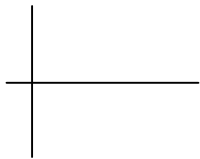




mindmap 2

The Visual Application
Assembly Environment

User's Guide



Copyright © 1990-1996, 1997 mindmap Software Corporation.

All Rights Reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of this agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of mindmap Software Corporation.

mindmap Software Corporation
2520 Mission College Boulevard
Santa Clara, CA 95054
USA

LIMITED WARRANTY


mindmap Software warrants the software media to be free of defects in workmanship for a period of 30 days from purchase. During this period mindmap Software will replace at no charge any such media returned to mindmap Software, postage prepaid. This service is mindmap's sole liability under this warranty.

DISCLAIMER

LICENSE FEES FOR THE SOFTWARE DO NOT INCLUDE ANY CONSIDERATION FOR ASSUMPTION OF RISK BY MINDMAP SOFTWARE OR ITS LICENSOR, AND MINDMAP SOFTWARE AND ITS LICENSOR DISCLAIM ANY AND ALL LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR OPERATION OR INABILITY TO USE THE SOFTWARE, OR ARISING FROM THE NEGLIGENCE OF MINDMAP SOFTWARE AND ITS LICENSOR, OR THEIR EMPLOYEES, OFFICERS, DIRECTORS, CONSULTANTS OR DEALERS, EVEN IF ANY OF THESE PARTIES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. FURTHERMORE, LICENSEE INDEMNIFIES AND AGREES TO HOLD MINDMAP SOFTWARE AND ITS LICENSOR HARMLESS FROM SUCH CLAIMS. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY THE LICENSEE. THE WARRANTIES EXPRESSED IN THIS LICENSE ARE THE ONLY WARRANTIES MADE BY MINDMAP SOFTWARE AND ITS LICENSOR, AND ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR A PARTICULAR PURPOSE. THIS WARRANTY GIVES YOU SPECIFIED LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF WARRANTIES, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

We welcome your comments and suggestions about this manual. Please send your comments via email to support@mindmap-software.com or write to us at the above address.

Part number 01-100004-0001



Acknowledgments

Thanks to all who contributed to the success of this document. Special thanks to Frank ‘The Raver’ Ihringer, Martin ‘Mr. Detail’ Opitz, Felix ‘Mr. Arctanh’ Opitz, comical Martin Schlichenmayer, the suave Alexander Friedrich and reliable Thomas Schumann.

If you don’t know where you are going, all paths will lead you there.

About the Book

The manuscript for this manual was prepared in Microsoft Word for Windows 7.0b simultaneously in three countries on two continents. Formatting was done in this same great program, although we did, at times, feel like we were pushing the limits.

Drawings were done in Adobe Photoshop using the Wacom Digitizer.

The body has been set in Century Schoolbook. The headings are set in Humanist BT. Notations are set in Comic Sans MS.

Trademarks

Windows, Windows 3.1, Windows 3.11, Windows for Workgroups, Windows 95, Windows NT, Microsoft Office, Visual Basic, Video for Windows are registered trademarks of Microsoft Corporation.

PowerBuilder is a registered trademark of Sybase, Inc.

OPEN/image is a registered trademark of Wang Laboratories, Inc.

Notes, LotusScript, and Lotus SmartSuite are registered trademarks of Lotus Development Corporation.

SQLWindows is a registered trademark of Centura Software Corporation.

DBase and Delphi are registered trademarks of Borland International.

Photoshop is a registered trademark of Adobe Systems Inc.

LEGO is a registered trademark of LEGO, Denmark.

Contents

CHAPTER 1 WHAT IS MINDMAP ?	9
Background	9
Hardware and Software Requirements	12
Document Conventions	14
Suggested Reading	14
CHAPTER 2 GENERAL OVERVIEW	17
Introduction	17
Contents of Diskettes/CD	18
Installation Steps	18
Contents of the Program Group	26
mindmap Settings	26
Deinstalling mindmap	27
What can go wrong	28
CHAPTER 3 MENU BAR	31
Use of the mindmap Menu Bar	31
File Menu	33
Edit Menu	60
Objects Menu	76
Format Menu	78
Properties Menu	89
Page Menu	100
View Menu	102
Run Menu	104
Windows Menu	108
Help Menu	108
CHAPTER 4 TOOLBOX	111
The use of the Toolbox	111
Pointer Option	112
Link Option	113
Run Option	114
Zoom Option	115

CHAPTER 5 COMPONENT TYPES	117
Draw Component	117
Button Component	125
Text Component	134
Graphic Import Component	137
Printer Component	144
Database Manager Component	146
List Box/Combo Box Component	175
Data Table Component	184
Input Field Component	193
Input / Output Component	203
Menu Component	210
Output Page Component	215
MCI Component	221
Visual Basic Component	226
 CHAPTER 6 LINKS	 233
General Overview	233
Events	244
Messages	254
 CHAPTER 7 USING THE CLIENT/SERVER COMPONENT	 319
General Overview	319
Special Conditions	322
Creating a Client/Server Application	330
Attributes	348
Special Events	361
Special Messages	362
Parser Extensions	362
Limitations	363
 CHAPTER 8 MAKING AN EXECUTABLE FILE	 365
General Concepts	365
Test it First	365
Building the EXE	367
 CHAPTER 9 PRINTER LAYOUT	 381
General Concept	381
Changing an Existing Template	383
Adding a New Template	385

	Creating a New Template File	387
CHAPTER 10 THE PARSER		389
	Overview	389
	Formulas and functions	398
	General Functions	401
	Component Specific Functions	453
	Registering External Functions	480
APPENDIX	483	
	Glossary	483
	Entries in MINDMAP.INI	499
	Table of Parser Functions	506
	ANSI Character Set	509
	Index	513

Chapter I

What is mindmap ?

Background

What mindmap can do

First and foremost, **mindmap** is an environment in which applications can be developed. Since computers ultimately execute machine instructions, a development environment must eventually generate such instructions. Over the brief history of computers, major advances have been accomplished in isolating the developer from the low level of machine code. Each new level of isolation has been accompanied with more abstraction. At each new level the developer has to deal less and less with machine based concepts. In the very early days, it used to be AND and OR switches. Then came registers and machine instructions. These were followed by variables, conditionals, loops, and so forth.

mindmap is a member of the newest development level - graphical development. It is a language, albeit a graphical language.

The price we have paid in general for the increase in power has been a loss of control. The more complex the entities that are manipulated, the less influence one has regarding the details. All in all, more can be accomplished, but at the expense of being limited by the scope of the environment.

Let's briefly look at the power issue. What **mindmap** offers is twofold. Persons competent in development, in general, are now able to perform most tasks much easier, faster and will be less prone to errors. Persons not having a background in development, can now build their own applications, without having to resort to 'agents' to do the work for them.

mindmap is an environment in which applications are actually assembled. Existing building blocks (components) are used, ar-

ranged and connected, such that the result is an application. `mindmap` is not intended to be used as an environment in which building blocks are created.

Traditional development environments offer a language, a debugging facility and a variety of supplemental tools. The language is supplied in the form of a compiler or interpreter. The developer enters statements into an editor and has the statements translated - either up front (compiler) or at runtime (interpreter). The statements input reflect a certain, and hopefully intended, procedural flow of things. Therefore, the developer must conceive of the flow (semantics) and then mold it into the language (syntax). In order to easily locate and eliminate syntactic and semantic errors in the code, a debugger is most often supplied. This helps the developer step through the application and view isolated states. Finally, most environments offer various other tools that ease the general process of building applications (linkers, profilers, etc.).

Here, `mindmap` differs dramatically from the traditional approach to development, in that it shifts the emphasis of development. The syntax of `mindmap` is very limited. Statements are not actually keyed in. They are selected from various lists, which are displayed depending on the situation. It is not possible to create a syntactically invalid statement. `mindmap` is also non-procedural, so that the concept of mapping the flow to a series of statements is also completely different. The process is much more comparable to painting a picture. What You See is What You Get (WYSIWYG) best describes it.

The assembler of an application selects a building block, defines various attributes and then links it to other blocks. Think of it as a grown-up version of an erector set or Lego® blocks. First, one selects the type of block from an assortment of different blocks (wheels, plates, blocks, windows, etc.). Next, a decision is made regarding the attributes of the selected block type (color, size, number of panes, etc.). Finally, the block is combined with other blocks (knobs are plugged into holes, parts are screwed together, etc.).

Obviously, the types of applications that can be built with the environment is a function of the available components, their attributes and the ability to connect the components. This is fundamentally a question of expandability. `mindmap` ships with a large assortment of components, having many attributes, and numerous methods of being combined with one another. Never-

theless, the system offers diverse means of incorporating new components.

What we assume you know

The first major assumption is that you know how to handle MS Windows. We assume that you are familiar with MS-Windows concepts and associated conventions. This implies that you know how to deal with either a mouse and/or the equivalent keyboard entries.

If you will be using **mindmap** in conjunction with databases, we assume that you are knowledgeable of general database concepts. This means that you are comfortable with the concepts of fields, records, and tables. If the underlying database engine you are using supports SQL (Structured Query Language), and it becomes necessary to execute an action by employing SQL, we also assume that you are either familiar with the SQL language or have access to other sources of information about SQL.

If you will be using **mindmap** to construct multimedia applications and will be using MCI commands (Media Control Interface), we make the assumption that you are either already knowledgeable or have access to other sources of information on MCI.

It will be helpful if you have some experience in application development. This does not mean that we expect you to be able to write code, but having experience in programming usually means that you:

- ▶ know how to decompose complex tasks into smaller, more manageable pieces;
- ▶ are accustomed to first analyze and then design, before you jump to the actual realization;
- ▶ have been exposed to the concepts of variables, loops, sub-routines, etc.

Please do not misunderstand this point. It is absolutely not necessary to be a programmer in order to be able to build applications with **mindmap**. It simply makes it easier, in some cases, in that general programming concepts can be utilized. We must

point out, though, that sometimes programming experience is a hindrance, in that it becomes difficult to think in non-procedural terms.

Throughout this manual and your use of `mindmap`, we will discuss various options or alternatives in the use and configuration of the components. You are highly encouraged to try the alternatives discussed. Only by experimenting, will you truly come to realize the depth of possibilities offered in `mindmap`. We look forward to your comments, suggestions and general feedback.

Hardware and Software Requirements

`mindmap` appreciates powerful hardware, but it does not give up on systems that are not state-of-the-art. The faster, bigger and better your system is, the faster your `mindmap` applications will run. Being a little more specific...

The more RAM you have, the better it is. `mindmap` will function on a 4MB system, although you might not enjoy the performance. We suggest a minimum of 8MB - obviously the more, the better.

Additional RAM is more important to `mindmap` than a faster microprocessor. Obviously, the faster the processor, the faster `mindmap` will execute. `mindmap` will function on a 386SX, although we would suggest a 486 as a minimum processor.

Since `mindmap` itself, and most applications built with it, are very user interface oriented, and thus utilize graphics extensively, we suggest a fast graphics card. `mindmap` will function in 4-bit, 640x480 pixels graphic environments, but we again suggest at least 256 colors and a fast graphics card.

Since MS Windows is a virtual memory operating system, swapping from memory to the hard disk and back is quite common. `mindmap` and applications built with `mindmap` are generally compact in size, but, nonetheless, must often be swapped to a hard disk. Once `mindmap` has been installed, it does not make special demands concerning the hard disk. The faster the drive, the more free space it has, and the less fragmented it is,

the better. **mindmap** consumes approximately 5MB of hard disk space after installation.

Since **mindmap** is an MS-Windows application it does require one of the currently available MS-Windows environments. **mindmap** currently supports the 16-bit interface in MS-Windows, so it will run on the following versions of MS-Windows:

- ▶ MS-Windows 3.11
- ▶ MS-Windows for Workgroups
- ▶ MS Windows 95

mindmap will run as a 16-bit application on Windows 95. **mindmap** will also support the common dialog boxes and long file names. **mindmap** currently does not support MS-Windows for Pen.

If you plan on using the supplied database interface, you will require Microsoft's ODBC (Open Database Connection). **mindmap** installs a version on your system, but please take caution here; there might be a conflict with other ODBC applications on your system. **mindmap** supports the 16- and 32-bit implementations of ODBC. In order to have access to some of the more sophisticated features (scrollable cursors, etc.), you must use ODBC 2.x.

If you use videos (AVI files) or other multimedia features in **mindmap**, it will be necessary to have Video for Windows and/or any other required drivers installed (Sound card etc.). Video for Windows is included in many packages. Please refer to the vendor of your hardware for additional information.

If you have installed other software products, you might have opted to also install some supplied graphic filters. In this case, they will be recognized by **mindmap** and offered, along with the standard **mindmap** filters, when you attempt to import graphic files. These filters are denoted with an asterisk in front of the file type in the corresponding dialog box, while the **mindmap** filters appear without the asterisk. This applies to 16-bit versions of import filters which support the Lotus/Intel/ Microsoft standard.

In any event, it is important that you consider not only your system and its configuration in the use of **mindmap**, but more

importantly, the system configuration of the systems on which your completed application will be deployed.

Document Conventions



Through out the documentation, we have made it a point to supply you with as many Tips, Tricks, Hints and Warnings as possible. The intention is to offer as much help as possible where we expect you to need it. These inserts are easily recognized, as they are set off from the rest of the text as shown with this example.

Throughout this document and in the online Help, some conventions have been made to assist in their ease of use.

► Menu Options:

All selections from the menu are in a different typeface (Humanist Bold). In order to more easily distinguish the various levels, each entry is separated with a vertical line. An example is:

File | Open

► Keyboard Entries:

Whenever this documentation refers to a keyboard entry, the string is set in a different typeface (Courier). An example might be

F1

When you encounter a string set in this manner, you are expected to key in the appropriate characters.

► Buttons:

Generally, buttons are not displayed in their graphic representation. We normally display just the button label, but we use a different font. An example would be

OK

When you read a string in this setting, please press the associated button in mindmap.

Suggested Reading

If you will be dealing with databases in your applications, we strongly recommend that you make yourself familiar with SQL. Although mindmap attempts to isolate you from SQL as much as possible, it might become necessary to generate more complex

statements using SQL. A number of books and general introductory literature is available. Some of the titles we suggest include:

Author	Title
Bowman, et al	Practical SQL Handbook, Addison-Wesley, 1996
Gruber	Understanding SQL, Sybex 1990
Taylor	SQL for Dummies, IDG 1995

If you plan on creating applications that include multimedia features (sound, video, animation, etc.), you will be using Microsoft's MCI (Media Control Interface) language. The scope of this language is basically determined by Microsoft, but it has provisions for vendor specific extensions. Beside referring to the appropriate Microsoft literature, it might also be necessary to read the information provided by the vendor of your additional hardware/software.

Technically, **mindmap** is a visual programming language. It includes a syntax — which is graphical and does not permit you to create syntactically incorrect statements. It has a domain — which is limited to building a rather broad scope of applications under MS Windows. Since it does not do away with underlying concepts commonly found in application building (loops, variables, subroutines, etc.), we recommend that you eventually devote some time to these concepts. If you have a programming background, these concepts are more than familiar to you. If you are a novice, you might run into some roadblocks, while building applications. They most likely will be of a more conceptual nature. It is therefore recommended that you spend time at least browsing through some fundamental literature dealing with these basic concepts.

mindmap directs a major focus towards the user interface. Building appealing and functional user interfaces is part science and part art.

The scientific part is dealt with in the following representative publications:

Author	Title
Aaron Marcus, Nick Smilonich, Lynne Thompson	The Cross-GUI Handbook; Addison-Wesley, 1995
Susan L. Fowler, Victor R. Stanwick	The GUI Style Guide; Academic Press, 1995

The artistic side can be found in books such as:

Author	Title
Ray Kristof, Amy Satran	Interactivity by Design; Adobe Press, 1995