interviewer's life easier; these arrived after World War II, passing first through the wire recorder stage. These were ponderous machines, only gradually slimming down to today's light models. Many years later, following our dreams of computers that would ease our lives as qualitative social researchers, software was devised for us. As Tesch pointed out in a relatively recent book¹, most of the programs were limited as tools for helping us in our qualitative analysis. At the time, she named only two or three programs that could be of much aid in generating social theory, one of them the ATLAS/ti program. This was based in part on Grounded Theory methodology and methods-methodology because of its flexible mode of operation, adapted to some sort of conceptual ordering, including the systematic generation of theory through the interplay of the researcher's brains and skills with the data; and Grounded Theory methods because many of these had been incorporated into the Atlas program.

Now the program has been further elaborated and improved. It should be of considerable aid in providing both greater efficiency and more elaboration for social scientists who master its intricacies. I myself am no expert computer-based researcher, and at my age am not likely to become one, but have found no great mystery in piloting my way, during session sessions, around the current program's predecessor.

Thomas Muhr, its author, makes no claims whatever to having produced a program that will perform miracles for your research – you still have to have the ideas and the gifts to do exceptional research. But you may find the ATLAS/ti program of measurable help and stimulation for your work. Surely it is not the only program that may be useful to you, given the various purposes of social research, but it is among those that should be seriously considered no matter what your research aims."

Anselm Strauss
Department of Social and Behavioral Sciences
University of California, San Francisco, California
March 1996

This handbook is dedicated to Prof. Anselm Strauss, who encouraged me to continue the work on ATLAS/ti and who kindly shared his ideas on how to improve this tool.


Objectives

ATLAS/ti is a powerful workbench for the qualitative analysis of large bodies of textual, graphical and audio data. It offers a variety of tools for accomplishing the tasks associated with any systematic approach to "soft" data, e.g., material which cannot be analyzed by formal, statistical approaches in meaningful ways. In the course of such a qualitative analysis ATLAS/ti helps you to uncover the complex phenomena hidden in your data in an exploratory way. For coping with the inherent complexity of the tasks and the data, ATLAS/ti offers

a powerful and intuitive environment that keeps you focused on the analyzed materials. It offers tools to manage, extract, compare, explore, and reassemble meaningful pieces from your often extensive amounts of data in a creative, flexible, yet systematic way.

ATLAS/ti imposes virtually no restrictions to the size of your data, the number of entities created, the complexity of the structures and theories derived.

**A VISE Principle**

The main principles of the ATLAS/ti "methodology" can be termed **VISE**: *Visualization, Integration, Serendipity and Exploration*.

**Visualization**

Visualization means direct support of the way human beings think, plan, and approach solutions in creative, yet systematic ways. Tools are offered to visualize complex properties and relations between the objects which accumulate during the process of eliciting meaning and structure from the analyzed data.

The *object-oriented* design of ATLAS/ti tries to keep the necessary operations close to the data they are to be applied to. The visual approach of the interface keeps you focused on the data and quite often, the functions you need are often just a few clicks away.

**Integration**

Not to loose the feeling for the whole when working on details - this is another fundamental design aspect of this software. Given the often analytical operations needed especially in early stages of an interpretation, synthetic operations keep and bring together the pieces. The main "container" object integrating all the other entities your studies are made off is the "Hermeneutic Unit".

Resuming work on a complex project with hundreds, or thousands of files is only a matter of loading one file.

**Serendipity**

In Websters Dictionary we find for "serendipity": "a seeming gift for making fortunate discoveries accidentally". In the context of information systems we can add: To find something without having searched for it. The term "serendipity" stands for an intuitive approach to data. A common operation making use of the serendipity effect is "browsing". This information seeking method is a genuine humane activity: when you are spending a day in the local library (or on the Internet ...) you often start with searching using a catalog, but you will find yourself increasingly engaged in browsing through the shelves neighborhood. Of course, this creative work style has its drawbacks: you sometimes spend some time...

**Exploration**

I needed an "e" to make the acronym complete!-) Seriously, exploration has close relations to the VISE principles already explained. Through an exploratory yet systematic approach to your data as opposed to a more "bureaucratic" handling of data it is assumed that especially your theory building, constructive activities will greatly benefit. The whole conception of the program, including
getting acquainted with its own idiosyncrasies, is aimed towards an exploratory, discovery oriented approach.

**Design Objectives**

*ATLAS/ti offers support to the researcher - without wresting control of the intellectual process.*

The fundamental design objective we applied when creating ATLAS/ti was to develop a tool which effectively supports the human interpreter, particularly in handling relatively large amounts of research material, notes, and associated theories. Although ATLAS/ti facilitates considerably many of the activities involved in qualitative data analysis and interpretation (particularly selecting, coding, and annotating), its purpose is not to "automatize" these processes. Automatic interpretation of text cannot succeed in grasping the complexity, lack of explicitness, and contextuality of everyday or scientific knowledge.

**About this Manual**

This printed short manual has been created from the full 320 page manual by a rather ruthless chopping procedure. The objective was to provide printed material to be easily carried around and to serve as an introductory guide to ATLAS/ti.

The structure of the main manual resembles this short manual to some extent. However, topics which were not considered as basic have been completely omitted, and you will find the symbol to the left (definitely the most frequent you will find in this text) referring you to the main manual:

Other topics have been reduced in length rather extensively and quite often, some more enlightening graphics were also omitted for space reasons.

As the text has not been fully revised to take the reduction from the full manual into account, you will find occurrences of inconsistency, especially concerning cross references to missing or reduced sections.

English native speakers will notice, that this manual was not written by a native speaker. It was necessary to have this first release of the manual written by the author of the program and there was simply no time to translate it into "real" English. I gave my best and hope you will understand most of it.

**Viewing and Printing the "real" Thing**

The "real" online manual and reference is provided on the ATLAS/ti CD-ROM. It uses the wide spread Adobe Acrobat Portable Document Format (PDF). On the CD-ROM you will also find the reader software (Acrobat Reader) to view and print the manual. Acrobat Reader not guarantees a precise printout as intended by the author, but is also a powerful viewing device with a clickable contents table and easy overview through "thumbnail" pages.

Print the complete or parts of the long manual to get detailed information about ATLAS/ti's special topics when needed. However, simply browsing through the online manual serves your information needs quite well in many cases. There are two ways to gain access to the manual:
Viewing the Manual from CD-ROM

This mode lets you browse through the manual directly from the CD-ROM. This benefit of not wasting any space on your hard disk also has its downsides: You have to have the original ATLAS/ti CD-ROM available each time you want to view the manual and loading, browsing and printing will be considerably slower than if accessed from the hard disk.

To view the manual on CD-ROM

1. Start the SETUP program from the ATLAS/ti CD-ROM.
2. From the main menu choose ATLAS/ti ONLINE MANUAL.
3. From the ONLINE MANUAL menu choose VIEW THE ATLAS/ti ONLINE MANUAL.

The pages of the original document are formatted in A4. If you are using a different format on your printer make sure to check the option Shrink to Fit in the printer dialog of the Acrobat reader.

Note, that the page numbers generated do not correspond to the layouted pages in the manual.

Installing the Manual on your Computer

Installing the manual and the reader software (if you have not already installed it previously) will occupy about 8 MB on your hard disk. The advantages are: speed, no need for the CD-ROM (mandatory if you are accessing it from a workstation with no easy access to a CD-ROM drive).

For detailed information about the installation of the Acrobat viewing software read the file readme.wri in directory ACREORD\mmBIT\ on the CD-ROM.

To install the Acrobat reader on your hard disk

1. Follow steps 1 and 2 above but choose INSTALL THE ACROBAT READER ... instead.

To copy the manual onto your hard disk

1. Either copy the file ATLMAN.PDF directly from the CD-ROM's root directory to your hard disk (using the Windows Explorer) or choose COPY THE ONLINE MANUAL... from the ONLINE MANUAL menu.

Manual on the Web

New and modified chapters will be available from our web site in the future. After installing the Acrobat reader, a "plug-in" will be optionally installed for your web browser to allow direct access to these materials without the need to download.

Online Help

Another useful information resource is the help file ATLASTI.HLP that can be launched from ATLAS/ti (F1) or directly from the Scientific Software folder. This help file has been created from the long manual and offers quick find and index capabilities.
Installing ATLAS/ti

ATLAS/ti for Windows uses an industry standard setup tool known from other installations. Just follow the on-screen instructions but read the following carefully.

In any case, because printed materials are almost always out of date in details, read the file readme.txt on the CD for up-to-date information before starting the installation.

System requirements

**Hardware:** PC with 486 (Pentium with 166 MHz recommended), 16 MB of RAM, between 15 and 25 MB of available hard disk space are required. Display minimum resolution of at 800x600 pixel recommended. A minimum of 15 MB Virtual Memory.

**Operating system:** One of the following operating systems should be installed on your computer: Windows 95, Windows NT 3.51, Windows NT 4.0, Windows 3.1 or Windows for Workgroups 3.11 with the 32-bit extensions Win32s 1.3c installed. (included on the CD-ROM) Windows 95 and Windows NT 4.0 are strongly recommended.

Previous Installations

SETUP will recognize existing installation of ATLAS/ti for Windows and will offer to upgrade. However, if you want to do a full and clean installation, it is recommended to UNINSTALL a previous version of ATLAS/ti before.

Upgrading from ATLAS/ti for Windows 4.0

The ATLAS/ti setup will determine if you are upgrading a previous version. It will also try to not overwrite any of those system files, that either have been configured to the environment or which were modified by the user. However, it is recommended that you copy (don't move them !) the files to a safe location outside the ATLAS/ti directory tree. Files to look for are: (hmerenc.hdb, blackboard.txt, *.hpr, *.skt, *.rel, *.hyp). Of course, no system files (*.EXE, *.DLL, *.SLL) should be backed up and copied over a new installation.

Starting the Installation

1. Have the serial number ready. If you are doing an upgrade on a previous version of ATLAS/ti for Windows, SETUP will determine the serial number and installation paths automatically.
2. Insert the ATLAS/ti CD into a CD-ROM drive.
3. Start SETUP from the CD-ROM (if "autostart" is active, SETUP will start automatically).
4. In addition to the installation procedure, the SETUP program's start screen offers you options like viewing and installing the full manual and online tutorials.
5. Setup will now guide you through a sequence of steps. From almost all situations, a BACK button takes you back to the previous step.
6. After the creation of the ATLAS/ti directory structure (cf. "Paths & Folders" on page 100), the copying of all files and the creation of registry entries, SETUP will terminate.
You will find a new folder entry in the start menu of Windows 95 and Windows NT 4.0 and a program manager group "Scientific Software" under all other Windows dialects. Optionally you will have a new entry in the Start Menu.

**Windows 3.1 and Windows for Workgroups 3.11**

SETUP will check your system and will optionally upgrade an existing 16-bit Windows (3.1 and WFW 3.11) with the Win32s subsystem. In case an incompatible version of the Win32s subsystem (anything less than 1.3c) was detected, SETUP will refer you to the text file (Win32s.txt) explaining the (rather tedious) deinstallation of the previous system and will cancel the setup. There is no automatic method to get rid of all the files and entries created by this add-on. It is recommended to start he computer in DOS mode to remove the old Win32s files (not is a DOS session under Windows), because while running Windows keeps hold onto some of the files.

Some provisions have been taken, that ATLAS/ti for Windows as a true 32-bit program still runs under Windows 3.x with the Win32s 32-bit extension kit quite satisfactorily. However, 16-bit Windows is far less efficient and stable than the 32-bit based operating systems Windows 95 and Windows NT. Among the known problems of Win32s are those related to S3 based video cards.

Under Windows 3.1 and Windows for Workgroups 3.11 you have to take a number of constraints into account. The system is less stable (especially if the installation is a bit old), extremely less efficient and it does not have enough space for "resources" (bitmaps, windows, colors, fonts, menus, items in lists, etc). This is totally independent of the size of RAM you have in your system. Do not use other applications concurrently when running ATLAS/ti. ATLAS/ti's help file might not get started properly as well.

**Important:** Under Win32s include the following line (or change an existing SHARE instruction) in file AUTOEXEC.BAT (located in your disk’s root directory):

```
SHARE /L:500 /F:5100
```

If SHARE is already used in the autoexec.bat file, add the parameters above.

**Network Installations**

Network aware installations (site licenses only) are different from stand-alone installations. After a full server installation of all program files, lightweight client installations are offered for easy installation of necessary links and registry entries.

Please read file NETINST.TXT on the CD-ROM for further details.

**Uninstalling ATLAS/ti**

Windows 95 and Windows NT 4.0: From the Control Panel open program **ADD/REMOVE PROGRAMS**, Click on the ATLAS/ti for Windows 95 entry and the **ADD/REMOVE** button.

Windows 3.1 and Windows NT 3.51: Double-click the "Uninstall ATLASTi" icon in Scientific Software's program group.

Do NOT uninstall ATLAS/ti by simply removing files from the hard disk. However if files and directory entries were left after installation, you can remove these manually.
Main Concepts

The Process

Two principal modes plus a supplemental level of working with ATLAS/ti are distinguished. The *textual* level includes activities like segmentation of data files, coding of text, image and audio passages and the writing of memos. The conceptual level focuses on model building activities such as linking codes to form semantic networks. The organizational level offers the necessary "infrastructure" for the other two.

Textual Level Work

Textual research activities include the breaking down, or segmenting, of the primary documents into passages (selections to be indexed), the adding of your own comments to respective passages (note-making/annotating), as well as the filing or indexing of all selected primary document passages, secondary text materials, annotations, and memos to facilitate their retrieval. The act of comparing noteworthy segments leads to what we might all see as the start of actual theory-building; a creative conceptualization phase or moment where one's own ideas begin to materialize.

![The overall process of text interpretation with ATLAS/ti proceeds from text to structure to text.](image)

*Figure 1 - Text-Structure-Text.*

Conceptual Level Work

Beyond mere code & retrieve, ATLAS/ti's unique networking feature allows you to visually "connect" selected passages, memos, and codes, into diagrams which graphically outline complex relations. This feature virtually transforms your text-based work space into a graphical "playground" for constructing concepts and theories based on relationships. This process sometimes renders yet other relations even more obvious than before, with the ability to instantly revert back to your notes or primary text selection. Such textual/conceptual traversability is unique to ATLAS/ti.

See "Conceptual Level - Networks & Co" on page 61 for more detail.

Organizational Level

A third level can easily be distinguished from the levels explained above. The organizational level is where you prepare the materials, organize the database, change the encoding type of documents, migrate complete projects to other

---

2 Although other media types, like graphic and audio files may also be used as „primary documents“, we will often use the terms "text" and "textual" regardless of the specific media.
computers. It also includes the administration of users in multi-author environments, the merging of Hermeneutic Units, the creation of World Wide Web publications.

**General Steps when Working with ATLAS/ti**

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

1. Create a project, an "idea container", which is meant to enclose your data, all your findings, codes, memos, structures under a single name. We call this project object a **Hermeneutic Unit** (HU).

2. Associate your data files, your **Primary Documents** (text, graphics, audio) located anywhere on your computer or network with your Hermeneutic Unit.

3. Read and select text passages (or identify areas in a graphic document), that are of further interest, assigning code words ("coding") or memos - the textual working phase.

4. Comparing data segments differently or equally coded, assigning more files to the project.

5. Building semantic, prepositional or terminological networks from the codes you created in the first phase. These networks, together with your codes, super codes, and memos, are cornerstones of the emerging theory.

6. Finally you compile a written report or you decide to publish your project as a World Wide Web document.

The ATLAS/ti workbench offers a multitude of tools to accomplish all of the tasks above - except the finding of relevant text passages!-

**The Hermeneutic Unit**

The most prominent "data structure" you are working with is the **Hermeneutic Unit** (HU). The name was chosen to reflect the approach we had taken when building a support tool for text interpretation. There was no intention to frighten potential users with this admittedly "tongue breaking" name.

If you understand this concept, then you have understood almost everything you need to work with ATLAS/ti. And - despite its impressing name - it is quite practical in everyday work.

Everything that is of relevance to one project (e.g., one research topic) is treated as one entity, regardless of the space in which it resides in the electronic environment. For instance, the **primary documents**, as well as relevant segments (quotations) of these documents, the **codes**, the developing concepts, the linkages between the concepts, the **families, networks, memos**, etc., are all part of one hermeneutic unit. One obvious advantage of this bundling is that the user often only has to deal with and to think of one entity. The activation of a hermeneutic unit is a straightforward selection of a single file; all associated material is then activated automatically.
The lowest level of a Hermeneutic Unit contains the primary documents assigned, followed closely by the "quotations" as selections of primary documents. Codes refer to quotations. Memos - you meet them everywhere.

Figure 2 - The hierarchy of objects inside a Hermeneutic Unit

The Hermeneutic Unit is the "spider in the web".

A Hermeneutic Unit 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, appropriate browsing and editing tools are offered.

Primary Documents

The text, graphical and audio materials you wish to interpret are the primary documents, meaning firsthand materials. Some research fields may also know these as the "raw data", and yet others even as "field notes". Assigning primary documents to a Hermeneutic Unit is often the next logical step after its creation.

Primary Document Media Types

Three media types of documents are currently supported: text, graphic, and audio data.

Textual Data

Proprietary word processor formats are not directly supported by ATLAS/ti. Textual primary documents have to be either using the ASCII (DOS, OEM) or ANSI (Windows) character code table. These "plain vanilla" text formats are the least common denominator for textual data. All word processors are capable of storing text in this format.

Graphical Data

The common file types Windows Bitmap (BMP), TIFF, and Kodak Photo CD are three of more than twenty graphic file formats that are directly supported by ATLAS/ti.

Audio Data

Sound files of type WAV can be used as primary documents. Of course, a sound device must be installed on your computer to make use of those.
Quotations

A quotation is a segment from a primary document which was considered as interesting or important. For textual documents this is an arbitrary sequence of characters ranging from a single character, words, sentences, paragraphs or the whole file. Usually those selections are created by the human interpreter, as recognizing a useful piece of data worth selecting is already an interpretative act. However, if enough evidence is available not only "between the lines" but in the lines themselves (as is often the case in linguistic studies), the auto coding feature of ATLAS/ti can do automatic repetitive segmentation into quotations (and their automatic assignment to a code) for you.

When a quotation is created, an identifier is automatically assigned. This identifier, which is the display name of a quotation in list windows and printouts, is built from the index of the primary text it belongs to, a consecutive number within the primary document and the first 20 letters of the text segment, e.g., "1:21 And the first part ..".

Quotations as Layers

Quotations are stored independently of the primary document they are "part of". In fact, quotations are stored inside the Hermeneutic Unit.

Just like the primary documents are not actually part of the Hermeneutic Unit, a quotation is not represented by inserting extra mark-up code within its primary document. A primary document file's content is therefor not altered by the creation of quotations. You can view the quotations as a transparency layer on top of the primary document. The latter does not "know" anything of quotations.

Technically, a quotation has three important attributes: the identifier (a number) of its primary document, and the two coordinates (start line and column, end line and column) that define the range of characters.

Types of Quotations

As quotations are part of a primary document, there are also textual, graphical and audio quotations.

Textual Quotations

Regarding textual primary documents as the prominent material of interpretative analysis, so are the textual quotations. As sequences of characters ("strings") they can be of arbitrary size, often sentences, speech turns, paragraphs are the grain size of textual quotations. Only text offers enough "syntactical clues" to allow searches for the occurrence of specific evidence. Text also offers the option for automatic segmentation as used by the Auto Coding procedure (see also "The Auto Coding Tool" on page 56).
**Graphical Quotations**

Graphical quotations are rectangular regions.

The creation, the activation and the display of graphical quotations has similarities with but also differs from their textual counterparts. A graphical quotation is a rectangular region inside a graphical primary document. From its data structure, it is identical to the textual quotations, as their main attributes are also the identifier of its hosting primary document and two coordinates that mark the beginning and end, defining a rectangle through its upper left and lower right corner.

**Audio Quotations**

Audio quotations differ significantly from the other two media in one aspect. Only one audio segment is currently available for a given audio primary document. This feature is a little light weight currently, but development is already on its way to have full access on arbitrary "frames" inside an audio file.

**Codes**

Code - what a magic word! The term **code** is used in so many different domains that it should be made clear what our understanding is in the context of qualitative research or even more specific, in the context of ATLAS/ti.

From a methodological standpoint, codes serve a variety of different purposes. They are meant to capture some meaning in the data.

They are used as "handles" to find specific occurrences in the data which cannot be searched by simply applying text based search techniques, because this would require sufficient "match-able" information in the text itself (e.g., "Statement No. 124"). They are used as classification devices of different level of abstraction to create sets of related information pieces for the purpose of their comparison (e.g., "Coping pain with medicine", "Coping Strategy").

**Keep a code's wording brief and succinct. Use memos or the code's comment for longer elaboration.**

From a "low level" tool perspective, codes are simply - usually quite short - pieces of text referencing other pieces of text. Their purpose here often resembles the latter definition, as set creating classifiers of often large numbers of textual units or other segments. In the realm of information retrieval systems, the term "index" or "keyword" is often used for what we term "code" (and "indexing" for "coding").

Codes are even more interpretations and findings compared with the primary documents and the selections thereof which they refer to. The length of a code should be restricted and not be too verbose. If textual annotations are what you want, you would use **memos** instead.

**Super Codes**

A Super Code differs from normal codes, as it stores a query formed using other codes. See details in section "Super Codes" on page 86 and 28.

**Memos**

Memos capture the textual "flesh" of your findings and are an important device for creating a theory. A "memo" is quite similar to a code, but usually contains
longer passages of text. Memos may "stand-alone", refer to quotations, codes and even other memos. Memos may have a user-definable type (theoretical, descriptive, etc.). Types may be used to filter memos.

**Families**

One purpose of codes is to sort interesting segments of primary data into meaningful subsets.

But who sorts the primary documents, codes and memos?

Families are a device to form clusters of those entities for easier handling of especially large number of those interpretative objects. Like the objects they host, families are named "containers". For more detail see chapter "Family Life" on page 56.

**Networks**

More sophisticated than families, networks allow a stronger structure than just treating sets of elements as similar. This is what codes do with quotations and families do with codes, primary documents, and memos. With the aid of networks you can express meaningful "semantic" relationships between elements. Almost everything can be connected in a network: codes, quotations, memos.

**General Properties of Objects**

### Comments

*Everything can be commented in ATLAS/ti*

Comments are not "first class objects" themselves (they do not have a name, they do not have their own list window, nor do they have any other property than the text they contain), they are an important *attribute* of any of the objects described so far.

Writing is one of the main activities when working with ATLAS/ti and although the system does not give you much help in *what* you write, it does support you to apply your writings to the adequate locations. All entities that were described in the former, have a "slot" that can be filled with commenting text.

### Authorship

Every object created during your work with ATLAS/ti (including the internal representation of a primary document) gets an electronic stamp with the current date & time and the currently logged in user's name. As different objects may have different "authors" (even when you just assign a primary document to the Hermeneutic Unit, you are considered this primary document's author) the productions of different contributors are traceable. By using special filters, it is possible to compare the work of different authors on the same project.
Overview

The ATLAS/ti workbench offers a variety of tools to serve your needs. The Hermeneutic Unit Editor is the main workspace, it is opened after the start of the program, and will be your „home page” most of the time.

Besides being the main editing tool, it also offers access to all the other tools from the workbench.

With all optional controls, buttons and "panes" visible, this window looks rather overwhelming at first sight. You can decide which parts of the window should be visible. Tailor its appearance to better fit your needs. Use the View menu to hide some of the buttons, windows etc.

Components of the HU Editor

Title Bar

The title bar is common to all windows and dialog boxes. It displays the name of the Hermeneutic Unit that is currently being edited. Windows can be dragged and moved around with the mouse by "grabbing" the window at its title bar.
Main Menu

Many ways lead to Rome. Commands found in the main menu are often also available from context menus and toolbar buttons.

The main menu bar underneath the title bar offers access to the different kind of objects that populate a Hermeneutic Unit.

The main menu (and the tool bars) is also the central repository for all other tools in the work bench.

Tool Bars

Toolbars offer functions that are generally also available as menu commands. A short help text is displayed when the mouse pointer rests on a tool button. This "tooltip" help text can also be displayed with a click of the right mouse button.

The Main Tool Bar

Underneath the menu bar is the main tool bar.

The Primary Document Toolbar

The primary document toolbar is left to the line numbers. Note, that it is disabled and will not even displays help information unless a document is loaded.

Go to Line displays the text at a specific line number.

Search activates the text search tool.

Free Quotation creates a - correct - free quotation.

The four coding buttons allow easy access to the most common coding functions.

Create Memo attaches a newly created memo to the current document selection.

Modify Quotation resets the size of a quotation

The next two buttons toggle the display of Line Numbers and Margin Area

Hyper Source and Hyper Target make the selection part of a hypertext structure

Show Quotations displays the list of all quotations surrounding the current text cursor position.
Object Drop-Down Lists

Below the main tool bar are four drop-down lists for the Hermeneutic Unit’s main entities. From left to right: **primary documents, quotations, codes,** and **memos.** The list of objects drops down, when clicking into its text field or onto the little drop-down arrow at its right side.

From the dropped down list, an object can be selected. Immediately after the object is selected the drop down list returns to its previous state.

Drop-down lists are very common in Windows applications for managing large numbers of objects in sparse space.

**Primary Document Pane**

The large window area is the **primary document pane** - in "Figure 3 - The HU Editor" on page 13 above a text is already loaded and displayed. It is safe to say that this pane is the core work space. Here is where text, graphic and audio material is reviewed, marked, coded, annotated.

**Line Number Area**

Left to the text area, but inside the primary documents pane client area, **line numbers** may optionally be displayed. Line numbers are generated dynamically during scrolling of the text.

**Margin Area**

To the right hand side of the primary text area is the optional **margin area**: it initially displays "serpents" were interesting text selections have already been created and the codes that refer to the text displayed. This area is multifunctional: the kind of referenced objects that are displayed (codes, memos, quotations) can be selected. Furthermore, this area can be used as an entity list for any of the types of objects usually displayed in the drop down lists: primary documents, quotations, code, and memos.

One of the great advantages of the margin area is that it is interactive: the displayed objects are sensitive to mouse clicks and have their own context menus.

Objects in the margin area are "click-sensitive". Their comments are displayed with a double-click and a context menu can be popped up.

*Figure 4 - Context menu for a code in margin area*
**Note:** A general context menu can be opened inside the margin area by moving the mouse pointer somewhere between the displayed objects. It allows to select different display properties and the kind of objects displayed.

![Margin area properties menus](image)

*Figure 5 - Margin area properties menus*

See "The Margin Area Context Menus" on page 98 for more detail.

---

**Note:** for graphical and audio documents no line number and margin area is displayed. However, in graphical document’s selections, a number and comment can be displayed „in place“.

---

**The Pop-up Menus**

*Pop-up or "context" menus are activated with a right mouse button click.*

Context - or pop-up - menus are an important repository for commands, that apply to a certain selected object, or to a specific pane inside a multi-pane window. Unlike the main menu bar which offers drop-down menus when clicking on any of its entries, a pop-up menu is activated by a right mouse button click. Pop-up menus are displayed at the mouse pointer's current location and eliminate the need to move to the menu bar. They are context sensitive because the pop-up menu's commands are dependent on the type and the current state of the object or the context in which the menu was activated.

Using pop-up menus to initiate procedures is a three step procedure typical for window-mouse user interfaces:

1. Select an object.
2. Open the pop-up menu with the right mouse button.
3. Select a command.

The HU editor provides specific pop-up menus in each of its sub panes. The primary document pane offers a variety of pop-up menus, depending on what is currently displayed. With no primary document selected and displayed (e.g., directly after starting the session or after closing a primary document) you get the wallpaper menu:

![Wallpaper pop-up menu](image)
After a primary text is loaded the primary document pop-up menu is displayed when clicking the right mouse button anywhere inside the pane:

*Pop-up menu for unselected primary text.*

When you select a piece of text and right click inside the text selection, another pop-up menu is displayed:

*Pop-up menu for selected text.*

If the selection is an existing *quotation*, the pop-up menu gets even more specific:

*Pop-up menu for a quotation.*

Without going into detail here concerning the commands offered, you can see that the commands are tailored to the needs of the objects under the cursor at the time of activation.

*Pop-up menus are here, there, almost everywhere to serve your needs - if you know how...*

One great advantage of pop-up menus is also often a disadvantage when you start working with the program. Pop-up menus do not permanently require space as the main menu does and so help in minimizing screen clutter. On the other hand, you do not have a visual indication, that there is a menu available. You simply have to remember, that it is worth using the right mouse button in different windows, even on specific objects selected or touched by the cursor.

The margin area and the network editor with all its different types of objects gathered in one window are nice examples for the flexibility of pop-up menus.
The Object Lists

The advantage of the drop-down lists for the four main entity types is also a disadvantage when the current work phase includes a lot of selecting of those objects. Drop-down lists do only display one object and so do not need much screen space To view more than one object is done by clicking the drop-down arrow. To make access to listed objects more efficient, object lists can be opened in addition. Although not an integral part of the HU editor, the object lists are common "companions" to the main editor and therefore described in this section.

Opening an Object List

Left of every drop-down list is a button to launch the corresponding object list as a detached window. Alternatively, the object list can be invoked by choosing EXTRA LIST from the corresponding main menu entry.

![Open Extra List](image)

The figure above shows how a code is opened by clicking on the object list button. Alternatively, the object list can be invoked by choosing EXTRA LIST from the corresponding main menu entry.

Selecting Items in Object Lists

Besides thinking, clicking will be one of the most frequently pursued activities when working with ATLAS/ti. Clicking comes in different flavors, like single and double-click, clicking with the left or the right mouse button. The "semantics" of a mouse-click is not perfectly straightforward, and a few deviations must be learned.

*Single click selects an object, double-click executes an associated action for the object.*

A single click with the left mouse button selects an item in each of the object lists. The selection is signaled by highlighting the selected item in the list pane. Many of the operations (like editing the comment, renaming the entry, opening a focused network view) offered by the context menus and tool bar buttons affect an item selected this way.
A double-click (two rapidly issued clicks) on an item does two things: the object gets selected and an additional activity is launched, depending on the type of the object. The effect of a double-click is explained in the respective list descriptions below.

**Primary Document List**

During the phase of assigning primary documents the primary document list is more convenient than the drop-down list. Using drag & drop, files can be directly and very efficiently assigned to the Hermeneutic Unit. Loading a PD is done by a double-click on a list entry (see also "To load a primary document using the object list" on page 34

A double-click on a primary document's list entry loads the file and displays it in the primary document pane.

**Quotation List**

Presumably, this list has to hold the most elements of all. With fifty primary documents and an average of 20 quotations per document, one thousand entries can be scrolled through.

The quotation list's network button opens a focused network view with the selected quotation and all its direct neighbors (other quotations, codes, memos).

Select and display one quotation after the other using the next and previous buttons.

A double-click on a quotation's list entry loads its primary document and then displays itself highlighted in context.

**Code List**

As coding and "decoding" are major components of the interpretative tasks carried out on the primary data, this list might be the most frequently displayed during everyday work. Creation of new codes and the invocation of coded data segments are quite easily done with the code list.

The code list's next and previous buttons display the quotations coded with the selected code one after the other.

A double-click on a code displays the coded quotation highlighted in the context of its primary document. If more than one quotation are associated with this code, a list pops up from which a quotation can be selected and displayed.

**Memo List**

The memo list displays the text body of a selected memo. Memos themselves have no comment but can be linked with other memos.

A double-click on a memo displays the quotation associated with this memo highlighted in the context of its primary document. If more than one quotation are associated with this memo, a list pops up from which a quotation can be selected and displayed.
Lists in Margin Area

You can also have a list of objects displayed in the margin area. If you do not need the margin area for anything else and you are currently working with one specific object type (e.g., concerned with heavy coding using the code list frequently) this method offers easy access to list elements and it does not clutter your screen. As an integral part of the main editor the list moves with HU editor and it does not obscure any other screen elements.

Resize the margin list using the split bar between the primary document pane and the list.

To load an object list into the margin area

1. Load a textual primary document.
2. Switch on the margin area.
3. Open the margin area's context menu and choose Lists/<ObjectType>.

To switch back into standard margin mode, open the context menu in the object list and select command BACK TO ANNO.
A Quick Tour for Beginners

This section redundantly repeats the description of operations also explained elsewhere. However, the sequence of the operations in this tour is guided along some simple tasks. You should be able to follow the instructions without having studied the other chapters.³

Getting Started

We will not dive into Hypertext, Relations, Multi-Authoring, Auto-Coding, User Management, Regular Expressions, Topological Sorts, SPSS and HTML conversion in this introductory section. Its single purpose is to guide you through some VERY basic but nonetheless important steps. You will learn how to assign primary documents, to select passages, create quotations, assign codes in different ways, build a simple network, do a query, and create a super code. For an overview of the general work process have a look at section "General Steps when Working with ATLAS/ti" on page 8.

In this first part, we will look at one of the sample projects that came with the software.

Starting the System

Start ATLAS/ti by

- double clicking on the ATLAS/ti for Windows icon in the program folder (program manager in Windows 3.x), or by
- selecting it from the start menu (Windows 95, Windows NT 4.0).

You will be greeted by the Welcome "wizard" which offers some options for beginning a session. In the background you can already see the main work space, the editor for Hermeneutic Units.

Note: if you do not choose anything from the "Wizard" it will disappear after a while. You can always get it back by choosing HELP/OPEN WELCOMEWIZ from the HU Editor's main menu.

The "wizard“ offers four options to start (you can display a short help text explaining these options by clicking on the Help button). The first option "Open Hermeneutic Unit from Picklist" is already checked and that is just fine. Click on OK to pick from the list. You are now offered a selection of example projects (Hermeneutic Units) which are not really examples of good interpretative work, but are sufficient to get an impression.

Choose "Bible Kabala and Apocalyptic Visions" from the pick list.

³ If you were an "Early Bird" user, you will note the similarities to the Early Bird Crash Course.
The Welcome Wizard disappears and you can see some activity in the HU Editor while loading the selected HU within a few seconds. The main area of the HU editor still displays the logo and some licensing information.

**Select a Primary Document**

The primary documents that are bundled within this Hermeneutic Unit are managed in the left most "combo" list box below the main tool bar. Click into the combo and select the first document from the list that drops down. The file is loaded and displayed in the main area. Just plain text is visible and it looks as if it has not been worked with yet.

**Display a Quotation**

Now select with the same method as above the first entry in the second combo list just beside the primary does list. This list keeps track of the "quotations", segments of primary documents (including text, graphic and audio) that have been created by selecting passages of interest, often assigning codes simultaneously.

The selected quotation "And the name ..." will be displayed highlighted and centered on screen within the text it stems from.

**Display Codes for a Quotation**

To check if there are any codes attached to this quotation, move the mouse pointer over the highlighted selection and pop up its menu with a right mouse click. Select item "Show Links". Three codes are displayed which are all connected to this text passage.

Switch on the "Margin Area" in the View menu above the main toolbar. You can now see the codes alongside their text passages.

**Display quotations for codes.**

Retrieving coded text segments in context is easy. For this purpose look at the third combo list and click on the little button to the left of it. This opens up an "extra" list detached from the main window that allows more freedom in selecting the items. Extra lists are available for almost all relevant object classes managed by a Hermeneutic Unit.

In the code list double click on code "Alchemy". A new quotation is displayed in the primary text area. Note, that the entry in the quotation list has also changed to "1:16 and there were ...". Every code has a postfix of two numbers in curly brackets. The first displays the number of quotations ("Alchemy" has exactly one quotation it references) and the second displays the number of other codes that are connected with this code ("semantic networks"). Alchemy has three neighbors.

Note, that some codes do not have any quotation at all, and are just building stones of a theory, connecting other codes.
Memos

The only combo list left is the rightmost. It hosts the memos, another major object type. Memos are similar to codes, but their main purpose is to capture your longer textual productions.

The Object Explorer

The Object Explorer is quite similar in "look & feel" to the Explorer used by Windows 95. The hierarchy of objects can be expanded or collapsed using the little boxes left to an object (with neighbors). The "raw" list displays the main object types quite nicely: primary documents, quotations, codes, memos, the "families" and the Network Views.

Click on the root "Bible, Kabala and ....". In the lower text pane the Hermeneutic Unit's commentary is displayed.

Click on the expand box left of "Primary Docs". Click on P1 and you can see the primary document's commentary (NOT the contents of the primary itself) displayed below.

Every object type has its own context menu associated with it.

Creating a New Hermeneutic Unit

Browsing around in a given Hermeneutic Unit is good for the very first impressions. Now lets create a new one.

To start a new Hermeneutic Unit

1. Select NEW HERMENETIC UNIT from the main editor's FILE menu. If changes are detected in "Bible..." you will be asked if you want to save these. Answer NO.

2. Enter a title for the new Hermeneutic Unit. Something like: "This is a very, very new HU". Seriously, titles should reflect the theme of your research undertaking in a concise way.

3. It is a good idea and good style to further elaborate the short title with a longer abstract. This is especially useful when working in multi-author settings, when generating HTML documents, and when Hermeneutic Units are displayed in the HU browser. Click on the second to left button in the main toolbar or select "Edit Comment" from the File menu.

4. Write some text like "This is my very first HU and I will promise to write comments every time I create a new one."

5. SAVE from the Texteditor's FILE menu stores this text in the Hermeneutic Unit. A good idea to save the Hermeneutic Unit on disk to make our work permanent (not too much yet, though).

6. Close the Texteditor (you may leave it open) and choose SAVE AS from the Hermeneutic Unit Editor's File menu. Under Windows 95 and Windows NT you will be offered the full name as a file name: "This is a very, very new HU.hpr". Under Windows 3.1, this name will be shortened to the usual 8.3 syntax.

7. Click SAVE. Your Hermeneutic Unit will also become an entry of the quick loader pick list.
Assign Primary Documents

The next logical step in most cases is to assign documents to be analyzed to the Hermeneutic Unit. Assigning texts, graphics or audio files to a Hermeneutic Unit is easy. Use Assign from the Documents menu. When choosing textual documents be careful to assign only text files that are plain ASCII ("DOS") or ANSI ("WINDOWS").

Choose text Rev-8.txt from the file dialog. The file name of this text is instantly displayed in the left combo box. Select it from the combo list using the mouse.

By the way, you may select and assign more than one document by simply selecting multiple files in the file dialog.

Creating a Quotation

The contents of file rev-8.txt should now be visible in the primary text area. Using the scrollbar, scroll the paragraph number 7 ("The first angel sounded...") into the visible area. We will now explore different selection techniques.

Semi-automatic selection:

1. Left click the mouse on the word angel (after "first"). This will move the text cursor into this word. Double click on "angel": the word angel will be highlighted.

2. Double click on the highlighted "angel": the sentence including "angel" is highlighted (well, up to the colon).

3. Double click on the highlighted sentence: The paragraph is selected. (A paragraph is any text enclosed with at least one empty line above and below).

4. One more double click selects the whole file.

5. The last double click brings you back to step 1.

The advantage of this selection technique is that you can be sure that a paragraph is selected in a unique way, not a space missing, not a period too much.

Manual selection:

Of course you can do manual selection as you are used to in any other Windows program.

1. Move the mouse pointer to the beginning of the text you want to select

2. Drag the mouse cursor (click and hold the left mouse button, then move it) to the end of the intended selection and release the mouse button.

Note: if you drag from within a selection, the system interprets this as the start of a drag & drop operation which is not what we want now.

To create a "free" quotation:

1. Select the complete paragraph using what you just learned. Everything from "7" to "burnt up." should now be highlighted.
2. Move the mouse cursor into the highlighted text and pop up the context menu for selections with the right mouse button.

3. Choose "Create Free Quotation". A serpent appears in the margin area, and a new entry appears in the quotation's combo box.

Creating a "free" quotation is similar to what we do when reading a book with a pencil: you make a mark of some sort in the margin or you underline what you find significant.

Coding

Often you do two things simultaneously: create a quotation and assign a code to it, in other words: you code a text passage, create a coding. Three ways to code are described in the following: **Open Coding, Code by List, In-Vivo Coding.**

**Open Coding**

1. Open the selection's (quotation number 1:17 should still be highlighted) context menu again and choose **Coding** this time. Two active options should be displayed.

2. Choose **Open Coding**.

3. Enter "Musical Disaster" and click OK.
   You have created your first code. The third combo box shows the code just created. Doesn't the margin look much better now?

Of course you can combine the creation of a quotation and a code:

1. Select the paragraph preceding the one that is currently selected (Number 6).

2. Select **Coding** again from this selection's context menu. All four coding options are now enabled.

3. Choose **Open Coding** again.

4. Enter "Testing Trumpets | Seven Angels". Note the vertical OR bar between the two codes. You should find it somewhere on your keyboard. With the help of this special character you can create as many codes as you want in one turn.

5. Click OK.

**Code by list**

1. Select the preceding paragraph starting with "5 And the angel....".

2. Open the **Coding** menu and select **Code by List**. The selection list that is now displayed shows the three codes that already exist. Of course, you will find only two codes if you have not entered the correct separator character in the previous coding.

3. Select all three codes and click OK.

You have created a quotation coded with three codes. Note, that the multiple usage of the codes is reflected in the margin area. Each appears more than once.
**In-Vivo coding**

1. In paragraph number 4 select the word "smoke" with a double-click.

2. Select **Code In-Vivo** from the **Coding** menu. You have created a code named as the coded text itself.

Quite often you want to capture the word, but you would like the actual quotation to contain more text:

1. Select and highlight the whole paragraph containing "smoke".

2. From the main menu's **Quotation** menu, select **Modify**. Without altering the connection between the quotation and the code "smoke" that you created before, the extent of the quotation has been enlarged to be more meaningful.

Open the code extra list by clicking on the list button left to the code combo. 4 codes are in the list and the "suffix" shows that three reference two quotations each, and one "smoke" refers to one. "smoke" should also be highlighted as the selected code. If not, double-click on the code "smoke". If you want to rename this code to start with an uppercase letter (you want it), open the context menu within the list area of the code list and select "Rename". Enter "Smoke" and click OK. Even if "smoke" was used in a few hundred codings all of these would now be updated to the new name.

Save the Hermeneutic Unit now by choosing **Save** from the **File** menu.

Double-clicking on entries in the code list does the following: the code is selected and the coded segment is displayed in context. If a code is associated with more than one quotation, then you are offered a list to choose from.

Double-click on code "Musical Disaster". A list with two quotations is presented. You can browse through these quotations by simply clicking on each. A double-click selects the quotation AND closes the list. As this "chooser" is not modal (blocking anything else) you can do something with the quotations that are displayed in context, e.g., assign more codes, delete them, modify,...

**A Little Drag & Drop**

Many operations requiring some clicks and menu pops can be cut short by way of Drag & Drop.

**To Code with Drag & Drop**

1. Select paragraph Number 2 ("And I saw..")..

2. From the extra code list, select (don't double-click) and drag code "Seven Angels" into the primary documents pane (you do not have to aim at the selected passage).

3. Release the mouse button.

Voilá, this piece of text has been coded. Look at "Seven Angels" suffix.

Double-click on "Seven Angels".

**Note:** you can select more than one code before dragging using a standard Windows selection techniques (see "Selection Techniques" on page 101)
To Code In-Vivo using Drag & Drop

1. Switch on line numbers with the "99" button at the left.
2. In the primary text double click and select the word "Revelation" in line 3.
3. Move the mouse pointer inside the selected text and drag it into the code list.
4. Release the mouse button.

This is the quick way to do "In-Vivo coding".

A Short Excursion into Networking

We already have collected some material that we can use for playing a little with the network editor. Select code "Seven Angels" in the code list. Open a focused network view by clicking the network button. One lonesome "node" is now displayed in the network editor (NWE). We need some company so you can just drag code "Musical Disaster" from the code list into the NWE. It will be placed where you release the mouse button. But of course, you can move the nodes around any place by dragging (hold left mouse button and move) the node.

Place the nodes so that you have some distance between the two.

Just like list entries in a list you can select nodes with a single click. Try it.

But what a sad network with only a single connection between the nodes:

To link two nodes

1. Move the mouse pointer onto node "Seven Angels".
2. Hold down the shift key.
3. Hold down the left mouse button and move it onto the other node, just as if you wanted to move the node. A "rubber band" tracks the movement of the mouse.
4. Move the cursor on node "Musical Disaster".
5. Release the shift key.
6. Release the mouse button. A selection of "relations" pops up and you can choose the "is cause of" link.

You have created a network, your first "theory" (about music or about angels?).

Of course, even the links have a menu....

This new network's changes should be made permanent. Select SAVE from the Network editor's NETWORK menu. You will be prompted for a name, but if you prefer you can accept the default with the OK button.

Now save the complete Hermeneutic Unit again.

Don't be Afraid of the QueryTool

Just a very tiny excursion to one of the most powerful yet rather complex tools from the workbench. You already "retrieved" quotations for codes in the most simple way: double clicking on one code. This translates into a query: "Give me all quotations for code X". Sometimes you need more complexity.
1. Open the QueryTool with a click on the "binoculars". Only the lower left list window shows some entries yet. Let's click together a query:

2. Double-click on "Testing Trumpets". It is getting crowded. The upper right pane displays the current query. The lower right pane displays the two quotations coded with the selected code.
   
   Not much different from the clicking on codes in the extra list.

3. Double-click on "Smoke". The current query now is "Smoke" and the one quotation is displayed in the result list. However, you can now see two "terms" in the upper right list.

4. Click (single) on the uppermost button on the left hand side. This is the OR operator. The result list now displays three quotations, the result of the query: "All quotations coded with Smoke or Testing Trumpets". The "term-stack" pane now displays the query, for which the result is displayed in the result list. It has "eaten" the two previous terms after you combined them with an OR.

5. Click on an entry in the result list. The "hit" will be displayed immediately in the primary text pane. Of course you can also make a print out by clicking the "printer button".

6. Click on the fourth button from the top, the last one in the group of four. This the negation (or "NOT") operator. What do you see? The query gets longer, and the entries in the hit list are now all quotations which you did not see without the NOT. This equals the query: "All quotations not coded with any of Smoke or Testing Trumpets".

**Create a Super Code**

Since we have gone that far, we will now create a Super Code. Such a "code" has the "look & feel" of the normal codes, but it does not store hard wired links to data segments, it stores the method to calculate these links only when you need it. It stores your hypothesis, it is an intensional code.

**To Create a Super Code**

1. Click on "Super-Code".

2. Accept the default name for now and click OK.

What have you got? A new code! You can use it like any other code.

Save your Hermeneutic Unit.

We will stop here with our Beginners Guided Tour.
The Textual Level-Basic Functions

Overview

In this chapter you will learn how to start and end an ATLAS/ti session, and you will learn about the main procedures needed to build projects ("Hermeneutic Units"), prepare and assign primary documents, mark, code, and comment text, graphic and audio segments, generate output, and make good use of the tools offered for keeping track of all your productions.

Starting and Closing an ATLAS/ti Session

Basic functions explain how to start and end an ATLAS/ti session, how to open existing projects, how to create new ones and how to prepare and assign primary documents.

Starting an ATLAS/ti Session

See "Starting the System" on page 21

Logging In

By default (e.g. after installation), automatic login is enabled. ATLAS/ti supports multiple users and associates the currently logged user with all objects created during the session.

However, if you have disabled automatic logon you are presented the Login dialog box and you have to enter (or select) your name and enter the password before proceeding with the welcome dialog above.

Please refer to "User Management" on page 92 for more information.

Leaving ATLAS/ti

When you have finished working, make sure to save all pending changes of the Hermeneutic Units you modified during the session. Then select File/Quit from the HU editors main menu to quit the session.

Creating, Opening and Saving a Hermeneutic Unit

Creating a Hermeneutic Unit

To start a new project you need to create a new Hermeneutic Unit.
Figure 7 - Enter a name for the new HU

**To create a new Hermeneutic Unit**

1. Choose **FILE/NEW HERMENEUTIC UNIT** from the main menu. You are prompted for a name.
2. Enter a name for the new HU and click OK. Because this title can be changed later, any preliminary title will do for now.
3. Optional but highly recommended: Write a comment

Write a comment for your HU - your team members - and last not least you - will benefit from this discipline.

(Note: You should make writing a comment for the newly created HU a habit. An editor for the HU's comment can be opened by clicking on the button in the main tool bar. The HU's comment will be displayed in the HU-browser when it is later opened again. This comment might be used to describe the project's goals and intentions.

Opening and saving Hermeneutic Units see "Getting Started" on page 21.

**Preparing Primary Documents**

Before you assign a primary document to a Hermeneutic Unit, certain provisions have to be taken. As ATLAS/ti does not process any proprietary text processor formats, textual PDs have to be converted to plain text using either ANSI (Windows) or ASCII (OEM, DOS) as their encoding scheme.

When you assign a primary document which does not have a suitable file format, you will either see nothing (wrong graphical file formats), all or partly garbled characters (proprietary text processor format), or only a few characters incorrectly displayed (ANSI/ASCII). The latter behavior is often not due to wrong conversion but can be corrected by setting the encoding type correctly (see "ANSI or ASCII? Changing the Encoding Type" on page 33).

**Preparing Textual PDs**

ANSI or ASCII texts are the most common form of electronically archived and transmitted textual documents. If your document source is the Internet, or you will be analyzing ready archived texts on CD ROM, you most probably will not have to convert the file's encoding format. However, if the origin of your documents is a word processor like Word or WordPerfect you have to create a file suitable for further processing with ATLAS/ti. All major text processors do have options to strip of all proprietary formatting information from the documents resulting in "plain vanilla" text only files with only tabs and line breaks as special formatting characters.
Before generating the text-only version of the file, you should format the text conforming to specific recommendations that will make life much easier when finally assigned to the Hermeneutic Unit.

Two characteristics of ATLAS/ti are important to understand how text should be prepared.

**How ATLAS/ti interprets plain text**

**Hard line breaks** needed. ATLAS/ti does not use automatic line-wrapping, as this would make problems with the overlay technique used for associating quotations with their primary documents. Therefore, make sure that you create text files with the hard line-feed option. If you do not do this, you will see every paragraph as one giant line.

**To create a text-only version from a WinWord document**

1. Load your text document into Word.
2. Set the font face and size to the font you use within ATLAS/ti. Fixed fonts are recommended (like Courier).
3. Switch hyphenation off.
4. Set the right margin so that the size of a line does not exceed the size of the primary document pane's width. *You should open ATLAS/ti at the same time so you can directly compare the width of the PD pane and the line size set in Word.*
5. Insert an extra line break between all text that you want to be paragraphs in ATLAS/ti. Natural "break-points" are speech turns in interviews or the already existing paragraphs in your Word document. *You can use Word's find/replace to automate this process.*
6. Choose **FILE/SAVE AS** from Word's main menu.
7. In the lower part of the Save As dialog box set file type to: "Plain text + Line Feed" (ANSI) or "DOS text + Line Feed" (ASCII). *The extension TXT will be used by Word when creating the text file. Be careful not to use the same name as your original Word document, because you will then have no version left that still contains formatting codes.*
8. Set the directory to your default textbank directory. This is not necessary but will make assigning the documents to the Hermeneutic Unit more comfortable, as the textbank directory is the one displayed first.

The exact procedure for creating plain text files may vary between different versions of the same program (the above was done with WinWord 7) and even more between different word processor brands.

**Preparing Graphical and Audio PDs**
Primary Document Management

Assigning Primary Documents

This is presumably the very next step after creating a Hermeneutic Unit. One or more files can be assigned to the Hermeneutic Unit at the same time. Simultaneously assigned files are ordered alphabetically.

To assign primary documents

1. Choose DOCUMENTS/ASSIGN from the main menu.
2. From the file dialog box select one or more primary documents to be assigned. For multiple selection techniques consult page 101.
3. You do not need to have file names displayed inside the Hermeneutic Unit. You can use DOCUMENTS/RENAME to change the name created by default to something that makes more sense.

Only files with legal extensions for primary documents (txt, asc, bmp, tif, jpg, wav, etc.) from the default textbank directory are presented by default.

Primary Documents are Accessed by Reference

An internal prefix for each assigned PD is created by the program (e.g., "P 23"). Even large numbers of primary documents assigned do not blow up the size of their hosting Hermeneutic Unit as one would expect when looking at the sizes of the individual files. This is caused by ATLAS/ti’s strategy to only keep references to the files.

It is important to understand, that the original files do not actually become part of the Hermeneutic Unit by this assignment. Several important characteristics are connected to this fact:

Disconnecting Primary Documents

Every now and then it may become necessary to remove unwanted primary documents from a Hermeneutic Unit.

This procedure does not remove the file from the disk. It solely removes the pd, its comment and all quotations from this Hermeneutic Unit.

To disconnect a primary document

1. Select the primary document to be removed from the Hermeneutic Unit.
2. Choose DOCUMENTS/DISCONNECT from the main menu.

Disconnecting a primary document leaves a "whole" in the sequence of PD ids. If you have three PDs indexed P 1, P 2 and P 3, disconnecting P 2 will result in a list of two PDs numbered P1 and P 3. The next primary document assigned will get a new id P 4. To get rid of such wholes, use the procedure described in "Renumbering all Primary Documents" on page 33.
Changing Path or Filename

Because primary documents are only included "by reference" any relocation of a primary document will result in a "file not found" error when trying to open it....

ANSI or ASCII? Changing the Encoding Type

The ANSI and ASCII encoding tables for plain text use different "codes" for special characters like German "Umlauts" (Ää, Üü,Ää,) or other languages specialties. A text prepared using a DOS editor or by exporting from a DOS based work processor which looked well under ATLAS/ti DOS will probably look weird in ATLAS/ti for Windows when it comes to displaying such special characters (technically: all characters outside the standard ASCII 7-bit coding table), e.g., like little squares.

Rearranging PDs

When assigning primary documents, they are sorted by their id (essentially a position index) initially. Although you can sort this list using different criteria during a session, the standard sort criterion will be the id when you start ATLAS/ti the next time.

To permanently rearrange primary documents you have two options available:

Renumbering all Primary Documents

Primary documents are loaded in read-only mode by default.

Loaded a Primary Document

Primary documents are loaded whenever they are needed for displaying, segmenting, coding, searching and printing. As long as you are working with only one Hermeneutic Unit simultaneously, only one primary document is loaded at any one time and displayed in the primary document pane.

Primary document are loaded in read-only mode by default, assuming that you are working on shared or archived material. This mode is perfectly OK for all common operations described later, like creating quotations, coding, and annotating. However, you can switch to edit mode if you need to do so.

There are several common methods to load a primary document:

- Clicking on its entry in the drop-down list or any other list where PDs appear.
- Selecting a quotation directly or by displaying quotations for a code or memo.
- From a network editor displaying the primary document's node.
To load a primary document using the drop-down list

1. Click on the down-arrow button to the right of the primary document drop-down list located underneath the HU editor's main tool bar.

2. Select a primary document by clicking on an entry in the list that dropped down in step 1. If the list does not display all PDs assigned, a scrollbar appears which you can use the to scroll through the list of entries. The selected primary document is now displayed inside the primary document pane.

To load a primary document using the object list

1. Open the primary document object list by clicking the extra list button to the left of the primary document drop-down list located underneath the HU editor's main tool bar.

2. Double-click an entry in the object list to load and display the primary document.

Navigating within Primary Documents

Once loaded you can scroll through the primary document. The options available depend on the media type of the selected PD.

Navigating within textual primary documents

- Textual primary documents allow for a large variety of options to display different parts of the text:

- A vertical and horizontal scrollbar is located to the right and at the bottom (when activated in the Views menu) of the primary document pane respectively.

- Once the primary document pane has the "input focus" (click into the pane with the left mouse button), the cursor (arrow) and other navigational keys on the keyboard can be used to move through the text. With the CTRL key as a modifier, the right and left arrow keys jumps to the beginning of the next and the previous word respectively.

- Jump to specific line numbers using the Go-to-Line button left of the primary document pane.

Navigating within graphical primary documents

- Displaying different parts of a graphical primary document is only possible by using the horizontal and vertical scrollbars.

Navigating within audio primary documents

- Not applicable.

---

4 In windowed environments like Windows, only one window or window pane can have the input focus and react to keyboard strokes or shortcuts.
Marking Text Segments

Probably the most common operation when working on primary documents is the marking of interesting sections and assigning codes and/or memos to them. Marking a segment does not by itself create a quotation - this must be explicitly requested, which is often the very next step after making a selection.

Just as with navigation, the options for marking segments differ between the three media types. The following describes the procedures offered to make selections in textual primary documents. For graphical and audio documents refer to "Special Handling of Non-Textual Primary Documents" on page 37.

The methods to make text selections are manifold:

- Making free-hand selections of arbitrary size.
- Using semi-automatic selections of words, sentences, paragraphs and files.

To mark text using the "free-hand" method

This method allows an arbitrary sequence of characters to be marked. Such a sequence can be as small as one character or as large as the complete text.

1. Load the primary text and scroll to passage that you want to mark.
2. Position the mouse-pointer at the beginning of the text to be selected.
3. Press the left mouse button (thereby establishing the anchor point of the selection).
4. While pressing the left mouse button drag the mouse to the end of the text passage.
5. Release the mouse button. The text passage is now marked which is indicated by a highlighted (or reversed) display of the selected text.

To mark text using semi-automatic click selection

As efficient short-cuts to selecting text segments of certain sizes, this technique also assures exact boundaries for the selected regions. The following sequence of steps selects incrementing portions of the text:

Double-click selection sequence:

1. Move the mouse pointer on a word.
2. Double-click to select a word.
3. Double-click again to select the sentence surrounding the current selection.
4. Double-click again to select the paragraph surrounding the selection. An empty line delimits a paragraph in ATLAS/ti (cf. page 31).
5. Double-click again to select the complete text.
6. Double-click once more to unselect the selected area.

The selection types word, sentence, paragraph are also used by the Auto Coding procedure (page 56) for marking and coding matching text passages automatically.
Note: If you wish to use quotation retrieval in later stages of your analysis, using semi-automatic selection guarantees the needed precision for queries with code combinations.

Creating Quotations

There is no other reason for making a selection in a primary document than to create a quotation (see page 10 for a definition). And even a quotation is commonly not an end in itself. Its creation is often an integral part of a higher-level procedure like coding or memoing described later.

Correct! There are many ways to create a quotation from a selection and these options are available via the tool bar, menus, and drag & drop.

Described below are the different procedures to create a quotation. Creating only a quotation, no code, no memo resembles making a "doodle" mark in the margin of a book to mark an interesting passage. There is no need to think of anything else.

For better feedback, activate the margin area

You might want to switch on the margin display to have this "doodle" feeling when creating a quotation.

To switch on the margin display

- Choose VIEWS/MARGIN AREA from the main menu or
- Click the Margin Area button in the tool bar left of the primary document pane.

To create a quotation using the Create Quotation button

1. Mark a section in the displayed primary document.
2. Click the CREATE FREE QUOTATION button in the tool bar left to the primary document pane or

To create a quotation using the context menu

1. Mark a section in the displayed primary document.
2. Move the mouse pointer into the marked region.
3. Open the selection's context menu with a right mouse button click.
4. Choose CREATE FREE QUOTATION.

Note: If you have accidentally chosen a section for which a quotation already exists, the context menu will be different and will not include the CREATE FREE QUOTATION command.

To create a quotation using Drag & Drop


Modifying Quotations

This feature lets you change the boundaries (start and end lines and columns) of a quotation.

Deleting Quotations

Quotations can be deleted. Of course, this will not remove anything from the primary document it is part of.

Affected by a deletion of a quotation are its references. For "free" quotations, the process will finish with no prompting. However, if it does have reference to one or more codes, linked quotations or memos, this will be indicated to you and you are prompted for confirmation before proceeding.

To delete a quotation

1. Select the quotation in question.

2. Choose QUOTATIONS/DELETE from the main menu or use the REMOVE button of the quotation object list.

3. Enter Yes or No in case the quotation has references.

Note: all quotations of a primary document are deleted when the latter is removed from the Hermeneutic Unit.

Special Handling of Non-Textual Primary Documents

Described here are special considerations when working with graphical and audio documents.

Handling of Graphical Primary Documents

Although you will note many similarities in the handling of graphical primary documents compared to textual PDs, there are also notable differences. Especially the creation of graphical quotations in graphical primary documents is different form the above described procedure for textual documents. Graphical quotations are rectangular sections of the original image.

To create a graphical quotation

1. Load the primary document as described in "Loading a Primary Document" on page 33.

2. Move the mouse pointer to the upper left corner of the rectangular section that you are going to create.

3. Drag (holding down the left mouse button) the mouse to the lower right corner of the rectangle.

4. Release the mouse button. You have now created a selection and the rectangle will be highlighted (inverse display or thicker border).
5. To register this selection as a quotation use the techniques described under "Creating a Quotation" on page 24. However, drag & drop creation is not possible for graphical PDs.

6. The newly created quotation can now be treated like any other quotation and will of course appear in the list of quotations.

Selecting graphical quotations

In addition to activating and displaying quotations by selecting it from the drop-down or double-clicking on its entry in the object list, graphical quotations can be activated by double-clicking inside its visible rectangle on the screen. If you click on an area where two or more quotations intersect, you are presented a list of quotations from which the one to be selected can be chosen.

Display of graphical quotations

All graphical quotations - not only the one currently selected - are indicated on screen by a border. A selected rectangle is visualized as a negative image or with a thicker frame.

The permanent display of the quotations' frame can be toggled on and off. Choose **TOGGLE AREAS DISPLAY** from the menu that pops up with the mouse pointer between graphical quotations on the graphical primary document area.

The style of the selected quotation can be controlled via the **SELECTION-DISPLAY TYPE** sub menu.

Special Handling of Audio PDs

Editing Primary Documents

Primary Document are Read-Only by Nature

One of the core ideas of ATLAS/ti viewed primary documents as parts of an archive, as shared documents not to be changed any more, with all further work being done on, not **within** the documents. As you already know, all quotations, codes and memos created are not actual parts of the primary documents, but are "transparency layers" defined within the Hermeneutic Unit referencing the primary documents.

Because of this primary documents are loaded in read-only mode by default, guarding the document from accidental modifications that might affect other users of the same document.
Coding Techniques

Introduction

The procedures of coding described in the following do not grasp the complexity of the intellectual efforts of coding as understood in the framework of Grounded Theory (GLASER & STRAUSS, 1967). However, they are also not mere indexing procedures, as annotating and commenting in every stage of the process is encouraged, supported by the system and should accompany the coding procedure regularly.

There is a variety of coding techniques available, some of which we have given names reflecting the impact of ideas and terminology of Grounded Theory on the design of the ATLAS/ti system.

The impact of Grounded Theory on ATLAS/ti's coding model is apparent.

During coding, a selection is registered as a quotation if necessary.

Four coding techniques are described in the following: Open Coding, In-Vivo Coding, Code-by-List, Quick Coding. In addition, the purpose and the creation of free codes is explained. The automatic coding procedure has its own chapter ("The Auto Coding Tool" on page 56) and needs some understanding of text search procedures as well.

All coding techniques result in quotations being linked with codes. There is no restriction on the number of codes assigned to a quotation and vice versa: a code my refer to an arbitrary number of quotations. If, in the process of coding, a selection does not refer to an exiting quotation, than it is automatically created.

General Coding Procedure

There are some common steps with every coding procedure described below.

First of all, it is quite useful to switch on margin display to see the immediate effects of coding text passages (the margin area is not displayed for graphic and audio documents). Furthermore, the margin area permits the direct manipulation of the displayed codes and the association between the quotation and the codes.

To switch on the margin area

- Choose Views/Margin Area from the main menu or
- Click the Margin Area button left of the primary document pane.

The margin area is a useful feedback device during coding. Codes attached to a text passage are displayed and scroll alongside.

Use the red split bar to change the size of the margin area.

**Note:** the relative size of the margin area can be changed by dragging the red "split bar" (red on screen, almost black in the print-out) between the primary document pane and the margin area with the left mouse button.
To code a selection of a primary document

1. Select a primary document.
2. Mark a section of the primary document you want to code.
3. Apply a coding procedure. You find the coding procedures either in the tool bar left to the primary document pane or in the context menu which pops up when moving the mouse-pointer into the selection and clicking the right mouse button.

A new quotation is created (if the selection did not match a previously created quotation) coded with the code(s) created or used in the coding procedure.

To code an existing quotation

You may also code an already coded or "free" quotation.

1. Select the quotation (this loads the primary document as well if necessary). The selection of a quotation can be done directly from the quotations list or by displaying through an already attached code.
2. Apply a coding procedure as described above.
3. The quotation is coded with the code(s) created or used in the coding procedure.

How Codes are Displayed

After a code has been created it appears as a new entry in the drop-down list and in any opened object list. You might have already noticed the two numbers separated by a dash enclosed by curly parentheses appended to the code's name by the system. Eventually, a "tilde" character is displayed after the closing parentheses.

Groundedness. The first number displays the number of quotations already coded with this code. The larger this number, the more evidence has already been found for this code in the data.

Density. The number following the dash is the number of other codes linked with this code. Codes with large numbers can be interpreted as having a high degree of theoretical density. This latter characteristic belongs to the networking procedures explained later.

Comment. The tilde character "~" marks commented codes. It is used for all kinds of objects for which comments were written.
Open Coding

Use Open Coding when you want to create a new code at the same time you are coding a section or exiting quotation of a primary document. The only other coding technique that creates new codes is described below ("In-Vivo coding" on page 26).

A code may contain more than a single word, but should be concise. If you want to use prose as code you might in fact want to annotate and should use memos instead.

To apply the Open Coding technique

1. Select the passage or quotation you want to code.
2. Click the Open Coding button or choose CODING/OPEN CODING from the selection's context menu.
3. In the prompter enter a descriptive name for the code to be created and click OK.

You will see some activity in the code- and quotation lists. The new code is now listed in the code list. If a quotation was created during the above step, a new entry for the quotation is inserted in the quotation list.

You should make it a habit to write a comment for every new code as explained below (see "Writing Code Comments" on page 43).

Creating more than one code concurrently.

More than one code can be created with open coding if you separate them with the special character "|" (ASCII 124).

Anxiety | Fear | Stress creates three new codes simultaneously. It is also possible to enter already existing codes which of course are not created anew.

In-Vivo Coding

Use In-Vivo coding, when the text itself contains a good name for the code.

The selected text itself is registered as a quotation and used as the code's name. Because the piece of text (often only one word) used to create the in-vivo code is often to sparse to serve as a quotation, this operation is often followed by a modification of the quotation (see also "Modifying Quotations" on page 37). This changes the size of the referenced quotation without touching its previously assigned code(s).
Note: In-Vivo coding can only be applied to textual primary documents.

To apply In-Vivo coding

1. Mark the text passage (no more than 40 characters) that you want to use as the new code's name.

2. Click the In-Vivo coding button or choose CODING/IN-VIVO CODING from the selection's context menu.

3. If you want to alter the quotation's size use the technique described under "Modifying Quotations" on page 37

In-Vivo coding can also be done quite efficiently by using the drag & drop technique, especially when creating more than one in-vivo code consecutively.

To apply In-Vivo coding using drag & drop

1. Mark the text passage that you want to use as the new code's name.

2. Open the object list for codes by clicking the button left of the codes drop-down list.

3. Position the mouse pointer into the selected text and drag it into the code list using the left mouse button.

4. Release the mouse button.

The Drag & Drop version of In-Vivo coding lifts the restriction of using a maximum of 40 characters.

Code-by-List

Use this technique if you want to assign existing codes to a primary document selection.

Any number of codes from a multiple-choice list can be chosen. If the selection equals and existing quotation only those codes which have not already been assigned to the quotation are offered in the list.

To apply the code-by-list technique

1. Select the passage or quotation you want to code.

2. Click the multi-coding button or choose CODING/CODE BY LIST from the selection's context menu.

3. A multiple-choice window offers all codes that can be assigned.

4. Click OK after selecting the codes. For multiple selection techniques consult page 101

Drag & drop is available for the code-by-list technique. The object list for codes has bi-directional drag & drop capabilities. It allows to drop text to create in-vivo codes and it allows to drag codes as a short-cut to the code-by-list method.
To code-by-list using drag & drop

1. Open the extra list for codes by clicking the object list button left of the codes drop-down list.
3. Select one or more codes in the object list.
4. Hold down the left mouse button and drag the codes into the primary document pane.
5. Release the mouse button.

Code-by-List - Inverse

The standard code-by-list technique associates a list of codes to one selected quotation. Once in a while, it may make more sense to associate a list of quotations to one code.

To link a number of quotations to a code

1. Select the code to be associated with one or more quotations.
2. Choose CODES/CODING/LINK CODE TO: from the main menu.
3. From the list of quotations select one ore more to be coded with the selected code and click OK. For multiple selection techniques consult page 101

Quick Coding

Purpose: to assigning the currently selected code. This is an efficient method for the consecutive coding of segments using the same code.

To apply the quick-coding method

1. Select a code in the code list window to make it the current code. The code may already be selected from a previous coding procedure.
3. Click the Quick coding button or choose CODING/QUICK CODING from the selection's context menu.

Writing Code Comments

Like primary texts, codes can be commented. Use this option to clarify the meaning of a code or to explain, how it is to be used for coding. Commenting objects greatly supports a cooperative working style.

To create or edit a code comment

1. Select the code.
2. Choose CODES/EDIT COMMENT from the main menu.
3. In the text editor write or edit an existing comment.
4. Save the changes and close the editor.
To edit a code comment using the code extra list

When you are working with the code extra list, you can access the selected code's comment in the yellow text pane below the list pane. Although not as comfortable as the text editor, it is useful for quickly browsing through a number of codes, displaying and editing their comments. To save any changes, open the text panes' context menu and choose SAVE.

More Coding Related Functions

Described in this section are some additional coding-related house-keeping functions like creating free codes, importing lists of codes, renaming and removing codes

Free Codes

Just as quotations do not necessarily have to be coded to justify their existence, so you can create codes that have not (yet) been used for coding purposes. All the methods described above create links between codes and quotations.

Reasons for creating free codes

- Prepare a stock of predefined codes in the framework of a given theory. Start coding in a "top-down" way with all necessary concepts ready at hand. This complements the "bottom up" open coding stage in which concepts emerge from the date.
- Create codes that come to your mind during the normal coding work which cannot be applied to the current segment but will be useful later.

To create a free code

1. From the main menu choose Codes/Create Free Code or click the Create Code button in the codes extra list (not the similar button left of the primary document pane).
2. In the prompter one or more codes (separated by the "|" character).

The newly created code's suffix displays {0-0}: no quotation, no code neighbor.

Importing a List of Codes

Renaming a Code

Renaming a code or deleting a code are procedures that look trivial, but sometimes understanding the "scope" of these operations poses a problem for the new user. For both operations you must understand, that there is only ONE code named "Happiness" in a Hermeneutic Unit, even if you applied this code many times to the primary documents.

To rename a code

1. Select the code in the code list (drop-down, extra list, anywhere).
2. Select Codes/Rename from the main menu.
3. Enter the new name and click OK.

**Note:** A nice method to rename a code (or other entities) using "In-Place" editing is to use the **Object Explorer** (see page 77).

### Deleting Codes

Deleting a code removes the code not only from a specific quotation, but removes it from any linkages it played a part in. If you want to remove a code only from a given quotation use the procedures described in "To unlink one or more codes from a quotation" on page 45.

Removing a code implies the removal of all existing associations between this code and the coded-quotations.

Please read the preliminary comments in "Renaming a Code" above.

**To delete a code**

1. Select the code in the code list (drop-down, extra list, anywhere).
2. Select Codes/Delete from the main menu. When you are working with the codes extra list, you can alternatively choose the **DELETE** button.
3. If the code has been used (assigned to quotations, linked with other codes, used by super codes) you are asked for confirmation before the process continues.
4. Furthermore, if quotations coded with the deleted code do not have any other context in which they are used, you are asked for confirmation to remove those quotations as well.

### Unlinking Codes

Removes the links between codes and quotations. Unlike with the remove function, neither codes nor quotations are removed, only the associations between codes and quotations is affected.

Two methods are described in the following.

**To unlink one or more codes from a quotation**

This method is very efficient if you need to remove more than one code association. To unlink single codes, the method described next might be more appropriate.

1. Select the quotation in question.
2. From the quotation's context-menu (move the mouse pointer into the selected area and click the right mouse button) choose **UNLINK CODES**.
3. From the multiple-choice list select one or more codes and click OK. *For multiple selection techniques consult page 101*

4. With the margin area switched on you should see the immediate effect of this operation: the unlinked codes are removed from the margin display.

**To unlink a code displayed in the margin area**

This method is an even more direct manipulation of the association between a code and a specific quotation.

1. Switch on the margin area display.
2. Open the context menu for the code in the margin area.
3. Choose **UNLINK**. The code's icon in the margin area disappears.

Step 2 also selects and highlights the quotation this *specific* code occurrence refers to.

*Figure 8 - Context menu for code in margin area*

**Decoding: Retrieving Coded Quotations**

This operation is the inverse function of coding and, as such, could be labeled "decoding". Such decoding offers immediate access on the primary data even when already concerned with higher level tasks, such as constructing theories. As with coding, there are also different ways to retrieve coded segments.
Click-Retrieval in Context

Depending on the media, quotations are displayed differently in context.

A straightforward method is to display quotations in context: a textual quotation will appear highlighted within its surrounding "con"text. Graphical quotations are displayed inverted or with a thicker frame when selected. A audio quotation will just play its contents when selected (you will not hear anything unless a sound device is attached to your system).

When you select a code in the code drop-down list or double click on an item in the code object list, the following will happen:

**Only one quotation coded with a code**

If there is only one quotation for the selected code, it is shown highlighted in the primary document window surrounded by its context.

**More than one quotation coded with a code**

However, if more than one quotation is associated with this code, a list is offered for selection. Selecting one of the entries displays the quotation in its context.

Quite comfortable is the use of the NEXT or PREVIOUS button of the code object list window, which allows to sequentially display the selected code's quotations.

Print Report of Coded Quotations

As an alternative to the in-context display of quotations, you can get a compressed version of all primary document passages assigned to a code by selecting "Quotations" in the code menu's print section.

Complex Queries

Retrieving only those data segments coded with ONE code is clearly not always sufficient and often, combinations of codes may seem more appropriate for specifying coded quotations. The retrieval of quotations using an arbitrary number of codes combined in complex "queries" may be achieved using the QueryTool, explained in detail in section "The QueryTool" on page 79.

Other Places to Activate Quotations

The code lists are not the only devices that offer the activation of coded quotations. First of all, it is equally simple to activate and display an arbitrary quotation, either coded or not. Simply select a quotation in the drop-down list or double-click on a quotation in the code extra list.

The Margin Area (page 15) displays codes (or other objects) associated with quotations. Clicking on an object in the margin area selects and highlights the referenced quotation in the primary document area. Because the margin area is only displayed for textual PDs, this method cannot be used with graphical or audio PDs.
The **Object Explorer** (page 77) also offers the display of quotations in context.

The **Network Editor** (page 64) also offers a path back to the primary data by double-clicking on nodes or by issuing command offered by the nodes' pop-up menus.

## Writing Memos

The writing of memos is an important task applicable in every working phase in a qualitative research project. Theory building, often only associated with the building of networks of codes, is also a case for using memos to capture longer elaborations of ideas. What has been captured in memos, are often the "puzzle pieces" to be put together in the phase of writing the final report.

Memos are generally relatively long explanatory and descriptive texts associated with other "objects" like quotations, codes, other memos. Memos can be "stand alone" - simply associated with the project.

### Difference between Memos and Codes

Codes are (or should be) succinct, dense descriptions of concepts emerging during the stage of closely studying the data. They often reduce complex and longish findings to crisp placeholders or even theoretically relevant concepts.

Beginners often stuff lengthy treatises into a code's name, making the distinction between codes, comments and memos less clear and often misusing codes for their more appropriate siblings.

If you find yourself using more than a few words for a code consider using memos or the code's comment instead.

Like codes memos have titles in addition. These titles are used to display lists of memos and help to find specific memos. These titles should also not be given lengthy names and should be used for easy identification of memos.

---

**Note:** Beginners sometimes use the memo's title as the actual text storage. This results in empty memos with giant one-line titles that are difficult to handle, almost impossible to print legible and creating giant nodes in network views.

---

### Difference between Memos and Comments

Memos are very similar to comments, in that both are devices to store longer (compared to concept names) texts. Comments exclusively belong to one entity (the primary text comment, etc). Unlike memos, comments are not displayed in a list separately from the object they are attached to. Memos can be assigned to more than one object and may have attributes.

### Memoing Primary Document Selections

The procedure to create and associate a memo with a selected passage of a primary document is similar to the coding procedures described above ("General Coding Procedure" on page 39).
To write a memo for a primary document selection

1. Select a primary document.
2. Mark a section of the primary document you want to memo.
3. Select the memo button left of the primary document pane.
4. Enter a title for the memo.
5. To edit the text body, choose MEMOS/EDIT from the main menu.

Memos may be assigned an attribute or "memo type". Using this type attribute, memos can be filtered.

Creating Reports

In fact, ATLAS/ti’s main strength is the support of online-activities like searching, browsing, creating links. However, once in a while you need to create something on paper or at least assemble results to be included in research reports or which can be studied off-line.

There is not one single location where you find report related commands, but for every object class there is an output menu which offers different options for creating "printed matters".

The following describes some general characteristics and some selected report procedures.

Output Targets

Test report options by directing output to the editor first. This might save a lot of paper!

All textual output can be directed to one of three devices: printer, text editor, and ASCII-file. With output redirected to editor you can modify the contents before finally printing or saving the output to an ASCII file. For inclusion in memos, other reports, etc. you would also choose the editor or a file as the target.

Figure 9 - Textual Output Targets

All output is preceded by a header containing the name of the hermeneutic unit, the current date and the name of the current author.
History of All Objects

The "evolution" of a Hermeneutic Unit measured by the creation date of objects assigned and created during the HU’s lifetime can be reported by choosing FILE/OUTPUT/ALL OBJECTS from the HU editor's main menu.

Frequencies: The Codes - Primary Document Matrix

A special output is generated by the choosing CODES/OUTPUT/CODES-PRIMARY-DOCUMENTS-TABLE. It cross-tabulates the primary documents (columns) and the codes (lines) with each cell counting the occurrences of the code in the primary document. Sums are computed for each row and column. Combined with sorting and filtering this is a useful "quantitative" option offered by ATLAS/ti. For more elaborate statistical analysis, the SPSS interface (cf. "SPSS Export" on page 92) should be used.

Generating Code Lists

This procedure creates an alphabetically sorted list of all currently filtered codes. Such a list is useful in two ways: It may be used to get a printout of all codes used in the hermeneutic unit, and it may be used as a "library" of codes, which can later be imported into another Hermeneutic Unit (see "Importing a List of Codes" on page 44). However, this method ignores any information except the name of each code. Additional information below the "tip of the iceberg", like time of creation, creator, comment or even structural links between codes, is ignored. A "flat", unstructured list of codes is created. To be able to reuse the code list later, it must be directed to a file (extension "COD").

To export complete code information including network connections see section "Theory Transfer" on page 75.

To create a code list

1. Set the code filter to include only those codes you want to be included. Without a code filter, all codes in the Hermeneutic Unit are processed.
2. Choose CODES/OUTPUT/CODE LIST from the main menu.
3. Select File as the output target if you want a code file for later re-import. Select the editor or the printer otherwise.

Code Output Dialog

As an alternative to the output oriented menu commands, a special "report generator" for codes can be accessed using the PRINT button in the code object list window.
Text Search

The Text Search Tool is used to search for the occurrence of specific text strings matching a search string or search pattern within primary texts. The ability to be searched is unique for "text" and makes this type of "media" the most sophisticated of all.

The Text Search Tool offers three different methods named **Simple Search**, **Category Search** and **GREP Search** to scan the primary text for specified text patterns. The functionality of the Text Search Tool is also fully integrated into the Auto Coding Tool (cf. page 56).

The mode of search is automatically determined by the kind of search string entered. Regular Expression search (GREP) must be explicitly selected with a checkbox.

The Text Search Tool

All types of text searches described in the following use the Text Search Tool below. Click the Text Search button left to the primary document pane, choose **DOCUMENTS/SEARCH** from the main menu or use the key combination **CTRL-F** to open the text search tool.

![Text Search Dialog](image)

*Figure 10 - The Text Search Tool with a search swarm named „PRONOUN“ currently selected*

Not all options are available with every type of search. For instance, the backward searches and case sensitivity is not available for Category Search.

Initially the status window display the current search mode. During search activity it might display warnings and error messages.

**Note:** The text search dialog box is "modeless". Therefore, you can leave the dialog box open, if you want to do different things (like assigning a code to the current text passage) and then continue searching. Use the **CLOSE** button to get rid of this window.

**Example:** *if your search specification just matched an interesting piece of text in your primary document, you can alter the selection of the found string, code it or do anything you consider useful. Then proceed to the next matching string by clicking NEXT/PREVIOUS.*
Text search starts at the current position of the text cursor in the currently selected primary text. When the end or - when searching backwards - the start of the current text is reached, you are asked if the search should continue through the rest of the primary texts:

![Search reached end of current PD](image)

### Search Options

Check the **CASE SENSITIVE** check box if you want characters to exactly match the searched text. With this option set, you will not find "Love" when you search for "love".

Not available in Category Search mode.

### GREP Mode

Checking the GREP check box enables GREP mode. If unchecked, the Search Tool is in "simple" mode. However, you may use the wildcard *, even without GREP. Not available in Category Search mode.

### Simple Search

Simple search searches for exactly the string specified as a search request. Note, that word boundaries are not recognized, so that searching for "the" would also signal success on "bother", "there" and "theta". Search with word boundaries is only possible with the Category Search.

**To start a simple search**

1. Open the text search tool.
2. Enter a string (not containing the special character ")") into the entry field.
3. Uncheck the GREP mode check box.
4. Set **CASE SENSITIVE** as needed.
5. Start the search by clicking the **NEXT** or **PREVIOUS** button.

### Category Search

The Category Search method combines two powerful techniques.

**Parallel search** is a sophisticated method to handle textual variations. You may specify a number of **patterns** that scan the text simultaneously. Secondly, such search expressions ("search swarms") can be stored under descriptive names ("category" or "macro") and are managed in search libraries.
Categories allow abstraction from the search string itself and permit easy reuse of complex expressions in later searches. With the simultaneous use of wildcards a search request using a search swarm might look as follows:

\*caus* | why | *efore | since.

The individual elements of the search swarm are delimited by the vertical bar character "|" (ASCII value 124), which can be read as a (Boolean) "OR". The search defined above might yield text passages including any of the following strings: "because", "causes", "causation", "before", or "therefore", as well as "why" and "since".

### Defining Categories

Search swarms which have proven to be useful can be assigned names in ATLAS/ti (search categories) to ease later re-use. E.g.,

```
CAUSE := *caus* | why | *efore | since
```

defines CAUSE as a placeholder for the original search swarm.

Therefore, instead of retyping lengthy search strings, the user may later simply type the search category "$CAUSE" into the entry field. The dollar sign is a special character which indicates that the content of the CAUSE category is to be used for the search and not the string "CAUSE" itself. Categories can be mixed with ordinary strings to build complex search requests:

```
until | $CAUSE | when
```

**Note:** no $ prefix is used when defining a new category.

As categories may be elements of other categories, recursive search trees of arbitrary depth may be defined. Another advantage of categories besides being short and handy, is, that you can modify these definitions later without the need to change a single character in other expressions using these.

### Hierarchies of Search Categories

**Note:** Do not mistake search categories for codes. Search categories are short cuts for complex search procedures, while codes are concepts representing some level of interpretation.

However, when using search categories extensively, building hierarchies of search categories, such a "para" taxonomy might stimulate the construction of a matching network of codes.

---

**Figure 11 - Hierarchy of Search Categories**

---
The definition for the category taxonomy above:

EMOTION = $POSEMO|$NEGEMO
POSEMO = $LOVE|.....others.....
NEGEMO = $ANGER
LOVE = love|loving|beloved|lover

Such category structures are efficient tools in exploiting the syntactical properties of the textual materials you use in your research. Of course, even these gadgets will not find any relevant information "between the lines".

**How To Use Category Search**

**To start a parallel search**

1. Start the text search tool if not already opened.
2. Enter a new search swarm, a category or select one from the library of search swarms by clicking on the drop down button at the right of the entry field.
3. Click NEXT.

*The Text Search Tool displays the list of search swarms contained in the current library.*

Some of the other controls become disabled because parallel search cannot be combined with GREP mode and it is never case sensitive. Also, the Previous button will be disabled. Currently the parallel search does only work forward (toward the end of the text).

**To define a new category**

1. Open the Text Search Tool.
2. Enter a new definition into the entry field using the syntax described above ("To define a new category" on page 54).
3. As soon as you press the NEXT button, a new category in the database of search expressions is created and a search for the first occurrence starts.

---

**Note:** The new category will be available for future searches only if you save them as a search library (see below).

**Managing Categories**

☆☆☆
**GREP Search**

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

The GREP search offers a subset of the Regular Expression language used in sophisticated text search systems. The core of a GREP search is the inclusion of special characters which control the matching process in the search string. Such „special expressions“ are frequently named „Patterns“. Examples: GREP search allows to find all occurrences of anything within parentheses (e.g., meta-linguistic paraphrases), or all occurrences of two-digit numbers followed by a period only if at the beginning of a line (e.g. speech turns of interviewees). (See examples below).

**Supported GREP Expressions**

\^ Anchors the following pattern to the beginning of a line. As the first character within brackets, it excludes the following characters (or range) from the search.

\$ Anchors the preceding search pattern to the end of a line

. The period matches any (single) character

* Matches any number (including zero) of the preceding expression

+ Matches at least one occurrence of the preceding expression

? Matches zero or one occurrence of the preceding expression: stones? matches stone and stones

[] Matches a range or set of characters: [a-z] or [0-9] or [aeiou]. [^0-9] = all nonnumeric characters

:d Matches any digit (equivalent to [0-9])

\ The escape character disables the special GREP functionality of the following character: \[ matches an opening bracket.

**Examples of GREP Searches**

**What to search for** | **GREP**
--- | ---
Find text (of arbitrary length) within brackets. Note, that the brackets had to be "escaped" with \\, as they are control characters themselves: | \\[^*\]*\]
Find all years between 1992 and 1997: | 199[2-7]
Find all numbers with 2 digits at the end of a line: | :d:d$ 
Find all "Meier"s | M[ae][iy]er
Find all lines beginning with one letter followed by a colon: | ^: 
(If you search for a "d" following the colon you will have to use the Escape character: ^\d Otherwise, :d will be interpreted as a digit.

**To start a GREP search**

1. Open the Text Search Tool.
2. Enter a search pattern.
3. Check the GREP box.
4. Click Next or Previous.
The Auto Coding Tool

The auto coding tool allows for automatic collection of text passages in one or more text documents ("Scope of Search"). It combines text search and code assignment.

All found passages are selected according to the specified setting (Matched String, Word, Sentence, Paragraph) and coded with the selected code.

Family Life

Families are containers for objects of type primary document, code and memo. They serve a variety of purposes. One important objective is to cope with possibly large amounts of objects by classifying them into sub sets, e.g., all theoretical codes, or all textual primary documents.

As named - and of course annotatable - "drawers" families can also be looked upon as nominal attributes. Furthermore, as one entity can be a member of more than one family, it so does have an arbitrary number of attributes. The partitioning of a set of objects into families and reduces the number of "chunks" requesting the researcher's attention. Families are often used in connection with filtering (cf. Sorting and filtering). Families (codes, primary documents) are also used for formulating complex queries using the Query Tool.

Common Procedures

Although the handling of families and their members (items, objects) is identical, their usage differs between the types of families. In the following, the general aspects and procedures on families are described. Hereafter, a description of specific characteristics of the different family types follows.

All procedures for displaying, editing, and creating families are found in the menus of their object type. To access code families, you would select Codes/Edit Families from the HU editor's main menu.

Open the Family Browser

The Family Browser is the tool for all editing of families and offers all the functions needed to create, remove, add and remove items, to filter objects and to write comments for families. It looks the same for all family types, except the title bar which displays the currently edited kind of families. All procedures here apply to all types of families.
To open a family browser

Select <OBJECT TYPE>/EDIT FAMILIES/OPEN FAMILY BROWSER from the main menu. Replace <ObjectType> with the type of families you want to edit: Documents, Codes, Memos.

The family browser menu

Some of the functions described below are accessible from the context menu that is available in the family list pane (move the mouse pointer into the list pane and click the right mouse button).

Creating Families

Before using a family, it must first be created using the family browser.

To create a new family

1. Open the family browser for the class of objects you want a new family created for.
2. Click the New Family button or choose New Family from the family list's pop-up menu.
3. Enter a name when prompted and click OK. If you enter the name of an already existing family, you will be asked if it is okay to overwrite. If you answer affirmatively, the existing family is cleared of all codes and its comment is erased. The new family will appear in the family list. The members list will be empty, just like the comment pane.
4. The next plausible step will be adding items to the newly created family and writing a preliminary comment.
Add Items

After the creation of a family this is often the next logical step. However, assignment and removal of items can be done at any time during the life cycle of a family.

To add new items to a family

1. Open the family browser for the relevant class of objects.
2. Select the family you want to add items to. If you have just created the family, it will already be selected.
3. Select one or more items in the list titled "Non-Members".
4. Click the Add Items button to make the selected items new members of the selected family. Watch the members' list being filled with the items just assigned.

Note: If you want to add one item after the other, you can simply double-click on each item to be assigned.

Removing Items

Removing items from a family is similar in "look & feel" to "Adding ...".

To remove items from a family

1. Open the family browser for the relevant class of objects.
2. Select the family you want to remove items from.
3. Select one or more items in the list titled "Members".
4. Click the REMOVE ITEMS button to remove the selected items from of the selected family. The items are not deleted from the system; only from this family. However, if you delete items somewhere else, they will be removed automatically from all their hosting families.

Note: If you want to remove one item after the other, you can simply double-click on each item to be removed.

Commenting a Family

A family's comment can be used to describe its applicability to others. This is especially useful, when treating families as attributes, which might need further explanation.

To create or edit a family's comment

1. Open the family browser for the relevant class of objects.
2. Select the family you want to edit the comment for.
3. In the yellow text pane write or edit the displayed comment.
4. Open the pop-up menu in this text area and choose SAVE.
Assigning Families to Objects

Using Families as Filters
One "added value" of families is the filtering of their respective objects, such reducing the amount of objects displayed in the lists. Example: With a double-click, only "Theoretical" codes are displayed in the code list.

To filter items using the family browser
1. Open the family browser for the relevant class of objects.
2. Open the extra list for this class of objects.
3. Double-click on a family in the family list. Watch how the display in the extra list changes to only display the items of the selected family.
4. Double-click on the family again to revert the filter back to "All".

Removing Families
Removing a family does only remove this container. It does not affect any of the contained items, other than that they lose this "attribute".

To remove a family
1. Open the family browser for the relevant class of objects.
2. Click the Remove button or choose Delete Family from the family list's pop-up menu.

Create Output
From the family list's pop-up menu two output options are available:
• Output
• Output all Families

Output
Output creates a report for the selected family including the name of the Hermeneutic Unit, the name of the family, its comment and all contained items. Optionally (you are prompted for confirmation) all quotations are included in the output as well.

Output all families
This kind of report generates a short information for all families of the selected object type. Quotations are not included.
Using Families for Merging Hermeneutic Units

Special families can be used to specify sets of objects, which shall not be included in the process of merging Hermeneutic Units (see "Merging Hermeneutic Units" on page 92).

Code Families

Code Families in Queries

The Query Tool not only allows the combination of codes to form complex queries, but does also accept code families as components. The interpretation of code families in queries is simple: a code family is treated as its elements combined with the Boolean operator OR. For more details refer to "The QueryTool" on page 79).

Code Families in SPSS Jobs

Another place were code families are exploited is when exporting data to SPSS syntax files (see SPSS Export on page 92 for details).

Creating Code Families from Network Views

Memo Families

Like codes, memos can also be grouped into families. Memo families as a grouping device allow the separation of theoretical memos from those referring to linguistic aspects of the text. The procedures for creating memo families and their handling resemble those already described for code families.

Primary Document Families

Primary Document Families as "Face Sheet" Variables

Besides sharing all the characteristics of code- and memo families, primary text families have an additional aspect: they may be used as selection criteria for limiting the Boolean search for quotations (cf. "Textbase Selection" on page 88). You may use primary document families as global attributes for the primary texts - supplementing codes, which may be viewed as attributes for quotations as parts of primary documents. If you have assigned the primary document family "female" and "under 25" to a number of interviews, you can then formulate queries like: "all quotations from interviews with females under 25 coded with "coping" or "power".

Primary Document Families in SPSS Jobs

In addition to being used within ATLAS/ti, primary document families, as well as code families, are utilized when creating a SPSS job (cf. "SPSS Export" on page 92).
Networks are the main ingredients for constructing theoretical models with ATLAS/ti. The establishment and refinement of relationships between the concepts found in the textual phase is now increasingly becoming more important. The "network" is an ubiquitous and powerful metaphor found in many different fields of research and application. Flow charts in project planning, text graphs in hypertext systems, cognitive models of memory and knowledge representation (semantic networks) are all networks that serve to represent complex information by intuitively accessible graphic means. One of the most attractive properties of graphs is their intuitive graphical presentation, mostly in form of two-dimensional layouts of labeled nodes and links.

In contrast to linear, sequential representations (e.g., text), network representations of knowledge resemble more closely the way human memory and thought is structured. Some cognitive "load" in handling complex relationships is reduced with the aid of spatial representation techniques. ATLAS/ti uses networks to help explore conceptual structures and to make them transparent. The networks add a heuristic "right brain" approach to qualitative analysis.

However, besides using networks for "mind mapping" and the visual design of theoretical models, you can also exploit the structural properties of code-networks to enhance the retrieval of quotations. Using networks of terms for retrieval purposes is a well known technique in information retrieval. In that domain, a structured sets of keywords is usually referred to as a thesaurus. This usage of networks for semantic retrieval is described in chapter "Semantic Operators" on page 82 in more detail.

The network of terms also allows "browsing" in addition to searching. Because nodes and links in ATLAS/ti do have properties beside their names (e.g., date, author, comment) the correct name for the networks used here is attributed graph.

When you have worked through this chapter, you should have learned about the following concepts and distinctions: network vs. network view, relation vs. link, nodes, layout, topological sort. The tools offered to create and manipulate networks: the network editor, the alignment tool, the tool palette, the relation editor, the import dialog, the setup dialog are described in later chapters.

**Nodes and Links**

The term network is formally defined within graph theory, a branch of discrete mathematics. A network is defined as a set of nodes (or "vertices") and links. A node in a network may be linked to an arbitrary number of other nodes.

The number of links for any one node is called its degree; e.g., a node with a degree of zero is not linked at all. Another simple formal property of a network is its order; the number of its nodes. You may make practical use of the degree of nodes by using it as a sort criterion in the codes list window. By the way, the second number in parentheses behind each code in the code list is the degree of the code.
**Directed Links**

Links are usually drawn as lines or arcs between the connected nodes in graphical presentations of networks. Furthermore, a link between two nodes may be directed or undirected.

A directed connection is drawn as a line with an arrow at one end. With directed links, source and target nodes must be distinguished. The **source node** is where the link starts and the **target node** is where it ends; where the arrow points to. **Connection** and **link** are used as synonyms.

Links are created either implicitly (e.g. coding a quotation "links" a code to a quotation without a full-blown link object) or explicitly by the user. The latter are described now in fuller detail.

*Strictly speaking do code-quotation associations also form a network.*

![Diagram of directed links]

*Figure 13 - Code-Quotations link*

**Node Types**

The user can manipulate and display almost all objects within the hermeneutic unit as network nodes: codes, memos, quotations, or primary texts. See "Node Types" on page 65 for details.

**Relations**

Nodes and links were already briefly explained above. 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").

*Meaningful relations have names*

The name of a link is displayed in the network editor as a little label attached to the link midway between the two connected nodes. There are six default relations - or link types - loaded at start-up time, which can be substituted or supplemented by **user-defined** relations. The relations initially available are (C1 and C2 are source and target nodes, respectively):

<table>
<thead>
<tr>
<th>Relations</th>
<th>Symbols</th>
<th>Width</th>
<th>Color</th>
<th>Formal Attribute</th>
<th>Layout Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 is-associated-with C2</td>
<td>==</td>
<td>1</td>
<td>black</td>
<td>symmetric</td>
<td>→</td>
</tr>
<tr>
<td>C1 is-part-of C2</td>
<td>[], P</td>
<td>1</td>
<td>black</td>
<td>transitive</td>
<td>↑</td>
</tr>
<tr>
<td>C1 is-cause-of C2</td>
<td>=&gt;, C</td>
<td>1</td>
<td>black</td>
<td>transitive</td>
<td>→</td>
</tr>
<tr>
<td>C1 contradicts C2</td>
<td>&lt;&gt;</td>
<td>1</td>
<td>black</td>
<td>symmetric</td>
<td>→</td>
</tr>
<tr>
<td>C1 is-a C2</td>
<td>isa</td>
<td>2</td>
<td>black</td>
<td>transitive</td>
<td>↑</td>
</tr>
<tr>
<td>C1 is-property-of C2</td>
<td>*</td>
<td>1</td>
<td>black</td>
<td>asymmetric</td>
<td>↑</td>
</tr>
</tbody>
</table>

Some of these characteristics of relations affect the display of links, others affect processing (search routines, automatic layout). A link between concepts
is displayed in a net-work editor by a line with the symbol as its label. The 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 head (->) directed towards the target code (C2). Symmetric relations are displayed without arrows in the network editor. An example of a transitive relation is the is-cause-of relation: from C1 is-cause-of C2 and C2 is-cause-of C3 follows C1 is-cause-of C3. Transitive relations also enable the "semantic retrieval" described in chapter Query Tool.

There are some additional properties that the user may define when building new relations: a comment explaining the relation, an alternative display-symbol for the network editor, the text which appears in the menu by which you select the relation to be used by a link and the width and color of the line that is displayed between two linked nodes using this relation and the preferred layout direction which affects the automatic layout that can be used for rearranging the nodes.

**Link vs. Relation**

It is important to understand the difference between a relation (or a link's type) and the link itself: There is only one "isa" relation, but maybe dozens of links using it. In the network view below, "isa" is used only once while "]" (is-part-of) is used seven times.

Another way to think of links and relations is to view the former as instances of the latter. Or you can view relations as templates from which links are generated. This view has a minor flaw: Unlike other things produced from templates, links are well informed about the characteristics of their templates. If a characteristic of a relation is changed (e.g., line width, color, symbol), these changes are propagated immediately to all links using it.

**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 "epistemological primitives" they constitute the main questions that guide the development of a model, a theory.

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 phases ("open coding") of this approach. In the analysis of argumentation structures other relations are more suitable: "is evidence of", "is contradictory to", "warrants". A medical expert attempting to capture diagnostic knowledge would use: "is-symptom-of", "is medication for". Relations can also be viewed as role indicators for concepts: e.g., the concept "Broken Leg" is assigned the role "condition" in the example below:
The Network Editor

Introduction

The network editor visually connects 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 and memos with the mouse, move them around the screen, draw, and cut links between them.

The following describes various methods available for creating and editing network views.

Network vs. Network View

This is an important distinction that is necessary for understanding the way networks are handled within ATLAS/ti.

Network view = logical structure + visual layout

A network in ATLAS/ti is the set of all objects and their links inside the Hermeneutic Unit. It does exist independent of any display oriented characteristics (layout, color, line width, etc.). It is the logical structure of the HU's objects. It does exist even with no network view yet created (although this is rarely the case, as network views are the preferred context to link nodes).

A network view ("netview") is often only a subset of this global structure of nodes and links combined with an individual layout of the nodes. A network editor is a "window" on the network or certain parts thereof.

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 their "parent" Hermeneutic Unit.
- Network views can be commented.
- They are displayed and edited in the Network editor.
- Network views allow the manual placement of the nodes
- One object can be a member of an arbitrary number of network views, just like a code can be an element of more than one code family.

Although network views allow for a very flexible display of the logical object structure, there are a few constraints that have to be kept in mind:

If a code A is linked to code B with relation R, then every network view that contains code A and code B will also include relation R. Furthermore, as only one link can exist between any two nodes, no network view will display any other relation between those two nodes.

**Figure 15 - A network view consistent with the view shown in Figure 14**

**Node Types**

The following object classes can be displayed and edited as nodes within the network editor:

**Codes as Nodes**

Codes are the most prominent objects that form the ingredients for an ATLAS/ti network. Terminologies, semantic networks - are all made up of codes.

**Memos as Nodes**

Memos in networks are often an important supplement for code networks. However, network views can also be misused to establish a comfortable "pick list" for a group of conveniently arranged memos.

When dragging a piece of text from a "drag enabled" application (like WinWord) into a network editor it becomes a memo with an automatically assigned label, which can be renamed later.
**Primary Documents as Nodes**

Primary documents as nodes are useful sometimes, but in the presence of quotations may clutter the view because of the possibly many links displayed. However, just as with the memos, networks make a nice graphical content table for graphical primary documents. With the "Full Image for PDs" option enabled, "thumbnail" images of the primary documents are displayed. Note, that this option is only available for graphical PDs.

**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. The inclusion of quotations in a network view supports the construction and the browsing of hypertext structures. Quotations can be included in a network view by simply dragging a piece of text from the primary document (drag possible from text documents only).

**Basics**

**Creating Network Views**

Two methods for creating network views are available. The first one creates an empty network view into which objects are imported in a subsequent step. The other method creates a network view from a selected object and its direct neighbors.

**To create a "tabula rasa" network view**

1. Select **NETWORKS/NEW NETWORK VIEW** from the HU editor's main menu.
2. Enter a name for the new net view. The new view is now displayed in a network editor with no nodes yet.
3. Import nodes with any of the methods described in section "Importing Nodes" on page 71.
4. Arrange the nodes.
5. Save the network view. This view will now be a part of the Hermeneutic Unit and is displayed in the pick-lst and netview browser (don't forget to save the HU at the end of your session, of course).

**To create a focused network view**

A focused network view is created "on-the-fly" from the selected object and its direct neighbors. Proceed as follows:

1. Open an object browser (e.g., a code extra list) or the Object Explorer.
2. Select an object with a left mouse click.
3. If available, click on the browser's network button. Alternatively open the object pop-up menu. *A network editor opens with the selected object and its neighbors.*
The nodes are initially placed using the semantic layout procedure but can be rearranged manually.

More nodes can be added to this network view using different techniques (see "Importing Nodes" on page 71).

**Note:** each time a network is opened on a selected object, a new focused network view is created. To open a previously created and stored focused network view, choose it from the pick list or network view browser.

### Opening a Network View

If there are already named network views in the Hermeneutic Unit you can open a network editor on a specific network view by one of the following methods.

**To open a network view from the pick list**

1. Click on the main toolbar's network button or choose **NETWORKS/Pick Network View** from the main menu.
2. Select one of the listed network views.
3. A network editor will open on the selected network view

**To open a network view using the NetView Browser**

Use this method if the number of network views already defined is too large to be comfortably displayed using the pick-list method.

It also allows to access some properties of network views without the need to open a network editor.

1. Select **NETWORKS/Open Network Browser** from the main menu.
2. Click on one of the displayed network views.
3. Click on the network button above the list

### Selecting Nodes and Links

Selecting nodes is an important first step for all subsequent operations targeted at individual objects.\(^5\) Such operations will have their corresponding menu commands in the network editor's main menu grayed out (disabled) if objects were not selected before.

**To select a single node**

1. Move the mouse pointer onto the node to be selected. The mouse pointer will change its appearance to .
2. Click the left mouse button.
3. The selected node will be displayed inverted.
4. All previously selected nodes will be deselected.

---

\(^5\) This precedence of "operands" before the operator is typical for many window/mouse based techniques and it closely resembles the Reversed Polish Notation used for creating queries in the Query Tool.
To select multiple nodes - method 1

1. Hold down the CTRL key on your keyboard.
2. Select a node as described above.
3. Repeat steps 1 and 2 for every node to be selected.

To select multiple nodes - method 2 ("marquee selection"):  
This method is very efficient if the nodes to be selected fit into an imaginary rectangle

1. Move the mouse pointer to the upper left corner of a rectangle representing a bounding box for all nodes to be selected
2. Hold down the left mouse button and drag the mouse pointer to the lower right corner of this rectangle.
3. Select a node as described above.
4. Repeat steps 1 and 2 for every node to be selected.

Selecting Neighbors

Selecting a Link

Links are "first class" objects just like the nodes. Selecting links is similar to selecting nodes.

To select a link

1. Move the mouse pointer onto the label of the link to be selected. The mouse pointer will change its appearance.
2. Click the left mouse button.
   
   The selected link label will be displayed inverted. All previously selected nodes will be deselected.

Selecting links is a convenient way to cut multiple links (main menu or context menu on link).

Deselecting nodes or links

To deselect a selected node or link

1. Hold down the CTRL key on your keyboard
2. Click on a selected node or link.

To deselect all nodes and links

1. Move the mouse cursor over the network editor's background.
2. Double-click the left mouse button

This function is also available from the menu or can be invoked by using the key combination CTRL-A.
All selected links will also be deselected.

**Moving Nodes**

By moving nodes to different positions you can fine-tune an initial layout created by the automatic layout procedure.

For precision placement of nodes use the node alignment procedures and tool.

**Note:** Do not forget to save the network view (and the HU itself at the end of the session) if you want to make the new layout permanent.

**To move a single node**

1. Move the mouse pointer onto the node to be selected.
2. Hold down the left mouse button.
3. Drag the selected nodes to its new position.

**To move multiple nodes**

1. Draw a frame around the nodes to be moved, or
2. Click on each node one at a time holding down the **CTRL** key to prevent previously selected nodes from being deselected.
3. Press the **CTRL**-key and drag the selected nodes to their new position.

**To move nodes with the arrow keys**

1. Select one or more nodes using the selection methods described in "Selection Techniques" on page 101.
2. Use the arrow keys in combination with the **CTRL** key to move the node(s) "pixelwise" in all four directions.

*Ctrl-Z - Undo node placements*

To undo an erroneous placement, use function **UNDO POSITIONING** or key combination **CTRL+Z**.

**Linking Nodes**

The links between the nodes in a network correspond to "hard-wired" connections between the objects. Thus, creating and removing links should not be regarded as solely "cosmetic" operations. Links make changes to the hermeneutic unit's "database".

There are several ways to link nodes but this one is recommended for linking exactly two nodes:

**To link two nodes**

1. Place the mouse pointer over the source node. The mouse pointer changes its appearance.
2. Hold down the **SHIFT** key on your keyboard.
3. Hold down the left mouse button and drag the mouse pointer to the target node. A "rubber band" between the mouse pointer and the source node gives some feedback for this procedure.

4. Release the **SHIFT** key.

5. Release the left mouse button.

6. Select the link type (for code-code and quotation-quotation-links only).

---

**Note:** A selection of link types is only offered if either two codes or two quotations are linked.

If more than one source node is to be linked with a target node use the Link Nodes method described next

**To link more than two nodes**

1. Choose the *source* nodes.

2. Choose **LINKS/LINK NODES** from the network editor's main menu.

3. Move the mouse pointer with the "rubber bands" to the *target* node and click the left mouse button.

4. In case of code-code and quotation-quotations links you are prompted to select a relation for each pair of nodes. If you want to connect two source nodes A and B to target node C, you are first asked for the relation between A & C and then for the one to be used between B & C.

**To link nodes using the list method**

[Icons for list method]

**Creating New Relations**

[Icons for list method]

**Cutting Links**

Several approaches for disconnecting previously linked nodes are possible.

The first method works for all types of links and is useful when many nodes linked to one other node are to be disconnected:

1. Select one or more nodes whose connections to one other node are to be removed.

2. Choose **LINKS/CUT LINKS** from the network editor's menu.

3. Move the mouse pointer with the "rubber bands" to the target node.

4. Click the left mouse button

Alternatively:

1. Click on one or more link labels.

2. Choose **LINKS/CUT LINKS** from the network editor's menu.
And:

1. Move the mouse pointer over an existing link.
2. Open the context menu and choose **CUT LINK**.

**Note:** The latter two methods work on "first class" links only: code-code or quotation-quotation ("hyper") links.

## Importing Nodes

As usual, there are also several options available for including "networkable" objects in a network view.

**To import nodes**

1. Choose **NODES/IMPORT NODES** from the network editor's menu bar. This opens a window offering to include nodes of different kinds into the network view. Only those objects are listed, which are not already a member of this view.
2. Select the object class from which you want to choose.
3. Select the object(s) to be imported into the current network view. *For multiple selection techniques consult page 101.*
4. Click the **IMPORT** button.

After importing, the nodes are placed in the upper left corner of the network editor.

*The Import Nodes dialog box offers access to all types of "nodable" objects. The list pane offers a context menu to select or deselect all displayed objects.*

![Import Nodes window](image)

**Figure 16 - Import Nodes window**

**To import nodes from object lists using drag & drop**

You can alternatively import nodes by dragging objects from extra lists into the network editor. Drag & drop gives you better control of the initial position of the imported nodes.

1. Open the network view you want new objects to be imported to.
2. Open an extra list for objects (code list, memo list, etc.) which you want to import into a network view.
3. Select the node(s) you want to import into the network view.
4. Drag the selected objects into the network editor.
To drag selections of a primary text into a network editor

Drag text from other applications into a network editor

Import Node Neighbors

Excluding Nodes from Network View

Removing nodes does not remove the objects represented by these nodes from the database. An object which has been removed, can be "re-imported" using the node import functions described above at any time. You are only making changes to the network view, not to the database of nodes. By incrementally importing and removing of nodes you can construct "well formed" network views.

To exclude nodes from a network view

1. Open the network view.
2. Select one or more nodes to be excluded using the selection methods described in "Selection Techniques" on page 101.
3. Choose NODES/REMOVE NODES FROM VIEW from the network editor's menu or use the CTRL-DEL key combination on the keyboard.

Note: To exclude a single node, open its context menu and choose REMOVE FROM VIEW.

Node and Link Actions

Commands affecting selected codes or links are available via their respective context menus. Via left mouse button double-clicks a selected procedure is "fired".

Actions via Context Menus

Context menus can be activated on nodes and links. Dependent on the type of the node or link selected, these menus will offer specialized options to choose from.

To open a context menu, move the mouse-pointer onto the object and click the right mouse button.
The two menus popped up when the mouse pointer touched the node/link and the right mouse button was clicked. Note, that the Edit Comment option is not available. Both objects simply do not have a comment yet.

To execute node actions per double-click

1. Position the mouse pointer over a node.
2. Double-click the left mouse button

The table below describes the specific actions launched for the different node types.

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codes</td>
<td>Display the code comment.</td>
</tr>
<tr>
<td>Memos</td>
<td>Display the memos contents</td>
</tr>
<tr>
<td>Quotations</td>
<td>Display the full text of the quotation (load primary document file if necessary)</td>
</tr>
<tr>
<td>Primary Documents</td>
<td>Display the PD's comment</td>
</tr>
</tbody>
</table>

Printing Networks

Printing networks is currently done rather primitive by "dumping" the screen representation of a network's visible area onto the printer.

To print a network view

1. Open the network view and arrange all nodes to be printed.
2. Choose NETWORK/PRINT NETWORK VIEW from the network editor's menu.

Open SPECIALS/PREFERENCES from the menu and click the PRINTING tab to set some print options like inclusion of the network view's title, its comment and an optional frame around the network view.

As the print is a 1-to-1 representation of the screen image, the quality of the print depends on the screen resolution.

A better solution would print the network views as "vector graphics" with print quality only restricted by the resolution and other capabilities of the printer.

In some cases network views do not print at all, just showing an empty frame.

Do the following to get them on paper:
Cosmetics - Network Display Properties

Before creating transparencies from network views for your next presentation, there are different options available to alter the appearance of nodes and links and even the background.

However, all settings created using the Display menu of network editor are lost after closing this editor. If you want to change colors and fonts globally for all network views and permanently use the options offered by network preferences.

Colors

The color of the network editor's background and the color of nodes can be set independently.

Choose DISPLAY/SET COLORS and then one of the options offered. You will be offered a standard color chooser dialog from which you pick the color to be assigned.

Fonts

The font used for nodes and links can be set independently choosing DISPLAY/SET FONTS from the network editors main menu.

Node Appearance

Under the Display menu of the network editor you will find a large variety of options to alter the display characteristics of nodes and links. Some of these options affect all nodes regardless of their type. Other options change the display of certain node types only.

Layout Procedures

Two methods are available for the automatic placement of nodes.

Semantic Layout

Shortcut: 

CTRL-L

Place the nodes within the window using the semantic layout algorithm. This algorithm tries to place the nodes into optimal positions using an invisible matrix of default positions. It tries to place the nodes with the highest connectivity into center positions. By recursively applying the same method to the rest of the nodes using neighboring positions of the first node placed, the algorithm tries to avoid overlapping nodes and too many crossing links. However, an optimal solution cannot be found in a reasonable time.

The results of the automatic layout procedure are often quite reasonable and at least a good starting point for subsequent manual refinement of the nodes' placement. If you are not happy with the layout produced, you can revert to the previous placement by using UNDO POSITIONING (Ctrl-Z).
Topological Layout

This special layout procedure tries to create a linear list of nodes positioned from the upper left to the lower right. This sequence is the result of a depth first traversal of the graph. The algorithm tries to resolve as many constraints between any two nodes so that a node with the least dependencies is made the first node positioned in the upper left corner, and the node with the most dependencies on other nodes is positioned in the lower right corner of the network editor.

Theory Transfer

By "Theory Transfer" we understand the reuse of codes, relations and code networks produced in one project in subsequent projects.

Two different strategies are supported: The reuse of "flat" (unstructured) lists of codes (names only) in another Hermeneutic Unit and the transfer of rich representations of codes (including comments, authorship, creation date) and the network they are part of.

Merging Codes

One or more codes may be incorporated into one other code. This code then "inherits" all the references (quotations, other codes, memos) and comments of the incorporated codes.

This operation is not an abstraction function, but for cleaning-up purposes. One common use is the unification of codes of similar meaning but different naming (synonyms), especially after theory import or merging of Hermeneutic Unit created by different authors.

Two different ways to merge codes are described: a list based method and one that works from within a network editor.

The two images below show three codes and their links to other objects before and after merging two of the codes into the third.

Two codes "Magic 7" and "Magic 3" have been selected. Using "Merge Codes", the two codes will be merged into "Number Magic".

Figure 17 - Code Merge: Before the merge
After the merge: The two former sub codes of "Number Magic" - "Magic 3" and "Magic 7" have been merged into "Number Magic". All references to quotations, other codes and memos have been "inherited" by "Number Magic".

Figure 18 - Code Merge: After the merge

**To merge codes using the list method**

1. Select a code into which a number of other codes is to be merged.
2. Choose **CODES/MISCELLANEOUS/MERGE CODES** from the HU editor's main menu.
3. In the multiple choice window select the codes to be merged into the selected code.

**Note:** Using the list method, you select the target code first. Then you choose the codes to be merged. This is "inverse" to the method described in the following.

**To merge codes using the network editor**

1. Open or create the network view which contains the codes to be merged.
2. Select all "source" codes to be merged into one target code. Make sure that the target node is already visible in the network editor.
3. Choose **NODES/MERGE CODES** from the network editor's menu bar.
4. Move the mouse to the target node and click the left mouse button.

**Note:** The codes incorporated by the merge procedure are deleted from the Hermeneutic Unit's internal code database. As you cannot revert this procedure easily, and a lot is modified by this procedure, save the Hermeneutic Unit before doing the merge. When you are not satisfied with the result, you can then revert to the previous state by reloading this HU.

**The Relation Editor**

The Relation Editor allows the creation, display, and editing of relations used for linking codes to codes or quotations to quotations. Only these two types of links use explicit relations that can be edited (unlike the links between codes and quotations, that are unnamed, type-less relations).
The Object Explorer

Although the structures that can be created with ATLAS/ti are not necessarily restricted to hierarchical trees, the ATLAS/ti Object Explorer provides a strictly hierarchical view on the Hermeneutic Unit and all the objects it references (primary documents) and contains. Its main purpose is a powerful content table which may be used to navigate between the different parts of the HU. Unlike the „pure“ extra lists which display only one type of objects, the Object Explorer allows the integrated and structured display of all the heterogeneous objects that make up the HU: primary documents, quotations, codes, memos, families and networks.

Because of the non-hierarchical structure, you will notice some redundancy when expanding the tree view. Quotations are listed under each primary document, a primary doc can be a member of a primary document family, so it appears there as well, and so do all its quotations on the next hierarchical level.

Divided into two „panes“ like many tools ATLAS/ti offers for handling list of objects, the upper pane displays the tree structured contents of the Hermeneutic Unit and the text pane below displays a commentary associated with the currently activated object. The red split bar permits relative resizing of the two panes.

Figure 19 - The Object Explorer

The Object Explorer’s „Look & Feel“ resembles the Explorer that can be launched from Windows 95 and Windows NT desktops to display and manipulate computers, disks, directories, and files.

The root object of the tree is - of course - the Hermeneutic Unit itself. The direct branches from this root are: Primary Docs, Codes, Memos, Primary Doc Families, Code Families, Memo Families and Network Views.

How to?

The Object Explorer can be launched by clicking on the button in the main toolbar or by selecting EXTRAS/OBJECT EXPLORER from the main menu.

Note: more than one Object Explorer can be opened at a time (pressing the Ctrl key when opening). This is useful if different parts of the HU are to be observed concurrently. However, you should be careful when editing objects in parallel instances of the Object Explorer as changes are not “broadcasted” to other instances to synchronize the information displayed.
Renaming Objects

Almost all objects can be renamed using the „in-place“ technique which is also used by the Windows Explorer and other Windows 95 file dialog boxes to rename files and directories.

To rename an object in the Object Explorer

1. Click on an object's name to select it.
2. Click on the selected object's name again. After a short delay, the list entry changes its appearance into a small text entry field.
3. Edit the name of the object (you can use the arrow keys, and cut-copy-paste).
4. Click on the objects iconic representation to stop editing.

Opening Object Menus

Click on an object with the right mouse button, which will select the object if not already selected, and will open a context menu if one is defined for the object type. The menu displayed offers a few selected object specific operations.

**Note:** a very useful operation is the display of quotations in context. Choose „Display Quotations“ for codes and memos, „Display in Context“ for quotations.

Displaying and Editing Comments

To edit an object’s comment, select it and edit it in the text pane below the tree view. Save the result by opening the context menu inside the text area and choose „Save“.

Code Trees & Forests

Special Object Explorers can be invoked to display codes only. From the CODES/MISCELLANEOUS menu choose CODE TREE to display the currently selected code as a root of a tree of connected codes.

To display all root codes with their trees, choose CODES/MISCELLANEOUS/CODE FOREST.

Except the restriction of the display to codes, the procedures described for the Object Explorer apply as well.

**Note:** the conversion of an arbitrary network of codes into a hierarchy results in redundant replication of codes in sub branches. Only the Network Editor is capable of displaying non-redundant networks.
The QueryTool

**Searching for quotations with codes**

The QueryTool is used for the retrieval of coded text, graphic or audio segments (= quotations) using the codes they were associated with during the process of coding. This is different from text search: To search for occurrences of text matching a specified pattern or string, you have to use the search command instead.

A *query* is a search expression build from operands and operators that define the conditions that quotations must meet to be retrieved.

### The Screen

The QueryTool is activated by clicking the QueryTool button or by choosing **CODES/OUTPUT/QUOTATION RETRIEVAL** from the HU editor's main menu.

![Figure 20 - The QueryTool](image)

The QueryTool is a multi-pane window consisting of the following main parts:

- **The operator toolbar**, located near the left margin of the window.
- The **code-family pane** in the upper left lists code-families to be used in queries.
- The **codes pane** underneath the code-family pane contains all codes currently in the Hermeneutic Unit.
- The **term-stack pane** in the upper right displays the stack of all expressions entered in the current query. If more than one entry is visible, there are arguments still waiting to be used in the query. The topmost entry is the current query and its result is displayed in the **result list** located in the lower right of the window.

Directly underneath the term-stack pane is a **feedback pane** displaying the topmost entry of the term stack in two different notations. The red split bar lets you change the relative size of the two panes.
Operands

*Operands*, also called arguments or descriptors, and *operators* are the only ingredients of *queries*. No parentheses or other purely "syntactical flavor" is needed to construct valid queries.

**Basic Operands**

Two sorts of *basic descriptors* may be used in a query: *Codes*, *super codes* and *code families*.

Code families may for the purpose of queries be interpreted as their member codes connected by the Boolean operator OR.

**Code Families as Operands**

*Note:* When you are using code families as terms in your query, you can look at them as their contained member codes being combined by an OR operator.

Simply selecting a code family F1 which contains five codes C1,...,C5 is equivalent to the query: C1 or C2 or C3 or C4 or C5.

**Complex Operands**

The term "Operand" does not only apply to basic descriptors. An operand can be *any expression* that itself is used as an argument for an operand forming a more complex expression. A complete term (e.g., "A and B") may be used in a more complex query as an operand: "NOT(A and B)", "(A and B) or (C and not D)"; etc.

All types of operands can be freely mixed in a query using any of the operators described below.

Operators

**Overview**

Three sets of operators are available. They are located within the toolbar at the left edge of the QueryTool screen. From top to bottom:

- **Boolean** operators allow combination 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. In other information systems these operators are often referred to as "Thesaurus Operators".

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

*Note:* You can display a short help for how to use each operator by right clicking on its corresponding button in the toolbar. These tooltips also pop up if "Display Tooltips" is switched on in the HU-Editor's View menu.
**Boolean Operators**

Four Boolean operators are available with the QueryTool: OR, XOR, AND, and NOT.

OR, XOR, and AND are *binary operators* which need two operands as input. NOT is a *unary operator* and needs exactly one operand. However, the operands themselves may be of arbitrary complexity. Codes, code families or arbitrary expressions can be used as operands: (A or B) and (not C and D).

**OR** The OR operator retrieves all data segments (quotations) that are coded with any of the codes used in the expression. Example: "All quotations coded with 'Earth' OR 'Fire'". As arguments can themselves be arbitrary expressions, another formulation is: "all quotations that are retrieved by any of the expressions the query combines". Example: "All quotations coded with 'Earth' OR 'Fire AND Water'".

**Note:** the OR operator potentially generates MANY hits. It has a high "recall", but low "precision".

**XOR** The OR operator does not really resemble the everyday usage of "OR". Its meaning is "At least one of.." including the case where ALL conditions match. The XOR operator in contrast 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** The AND operator finds quotations that match ALL the conditions specified in the query. 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.

**NOT** The NOT operator tests for the absence of a condition. Technically it subtracts the findings of the unnegated term from all text segments available (the "universe"). Given 120 quotations in the Hermeneutic Unit, 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 the argument (e.g., "NOT (Earth OR Fire)" (= neither Earth nor Fire).

**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).

![Figure 21 - Boolean queries depicted as Venn diagrams](image-url)
**Semantic Operators**

The operators in this section exploit the connected codes resulting from previous theory building work using the Network Editor. Unlike the Boolean operators, which closely resemble set operations, the semantic operators derive results from a - hopefully - more meaningful structure of semantically related concepts. While Boolean based queries are extensional by simply enumerating the elements of the combined sets (e.g., LOVE or KINDNESS), semantic operators are intensional, as they already capture some meaning expressed in the semantic network (e.g., sub(POSITIVE ATTITUDES)).

**SUB** The SUB (or DOWN) operator traverses the network from higher concepts to basic ones collecting all quotations from an of the sub codes. Only "transitive" relations between the codes are processed, all others are types ignored. When building a terminology from your codes, use the ISA relation for sub-term links.

Example: "All quotations coded with Magic or any (direct or indirect) sub-term of Magic". Like the OR operator in the set of Boolean operators, the SUB produces rather large result sets. However, because unlike the OR operator you make use of a theory using SUB, the "precision" (= to get only what you expect) is much better. Of course, if your network contains absurd connections ("computer ISA intelligent entity"), the quality of your retrieval will decline.

**UP** The UP operator looks at all direct super codes of the selected code. Unlike the SUB operator it does not recursively traverse the structure. Only the next level is considered.

**SIBlings** The SIBlings operator finds all quotations that are connected to the selected code or any other descendants of its parents. Example: "All quotations coded with Horror %4 or any other value of Horror". This works because all values of "Horror" have a common ancestor ("Dimension: Horror").

**Note:** only codes, neither code-families nor arbitrary expressions can be used as operands for semantic operators.

---

Sample queries on the network in Figure 22:

sub (Positive Attitude)
=> \{Q1, Q2, Q3, Q4, Q5\}

sub (Negative Attitude)
=> \{Q6, Q7, Q8\}

sub (Attitude)
=> \{Q1..., Q8\}

---

*Figure 22 - A hierarchy of concepts suitable for semantic retrieval*
Proximity Operators

The operators in this section need exactly two operands (of arbitrary complexity) as their arguments. They differ from the other operators in an important aspect: proximity operators are non-commutative. This property makes their usage a little more difficult to learn.

Non-commutativity dictates a certain sequence of entering 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 whose quotations you are interested in. „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 you are interested in. If B’s quotations are requested, you have to enter „B encloses A“ using the query language described below.

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

Embedding Operators

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

**WITHIN** A WITHIN B retrieves all quotations coded with A that are contained within text passages coded with B.

**ENCLOSES** A ENCLOSES B retrieves all quotations coded with A that contain text passages coded with B.

Overlap Operators

The overlap operators describe quotations that overlap one another.

**OVERLAPPED BY** A OVERLAPPED BY B retrieves all quotations coded with A that are overlapped by text passages coded with B.

**OVERLAPS** A OVERLAPS B retrieves all quotations coded with A that overlap text passages coded with B.

Distance Operators

The distance operators describe a sequence of disjoint quotations. The maximum distance in lines may be specified.
FOLLOWS  A FOLLOWS B retrieves all quotations coded with A that follow quotations coded with B.

PRECEDES  A PRECEDES B retrieves all quotations coded with A followed by quotations coded with B.

The Co-occurrence Operator

CO-OCCURRENCE  A CO-OCCURRING WITH B: Find all quotations that co-occur with B. This operator is essentially a short-cut for a combination of all the basic proximity operators via OR.

The Query Language

Queries are built from operands and operators using the principle of Reversed Polish Notation (RPN). This sounds difficult, but is not after getting used to it once. RPN, which was invented by the Polish mathematician Lukasiewicz, does not require any parentheses to control the priority of operators, nor does it require any other purely "syntactical sugar" like commas, periods, etc. Every click produces a meaningful result and it is impossible to create syntactically wrong queries (of course you can create a lot of semantic garbage).

Operands First, Operators Next

The most prominent - and also often confusing aspect for the beginner - is the order in which operands and operators of a search expression are entered. Using RPN operands are entered first, and then the operators. This is a very unusual method for most of us who have grown up and got used to a notation where operators are placed between the operands. Most calculators use this "infix" notation.

Infix: good for reading.
Suffix: good for clicking.

Two aspects must be distinguished: how we read expressions and how we formulate them with a "point and click" language. I believe that the infix notation is easier to read, but the "suffix" notation is far easier to use when creating queries using direct manipulation user interfaces like Windows.

Although maybe a bit weird in the beginning I am convinced that you will feel quite comfortable with this method once you got acquainted to it.

Creating a Query with the QueryTool

Other than in the arithmetic example above, the retrieval of quotations with the QueryTool is a bit different. We are really not interested in the operands themselves, but in the set of quotations that is the result of evaluating an operand. When we formulate a query „All A OR B“ we are indeed meaning: „All quotations coded with A OR quotations coded with B“. Thus, „entering“ the operand code „X“ displays all quotations which were coded with „X“ in the
result list. And even this list of the quotations’ short names is not the result we are actually looking for. With yet another step we can generate a report listing the full quotations or we can display each quotation in the context of its primary document for further refinement.

To assemble a query with the **QueryTool**, you also select the operands first, then the operators.

The most simple query is one without any explicit operators at all: Just double-click on a code or code family. The result (if any) is immediately displayed in result list.

Now double-click on another code or code family. The same thing happens: the result for this operand is displayed.

**Build complex queries incrementally with immediate feedback after each step.** Starting from such simple examples it is easy to construct a more complex query immediately: click on the OR operator (the topmost in the left toolbar). The combined result is instantly displayed in the result list.

**Query Examples**

The examples below use the Hermeneutic Unit „BIBLE.HPR“ which you should have loaded and displayed while reading the following.

**How to Formulate a Boolean Query**

Sample query using Boolean operators: „Give me all text, graphical or audio segments coded either with any of the codes included in code family BIG FAMILY or with those in code family MAGIC STUFF but which not coded with code MAGIC 3“

1. Double-click on code-family BIG FAMILY. The result list (in the lower right) immediately displays 13 hits (might vary with the HPR version).

2. Double-click on code-family MAGIC STUFF. The result list now displays all quotations coded with this code family. Two operands (BIG FAMILY and MAGIC STUFF) are now on our (term-) stack.

3. Click operator XOR (the „either“ in the query above) and watch the result of applying the XOR operator on the two previously entered operands. You have now created ONE new complex operand to be used in the following steps!

4. Double-click on code MAGIC 3. The result list now displays all quotations coded with 'Magic 3'.

5. Click the NOT operator. The result list now displays all quotations not coded with 'Magic 3'. We now have two operands on the "stack" and may use another - binary - operator:

6. Click the AND operator. The result of ANDing the two operands produced in the previous steps ("Big Family XOR Magic Stuff", "NOT Magic 3") is displayed in the result list.

**How to formulate a query with semantic operators**

Example using Boolean and semantic operators: „All text segments coded with MAGIC or any of its sub-terms, but not including BLACK MAGIC“

---

**ATLAS/ti Short User's Guide**  The QueryTool  •  85
1. Double-click on code MAGIC. One "hit" is displayed in the result list -
the only quotation directly referenced by this code.
2. Click on the SUB operator. The result list fills significantly (22 hits).
3. Double-click on BLACK MAGIC. Result: Nothing!
4. Select the NOT operator. Result: Everything!
5. Select the AND operator. Of course: 22 hits.

**Note:** Not the most clever query, but should you assign this little "program"
to a super code and use it again later (when Black Magic refers to something)
you will receive more meaningful results. Without any reformulation of the
original query!

**Creating a query using proximity operators**

A sample query using proximity operators: „Show me all text passages coded
with MAGIC or any of its sub-terms that are embedded within larger contexts
coded with HORROR %4“.

1. Double-click on code MAGIC
2. Click the DOWN operator.
3. Double-click on code HORROR %4.
4. Click the WITHIN operator.

A number of lines for the maximum distance between two segments can be
specified in the lower left of the QueryTool:

```
Distance in lines: 5
```

**Note:** Enter an appropriate line count *before* entering any of the distance
operators.

**Super Codes**

"Intelligent" super codes are „frozen“ hypotheses.

Super Codes are a convenient way to store your queries. Super Codes are very similar to normal codes with one important
difference: instead of „hard wired“ connections to quotations, super codes store a query to compute their virtual references when
needed.

They might be thought of as „frozen hypotheses“ to be "tested"
against the data repeatedly. They „automagically“ change their
behavior in the ongoing course of your theory building work. If you
have a super code „All about Magic“ with a query „SUB Magic“
and later add another sub code „White Magic“, all quotations the
latter code points to are also retrieved by the (unchanged) query of
„All about Magic“. Super codes can be clicked on in the code list
like any other code and they will display their quotations in an
identical way.
Normal codes have "hard-wired" quotation references.

Super codes recalculate the stored query "when-needed" and deliver the result "on demand".

Super Codes can contain Super Codes can contain.... Of course can super codes be used in code families, network views and even in other queries, giving you the option to incrementally build the most complex queries.

Creating Super Codes from Queries

To create a super code, you must have already constructed a query using the QueryTool which is displayed in the term stack. Note, that because super codes are "intensional", you can also create a valid and useful super code with an empty result list (which might well change in a later stage of your research work).

To create a super code

1. Click on the "Super-Code" button.
2. Enter a name for the new Super-Code or accept the default name created from the query expression.
3. Enter Return. The newly created super code immediately appears in the list of codes and can be used for new queries (and super codes) right away.

A default comment for the new super code is also created showing the assigned query in infix notation.

Auto-Optimization of Supercode Queries

What You Cant do with Super Codes

Coding

Some operations cannot be applied to super codes. The most important constraint is, that you cannot associate them with quotations directly. Therefore, super codes are not presented when doing "code by list" and drag & drop onto data selections is prohibited.

Merging

Merge operations including super codes is also not a legal operation. However, when we find a way to meaningfully and economically combine the query terms
of the participating super codes, such a procedure might be available in a future upgrade.

**Create Cycles**

If you created a super code whose query contains a reference to a code family, you cannot assign this super code to the code family later. This would create a cyclic structure and is therefore forbidden.

**Textbase Selection**

A window similar to the QueryTool’s lets you specify the texts which are to be included in any subsequent query. By default, the query's "document universe" is all primary documents in the Hermeneutic Unit. Clicking „Textbase Selection“ opens another window which shows the primary documents in the lower left pane and the primary document families in the upper left. As primary document families can be looked at as nominal variables, it is easy to pre-select "all interviews with male interviewees aged between twenty and thirty from small towns".

**Output**

**Cleaning up the Hitlist**

Before creating a report of all the quotations found by the query you have the option of removing entries in the hitlist before doing so. You can remove unwanted hits from the list using the erase button above the printer button. Especially after queries with low precision (lots of potential garbage), it is very useful to be able to sort out the false hits before sending the result to the printer.

Simply select a quotation in the hitlist and click the **Erase** button.

**Note:** Removing elements for the result list has implications for the creation of super codes.

**Creating a Report**

To print all "hits" found with a query, click on the **Printer** button at the right of the result list.

**Miscellaneous Functions**

Several functions are available in pocket calculator style as a series of buttons. Other functions can be activated by context menus opened in one of the list panes.
The structure generating procedures described so far have focused on the creation of code networks. But other types of nodes can be linked to networks as well.

You may already have tried to link memos to codes or memos to memos. These links cannot be explicitly specified, as is possible with code-code links. The direct linkage of data-segments (quotations) to data-segments offers similar flexibility in choosing and defining relations. Almost all of the editing facilities described for code networks also hold for text-links.

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

While a code offers fast access to sets of data segments, it defines only a simple relation between the latter: equivalence.

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

No code is needed to traverse from quotation Q1 to one that it contradicts (Q2).

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

---

6 The term "hyper" is based on this distance independent traveling through a hyper-space of text (hyper-space is a science fiction term describing a fast traveling method using higher dimensional "holes" in space as shortcuts).

7 It should be stressed once again, that we refer to textual, graphical and audio segments when we speak of "text segments".
Today's largest hypertext system: the World Wide Web

Computer related hypertext applications include the Windows online help system which displays operational information in suitable small chunks (compared to printed information) but with a considerable amount of linkage between the information nodes. A well known hyper media structure is the World Wide Web with textual, graphical and other multimedia information linked together not just across file boundaries but leaving the scope of a single computer with information nodes distributed world wide. The knowledge of the world is steadily compiling into a giant network, which can be accessed with any computer linked to the Internet "information highway". By the way, with ATLAS/ti's HTML generation feature, you may actively contribute your own knowledge to the "web".

Advantages of Hypertext

What are the advantages of direct connections between text segments, compared to the traditional procedures of qualitative text analysis?

Let us ask another question: How can you express, that a statement X in text A is in contradiction with another statement Y in another 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??

Pure classification is not always adequate

The "code & retrieve" paradigm, which is so prevalent for many systems supporting the qualitative researcher is not adequate for certain types of analyses. Formally looked at, attaching codes to pieces of data creates named sets of segments with almost no internal structure. No question: partitioning lots of text segments into sets is by itself very useful: classification leads to manageable amounts of segments which may be retrieved by using the key words (codes) which serve as their "handle".

Hypertext makes text-text relations explicit

The concept of hypertext introduces explicit relations between passages. These links have to be build 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 statements contrary to statement x". Hypertext allows to create different paths through the text you are analyzing. You may create a time line different from the strict sequence of the original text.

ATLAS/ti incorporates procedures for creating and browsing hypertext structures. It allows for named relations to link the selected quotations. Further, you can create graphical maps (using network views) to make parts of your hyperspace accessible in a comfortable way. Hypertext links may cross-ref to other documents (inter-textual links) or may link segments of the same primary document (intra-textual links).

Creating Hypertext Links

This method not only allows you to connect previously generated quotations, but also mark an arbitrary text passage in the primary text pane to become a text node. The newly created section is registered as a quotation first.

To create a hypertext link in context:

1. Mark a passage or select an existing quotation.
2. Choose option **CREATE LINK SOURCE** either
   - from the quotations list menu,
   - from the selection's context menu or
   - by clicking on the "source anchor" button
3. If a new passage was selected, a quotation is automatically created and appears as a new entry in the quotation lists (if visible)
4. Mark another passage or select a quotation to which the "source" is to be connected.
5. Choose "Create link target" (menu or "target anchor" button).
6. Select the type of relation to be used to connect the two quotations.
7. Select either one of three choices:
   - **Create a star:** Proceed linking with the original quotation still selected as the source anchor. Then proceed with step 3.
   - **Create a chain:** Make the target anchor the new source anchor. Proceed with step 3.
   - **Finish linking.**

To validate the established structure you can open a network editor on the target node immediately using the quotation's context menu. An immediate effect of creating hypertext links can be seen in the quotation list window, where linked quotations are prefixed with either: >, < or <> denoting source anchors, target anchors or quotations, which are both source and target.

**Traversing Hypertext Links**

1. Switch on the margin area.
2. Open up the properties context menu in the margin area and select **OBJECT TYPES/HYPER-LINKS**.
3. Double-click on a hyper link displayed in the margin. A pop-up window displays the hyper link's contents just like in the method described above.
4. To display the hyper link in context click into the pop-up window. Clicking outside the pop-up window cancels the process.

*The margin area permits a very intuitive traversal of hyper links.*

*Figure 24 - Direct hyper-link traversal using margin area*
WWW, Merging HUs, SPSS Export and Collaboration

World Wide Web Publishing
Create a structured publication for the World Wide Web (WWW)

Merging Hermeneutic Units
The Merge Tool (re-) unites Hermeneutic Units that were originally divided for analytical or economical reasons. Its main purpose is the support of teams. It links back together the contributions of different members of a research team. A common scenario is the analysis of different sets of documents by different team members, sharing a common code base. Of course, the single researcher can also benefit greatly, using the merge facility to help organize large projects.

SPSS Export
ATLAS/ti provides an export function to permit further processing of the syntax file by SPSS®, the widely used Statistical Package for the Social Sciences.

Collaboration
Collaboration or multi-authoring means that more than one author may work on a hermeneutic unit. ATLAS/ti supports this team work by systematically keeping track of each author's productions.

User Management
If you do not intend to use ATLAS/ti in a workgroup environment, the only procedure you might want to know is how to modify the default account Super - unless your name is "Monty".

ATLAS/ti knows two classes of users: administrators and all others. Administrators have more rights than "normal" users. The main task of administrators is to define new users.

Recommendations for Classroom Use
If ATLAS/ti is to be used in a project or a classroom situation and it is intended to have groups of authors collaborate, it is recommended that every user be registered in the users database. This is done by the administrator defining an account for every user. Accounts include the login-name, a password, the real name and the privilege level.
ATLAS/ti's user database is independent from the account database in a local area network.

Although user management is an increase of bureaucracy, its purpose is to support a cooperative working style. All hermeneutic units and the objects it contains are marked with the currently logged-in author's account name. The technical prerequisite for team work is either a single PC running ATLAS/ti or, much better, a local area network (LAN) with the system stored on a server and running on a number of workstation PCs.

All hermeneutic units and primary texts that the group is working with should be stored in one publicly accessible directory somewhere in the network. These hermeneutic units would still have access restrictions because of the administrative layer introduced by ATLAS/ti. Access to individual hermeneutic units is always controlled by their owners.

The procedures defined below are for use by the administrator only. They include the definition and modification of user accounts and passwords.

The User Database

All users who have been defined are maintained in a special "database"-file HERMENCN.HDB, located in the system directory of ATLAS/ti. This file is loaded when the system is started. Don't modify the file from outside ATLAS/ti, as this can corrupt it. After installation of ATLAS/ti there is already one "dummy" user with administrators privileges defined: Account = SUPER, password = USER. Because automatic login is the default mode, it is likely that you are presently working under this default account. Because of its administrative privileges, this mode lets you can gain access to the following user-management functions.

The User Administration Window

A special editor window for maintaining the user database is activated by choosing EXTRAS/USEREDITOR from the main menu. Without administrator privileges, this option will not even be included in the menu. In this case, please choose EXTRAS/LOGIN and enter SUPER and USER. If that doesn't work, you might not have sufficient rights to use administrative functions.

Before making changes to the user database, it is a good idea to make a backup copy of file HERMENCN.HDB located in the program's directory.

![User Administration Tools](image_url)

Figure 25 - User Administration Editor

The user administration tools allows to manage users even on non-networked computers to support team work.
**Sortage and Filtering**

What are the main strength of a computer? According to traditional criteria, these are the sorting and retrieval of data. To apply such techniques, the data must have attributes by which it can be ordered and filtered. The entries in the various list windows (codes, quotations, memos, etc.) display only the "tip of the iceberg" of the data objects. Various attributes provided by the system (date, time, author) and others specified by the user can be used for filtering and sorting data. Sorting and filtering is available for almost all entities within a hermeneutic unit: primary documents, quotations, codes, and memos.

**Pack & Go - Moving Hermeneutic Units**

Sometimes it is necessary to take a hermeneutic unit to another location. You might want to work at home with a Hermeneutic Unit that is stored on your computer at work. Simply copying the HU using the Windows explorer is not sufficient, because the primary documents referenced by the HU are not copied. To copy all primary documents along with the hermeneutic unit, you have to use **COPY BUNDLE** instead.

**Example:** Applying **COPY BUNDLE** on Hermeneutic Unit BIBLE.HPR will copy 5 files to the destination disk: BIBLE.HPR, REV1.TXT, REV2.TXT, KABBALA.BMP, INDIAN.TXT, and a batch file BIBLE.BAT containing the COPY commands needed to restore the files. In case your HU contains references to outsourced memos, these are copied as well.

**Primary Document Mapping**

Moving a project between different computers (and often different directory structures) is now possible without having to change the path for each PD every time.

**Example:** All or some primary documents in your Hermeneutic Unit reference files in directory H:\TEXTBANK. You want to work on this HU on your computer at home, but here you do not have a disk H:. Simply copying the files packed with Pack & Go into C:\TEXTBANK will not suffice. Every time a primary document is needed, e.g., when clicking on a PD or a quotation is to be displayed in context, the attempt to load the file will fail and produce an error message.

**Checking Redundant Codings**

This procedure identifies overlapping or embedded quotations that are associated with the same code. Such codings - which are considered as redundant - might have been the result of normal coding but can often occur unnoticed during merging procedures affecting quotations (merging HUs or modifying quotations).
Outsourcing and Inlining Memos

Memos, like commentaries, are an actual part of a Hermeneutic Unit. Unlike primary documents they contribute to the overall size of the HU. Analysis of existing HUs did show, that memos are often responsible for a significant share of the size of large Hermeneutic Units.

Outsourcing memos allows to convert single memos or all of them into external files, replacing the text body itself with internally generated unique file names. Inlining is the reverse operation to outsourcing. It reintegrates outsourced memos back into the Hermeneutic Unit.

What's not in the Menus - Drag & Drop

Drag & Drop is an interaction technique offered by Windows and accessible by many sophisticated programs. It offers a "speechless" access to functions that are often otherwise available using menus and tool bars. It is a "direct manipulation" technique, that is often more economical than having to issue a number

Applications of Drag and Drop

- Drag files from the (Windows) explorer or file manager into the primary doc list.
- Drag Hermeneutic Units into the Hermeneutic Unit editor (same as loading a new one or - with the Ctrl key pressed - merging another HU into the current)
- Drag entries within the primary document extra list to change their position.
- Drag text between text editing areas.
- Drag text from other applications that are OLE-2 drag servers (like WinWord) into text areas. If you can drag text from your word processor into another application, then it is such a "server". A very neat and fast way to produce plain ASCII texts from Word files!
- Drag all kinds of objects into network editors.
- Dragging text from Word for Windows into a network editor creates memos.
- Drag object lists into the margin area.

Personalize your Screen

Make yourself comfortable: If you prefer another image to be displayed as the start-up "wallpaper" you can specify any bitmap graphic of decent size. All file formats that are valid graphical primary documents are accepted.
Troubleshooting

Of Bugs & Nuisances

Sad but inevitable: no software and no operating system is totally error free. Software of a certain complexity (as ATLAS/ti) can by no means be formally verified. Only by way of empirical tests (by us developers and by you, the user) can such errors be found. In addition, the development of ATLAS/ti is a process of evolution with significant participation of the users providing feedback, making proposals for new features, reporting "bugs".

The majority of errors is harmless, most of them are never recognized as such under normal conditions.

Other "errors" are such only in the eye of the user and sometimes prove to be part of the software "by design".

The Error-Log File

Most errors that occur during the operation of ATLAS/ti are displayed and written to a special log file named ERROR.LOG. This text file is located in the user system directory (in single user versions, this is the PROGRAM directory of the ATLAS/ti path structure).

The contents of this file can only be interpreted by the developers of the program and is usually of no use for others. In case of bug reports you might be asked to send this file to a specified support e-mail address (usually support@atlasti.de). Please do NOT send such a file unasked and please do NOT send it to all members in the mailing list.

Known "Bugs" and Missing Features

Getting Support

Your primary resource for problems that arise during the use of ATLAS/ti should be our mailing list (see "The ATLAS/ti Mailing List" on page 105 for details).

Of course, please check the manual first for a description of your problem.

Service Packs & Patches

Updates in form of "patches" and service packs will be published on our web server once in a while and can be downloaded by users.
**Downloading Service Packs**

Check our web site now and then if new patches or service packs are available. If you are a member of the mailing list you will receive notification whenever new software was uploaded to our server.

The service pack page is directly accessible from our home page http://www.atlasti.de.

**To download a servicepack or patch**

2. Click the Download Section.
3. Read the information in the download section's page. It might well be that some patches are not applicable for your release of ATLAS/ti.
4. Download the text for the patch or service pack that you decided to download. Read this text carefully.
5. Download the software.

**Using the Service Pack Manager**

The Service Pack Manager installs the software that you have downloaded in the previous step and updates your system accordingly. Before you can use this feature, you must copy the downloaded patch into ATLAS/ti's root directory on the disk, which by default is: c:\program files\scientific software\atlasti.

After successful installation, you will have a slightly or significantly changed program with less bugs and added or modified functionality.

**To install a service pack or patch**

1. Download the patch and copy it to the ATLAS/ti directory
2. Start ATLAS/ti
3. Choose **EXTRAS/SERVICE PACK MANAGER** from the main menu.
4. Choose the service pack in the list displayed by the service pack manager.
5. Click OK.
7. Restart ATLAS/ti. Usually the installation of a patch or service pack increments the so called "build" counter. You can verify the new build number in the start-screen of ATLAS/ti. It is always displayed right behind the version number (e.g. 4.1 (build 51)).

**Note:** if you do not get offered any service packs through the service pack manager the reason might be one of the following:

- The patch does not apply to your current release. (has already been installed or your current build number is too low).
- The patch was not copied into the correct directory after downloading.
References

The HU Editor's Menus

Described here are the menus for the Hermeneutic Unit-Editor accessible via the main toolbar underneath the title bar. Some of the menus are also accessible as context (pop-up) menus in the respective panes using the right mouse button.

The Margin Area Context Menus

The margin area is a multi-purpose device and displays a variety of menus depending on the type of information currently displayed. By default, it shows codes used alongside the coded text passages.

The main context menu pops up when doing a right click in the margin area with the mouse pointer between any of the displayed objects. When clicking on an object, a specialized context menu for the selected object pops up.

Object Type sub menu

Set the type of objects (codes, memos, hyper links, just serpents or all objects) displayed in the margin area (codes by default). This setting can be made permanent using the general preferences HU Editor tab. With LINE Refs checked, the start and end line of the referenced quotation is displayed as a suffix for each object.

Lists sub menu

Choose an object list to be displayed in the margin area. This makes the functionality of the margin area identical to the extra object lists behavior. The context menu inside this area than changes as well, displaying the object types usual context menu. An added command at the bottom of the menu BACK TO ANNO lets you switch back to normal annotation mode.

One per Line

This is the default setting. When more than one object belongs to one quotation, they are displayed on an extra line each. Unchecking this option displays all these objects on one line (wrapped by the windows' border) separated by a comma.

Use Images

Displays the object's type icon along with its name. This improves discrimination of objects especially when selecting All. However, as the space needed for each object increases, clutter does as well.

Set Font

Set the font for the margin area. This and all other options above can be set by the general preferences dialog.
The larger the primary text font and the smaller the margin area font, the least clutter is generated.

**The Margin Area Object Menus**

When right-clicking not between but on an object, a specialized menu pops-up offering operations for the selected object.

![Menu Screenshot]

**Display Info**

Display name, creation date, date of last modification, author and comment for the object.

**Display Comment**

Display the comment (or the text body of a memo) in a pop-up window. This is also the double-click operation on an object in the margin area (except quotations).

**Edit Comment**

Open an editor for the selected object's comment (or memo text).

**Open Network Editor**

Create and open a focused network view for the selected object.

**List Quotations**

Display a list of other quotations the object refers to (not available for quotations).

**Unlink**

Remove the association between the object (e.g. a code) and the quotation.

**Edit Link Comment**

Quotations only. As an association between two quotations (the one selected in the primary document pane and the one in the margin area) is a fully qualified link, you have the option to edit this link's comment using this command.

**The Network Editor's Menus**

This section of the help file explains systematically the menus and options available from the network editor's menu bar. Note, that there are also object specific context menus available. These are activated by moving the mouse pointer over an object (node or relation label) and clicking the right mouse button.
Appendix

Paths & Folders

After the successful installation of ATLAS/ti for Windows, a directory structure is created where the program's system files and the user's files reside. Given that the default settings for paths during setup were accepted the following structure is created on your hard disk:

All files relevant for ATLAS/ti are stored in the **SCIENTIFIC SOFTWARE** branch under the **PROGRAM FILES** standard location for Windows 95 applications.

Folder **ATLASTi** is the root for all ATLAS/ti related files and directories - currently the only sub folder of Scientific Software. This will change with more tools Scientific Software will offer in the future. This folder is where you have to copy future service packs and patches to in order to update your system using the service pack manager.

By default this folder is also the location for the timer controlled backup copies created in specified intervals.

Folder **HTML** contains all auxiliary files needed and generated when using the HTML generator.

Folder **LOCLINST** will be created during network aware installations only (site license needed). It contains a specialized setup routine for quick & easy client installations.

Folder **PROGRAM** contains the executable files for ATLAS/ti, the database of users and the main configuration file ATLAS.INI. Do not move any of the files here to other locations. In single user installations it also contains default relation files, search string libraries and the user configuration file USER.INI.

Subfolder **SCIENTIFIC SOFTWARE** contains the short cuts for the example projects, the manual, and the program.

Folder **TEXTBANK** is of more relevance for your everyday work. It is the default location for all Hermeneutic Units and primary documents. This folder is used when you open the file dialogs for loading Hermeneutic Units, assign primary documents, etc.

The location of the **TEXTBANK** folder and the **HTML** folder can be changed using the general preferences' **PATH** tab.
Windows Interface Techniques

This section describes a few general procedures offered by Windows 95 and Windows NT. Although general knowledge of Windows is assumed for the reader of this documentation, a few of the most frequently used techniques are briefly described in the following.

Selection Techniques

Many of the windows and dialog boxes opened when working under the Windows operating system ask the user to select one or more items from a list for subsequent processing. These items are often files, but can be objects of any kind. When using ATLAS/ti, you will find yourself clicking and selecting files, texts, codes, memos, users, and other entities. Selection is the primary means by which the user identifies objects in the interface, and is therefore one of the most important aspects of the latter.

Single object selection

Selecting a **single object** is simple: all you do is click on it with the left mouse button, the selection button. When another object is selected, the previously selected object is unselected and the new one is highlighted in the list.

Multiple object selection using the **CTRL** and **SHIFT** modifier keys

How to select multiple objects? First of all, not all lists offer multiple selection capabilities, as this would not make any sense under some circumstances. Multiple selections can be either **disjoint** or **contiguous** or a mixture of both. In both cases it is necessary to prevent a previously selected item to be unselected. You have to use **modifier keys** on the keyboard in combination with mouse clicks to accomplish this effect.

When the user presses the mouse button, the starting point, or anchor point, of a selection is established.

The **CTRL** key is the is the disjoint, or toggle, modifier. When you press the **CTRL** key while making a new selection, any existing selection is preserved and the selection's "anchor point" is reset to the item selected last.

**To select a disjoint set of items:**

1. Select the first item in the list using the left mouse button.
2. Hold down the **CTRL** key on your keyboard.
3. Select other entries in the list.
4. Unselect entries you do not want to be part of the selection by clicking on them again.

**Note:** This selection technique is also available within the network editor for the selection of multiple nodes.

Contiguous or range selections are done with the **SHIFT** key. The current selection is extended from the current anchor point up to the location of the mouse pointer. The anchor point is NOT moved by this operation.

**To select a range of objects:**

1. Hold down the **SHIFT** key
2. Click on the item that you want to be the last one in the contiguous range of items. Any other disjoint selections are not affected unless the extent of the selection overlaps an existing disjoint selection.

3. Release the **SHIFT** key.

The **CTRL** and **SHIFT** key can be combined to create disjoint ranges of items. The figure to the left shows a disjoint selection containing one range of items.

**Note:** the network editor allows a specific kind of range selection, called region or *marquee* selection using a bounding box created with the mouse. This is explained in section "To select multiple nodes - method 2 ("marquee selection")." on page 68.

---

### Where are multiple selections possible and where not?

#### Single selection only

The Hermeneutic Unit editor's four **drop-down list** boxes underneath the main menu. Drop-down lists are single selection devices in general. Any pop-up menu based selection of choices is single selection only.

The pop-up menu disappears immediately after the item is selected. Examples: traversing hyper links with **SHOW LINKS**, selecting relation for linking codes or creating hypertext links.

The Object Explorer also only allows single selections. So does the margin area. All lists in the QueryTool.

All lists in "container" browsers: e.g., family browser, netview browser.

#### Multiple selection

Multiple selection is possible in the following places

- All extra lists and list views
- "Code by list" window
- All "Link to:" list windows (e.g., **CODES/CODING/LINK CODES TO** from the main menu)
- The Network Editor: disjoint and region node selections

#### ATLAS/ti and the Windows 95 Shell

The "shell" (e.g., the general user interface) introduced by Windows 95 and also used under Windows NT is taken advantage of by the ATLAS/ti installation procedure. The integration of ATLAS/ti into the Windows 95 (and Windows NT) shell takes place at several places:

- The program icon is placed in the start menu (optional).
- A folder is created with program icon, sample projects and help files.
- A new project can be started by simply choosing **NEW/HERMENEUTIC UNIT** from the screen backgrounds context menu.
• A Hermeneutic Unit can be opened by double clicking on its icon on the desktop, in the Explorer or other places where files are managed.

• A full Windows 95 compliant deinstallation procedure removes all but changed files and directories, entries in control files and system registry.

**Graphic File Formats**

**File Formats**
The following file formats are supported by ATLAS/ti and can be used as primary documents and background images:

Windows and OS/2 Bitmap (BMP, JPEG), PC Paintbrush (PCX), TIFF, Targa (TGA), Kodak Photo CD (PCD), MacPaint, FBM, Sun Raster, CMU WM Raster, Portable Bit Map (PBM, PGM, PPM), Faces Project, Utah RLE, X Window Dump, McIDAS areafile, G3 FAX, GEM Bit, X Pixmap, and X Bitmap.

**Recommendations for Graphical Documents**
While all the above graphical file types can be used, some recommendations are given:

BMP files which are most common under Windows are large but load very fast (fastest of all file formats supported).

JPG, JPEG files are compressed and occupy much less space than BMP files. However, the decompression time needed for the latter results in a noticeable delay between selecting a primary document and the display of the image in the primary document pane.

Size of a graphic file does not necessarily result in longer loading times. This is especially true for Kodak PhotoCD files which can be rather large (4 MB) but display in seconds. Highly compressed JPG files are very small but need their time until displayed.

JPG should be used for photographs and natural colors. However, the compression results in a loss of quality especially in areas of high contrast. Experiment your graphics tool's JPG compression factor for best results.

BMP files should be used for graphics with high contrasts.

**Frequently Asked Questions and More**

Find below a small selection of "frequently asked questions". A more complete set is evolving with more widely spread use of the ATLAS/ti. You will find a "FAQ" on our home page in the near future. Of course, your contribution is also much appreciated.

*When displaying primary texts, every paragraph seems to be one long line*

This is a broad hint, that the text used as a primary document has not been formatted according to the specifications required by ATLAS/ti.
One reason: when using "Save as" in your word processor, you must choose "text with line breaks" when creating the text file needed by ATLAS/ti. Otherwise, the original paragraphs are displayed one very long line each inside ATLAS/ti. The line breaks appearing your original word processor are ignored.

Please check section "Preparing Primary Documents" on page 30.

**Some characters of primary documents display as little squares**

Two encoding schemes are used on DOS/Windows based personal computers: ASCII (or DOS) and ANSI (or Windows). To correctly display both kinds of text (the former being created by a DOS based word processor or ol' ATLAS/ti DOS) when loaded as primary documents, you might have to change the encoding scheme of those primary documents that are displayed wrongly.

See "ANSI or ASCII? Changing the Encoding Type" on page 33 for details.

**The complete text looks rather weird**

This happens when the primary document is in its original word processor format (e.g. Word for Windows) and has not yet been prepared for use within ATLAS/ti at all. Special formatting codes spread throughout the document are then displayed as funny characters. See "Preparing Primary Documents" on page 30.

**ATLAS/ti cannot find the primary documents and displays an error message instead**

Primary documents are referenced via their path and file names by the Hermeneutic Unit they are associated with. Moving a HU to another computer and forgetting the primary documents is one reason. Another cause is renaming or restructuring the directory system on the computer where your HU resides.

Check the following sections for help: "Quotations as Layers" on page 10, and "Changing Path or Filename" on page 33.

**General Preferences**

All settings configurable with the Preferences dialog boxes are stored in file USER.INI, which resides in the system directory on single user systems and in a private user directory on network aware systems (all site licenses). The file USER.INI, which does also store a number of other system state variables is a plain text file which can be edited with a plain text editor (like the one that is offered in ATLAS/ti). However, such manual editing (which was often a must with ATLAS/ti DOS) is strongly discouraged and all modifications should be done exclusively through the preferences dialog boxes.

**Network Editor Preferences**

Unlike settings modified using the network editor's menu commands, several options can be made permanent using the network preferences dialog. Note, that most modifications are not immediately displayed in the network editors. Close and reopen a network editor to see the changes.
**Sorts and Filters**

Sorting and filtering is available for primary texts, quotations, codes, and memos and can be accessed via the HU editor's menu or the context menus of extra lists. Some of the options for sorting and filtering are common to all types of objects, while others are more specific. The following describes the options in more detail. Sort order: D = descending, A = ascending.

**The ATLAS Prolog Notation**

This section describes the storage representation format of ATLAS/ti Hermeneutic Units. It is included for those who are interested in technical details or who are planning to build knowledge bases using ATLAS/ti. It is not needed for everyday work.

**Smalltalk**

ATLAS/ti's implementation language is Smalltalk, the "mother" of all object-oriented languages and the source for all window and mouse based systems.

**The ATLAS/ti Mailing List**

**What the List is Good for**

During the beta-tests for the predecessor of ATLAS/ti, which began in early 1991, an electronic mailing-list was established for the users of the program. A mailing-list is a special e-mail address, that serves to distribute all messages sent to this address to all members assigned to the list. The main purpose of the list dedicated to ATLAS/ti is peer-to-peer support: to exchange experiences with the tool, discuss technical and methodological issues that arise with the use of the program.

**Resources - the ATLAS/ti Home Page**

The ATLAS/ti home page at http://www.atlasti.de should be a regular place to visit, even if you are already a member of the mailing list. Here you will already find "after sales" information like workshop announcements, special announcements (e.g, "The listserver is down for the next two days") or service packs and patches ready for download. In addition you will also find links to various resources concerning qualitative research.

If the address above is not valid anymore, because of changes the web server location, you will most likely get information about the new address. If not, please search for "ATLAS/ti" using one of the Internet search engines.
Index

A
Administrator 92
Anselm Strauss 1
ANSI 9
ASCII 9
Author 92
Authors 12
Auto Coding 56

B
Boolean operators 81
Bugs 96

C
Category search 52
Click retrieval 46
Code repositories 50
Codes
Comments 43
Definition 11
Deleting 45
Display 40
Free codes 44
List output 50
lists 50
Rename 44, 45
Renaming 44
Repositories of 50
Super Codes 86
Unlink from quotations 45
Coding 25, 39
Code-by-list 25, 42
Code-by-list-inverse 43
In-vivo 26, 41
Size restriction 42
Open coding 25, 41
Quick 43
Cognitive load 61
Collaboration 92
Comment
Hermeneutic Unit 30
Comments 12
Commutativity 83
Compatibility 33
Primary documents 33
Conceptual level 7, 61

D
Degree (nodes) 61
Directed links 62
Document See also Primary document
DOS text 9
Drag & Drop 19, 24, 26–27, 36, 38, 42, 71–72, 87, 95

E
Epistemological primitives 63
Error.log file 96
Exploration 2
Extension 82
Extra Lists 15

F
Families 12, 56
Add items 58
As filters 59
Create 57
Create reports 59
Family Browser 56
Remove 59
Remove items 58
FAQ 103
Filter 105
Focused network views 66
Folders 100
ATLASTi 100
HTML 100
Loclnst 100
Program 100
Textbank 100
Forest of codes 78
Frequencies table 50
Frozen Hypotheses 86

G
Graph theory 61
Graphic files
handling 37
recommendations 103
supported formats 103
GREP search 55
Grounded Theory 39, 63

H
Hermeneutic Unit
Create new 23, 29
definition 8
Open file 21
Hermeneutic Unit editor See also HU Editor
Hierarchical view 77
HU Editor
drop-down lists 15
launching extra lists 15
Line Numbers 15
Main Screen 13
Margin Area 15, 39
title bar 13
Hypertext 89
CHAIN structure 91
Source anchor 91
STAR structure 91
Target anchor 91

I
Indexing 11
Integration 2
Intension 82
Intensional Codes 87
In-Vivo coding 26, 41

K
Knowledge base 105

L
Layout 74
Semantic 74
Topological 75
Line Numbers 15
Links 61
Login 29

M
Mailing list 105
Manual 3
Margin Area 15, 39, 46, 47
Context menus 98
Font 98
List sub menu 98
Object Lists 20
Object menus 99
Object type 98
One per line 98
Use images 98
Memo families 60
Memos 95
Definition 11
vs. Codes 48
vs. Comments 48
Writing 48
Merge Codes 75
Merge HUs 92
Mind maps 61
Multi-authoring 92

N
Named links 62
Network
Merge Codes 75
Network Editor
Import neighbors 72
Layout 74
Links
Cut 70
Nodes
Appearance 74
Hide 72
Import 71
Open 67
Preferences 104
Network views 64
Creating 66
focused 66
Opening 67
Networks 12, 27, 61
Node types 62, 65
Nodes 61
Non-textual primary documents 37

O
Object Explorer 77
Object Lists
Codes 19
Margin area 20
Memos 19
Open 18
Primary documents 19
Quotations 19
Selecting items 18
Operands 80
Operators 80
Order (nodes) 61
Output targets 49

P
Parallel search 52
Paths 100
Patterns 55
Preferences 74, 104
Network Editor 104
Preparing Primary documents 30
Primary document 9
ANSI or ASCII 33
Assigning to Hermeneutic Unit 24, 32
attributes 60
Change encoding 33
Q

Qualitative Analysis 1
QueryTool 27, 79
  Boolean operators 81
    AND 81
    NOT 81
    OR 81
    XOR 81
  Boolean query 85
  Examples 85
  Operands 80
  Operators 80
  Print results 88
Proximity
  Co-occurrence Operator 84
  Distance Operators 83
  Embedding Operators 83
  Overlap Operators 83
  Proximity operators 83
  Proximity query 86
  Query 79
  Recall & Precision 81
Screen 79
  Semantic operators 82
    SIBlings 82
    SUB 82
    UP 82
  Semantic query 85
  Super Codes 86
    and code families 88
    Creation 87
    Cycle protection 88
    Merging 87
  Textbase selection 88
  Using Code Families 80
Quit program 29
Quotations 10

as layers 10
Audio 11
Create 24, 36
Delete 37
Graphical 11
Create 37
Selecting 38
Retrieval 46
Textual 10
Types of 10

R

Redundant Codings 94
Relations 62
  vs. links 63
Reports 49
  Code lists 50
  Code output dialog 50
  Code-PD Matrix 50
  Objects sorted by creation 50
  Output targets 49
Roles 63

S

Search swarm 53
Selection Techniques 101
Semantic Layout 74
Semantic Operators 82
Serendipity 2
Service Pack Manager 97
Service Packs 96
Smalltalk 105
Sorting 105
SPSS
  Primary document
    family 60
Start ATLAS/ti 21
Super Codes 11, 28, 86

T

Text Search 51
  Category search 52
  Defining categories 53
  GREP search 55
  Search swarm 53
  Search swarm delimiter 53
  Simple search 52
Text segments
  Free-hand selection 35
  Semi-automatic selection 24, 35
Textbank 100
Textual level 7
Theory Transfer 75
Thesaurus 61
Topological Layout 75
Transitive relations 82
Tree of codes 78

U

Unlink codes 45
User management 92
  administrator 92
  Classroom setup 92
  User database 93
  User editor 93

V

Venn diagrams 81
Vertices 61
VISE Principle 2
Visualization 2

W

Wallpaper 95
Windows
  Selection Techniques 101
Windows 95
  shell 102
Windows text 9
WinWord 95
  Preparing primary texts 31
Within operator 83
World Wide Web 90
A Word of Thanks

It is impossible to make a full list of people who in some way or other contributed to the current state of the software. I would like to thank all of them.

Thanks to all the "beta-testers" who worked intensively with the current system's prototype and gave valuable feedback. I am also indebted to the members of the ATLAS/ti e-mail discussion list, which started during project ATLAS and has proven to be a valuable resource both for the developer and the users of the program.

Thanks also to all customers who purchased the software without much written materials and waited patiently for a manual to be released.

Thanks to Carsten Schenke, who worked night and day during the last months under a considerable pressure to implement the setup program for ATLAS/ti.

A very special thanks goes to Hallock Hoffman of Qualitative Research Management, founded by Renata Tesch, the most engaged person in computer supported qualitative research until her death in 1994. As the representative of ATLAS/ti from 1994 until 1997 Hallock significantly contributed to the growing acceptance of this and other tools to support a methodology, which does not stop at "counting beans".

Last not least, I am especially indebted to the constant support by my wife Christine. Without her, this project could never have been completed and the simply would not be such a thing as ATLAS/ti for Windows.

Thomas Muhr, Berlin, November 1997