

MindMap 2

The Component Assembly Environment

Tutorial

Copyright

Copyright © 1990-1996 MindMap Software.

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.

MindMap Software
2540 Mission College Blvd.
Santa Clara, CA 95054
U.S.A.

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 cost 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.

Acknowledgements

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, and reliable Thomas Schumann.

If you don't know where to go, anyway is the right direction.

About the Book

The manuscript for this manual was prepared in Microsoft Word for Windows 7.0b on two continents in three countries simultaneously. 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 10 pts. The headings are set in Humanist XBd. The captions and notes are set in T-Makers Handwritten fonts.

Trademarks

Windows, Windows 3.1, Windows 3.11, Windows for Workgroups, Windows95, Windows NT, Microsoft Office, Visual Basic, Video for Windows are registered trademarks of **Microsoft Corp.**

PowerBuilder is a registered trademark of **Sybase.**

OPEN/image is a registered trademark of **Wang.**

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

SQLWindows is a registered trademark of **Centura.**

DBase, and Delphi is a registered trademark of **Borland.**

Photoshop is a registered trademark of **Adobe Systems.**

LEGO is a registered trademark of **LEGO**, Denmark.

Inhaltsverzeichnis

EINLEITUNG	7
ERSTE SCHRITTE	11
NACHRICHTEN UND LINKS	27
EIN PICTURE-VIEWER	39
EIN "TASCHENRECHNER"	55
EIN VIDEO-RECORDER	67
DATENBANK	91
BAU EINER ERWEITERTEN DATENBANKANWENDUNG	109
KLEINES DATENBANKBREVIER	145
ANHANG: BEISPIELANWENDUNGEN	159

Einleitung

Sicher werden Sie bereits einige Schritte in MINDMAP gegangen sein. Oder vielleicht gehören Sie ja zu den Menschen, die erst die Dokumentation lesen, dann das Tutorial durcharbeiten, um sich schließlich mit ernsthafter Arbeit über die Software herzumachen? Wir lieben beide Anwendertypen. Auch wenn die letzteren sehr selten sind... MINDMAP läßt sich wunderbar spielerisch erobern, aber wie bei jeder Software schleichen sich so auch gerne "schlechte Angewohnheiten" ein.

Wie auch immer Sie anfangen möchten, dieses Tutorial soll Ihnen Hinweise geben, mit welchen Methoden Sie in MINDMAP schnell zum Ziel kommen.

Zum einen sollen die Übungen praktisch verwertbar sein, zum anderen sollen auch die wichtigsten Funktionen von MINDMAP angesprochen werden. Daher befinden sich in diesem Tutorial verschiedene Beispiele, die Sie nachbauen können. Sie werden feststellen, daß sich der Aufbau der Beispiele systematisch steigert. So lernen Sie schrittweise neue Komponenten kennen, und bekommen ein Gefühl dafür, wie man mit MINDMAP Anwendungen baut.

Haben Sie bereits mit Programmiersprachen programmiert? Nein? Wir glauben, daß das nicht als Nachteil für den Einstieg in MINDMAP zu werten ist. Tatsächlich wollen wir denen, die über MINDMAP in die Welt der Programmierung finden Mut machen. Wir freuen uns, daß Sie sich unbefangen und frisch ans Werk machen. Wenn Sie mit MINDMAP arbeiten, wenden Sie heute schon eine Philosophie an, die sicher morgen ganz normal sein wird. Wenn Sie bereits programmiert haben, dann werden Ihnen die grundsätzlichen Dinge wie Schleifen oder Variablen bereits geläufig sein. Aber Vorsicht! So manchem fällt es schwer, sich von den prozeduralen Angewohnheiten zu trennen. Sie werden nach verschiedenen Dingen in MINDMAP vergeblich suchen, weil sie einfach nicht mehr benötigt sind.

Stellen Sie sich das Entwickeln eines Programms vor, wie die Herausforderung einen Fluß zu überqueren. Am anderen Ufer liegt Ihr Ziel. In dem Fluß schwimmen Baumstämme. Be trachten wir nun die möglichen Methoden, diesen Fluß zu überqueren, bzw an das andere Ufer zu kommen. In der Zeit, als man

Computer noch mit "K" schrieb, wanderte man hinauf, bis an die Quelle des Flusses und an der anderen Seite wieder hinab. Das kann ein langer Weg sein. In dieser Zeit waren Kartei - kisten und mechanische Schreibmaschinen sehr populär. Einigen erlauchten Personen wurde dann das Privileg zuteil, schwimmen, d.h. programmieren zu lernen. Man begab sich bis zum Hals in die Fluten und kämpfte gegen die anströmenden Baumstämme an. Auf der anderen Seite angekommen, achtete man sorgsam darauf, dass nicht zu viel über die Schwimmtechnik bekannt wurde, da dies ja sehr bald die Kompetenz in Frage stellen könnte. Vielleicht nennen wir diese Zeit die "Assemblerzeit".

Dicht auf die Assemblerzeit folgte die PL/1- oder COBOL-Zeit, die auch einige Spezies, heute als Mainframer bekannt, hervorbrachte. Bald gab es auch den bemerkenswerten Versuch, eine Sprache für Anfänger zu erfinden "BASIC". Diese Sprache arbeitete sich mit einem Vehikel Namens Commodore 64 sogar bis in die Wohnzimmer der Menschen vor.

In der C-Zeit, der Beginn einer sehr wichtigen Epoche, lernte man schon, sich an die vorbeischwimmenden Baumstämme zu klammern, um mit deren Hilfe den Fluß zu überqueren. Manche hangelten sich dabei schon von einem Stamm zum nächst günstigeren. Aber alle wurden naß. Um dies zu verhindern leitete man die "++"-Zeit ein. "Objektorientierte" und "objekt - angelehnte" Sprachen entstanden, und mit erheblich weniger Mühe und Risiko überquerte man den Fluß rittlings auf einem Baumstamm. Das können nun aber auch schon eine Menge Menschen, und demzufolge wundern sich auch viele über die fixe Person, die plötzlich von hinten links auftaucht, mit Schwung über die Stämme tippelt und schließlich trockenen Fußes und sehr schnell und elegant auf der anderen Seite ankommt.

Haben Sie sich in dieser Szene erkannt? Fein! Denn dann haben Sie bereits das erste Kapitel dieses Tutorials mit Erfolg gemeistert.

Wir fragen uns tatsächlich, warum jemand, der genug Mühe hat, sich ein Problem des ganz normalen Geschäftsalltags zu vergegenwärtigen, sich auch noch mit maschinen- oder betriebssystembedingten Schwierigkeiten herumplagen soll. Möchten Sie wirklich mehr über Akkumulatoren oder Speicheradressierung wissen? Vielleicht ist es dann besser, Sie legen dieses Buch erst einmal auf die Seite und besorgen sich

Lektüre wie "Mein BIOS und ich" oder "Mit ROM und RAM auf Du und Du". Wie die Wirklichkeit zeigt, gibt es heutzutage jedenfalls tatsächlich Menschen, die einen PKW lenken ohne KFZ-Mechaniker zu sein.

Vorläufig glauben wir, daß Sie MINDMAP erworben haben, um schnell und effizient Herausforderungen der täglichen Arbeit zu meistern. Und deshalb werden wir uns im zweiten Kapitel des Tutorials auf "echte" Probleme stürzen.

Die in diesem Tutorial erzeugten Anwendungen, finden Sie auch auf der beiliegenden CD.

Erste Schritte

Dieses Kapitel können Sie überspringen, wenn Sie bereits Bekanntschaft mit der Oberfläche von MINDMAP gemacht haben. Natürlich freuen wir uns in diesem Falle trotzdem, Sie auch hier wieder bei uns zu haben.

Wir verlieren keine Worte darüber, unter welcher Version MS-Windows Sie MINDMAP aufgerufen haben.



Der Trick mit dem Doppelklick auf das entsprechende Icon funktioniert überall.

Nach dem Aufruf finden Sie MINDMAP wahrscheinlich wie folgt vor:



Auf der rechten Seite befindet sich ein schmales Fenster mit Schaltflächen zur Benutzung verschiedener Werkzeuge. Die Funktionen dieser Werkzeuge lassen sich alle auch über die Auswahl der entsprechenden Menüpunkte erreichen. Insofern ist diese Box als Abkürzung zu verstehen. Die am häufigsten benutzten Funktionen finden sich hier wieder.

Von besonderer Bedeutung ist auch die unauffällige Zeile am Fuße des Hauptfensters. Hier findet man rechts den Namen

der jeweils ausgewählten Komponente. Unter Komponente verstehen wir jedwedes Element ihrer Bildschirmmaske. Das kann eine Schaltfläche, ein Rechteck oder ein Container zur Aufnahme eines Bildes sein. Hier kann man auch die durch das Programm standardmäßig zugeordneten Namen verändern. Hierbei gestatten Sie einen Hinweis, der sich in Zukunft als sehr förderlich zeigen wird. Benennen Sie Ihre Komponenten! Die durch MINDMAP automatisch generierten Namen sind sinnvoll und notwendig. Sie bestehen aus einem Kürzel, das den Komponententyp anzeigt und einer fortlaufenden Nummer. Sie sagen aber nichts darüber aus, welche Funktion inner halb Ihrer Anwendung die jeweilige Komponente hat. Gewöhnen Sie sich eine Praxis an, die Namen so zu vergeben, daß Sie schon am Namen der Komponente erkennen können, um was es sich handelt, auf welcher MINDMAP-Seite sich die Komponente befindet, etc. Diese Komponenten werden nämlich über ihren Namen angesprochen. Wenn man sich bei der Entwicklung an Konventionen hält, lassen sich die Applikationen später leichter verstehen. Für Sie und auch für andere.

Nach dem Start von MINDMAP wird der größte Teil des Fensters von einer leeren Fläche eingenommen. Diese Fläche ist wie ein Blatt zu verstehen, auf dem Sie vorgenannte Komponenten unterbringen und mit Funktionalität versehen. Nur wenige Ihrer zukünftigen Entwicklungen werden sich auf eine einzige Seite beschränken. In der folgenden Übung lernen Sie den Umgang mit verschiedenen Seiten kennen und so ganz nebenbei generieren Sie auch bereits ein paar Komponenten.

Übung 1:

Zeichnen eines Rechtecks

Wir gehen davon aus, daß Sie MINDMAP neu gestartet haben und eine leere Seite vor sich haben.

Klicken Sie mit der Maus auf das Symbol in der Toolbox zum Erzeugen eines grafischen Primitivs.

Ein kleines Fenster öffnet sich, welches die verschiedenen Möglichkeiten aufzeigt.



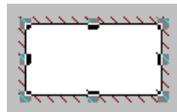
Begnügen Sie sich bitte vorerst mit einem ganz normalen Rechteck. Hierbei lassen wir Ihnen lediglich die Freiheit, ob es sich um ein "kantiges" Rechteck oder eines mit abgerundeten Ecken handeln soll. Haben Sie bemerkt, daß sich die Form des Cursors verändert hat?

Setzen Sie nun die Maus auf die freie Fläche und halten Sie den Mauszeiger gedrückt. An der Position, an der Sie mit dem Mauszeiger per Druck auf die linke Maustaste beginnen, legen Sie den Ort für die linke obere Ecke der Komponente fest. Ziehen Sie nun, ohne die Maustaste loszulassen, einige Zentimeter nach rechts unten auf Ihrem Bildschirm. Es entsteht ein gestricheltes Rechteck, welches Ihnen die Dimensionen der zukünftigen Komponente andeutet. Sie können nun versuchen die Maus wild über den Bildschirm bewegen. Halten Sie aber dabei die linke Maustaste weiterhin gedrückt. Wie Sie bemerken können ist es auch möglich die Maus auf eine Stelle links über der Startposition zu bewegen. Vielleicht ist Ihnen diese Methode angenehmer? MINDMAP läßt Ihnen freie Wahl. Egal, wo Sie zu zeichnen beginnen, der Endpunkt wird immer der diagonal gegenüberliegende sein.

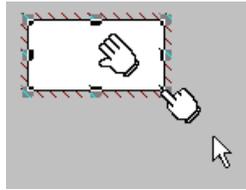
Sobald Sie die Maustaste loslassen, ist die gewünschte Komponente plaziert.



Bitte beachten Sie den Mauscursor! Er deutet durch seine Form nach wie vor darauf hin, was bei der nächsten Mausaktion geschieht. Das Fadenkreuz deutet darauf hin, daß Sie sich im Zeichenmodus befinden. Da neben befindet sich das Symbol der Komponente, die selektiert ist. Da wir davon ausgegangen sind, daß Sie mehrere Rechtecke in Folge produzieren, hat der Cursor immer noch die Form zum Zeichnen eines Rechteckes. Für unser Beispiel haben wir mit der soeben erzeugten Komponente jedoch genug. Wir möchten diese Komponente nun bearbeiten und schalten daher wieder auf den normalen Zeigecursor zurück. Klicken Sie mit der linken Maustaste auf das oberste Symbol der Toolbox, dem Zeiger. Nun befinden Sie sich wieder im sog. Edit Mode.



Klicken Sie als nächstes in das soeben gezeichnete Rechteck. Wie Sie feststellen können, bildet sich nun ein Rahmen um das Rechteck. Er deutet an, daß diese Komponente nun ausgewählt ist und zur Bearbeitung zur Verfügung steht.



Sicher haben Sie auch bemerkt, daß sich in dem gestrichelten Rahmen kleine schwarze Rechtecke befinden. Diese Rechtecke dienen zum "Anfassen" der Komponente. Fahren Sie bitte langsam und ohne eine Taste zu drücken mit der Maus in das Rechteck und wieder

hinaus. Bewegen Sie sich mit der Maus an einen der Anfasspunkte. Sie stellen fest, daß es hier drei Formen des Mauszeigers gibt. Innerhalb der Komponente erscheint der Mauszeiger als offene Hand. Außerhalb der Komponente erscheint er wie gewohnt als Pfeil und bei Berührung eines der Anfasspunkte erscheint die Maus als Hand mit ausgestrecktem Zeigefinger.

Wenn der Mauszeiger sich innerhalb der Komponente zu einer Hand verwandelt hat, halten Sie doch einmal die linke Maustaste gedrückt und bewegen Sie die Maus auf Ihrer Arbeitsfläche. Die Komponente verändert entsprechend seine Position. So können Sie also die Lage einer Komponente verändern, ohne seine Form zu beeinflussen.

Nähern Sie sich nun mit dem Mauszeiger einem der Anfasspunkte. Wenn der Mauszeiger sich in eine Hand mit Zeigefinger verwandelt, dann halten Sie in diesem Augenblick die linke Maustaste gedrückt und ziehen so an der Komponente. Wie Sie sehen, verändert sich nun nicht die Lage, sondern die Form der Komponente. Und zwar abhängig davon, welchen der Anfasspunkte Sie gewählt haben.

Bei Anwahl eines Anfasspunktes auf einer Geraden des Rechteckes können Sie die Form des Rechteckes beeinflussen indem Sie lediglich diese Gerade verschieben. Bei Anwahl eines der Eckpunkte beeinflussen Sie gleich die beiden Geraden, die diesen Eckpunkt bilden. Beide Möglichkeiten sind gleichermaßen wichtig. Um die Größe einer Komponente zu verändern, aber das bestehende Seitenverhältnis beizubehalten, drücken Sie bitte die `Shift`-Taste, während Sie die Größe der Komponente an einem der Eckpunkte verändern.

Bringen Sie das Rechteck in die Form Ihrer Wahl. Aber lassen Sie bitte noch etwas Platz auf der Arbeitsfläche.

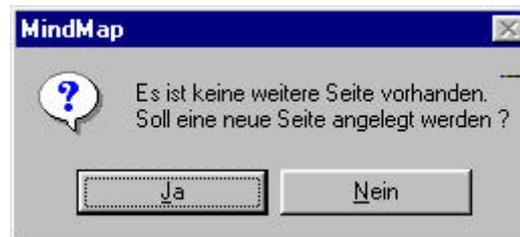
Jetzt dürfen Sie sich eine Tasse Kaffee einschenken!

Erzeugen einer weiteren Seite

Wir haben ja versprochen, daß wir auf die Verwendung mehrere Seiten eingehen, und genau das geschieht jetzt. Obwohl wir ja bisher nur eine Seite vor uns haben, betätigen wir die Schaltfläche, die uns auf die "nächste" Seite bringen soll. Sie finden diese am Fuß des Fensters in der Statuszeile.



Ein Dialog weist uns darauf hin, daß keine weitere Seite existiert und bietet an, eine solche zu erzeugen.



Wir nennen unser soeben gezeigtes Verhalten "Desperado - verhalten". Und zwar deshalb, weil wir eine Ermahnung in Kauf genommen haben, um bequem an unser Ziel zu kommen. Natürlich hätten wir auch eine neue Seite erzeugen können, indem wir im Menü auf **Seite** und dann auf den Menüeintrag **Seite einfügen** gegangen wären. Selbstverständlich dürfen Sie Ihre Methode wählen. Der erste Weg erzeugt Seiten mit aufsteigender Nummer "rechts" neben der ersten Seite, der zweite Weg erzeugt Seiten "links" neben der angezeigten Seite. Da es jedoch keinerlei Hierarchie unter den Seiten gibt ist es vollkommen egal welche Seite die erste und welche die "letzte" Seite ist. Denn auch bei den Seiten gilt es als ratsam, diese mit "sprechenden" Namen zu versehen. Damit beginnen wir sofort!

Bewegen Sie Ihren Mauszeiger in das Feld, welches den Namen der gegenwärtigen Komponente, also der neuen Seite zeigt.



Ersetzen Sie den Namen `pg2` durch den Namen `Target`, indem Sie diesen neuen Namen einfach über den alten Namen schreiben. Kehren Sie nun wieder auf die zuerst erzeugte Seite zurück. Verwenden Sie dazu die Schaltfläche mit dem Pfeil nach links.



Nennen Sie diese Seite, auf der Sie das von Ihnen plazierte Rechteck sehen müssten, `Source`.

Kehren Sie nun auf die Seite `Target` zurück und erzeugen Sie hier eine Komponente in Form eines Kreises.

Gelungen? Wunderbar! Wir haben nun in unserer Anwendung zwei Seiten, mit jeweils unterschiedlichen Komponenten. Wir werden diese Anwendung nun starten um zu sehen was passiert! Aber vorher -ganz wichtig- sichern wir unsere Kreation auf der Festplatte.

Speichern der Anwendung

Wählen Sie im Menü **Datei | Speichern als...** und vergeben Sie einen Namen, an den Sie sich später auch noch erinnern. Wir empfehlen **TUT_01** oder auch **TUT_01.MM**. Das **TUT** soll für Tutorial stehen und das **01** signalisiert unsere erste Kreation. **MINDMAP** benötigt keine speziellen Endungen bei der Vergabe von Dateinamen. Es ist jedoch empfehlenswert, die Endung **.MM** zu verwenden. Das hat auch mit den bereits erwähnten Namenskonventionen zu tun. Bei einer Datei mit der Endung **.MM** wissen wir, daß es sich um eine **MINDMAP** Datei handelt.

Starten einer Anwendung

Sie sehen in der Toolbox als drittes Element von oben gesehen eine Person, die sich beschleunigt bewegt. **Run!** Mit dieser

Schaltfläche bringen Sie eine Anwendung zur Ausführung. Es gibt zwei weitere Methoden Anwendungen zur Ausführung zu bringen. Über das Menü geschieht dies mit **Run | Run Mode** oder auch über die dort angedeutete Taste **F4**, welche besonders praktisch ist, weil sie uns aus dem Run Mode auch sicher wieder in den Edit Mode führt, aus dem wir den Run Mode aufgerufen haben.

Sie kennen nun drei Methoden, eine Anwendung in MINDMAP zu starten. Verwenden Sie eine der drei.

Hoppla! Das ging schnell. Sie sehen nun die erste Seite Ihrer Anwendung vor sich. Das Tempo in dem die Anwendung erscheint ist wahrscheinlich nur für die Personen aufregend, die an lange Kompilier- und Linkzeiten bei herkömmlichen Sprachen gewöhnt sind. Jedenfalls erscheint Ihre Seite **Source** genau so, wie Sie sie entworfen haben auf dem Bildschirm. Ihr Programm!

Aber wo ist die zweite Seite? Wo ist die Seite **Target**? Nun, die ist vorhanden, aber wir haben vergessen, etwas einzurichten, mit dem man sie aufruft! Kehren wir also zurück in den Edit Mode, um etwas Funktionalität in unsere Anwendung zu bringen. Hier gibt es zwei Methoden! Die eine ist die bereits erwähnte **F4**-Taste, die andere wäre der Doppelklick mit der Maus am äußersten rechten Rand des Bildschirms, vor ausgesetzt, MINDMAP ist bildschirmfüllend aufgerufen worden.

Erzeugen einer Schaltfläche

Wieder zurück im Edit Mode von MINDMAP installieren wir auf der Seite mit dem Namen **Source** eine Schaltfläche. Eigentlich besteht zunächst kein Unterschied zum Zeichnen eines Rechteckes. Klicken Sie in der Toolbox auf das Symbol für die Schaltfläche. Auch hier geht ein kleines Fenster mit verschiedenen Optionen auf. Wählen Sie die Option **Befehlsschaltfläche**.



Zeichnen Sie, wie bereits gewohnt, die Komponente auf den Bildschirm. Nachdem Sie die Maustaste losgelassen haben, erscheint eine typische Schaltfläche auf dem Bildschirm. Auch hier bleibt der Cursor zunächst in der Erwartung weiterer zu zeichnender Schaltflächen. Benennen Sie die Schaltfläche. Wir schlagen vor `btn_target01` oder `btn_source_target`. Sie dürfen auch den Namen `diesistdietaustediegedruecktwerdenuaufseitezweizugelangen` verwenden. Obwohl es möglich ist, bei der Namensvergabe Leer schritte oder Umlaute zu verwenden, können wir dies nicht empfehlen. Der Grund ist, daß solche Zeichen Konflikte bei der Verwendung von Formeln erzeugen können. Das probate Mittel, Elemente innerhalb der Namen zu trennen, ist der Unterstrich oder kurz `"_"`.

Diese Schaltfläche ist bereit alles für Sie zu tun! Alles was in ihrer Macht steht, versteht sich. Diese Aussage ist wichtig. MINDMAP erkennt nämlich an der erzeugten Komponente, zu welchen Funktionen es in der Lage ist und bietet unsinnige Funktionen garnicht erst an. Wir wollen ja garnicht „alles“, sondern etwas ganz bestimmtes. Nämlich auf unsere zweite Seite verzweigen.

Erzeugen eines Links - Wir "programmieren"

Bitte sorgen Sie dafür, daß wieder der normale Zeigecursor erscheint, d. h. bitte wechseln Sie in den Edit Mode, sofern Sie sich noch im Zeichen-Modus befinden. Selektieren Sie dann die soeben erstellte Schaltfläche durch Anklicken. Der Ihnen bereits bekannte Rand erscheint. Drücken Sie nun, während sich der Mauszeiger außerhalb der Komponente befindet, die **rechte** Maustaste.

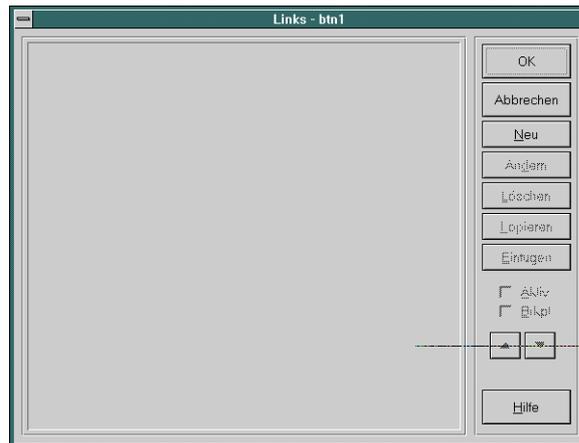
Wie durch ein Wunder erscheint ein neues Fenster, das Attribute-Fenster. Dieses Fenster bezieht sich immer auf die momentan selektierte Komponente, bzw. auf eine Gruppe momentan selektierter Komponenten. Die Elemente dieses Fensters können je nach Typ der Komponente verschieden sein. Einige der Elemente erscheinen jedoch immer wieder. Eines davon ist das sog. Link-Symbol, welches sich so re-präsentiert.



Auf alle übrigen Elemente werden wir noch detailliert eingehen.

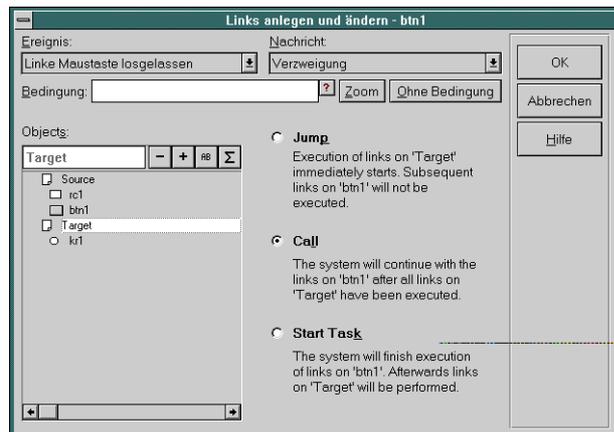
Was ist aber denn genau ein Link? Bei der Arbeit mit MINDMAP erzeugen wir Komponenten, die verschiedene Eigenschaften aufweisen und Container, die diese Komponenten beherbergen. Komponenten können grob gesagt zwei Dinge tun. Sie können einerseits eine Nachricht aussenden oder eine von einer anderen Komponente ausgesandte Nachricht erhalten und gegebenenfalls darauf reagieren. Eine Schaltfläche freilich erhält meist eine Nachricht vom Anwender. Sie wird gedrückt, losgelassen, doppelgeklickt (Sprache lebt...), um nur einen kleinen Ausschnitt zu nennen. Diese Seite der Komponente nimmt also Ereignisse wahr. Die andere Seite der Komponente versendet selbst Nachricht. Das könnte zum Beispiel eine Verzweigung sein oder eine Wertzuweisung an eine andere Komponente.

Nach Anwahl des Link-Symbols erscheint folgender Dialog:



In diesem Dialog werden die Links gezeigt, die an diese Komponente gebunden wurden. Sind noch keine Links definiert, bleibt eigentlich nur die Schaltfläche *Neu* oder *Abbruch*.

Nach Druck auf die Schaltfläche *Neu* finden wir uns in einem der wichtigsten Dialoge von MINDMAP wieder. Wir widmen uns deshalb diesem Dialog etwas genauer.



Betrachten wir die angebotenen Optionen, sehen wir oben auf der linken Seite, eine Liste der möglichen Ereignisse. Wie bereits erwähnt stehen hier nur die für die angewählte Komponente gültigen Ereignisse. Ausgewählt ist, weil bei der Schaltfläche am häufigsten verlangt, **Linke Maustaste losgelassen**. Schauen wir doch einmal nach, welche Ereignisse unsere Schaltfläche außerdem registrieren kann.

Klappen Sie bitte hierzu die Liste auf. Sie sehen, unsere Schaltfläche ist also in der Lage, auf all diese Ereignisse zu reagieren. Wie man erkennen kann, handelt es sich beileibe nicht nur um die im engeren Sinne typischen Merkmale von Schaltflächen. Verstehen Sie dies bitte als Hinweis auf die Flexibilität von MINDMAP.

Unterhalb der geschlossenen Liste erscheint ein Feld, in das noch zusätzliche Bedingungen eingetragen werden können. Hiermit kann die Reaktion auf ein Ereignis beispielsweise von dem Zustand einer anderen Komponente abhängig gemacht werden. Im Klartext: "Reagiere auf Tastendruck nur dann, wenn im Feld Name etwas steht".

Auf der rechten Seite erkennen Sie eine weitere Liste, die uns zeigt, welche Nachrichten eine Komponente auslösen kann. Auch hier stehen umfangreiche Optionen zur Verfügung, auf die wir jetzt nicht im Einzelnen eingehen wollen.

Betrachten wir zunächst die untere Hälfte des Dialogs. Hier finden wir eine Liste der für den Empfang der ausgewählten Nachricht geeigneten Komponenten. Wie wir ja bereits wissen, entsteht durch die Aktivierung von Ereignis/Nachricht-Zuständen eine Verkettung, oder besser Vernetzung von Komponenten. MINDMAP gibt durch ein Symbol, welches dem Komponentenamen vorangestellt ist, einen Hinweis darauf, von welcher Art diese Komponente ist. Sicher pflichten Sie mir nun bei, daß die Vergabe von aussagekräftigen Namen überaus wichtig ist. Besonders wenn, wie in Ihren Applikationen zu erwarten, einige hundert oder tausend Komponenten in der Liste lauern.

Also noch einmal. Unsere Komponente hat zwei Steckdosen und eine Reihe möglicher Funktionen. Mit der linken Steckdose wird die Komponente mit einer ereignisauslösenden Komponente verbunden; die Ereignisse werden durch die Liste links oben bestimmt. Schließlich wird an die rechte Steckdose die Komponente angeschlossen, welche die Nachricht erhalten soll. Bei dieser Komponente findet natürlich genau dasselbe wieder statt. Sie hat sogar die Möglichkeit, die aufrufende Komponente erneut aufzurufen. Das erscheint spontan etwas merkwürdig, nicht wahr? Aber die Programmierer unter den Lesern werden jetzt sagen: "Ach so". Sie haben ein Schleifenkonstrukt entdeckt! Die Abbruchbedingung steht im Bedingungsfeld. Aber um solche Dinge werden wir uns später intensiver kümmern.

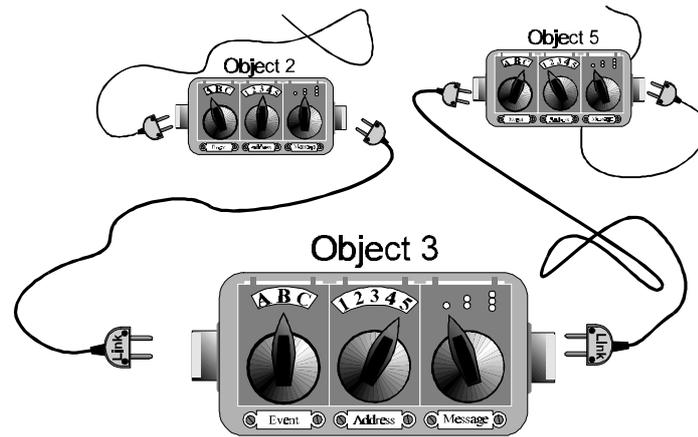
Vorläufig möchten wir ja nur die Seite wechseln. Deshalb be - lassen wir das Ereignis auf **Linke Maustaste losgelassen** oder stellen dieses Ereignis ein, sofern es verändert wurde. Dann wählen wir in der Liste der Nachrichten **Verzweigung** und wählen als Ziel für diese Verzweigung die Seite **Target**, indem wir in der Liste im unteren Dialogbereich auf den Namen dieser Seite klicken. Sie werden bemerken, daß sich die Einstellung auf der linken Seite des Dialogs von **Jump** auf **Call** ändert. Dies ist richtig so und sollte Sie an dieser Stelle nicht weiter beschäftigen.

Alle übrigen Optionen dieses Dialogs werden später erklärt. Bestätigen Sie nun Ihre Einstellungen indem Sie mit **OK** den Dialog schließen. Der Dialog mit dem gerade definierten Link erscheint wieder.



Bestätigen Sie auch diesen Dialog mittels der **OK**-Schaltfläche.

Die folgende Darstellung ist stark vereinfacht. Hier hat jede Komponente nur jeweils eine Schnittstelle. Im richtigen Leben wartet MINDMAP jedoch mit beliebig vielen Schnittstellen auf.



Titel für die Schaltfläche

Unsere Schaltfläche wirkt etwas nackt, nicht wahr? Und außer uns weiß ja nun keiner, welchem Zweck die Schaltfläche dient. Deshalb werden wir jetzt eine Aufschrift anbringen. Wenn die Schaltfläche selektiert ist, drücken Sie bitte auf der Arbeitsfläche die rechte Maustaste. Die daraufhin erscheinende Box **Attribute** kennen Sie bereits. Wählen Sie nun das Symbol für die objekt-spezifischen Attribute.



In dem erscheinenden Dialog kann in dem Feld **Inhalt** ein Titel für die Schaltfläche vergeben werden.

Hier können Sie nun eintragen *Ziel*. Bestätigen Sie den Dialog mit *OK* und sehen Sie nach, was daraus geworden ist. Wir wissen, daß das schwerfällt, anhand der vielen Optionen, die in diesem Dialog zum Testen einladen. Aber auf die anderen Punkte kommen wir später.

Jetzt ändern wir noch die Schriftart der Schaltflächen-Be - schriftung. Klicken Sie erneut auf die rechte Maustaste, während die Schaltfläche selektiert ist. Wählen Sie nun das Symbol für die Schrift.



Der folgende Dialog wird Ihnen aus anderen Anwendungen in großen Teilen bekannt vorkommen, deshalb gehen wir hier nicht näher darauf ein. Entscheiden Sie sich für Ihre Lieb - lingschrift und bestätigen Sie den Dialog.

Wir haben versprochen, noch eine andere Option anzusehen. Wählen Sie aus der Box, die erscheint, wenn die rechte Maus - taste gedrückt wird wieder das Symbol mit dem Hammer, also das Symbol für die objekt-spezifischen Attribute. Klappen Sie hier die Liste **Auswahl** auf. Entscheiden Sie sich für irgend - einen Eintrag in der Liste, vielleicht **Pfeil nach rechts**. In dem kleinen Grafikeditor sehen Sie, wie **Pfeil nach rechts** aussieht. Bestätigen Sie den Dialog mittels *OK*. Die Schaltfläche zeigt nun die kleine Grafik, und unser mühevoll formatierter Text scheint verloren zu sein. Mitnichten. Kehren Sie zurück in den entsprechenden Dialog und betrachten Sie die mittlere der drei Optionsboxen im oberen Teil. Die Option **Grafik** ist angewählt. Wenn Sie nun die Option **Text** an - wählen, erscheint wieder der Text, der vorher definiert wurde und immer noch in dem Feld **Inhalt** steht. Ob Sie das jetzt möchten, wissen wir nicht. Wir lassen Ihnen die Wahl.

Bitte fügen Sie nun auf der Zielseite (Target) auf eigene Faust eine Schaltfläche ein, die den Anwender wieder zurück zur Quellseite (Source) bringt. Das können Sie nun schon ganz alleine. Speichern Sie bitte Ihre Bemühungen ab. Unter **TUT_02.MM** finden Sie den Stand dieser Beispielapplikation.

Wenn beide Seiten nun entsprechende Schaltflächen zum Um - schalten auf die jeweils andere Seite haben, können wir die Applikation noch einmal zur Ausführung bringen. Sie kennen

bereits die Wege, die hierzu geeignet sind. Verwenden Sie die Schaltflächen, um zwischen den Seiten hin und her zu springen. Anhand der geschickt platzierten Komponenten Rechteck und Kreis erkennen wir, wo wir uns befinden.

Wenngleich diese Übung auch keine Anwendung her vorgebracht hat, wie wir sie im Alltag dringend benötigen, war sie doch dazu geeignet, einen wesentlichen Sachzusammenhang in MINDMAP darzustellen. Fassen wir die Erkenntnisse doch einmal zusammen:

Zusammenfassung Übung 1

- ▶ Die Toolbox auf der rechten Seite des Bildschirms (verschiebbar) beinhaltet Werkzeuge zur Erstellung von Komponenten. In vielen Komponentenfamilien gibt es mehrere Varianten.
- ▶ Verschiedene Dialoge werden in MINDMAP auf verschiedenen Seiten untergebracht.
- ▶ Komponenten, zu denen auch die Seiten gehören, sollen sprechende Namen tragen
- ▶ Komponenten sind untereinander verknüpfbar. Wir nennen diese Verbindungen Links.
- ▶ Der Dialog zum Definieren von Links wird nach Anwahl der Komponente mit der linken Maustaste, durch Druck auf die rechte Maustaste und Anwahl des Link-Symbolen aufgerufen.
- ▶ Ein Link besteht aus drei wesentlichen Bestandteilen: Ereignis, Nachricht, Komponente an die die Nachricht geschickt wird.

Nachrichten und Links

In diesem Kapitel werden Sie in die Geheimnisse von Nachrichten und Links eingeweiht. Das klingt spannend, wie so oft aber in MINDMAP steht der minimale Aufwand jedoch in keinerlei Verhältnis zu den aufregenden Ergebnissen. Bevor wir etwas richtig ein drucksvolles "programmieren", müssen wir uns aber gemeinsam mit vorläufig etwas trockenerer Materie auseinandersetzen. Versprochen: Im nächsten Kapitel wird es bunter, wenn Sie hier am Ball bleiben.

Eingabefeld und Text Komponente

Bitte sorgen Sie für eine freie Arbeitsumgebung. Wenn Sie MINDMAP eben erst gestartet haben, erledigt sich das von selbst. Wenn nicht, dann speichern Sie, wenn nötig, Ihr gegenwärtiges Projekt und schließen es. Wählen Sie dann über das Menü **Datei | Neu | Anwendung** eine neue Anwendung.



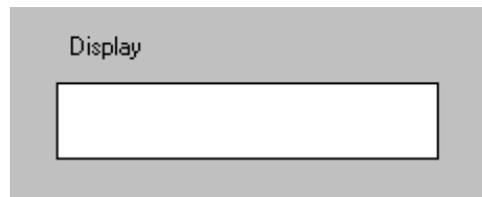
Bedienen Sie sich des Werkzeuges für ein **Eingabefeld** und zeichnen Sie ein schlankes Rechteck auf den Bildschirm. Dieses Feld soll die Texte und Werte sichtbar machen, die wir über noch zu erstellende Komponenten auswählen. Bei der Erzeugung eines Eingabefelds, werden Sie direkt nach einem neuen Namen für diesen Komponenten gefragt. Nennen Sie Ihr Feld **Display**. Sollten Sie bereits einen anderen Namen gewählt haben, können Sie die Komponente ja einfach umbenennen. Wenn Sie dies jedoch nicht möchten, dann beachten Sie bitte, daß in folgenden Ausführungen immer dieses Feld gemeint ist, wenn von **Display** die Rede ist.



Mit dem nächsten Schritt versehen wir das Eingabefeld mit einem Titel. Hierbei wird auch der Unterschied zwischen zwei erst einmal ganz ähnlichen

Komponenten deutlich. Wählen Sie das Werkzeug für die **Text-Komponente** und platzieren Sie diese in Form eines schmalen Rechteckes über dem Eingabefeld.

Dieses Feld präsentiert sich etwas anders als das Eingabefeld, denn der Hintergrund bleibt transparent. Der Hauptunterschied dieser beiden Komponenten besteht darin, daß das Eingabefeld der Interaktion zwischen Anwender und Anwendung dient. Die Text-Komponente hingegen dient ausschließlich der Anzeige von Daten. Da dieses Feld wenig aktive Aufgaben übernimmt, werden Sie auch nicht zur Eingabe eines Namens aufgefordert. Nach Platzierung der Text-Komponente blinkt in diesem ein Cursor, der dazu auffordert, den Text einzugeben, der dargestellt werden soll. Bitte tragen Sie Display ein. Sie können das Feld nun in die richtige Form bringen, so daß die beiden Komponenten etwa wie in folgender Darstellung aussehen.



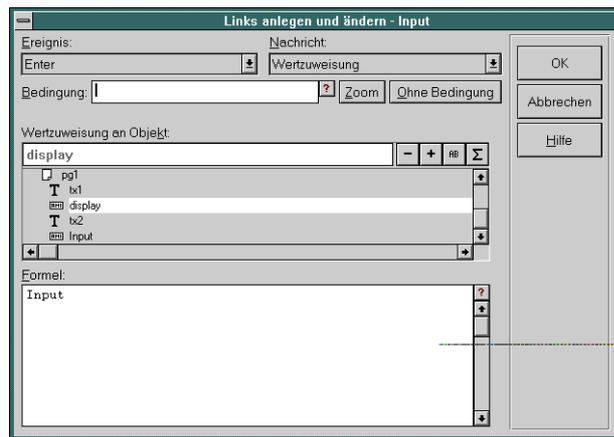
Erzeugen Sie nun ein weiteres Eingabefeld, irgendwo an einem freien Platz auf dem Bildschirm. Nennen Sie das Feld Input.



Klicken Sie, während das neue Eingabefeld markiert ist, mit der rechten Maustaste in den Bildschirm, um das Fenster **Attribute** zu aktivieren. Wählen Sie hier bitte das Werkzeug für Links aus.

Erzeugen Sie nun über die Schaltfläche *Neu* einen neuen Link für diese Komponente. Auf der linken Seite des Dialoges müßte das Ereignis **Enter** vorgeschlagen sein. Sollte dies nicht der Fall sein, wählen Sie diese Option bitte aus. Dieses **Enter** bedeutet: "wenn die Enter-Taste gedrückt wird". Auf der rechten Seite, der Seite für die Nachrichten, stellen Sie bitte **Wertzuweisung** ein. Dann wählen Sie in der Liste für das Ziel der Nachricht, das Feld Display. Jetzt muß noch festgelegt werden, welcher Wert an das Feld Display übertragen werden soll. Wir möchten den Inhalt des Eingabefelds Input übertragen. Daher schreiben Sie in das Formelfeld

einfach den Namen des Eingabefelds, also Input (ohne Anführungszeichen), hinein.



Bestätigen Sie die Dialoge mit *OK*, bis Sie sich wieder auf der Seite mit den Komponenten befinden. Jetzt können wir die Anwendung speichern und testweise starten. Die Anwendung liegt auch unter dem Namen **TUT_03.MM** als Datei vor.

Wenn Sie in den Run Mode gewechselt sind, geben Sie bitte in das Eingabefeld Input einige Buchstaben ein. Vielleicht "Hallo Welt" (ohne Anführungszeichen), denn diese Worte sind immer die ersten, die ein Programmierer bei einem Kurs verwendet. Das ist klassisch! Bestätigen Sie die Eingabe mit der *ENTER*-Taste. In dem Feld Display werden die hier eingegebenen Zeichen sichtbar.

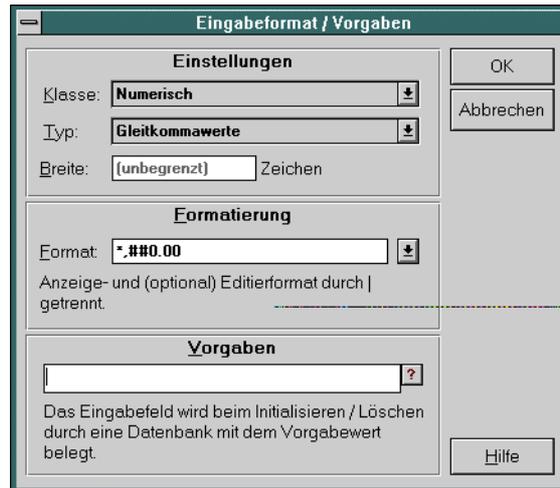
Beenden Sie die Ausführung durch Druck auf *F4*. Nun sind Sie wieder im Edit Mode. Verändern Sie den Link bitte, indem Sie in dem Formelfeld das Wort "Input" in Anführungszeichen setzen. Probieren Sie die Anwendung erneut aus.

Wie man sieht, unterscheidet MINDMAP anhand der Anführungszeichen, ob es sich um einen literalen Wert oder den Namen einer Komponente handelt.

Dem Begriff **Formel** sind wir aber noch nicht ganz gerecht geworden. Daher erweitern wir unsere Anwendung. Wir erzeugen einen kleinen Rechenapparat, der Grad-Celsius in Grad-Fahrenheit umrechnet und umgekehrt. Eine Anwendung die ja wirklich jeder Mensch nahezu täglich benötigt!

Plazieren Sie die Felder optisch wirksam nebeneinander. Ändern Sie die Beschriftung des Eingabefelds von *Display* in *Fahrenheit*. Hier ist nicht der Name des Eingabefelds gemeint, sondern der Inhalt der Text-Komponente. Passen Sie eventuell die Breite an. Versehen Sie das Eingabefeld Input mit dem Titel *Celsius*.

Wählen Sie nun das Eingabefeld mit dem Namen Input an, und drücken Sie die rechte Maustaste, um zu Attribute zu gelangen. Wählen Sie bitte das Symbol für **Eingabeformat/ Vorgaben**.



Bei der Umrechnung von Temperaturen benötigen wir nur Zahlen. Die Anzeige begrenzen wir auf zwei Nachkommastellen. Wählen Sie bitte in diesem Dialog aus den entsprechenden Listen die notwendigen Einstellungen aus:

- **Klasse** Numerisch
- **Typ** Gleitkommawerte.
- **Format** *,##0.00

Verändern Sie bitte analog hierzu die Einstellungen für das Eingabefeld Display.

Tragen Sie nun in das Formelfeld des Links für das Eingabefeld Input die folgende Formel ein:

$$\text{Input} * 1.8 + 32$$

Das ist die Formel für die Umrechnung von Grad-Celsius in Grad-Fahrenheit. Bestätigen Sie den Dialog mit *OK*.

Legen Sie nun auch einen Link für das Eingabefeld *Display* an. Auch hier ist das *Ereignis* **Enter** und die *Nachricht* **Wertzuweisung**. Die Wertzuweisung erfolgt an den Komponenten *Input* und die Formel lautet:

$$(\text{Display} - 32) / 1.8$$

Das war's.

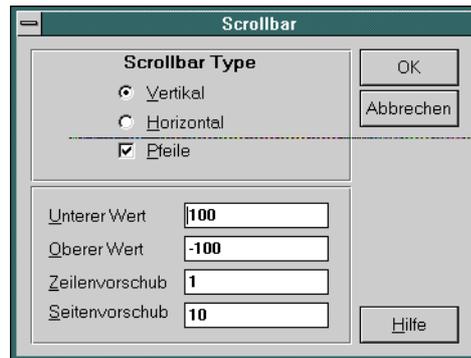
Probieren Sie jetzt die Anwendung aus. Egal, in welchem Feld Sie einen Eintrag vornehmen, in dem jeweils korrespondierenden Feld wird der umgerechnete Wert dargestellt. Diese Anwendung finden Sie unter **TUT_04.MM**.

Scrollbar



Um die Eingabe etwas komfortabler zu gestalten, installieren wir nun zwei Scrollbars. Sie finden das Werkzeug hierzu in der Gruppe **Schalter**. Bei einem Neustart erscheint in der Toolbox das Symbol der Befehlsschaltfläche, welches die Gruppe der Schalter symbolisiert. Ansonsten erscheint hier, wie bei jeder Gruppe der Tools, das während der Sitzung als letztes verwendete Symbol. Klicken Sie also das Fenster der Gruppe **Schalter** auf, und wählen Sie dann das Werkzeug für die Bildlaufleiste, die sog. **Scrollbar**.

Plazieren Sie die Scrollbar als schmales, aufrechtes Rechteck links neben dem Eingabefeld *Input*. Wenn sie Ihnen zu klobig erscheint, können Sie sie ja noch zurechtrücken. Nennen Sie die Scrollbar *scr1_C* (C für Celsius). Eine Scrollbar gibt numerische Werte zurück. Um einstellen zu können aus welchem Bereich diese Werte kommen, klicken Sie bitte die rechte Maustaste, während die Scrollbar angewählt ist. Aus dem so erzeugten Attribute-Fenster wählen Sie bitte das Symbol für die objekt-spezifischen Attribute aus. Sie müßten nun den folgenden Dialog sehen.



In diesem Dialog lassen sich nicht nur die Anfangs- und Endewerte (**Unterer Wert**, **Oberer Wert**) einstellen, sondern auch über **Zeilenvorschub** die Schrittweite bei einem Klick auf einen der Pfeile, die sich am Anfang und am Ende des Balkens befinden; natürlich nur, wenn diese Option auch an - geschaltet ist (**Pfeile**). Es läßt sich auch die Größe einer Seite bestimmen, wobei man hier den Begriff Seite nicht so genau nehmen darf. Zeile und Seite symbolisieren hier eher einen einzelnen Schritt oder einen "großen" Schritt. Der Effekt des **Seitenvorschubs** läßt sich erkennen, wenn man bei der Bedienung der Scrollbar mit der Maus in die freie Fläche zwischen den Pfeilen und dem Schieberegler klickt .

Geben Sie bitte als oberen Wert *-100* und als unteren Wert *100* ein, bestimmen Sie als **Zeilenvorschub** *1* und als **Seitenvorschub** *10*. Fertig! Bitte beachten Sie, daß sich **Oberer Wert** und **Unterer Wert** auf die Position am Bildschirm und nicht auf die Größe der Werte beziehen.

Jetzt stellen Sie einen Link her, der bei dem Ereignis **Scrollbar Änderung** eine **Wertzuweisung** von `scr1_C` an das Eingabefeld Input durchführt. Nun dürfen Sie sich zurück - lehnen und sich ein wenig darüber freuen, wie Ihnen das trotz der knappen Beschreibung gelungen ist.

Wenn die Anwendung jetzt gestartet werden würde, könnten wir zwar den Effekt genießen, daß der eingestellte Wert der Scrollbar in dem Eingabefeld Input erscheint, aber eine neue Berechnung und Abbildung des Wertes im Feld Display wird dadurch noch nicht durchgeführt. Den Link haben wir ja dem Feld Input zugeordnet und zwar nach Druck auf die Enter-Taste.

Kopieren von Links

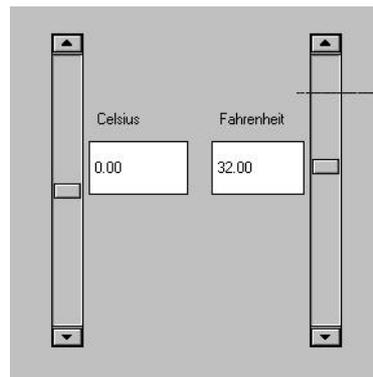
Wir können den Link des Eingabefeldes Input kopieren. Gehen Sie dazu in den Link-Dialog dieser Komponente. Markieren Sie den Link in der erscheinenden Liste durch Anklicken mit der Maus. Ist ein Link markiert, erscheint er hell unterlegt am Bildschirm. Sicher haben Sie auch bemerkt, daß durch das Markieren eines Links, verschiedene Schaltflächen aktiviert worden sind, die vorher inaktiv waren. Eine davon trägt die Aufschrift *Kopieren*. Klicken Sie bitte diese Schaltfläche. Verlassen Sie den Dialog mittels *Abbrechen*. Gehen Sie jetzt in den entsprechenden Dialog für den Scrollbar. Betätigen Sie dort die Schaltfläche *Einfügen*, und der kopierte Link erscheint. Er trägt jedoch kein Ereignis am Anfang der Zeile, da das Ereignis *Enter* für Scrollbar keinen Sinn macht. Sie können den Link nun doppelklicken oder die Schaltfläche *Ändern* drücken, wenn er markiert ist. Im Dialog **Links anlegen und ändern** wählen Sie als Ereignis **Scrollbar Änderung**. Achten Sie bitte darauf von wo und wohin die **Wertzuzuweisung** stattfindet. Beenden Sie den Dialog mit *OK*, und bleiben Sie noch ein wenig in der Liste der definierten Links. Hier ist es jetzt von entscheidender Bedeutung, an welcher Stelle der Link steht, da die Links der Reihe nach von oben nach unten abgearbeitet werden. Steht der Link mit der Formel an erster Stelle, dann findet zunächst die Berechnung statt und erst danach der Eintrag des neuen Wertes in das Eingabefeld Input. Das Ergebnis wäre falsch. Also ist es wichtig, daß zunächst der Eintrag in das Eingabefeld stattfindet und erst dann die Berechnung mit dem neuen, korrekten Wert und die Wertzuzuweisung. Einen markierten Eintrag in dieser Liste können Sie mittels der Schaltflächen im rechten unteren Teil des Dialoges verschieben. Diese Schaltflächen sind jeweils durch einen Pfeil nach oben oder unten gekennzeichnet. Wenn nötig, stellen Sie die erforderliche Reihenfolge her. Schließen sie den Dialog mit *OK*.

Testen Sie nun die neue Anwendung. Zum Vergleich liegt Ihnen diese Anwendung als **TUT_05.MM** vor.

Kopieren von Komponenten

Drücken Sie bitte, während die bereits erstellte Scrollbar angewählt ist, die Taste F2. Diese Taste dupliziert die selektierte Komponente. Verschieben Sie die so erzeugte neue Komponente rechts neben das Eingabefeld Display, und benennen Sie ihn scr1_F. Ändern Sie nun den beim Dupliziervorgang mit übernommenen Link der neuen Komponente so, daß in das Eingabefeld Display der Wert der Komponente scr1_F eingetragen wird, wenn man dessen Einstellung verändert. Auch der zweite Link muß dahingehend abgeändert werden, daß der neue Fahrenheit-Wert in Display mittels o. g. Formel in Celsius umgerechnet wird, und dann in Input erscheint. Mancher mag es auch vorziehen, den Link aus dem Eingabefeld Display zu kopieren, und ihn bei der Komponente scr1_F einzufügen und zu ändern. Das klappt ja wie am Schnürchen.

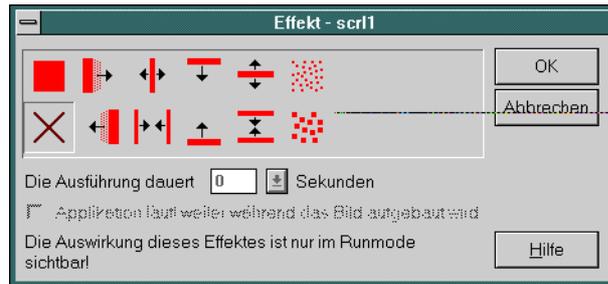
Die Anwendung müßte nun ungefähr wie folgt aussehen (s. Anwendung **TUT_06.MM**). Gehen Sie nun in den Run Mode, um Ihre Anwendung zu testen.



Attributänderung

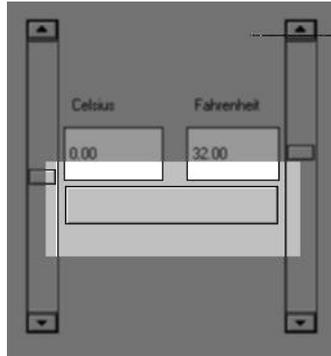
Im nächsten Schritt wollen wir erreichen, daß die Scrollbars zunächst unsichtbar sind und erst erscheinen, wenn eine

Taste gedrückt wird. Wieder im Edit Mode, erzeugen Sie bitte für jeden der beiden Scrollbars nacheinander einen neuen Link. Dieser soll bewirken, daß beim Ereignis **Anwendung gestartet** die Nachricht **Attributänderung** durchgeführt wird. Sobald Sie im Dialog, mittels dessen Links erzeugt werden, die Nachricht **Attributänderung** und die Komponente, der diese Nachricht geschickt werden soll ausgewählt haben, erscheint eine Liste der Attribute, die bei dieser Komponente geändert werden können. Im Falle einer Scrollbar sind dies die Attribute **Farben** und **Effekt**. Wählen Sie bitte **Effekt** und betätigen Sie die Schaltfläche **Ändern**, so daß der folgende Dialog erscheint.

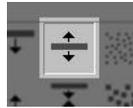


Wählen Sie hier unter all den Effekten, den gewünschten aus. Das Symbol für *unsichtbar* ist das Kreuz (erstes Icon in der zweiten Reihe). Schließen Sie nun bitte alle Dialoge mit **OK**.

Zum Glück verschwinden die Komponenten nicht gleich, sondern wirklich erst beim Starten der Anwendung. Jetzt benötigen wir noch eine Schaltfläche, die dafür sorgt, daß die Scrollbars eingeschaltet, d. h. sichtbar gemacht werden können. Zeichnen Sie also eine Befehlsschaltfläche, vielleicht gerade so unter die beiden Felder.



Benennen Sie die Schaltfläche `btn_Scroll_an` und versehen Sie sie mit dem Text "Scrollbar" oder "Scrollbars". Sie erreichen das durch Klick mit der rechten Maustaste, während die Taste angewählt ist und Auswahl des Symbols für die objekt-spezifischen Attribute. Aber das wissen Sie ja bereits aus den Kapiteln davor.



Stellen Sie einen Link her, der eine Attributänderung des Scrollbars `scr1_C` bewirkt. Hierzu können Sie einen Effekt Ihrer Wahl einstellen.

Sehr schick kommt der abgebildete Effekt mit einer Verzögerung von 0.25 Sekunden zur Geltung. Im Dialog, in dem die Links aufgelistet sind, können Sie den Link kopieren und gleich wieder einfügen. Den so eingefügten Link können Sie dann entsprechend ändern, damit auch der Scrollbar `scr1_F` eingeschaltet wird.

Jetzt sollten Sie die Anwendung einmal anschauen. Beim Start sind keine Scrollbars sichtbar, und bei Druck auf die Schaltfläche werden Sie in kaum zu übertreffender Dramatik auf dem Bildschirm erscheinen. (Anwendung **TUT_07.MM**)

Zeit für eine Pause. Sie können gerne einige Celsius und Fahrenheit Grade umrechnen oder über die entsprechenden Hilfsmittel, das Erscheinungsbild Ihrer Anwendung günstig beeinflussen. Vielleicht sieht das Ganze dann so aus?

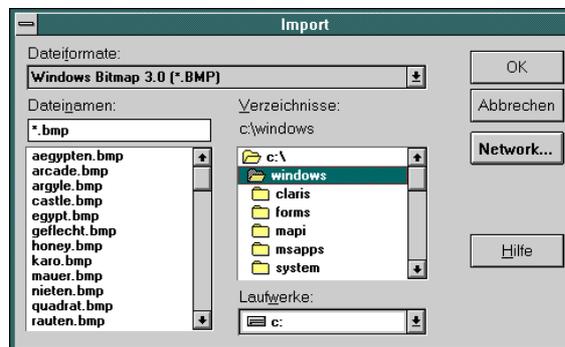


Ein Picture-Viewer

Graphik Import Komponente



In dieser Lektion befassen wir uns mit einem Thema, das wirklich Spaß macht. Wir erzeugen einen Viewer, um gespeicherte Bilder anzusehen. Das Symbol hierzu hat Sie schon eine ganze Weile gereizt, nicht wahr? Nach Auswahl des Symboles mit dem Kaktus erscheint der folgende Dialog:



MINDMAP fragt Sie zunächst nach einem Bild, welches die Komponente repräsentieren soll. In dem dafür vorgesehenen Dialog können Sie Laufwerk, Pfad und Dateiname einstellen. Bitte beachten Sie auch die verschiedenen Dateiformate. In der entsprechenden Liste finden Sie eine umfangreiche Auswahl an Grafikformaten. Wählen Sie für unser Beispiel *Windows Bitmap 3.0 (*.BMP)* und eine Datei aus dem Windows-Verzeichnis.



Nach Bestätigen des Dialogs mit *OK* oder einem Doppelklick auf die entsprechende Datei erscheint anstelle des Cursors ein Farbtopf, der bedrohlich im Begriff ist, umzukippen. Sie brauchen nun nichts anderes zu tun, als auf den Punkt zu klicken, den Sie als obere linke Ecke

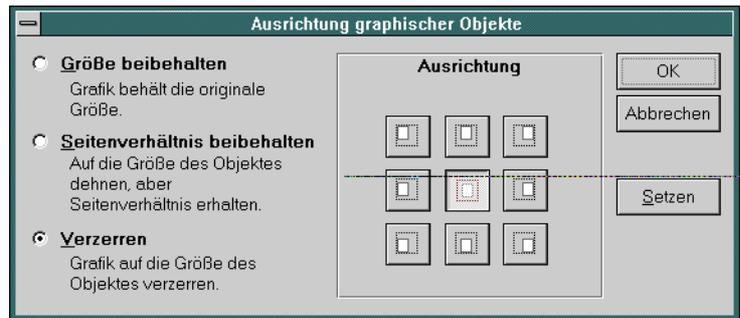
des Bildes für geeignet halten. MINDMAP wird die Komponente automatisch der Größe des Bildes anpassen. Deshalb ist es nicht erforderlich, daß Sie den Bereich durch Verziehen mit der Maus markieren.

Das Bild wird nach einer gewissen Ladezeit, die sich nach der Dateigröße richtet, auf dem Bildschirm sichtbar. Bitte vergessen Sie nicht, die Komponente zu benennen. Vorschlag: grf_Picture.

Betrachten wir nun die Attribute, die für eine Bild-Komponente einstellbar sind. Klicken Sie, während das Bild ausgewählt ist, mit der rechten Maustaste in den Bildschirm. Hier finden wir neben Attributen, die wir bereits von anderen Komponenten kennen, Attribute, die nur bei dieser Komponente zu finden sind.



Wenn Sie sich die verschiedenen Attribute anschauen, treffen Sie auch auf den Dialog **Ausrichtung graphischer Objekte**.



Die linke Seite des Dialogs enthält drei Optionen, in welcher Form das Bild behandelt wird. Um immer die korrekte Größe zu erhalten, wählen Sie bitte die Option *Größe beibehalten*. Auf der rechten Seite des Dialogs können Sie angeben, wie das Bild in der Komponente ausgerichtet wird. Wählen Sie für unser Beispiel die Ausrichtung nach der linken oberen Ecke.

Nun kann der anfängliche Erfolg, blitzschnell ein Bild in den Bildschirm einzubringen, nicht darüber hinwegtäuschen, daß unsere Anwendung etwas steif ist. Natürlich ist es nicht gewünscht, für jedes in Frage kommende Bild, eine solche Komponente zu installieren; womöglich noch eines auf jeweils einer dafür vorgesehenen Seite. Vielmehr erscheint es praktisch, eine Liste der in Frage kommenden Bilder zu haben, und jeweils die entsprechende Auswahl abzubilden. Eine solche Liste werden wir jetzt erstellen.

Dateiliste



Verwenden Sie das Werkzeug, um eine Listbox zu erzeugen. Nennen Sie diese Liste `lst_Pictures`. Haben Sie dies getan, handelt es sich bei dieser Komponente um eine ganz normale Liste, auch Listbox genannt. Um aus der Listbox eine Dateiliste zu machen, sind entsprechende Eintragungen bei den objekt-spezifischen Attributen zu machen. Wenn die Listbox gezeichnet und selektiert ist, wählen Sie bitte aus den Attributen das Symbol für die objekt-spezifischen Attribute aus.



In dem erscheinenden Dialog soll nun die Option **Dateiliste** mit allen darunter befindlichen Optionen, wie **Dateinamen**, **Laufwerke** und **Pfadnamen**, an-

geschaltet werden. Nach Beenden des Dialoges mit *OK* fragt MINDMAP freundlich nach, ob es den entsprechenden Link generieren soll. Wenn dies bestätigt wird, werden Sie auf gefordert, den Dateityp auszuwählen, bzw. einzutragen. MINDMAP wird sich erneut bei Ihnen melden und berichten, daß ab sofort beim Starten der Anwendung alle Dateien des aktuellen Verzeichnisses, die vom angegebenen Typ sind, in dieser Listbox angezeigt werden. Man wird aufgefordert, den automatisch generierten Link nach zubearbeiten, falls man ein anderes Verzeichnis wünscht, was ja bei uns der Fall ist.

Um nachzuvollziehen, was wirklich geschehen ist, rufen Sie bitte die Links der Dateiliste auf. Doppelklicken Sie bitte den Link, der sich dort befindet, um ihn zu editieren. Hier sollte nun als Ereignis **Anwendung gestartet** stehen, als Nachricht **Combo-/Listbox**. In Objekte sollte unsere Liste `lst_Pictures` und bei Kommando **Verzeichnis...** selektiert sein. Im Feld Text befindet sich vermutlich der Eintrag `"*. *"`, welcher nach unseren Anforderungen abzuändern ist. Tragen Sie bitte hier `"C:\\windows*.bmp"` ein, wobei wir hier davon ausgehen, daß sich die Dateien tatsächlich in diesem Verzeichnis befinden. Schließen Sie sämtliche Dialoge mit *OK*.

Bitte wechseln Sie in den Run Mode und testen Sie, wie die Anwendung reagiert (s. Anwendung **TUT_08.MM**).

Wir haben jetzt also eine Listbox, die die Dateinamen enthält, und einen Container, der das in der Listbox angewählte Bild repräsentieren soll. Mit den folgenden Schritten werden wir die beiden Komponenten zusammenbringen.

Drag&Drop

Drag&Drop per Link

Nun müssen wir der Anwendung noch beibringen, was sie tun soll. Wir möchten gerne, daß immer genau das Bild der Datei angezeigt wird, deren Dateiname in der Liste selektiert worden ist.

Hierzu erzeugen wir einen neuen Link auf unsere Listbox. Selektieren sie also bitte die Dateiliste, lassen Sie sich die

Attribute anzeigen und wählen Sie dort das Link-Symbol. Der Dialog mit der Linkliste wird sichtbar, wo Sie den von Ihnen vorher modifizierten Link sehen müßten. Wählen Sie nun bitte die Schaltfläche *Neu*, um den benötigten Link zu erzeugen.

Wählen Sie als Ereignis **Cursorwechsel** und legen Sie als Nachricht **Drag&Drop** fest. In den beiden dann erscheinenden Listen muß noch sichergestellt werden, daß der Sender die Listbox ist und der Empfänger die Bild-Komponente. Bestätigen Sie den Dialog und probieren Sie Ihre Anwendung aus (s. Anwendung **TUT_09.MM**).

Drag&Drop per Maus

Vielleicht ist Ihnen bei der Auswahl der Nachricht **Drag&Drop** das Drag&Drop-Symbol bekannt vorgekommen. Richtig! Sie haben es schon bei den Attributen verschiedener Komponenten gesehen, z. B. auch bei Ihrer Bild-Komponente und bei der Listbox.



Dieses Attribut dient dazu, eine Komponente zu veranlassen, auf Drag&Drop mit der Maus zu reagieren, wie wir Anwender von MS-Windows es kennen und lieben.

Um dies kurz auszuprobieren, modifizieren wir die bestehende Anwendung. Gehen Sie bitte in die Linkliste der Listbox *lst_Pictures*. Hier deaktivieren wir nun den Link, der die Drag&Drop-Nachricht enthält. Ihn brauchen wir ja jetzt nicht, da wir ja Drag-&Drop mit der Maus machen wollen. Haben Sie schon die Checkbox **Aktiv** in diesem Dialog bemerkt. Sie zeigt immer an, ob der gerade selektierte Link aktiv ist, d. h. ob er ausgeführt wird oder nicht. Setzen Sie bitte den Schalter für den betreffenden Link auf inaktiv. Sobald die Check box den Haken verliert, wird auch die Darstellung des Links verändert, so daß in der Linkliste der Unterschied zwischen Links, die ausgeführt werden, und welchen die nicht ausgeführt werden sofort erkennbar ist. Schließen Sie den Dialog mit *OK* und wechseln Sie in den Run Mode, um die Anwendung zu testen. Sie müßten nun feststellen, daß die Bilder der ausgewählten Dateien nicht mehr angezeigt werden.

Wieder zurück im Edit Mode ändern wir die Anwendung um, daß Drag&Drop mit der Maus stattfindet. Zunächst legen wir fest, wie die Komponenten auf die Nachricht **Drag&Drop** reagieren sollen. Wählen Sie bitte die Listbox an. Klicken Sie mit der rechten Maustaste, und entscheiden Sie sich für das Drag&Drop-Symbol. Legen Sie in dem darauf erscheinenden Dialog fest, daß **Daten senden (Maus)** möglich sein soll. Die anderen Optionen in diesem Dialog bleiben unverändert. Legen Sie nun für die Bild-Komponente auf die gleiche Weise fest, daß sie **Daten empfangen** soll.

Testen Sie nun die Anwendung (s. Anwendung **TUT_10.MM**). Wenn Sie nun in der Liste einen Dateinamen auswählen, und dann die Maus in Richtung der Bild-Komponente bewegen, während Sie gleichzeitig die linke Maustaste gedrückt halten, werden Sie bemerken, daß jetzt wieder die Bilder angezeigt werden.

Lassen Sie uns nicht darüber streiten, welche der beiden Methoden die elegantere ist.

Bildformat

Unsere Bild-Komponente hatte, da wir am Anfang den Cursor beim Erzeugen nicht verzogen haben, die Größe des ersten Bildes angenommen. Diese Größe stellt auch den Ausschnitt für die folgenden Bilder dar. Zwar werden kleinere Bilder nicht auf die Größe der Komponente gebracht, von größeren Bildern wird aber, ohne Verzerrung, nur der in die Komponente passende Ausschnitt gebracht. Und jetzt ist wieder Spielstunde. Da Sie ja bereits Erfahrung mit MINDMAP gemacht haben, habe wir keine Bedenken Sie jetzt unbeaufsichtigt mit einigen Optionen spielen zu lassen.



Wenn Sie sich in der zweifelhaften Lage befinden, ständig mit dem Eintreten Ihres Chefs zu rechnen, heißt das, was Sie jetzt tun, natürlich nicht "spielen" sondern etwa "evaluieren" oder "testen". Setzen Sie bitte im schlimmsten Falle ein ernstes Gesicht auf, werfen Sie die Stirn in Falten. Wir nehmen es Ihnen auch nicht übel, wenn Sie in einem solchen Falle den Kopf wiegen und leise "ts, ts, ts" sagen.



Testen Sie die Auswirkung von Änderungen der Optionen in dem Dialog, der erscheint, wenn Sie das Symbol für die Ausrichtung im *Attribute*-Fenster wählen.



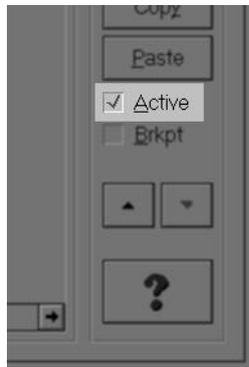
Beobachten Sie auch, was geschieht, wenn Sie Änderungen bei den objekt-spezifischen Attributen (Symbol mit dem Hammer) machen. So wird beispielsweise die Größe der plazierten Bild-Komponente der Größe der gegenwärtig geladenen Grafik angepaßt, wenn die Schaltfläche **Originalgröße** gedrückt wird. Das hat, wie wir ja bereits wissen, besonders Einfluß auf die Grafiken, die größer sind als die momentan geladene.

Können Sie nun wieder einen Zustand herstellen, der die Grafiken vernünftig abbildet? Das soll jetzt kein Tadel sein. Aber wir wollen ja weitermachen, nicht wahr? Stellen wir uns einmal vor, sie seien Art-Director Ihrer Firma (wenn Sie wirklich Art-Director sind, dann stellen Sie sich vor, Sie seien Programmierer). Sie möchten nun aus einer Anzahl Bilder diejenigen auswählen, die als Hintergrund für einen neuen Katalog in Frage kommen. Hierbei soll der Name des Bildes in eine Liste eingetragen werden, und diese Liste soll nach Beendigung Ihrer Auswahl in eine Datei geschrieben werden. Oder auch in die Zwischenablage, was nützlich ist, wenn parallel z. B. eine Textverarbeitung offen ist. Hierzu benötigen wir noch etwas Platz auf unserem Bildschirm. Alle diejenigen, die nicht mit einem 21 Zoll Bildschirm gesegnet sind, räumen also ein wenig auf. Bringen Sie neben der Listbox ein Eingabefeld `edt_Selection` an. Dieses Feld soll mehrzeilige Einträge zulassen. Stellen Sie also über die objekt-spezifischen Attribute die Option **Mehrzeilig** ein. Außerdem sollte es eine geeignete Größe zur Abbildung mehrerer Einträge haben. Damit man innerhalb der Einträge blättern kann, schalten Sie noch die Option **Vert. Scroll** ein.

Plazieren Sie eine Befehlsschaltfläche, die den ausgewählten Eintrag der Listbox in das Eingabefeld übertragen soll. Wählen Sie als Ort dafür vielleicht unter dem Input-Feld und neben der Listbox. Nennen Sie die Schaltfläche `btn_Select` und betiteln Sie sie mit *Select*.

Links deaktivieren

Bei den folgenden Versuchen kann es lästig sein, wenn nach jeder Wahl in der Dateiliste ein Bild geladen wird. Wir möchten schließlich nur den Link zwischen der Listbox und dem Eingabefeld ausprobieren. Andererseits wollen Sie ja auch sicher nicht auf den gelungenen Link auf der Picture-Komponente verzichten. Hierzu gibt es eine praktische Lösung, die Sie ja bereits kennengelernt haben. Wählen Sie die Listbox an. Gehen Sie in den Dialog mit der Linkliste. Wählen Sie den Link an, der das **Drag&Drop** auf die Komponente `grf_Picture` vornimmt und schalten Sie diesen temporär inaktiv. Jetzt bleibt der Link erhalten und ist dennoch nicht aktiv. Sie dürfen nur nicht vergessen, ihn im Ernstfall auch wieder anzuschalten.



Parser Funktionen

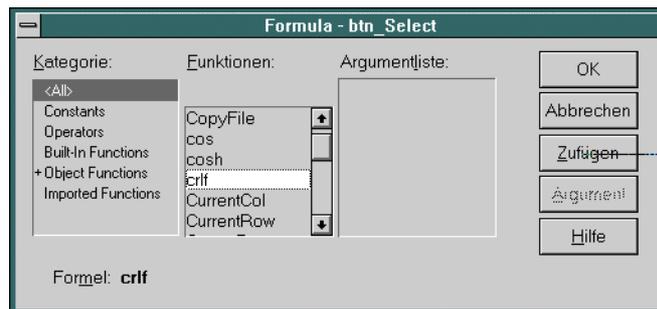
Erzeugen Sie nun einen Link für die neue Schaltfläche `btn_Select`, der eine **Wertzuweisung** von der Listbox zum Eingabefeld vornehmen soll. Vergessen Sie bitte nicht, die Komponente `edt_Selection` als Ziel für die Wertzuweisung anzugeben.

Spontan hat dieser Link einen kleinen Schönheitsfehler. Er überträgt zwar den gewählten Eintrag, fügt ihm aber nicht die eventuell bereits vorhandenen Einträgen hinzu. Deshalb soll

in dem Formelfeld auch nicht nur stehen `Ist_Pictures`, sondern wenigstens `edt_Selection + Ist_Pictures`. Das Plus-Zeichen in der Formel addiert nicht nur numerische Werte, sondern es verkettet auch Buchstabenfolgen. Bitte testen Sie die Arbeit der Schaltfläche.



Hier ergibt sich ein weiteres Manko. Denn der neue Eintrag wird zwar zu dem bereits vorhandenen hinzugefügt, aber es entsteht leider keine eigene Zeile pro Auswahl. Wir editieren den Link also erneut. Um zu erreichen, daß eine neue Zeile entsteht, bedienen wir uns einer sogenannten **Parser-Funktion**. Das ist neu! Die Parser-Funktionen von MINDMAP sind Funktionen, die nicht ohne weiteres über das Menü oder erstellte Komponenten erreichbar sind, sondern zusätzlich zur Verfügung stehen. Wenn Sie das Formelfeld des Dialogs zum Erstellen von Links vor sich sehen, dann beachten Sie in der rechten oberen Ecke des Feldes ein Fragezeichen. Wenn Sie dieses anklicken, bekommen Sie Unterstützung zu den Parser-Funktionen von MINDMAP. Hier können Sie sich die Syntax von eventuell in Vergessenheit geratenen Funktionen ansehen und auch gleich in die Formel übertragen lassen. Die meisten Funktionen des Parsers benötigen einen oder mehrere Parameter, die innerhalb von Klammerpaaren angegeben werden.



Der Parserdialog unterstützt Sie auch hier wirkungsvoll mit Hilfen und Hinweisen zu den erwarteten Parametern. Beim Blättern innerhalb der Liste finden Sie einen Eintrag mit der Bezeichnung `CrLf`. Diese Funktion benötigt keine weiteren Angaben. Sie erzeugt lediglich was die Buchstaben bereits andeuten, nämlich einen carriage return line feed, also einen Zeilenumbruch. Wählen Sie diese Funktion aus und betätigen

Sie die Schaltfläche *Zufügen*. Schließen Sie den Dialog dann mit *OK*.

In Ihrem Formelfeld sollte jetzt stehen



Bitte probieren Sie unseren neuen Link aus.

Bedingung

Kann es sein, daß Sie erst zufrieden mit diesem Link sind, wenn eine Auswahl nicht zweimal in die Liste übernommen wird? Hatten Sie im Verlauf der bisherigen Ausführungen bereits ähnliche Ansprüche, die wir überhört haben? Sie halten uns auf Trab, das ist gut. Den "echten" Programmieren unter Ihnen bieten wir jetzt wieder ein "Aha"-Erlebnis. Die Analogie auf ein *IF-THEN* Konstrukt. Die Nicht-Programmierer müssen bitte Verständnis dafür haben, daß wir den Programmierern immer mal ein Häppchen zuwerfen, damit sie am Ball bleiben und das erlernte Wissen anwenden können.

Unterhalb der Liste zum Einstellen der Events befindet sich ein Feld mit dem Titel **Bedingung**.

In dieses Feld können Bedingungen eingetragen werden, die entscheidend dafür sind, ob eine Nachricht gesendet wird oder nicht. Die erste Einschränkung erfährt eine Nachricht ja bereits durch das Ereignis selbst. Hier sagen wir ja beispielsweise *Wenn die linke Maustaste losgelassen wird, dann weise dem Objekt X einen Wert zu* . Mit einer zusätzlichen Bedingung schränken wir den Link noch weiter ein. Hier können wir vor der Ausführung des Links etwa den Zustand einer anderen Komponente überprüfen. Das könnte dann so klingen: *Wenn die linke Maustaste losgelassen wird, dann*

weise dem Objekt X einen Wert zu, aber nur wenn sich im Eingabefeld edt1 der numerische Wert 5 befindet.

Bezogen auf unser Beispiel müßte es heißen: *Wenn die linke Maustaste losgelassen wird, dann übertrage die Auswahl aus Ist_Pictures in edt_Selection, aber nur, wenn sie dort noch nicht vorhanden ist.* Hiermit werden doppelte Einträge vermieden.

Bei der Überprüfung einer Bedingung kann das Programm feststellen, ob eine Aussage wahr ist oder falsch. In unserem Falle wollen wir einen echten Vergleich zweier Komponenten oder deren Einträge durchführen. Wir bedienen uns hier wieder einer Parserfunktion. Am kleinen Fragezeichen neben dem Feld haben Sie vielleicht bereits erkannt, daß es sich auch hier wieder um ein Feld handelt, in dem Parserfunktionen Gültigkeit haben. Um die Formel komfortabel eingeben zu können, drücken Sie bitte auf die Schaltfläche *Zoom*, rechts neben dem Feld. Verwenden Sie die Schaltfläche mit dem Fragezeichen. Suchen Sie innerhalb der angebotenen Parserfunktionen nach `strpos` (*str* steht für String und *pos* steht für Position). Diese Funktion ist also dazu geeignet, die Position eines Teilstrings in einem anderen String festzustellen. Sie gibt eine Zahl zurück, welche die Position des ersten Buchstabens des Suchbegriffs innerhalb des zu durchsuchenden Textes an gibt. Wird der Text nicht gefunden, gibt sie den Wert 0 aus. `Strpos` benötigt zwei Parameter, nämlich den zu durchsuchenden Text und den Text nach dem gesucht werden soll. Die Funktion sieht also wie folgt aus:

```
strpos(text1, text2)
```

Dies bedeutet, daß die Funktion `strpos("Hallo Welt", "llo")` den numerischen Wert 3 zurückgeben würde, da der Teilstring "llo" im Suchtext an der dritten Stelle beginnt. Das Beispiel

```
strpos("MindMap ist stark", "schwach")
```

gibt selbstverständlich den Wert 0 oder besser *falsch* zurück. Was haben Sie denn gedacht?

In unserem Fall ist beides, der zu durchsuchende Text, sowie auch der gesuchte Text variabel. Daher können wir ihn auch nicht literal in die Formel eintragen. Anstelle der Texte kommen nun einfach die Namen der Komponenten, die diese Texte enthalten. Da wir eine Bedingung konzipieren, werden wir

noch einen Vergleichsoperator hinzufügen. Unsere Bedingung heißt ja *Wenn noch nicht vorhanden...*. Demzufolge sieht unsere Formel also folgendermaßen aus:

```
strpos(edt_Selection , lst_Pictures) = 0
```

Da wir schon einmal hier sind, betrachten wir noch die anderen Optionen, die mit Bedingungen zu tun haben. Schließen Sie bitte den Formeleditor. Im Feld der Bedingungen ist jetzt ein Teil der Formel zu sehen. Die *Zoom-Schaltfläche* haben wir bereits kennengelernt. Die Schaltfläche *Ohne Bedingung* rechts daneben ist etwas gefährlich. Sie löscht die Bedingung.

Bestätigen Sie nun den Dialog mit OK, um in die Übersicht der Links zu gelangen. Dort sehen Sie nun anstelle der ursprünglich einen Zeile, zwei Zeilen. Die IF-Bedingung *Wenn* ist dem Link vorangestellt.

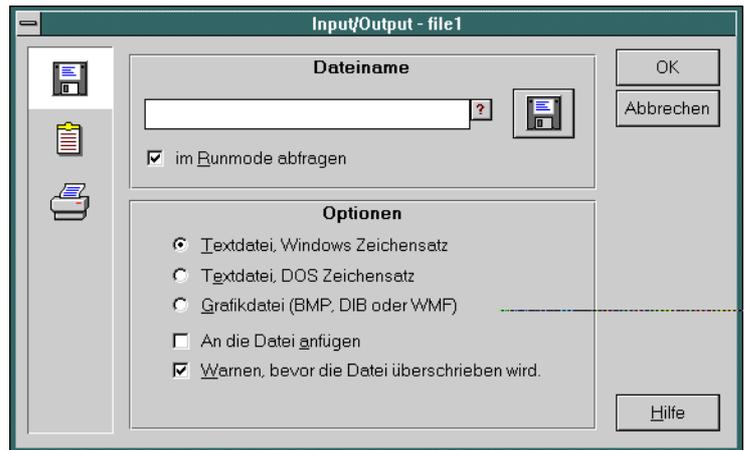
Bitte testen Sie nun unseren neuen Link. Er sollte nun die Einträge aus der Liste `lst_Pictures` nur dann in das Feld `edt_Selection` übernehmen, wenn diese dort noch nicht vorhanden sind. Klappt's? Na, prima! (s. Anwendung **TUT_11.MM**).

Input/Output Komponente

Jetzt fehlt noch eine Schaltfläche, die die Auswahl an Bild - dateien im Eingabefeld `edt_Selection` als Datei auf der Festplatte speichert. Bitte erzeugen Sie eine Schaltfläche mit dem Namen `btn_Save` und beschriften Sie sie mit *Save*. Diese Schaltfläche soll einen Link erhalten, der den Inhalt der Eingabefelds `edt_Selection` an eine Komponente überträgt, welche dazu in der Lage ist, eine Datei zu erzeugen.



Hierzu verwenden wir das Werkzeug **Input/Output**. Nach Auswahl mit der Maus erscheint ein Dialog, mittels dessen festgelegt wird, welcher "Kanal" verwendet werden soll. Es steht zur Auswahl **Datei**, was durch eine Diskette symbolisiert wird, **Clipboard** und **Drucker**. Wir entscheiden uns für *Datei*. Hat man *Datei* ausgewählt, sieht der Dialog wie folgt aus.



In dem oberen Feld läßt sich Pfad und Dateiname eintragen, der Ziel, oder Quelle des Inputs oder Outputs ist. Bitte setzen Sie solche Eingaben in Anführungszeichen. Das Disketten symbol auf der rechten Seite ermöglicht die interaktive Auswahl einer Datei. Wir markieren hier die Checkbox **im Runmode abfragen**, womit festgelegt wird, daß der Anwender im Run Mode nach der entsprechenden Datei gefragt wird. In diesem Falle wirkt die Eingabe im oberen Feld wie ein Default-Wert. Die Optionen im unteren Bereich legen fest, in welchem Dateiformat die Datei vorliegen wird. Im vorliegenden Falle ist es **Textdatei, Windows Zeichensatz**. Die Option **Textdatei, DOS Zeichensatz** erzeugt ebenfalls eine ASCII-Datei. Allerdings unterscheiden sich die beiden Formate hinsichtlich der Sonderzeichen. Wenn die Zielapplikation, also diejenige, welche eventuell die Daten weiterverarbeitet eine "Nicht-Windows"-Anwendung ist, wählen Sie bitte diese Option. Schließlich kann noch **Grafikdatei (BMP, DIB oder WMF)** für die Standard-Grafikformate von MS-Windows angewählt werden. Unterhalb der Radiobuttons gibt es noch zwei Checkboxes. Mit der einen wird festgelegt, ob die erzeugten Daten lediglich an die Zieldatei angehängt werden. Ist diese Option nicht ausgewählt, wird eine eventuell unter demselben Namen vorhandene Datei einfach überschrieben. Die andere Checkbox eröffnet die Möglichkeit, daß der Anwender vor dem Überschreiben einer bestehenden Datei gewarnt wird.

Nach Bestätigen des Dialoges kann die Komponente in den Bildschirm gebracht werden. Der Cursor hat sich verändert und wartet auf das Plazieren auf dem Bildschirm. Hier ist es

nicht erforderlich einen Bereich zu markieren. Sie müssen lediglich an einer geeigneten Stelle mit der Maus klicken. Die Größe der Komponente spielt überhaupt keine Rolle und schadet unserem empfindlichen Design überhaupt nicht, denn wir lassen es zur Laufzeit optisch verschwinden. Dann macht es sich nur durch seine Wirkung bemerkbar. Hierzu können Sie das Attribut *Effekt* auf "unsichtbar" einstellen. Das erreichen Sie über das Symbol für Effekte und die Einstellung, die durch das Kreuz symbolisiert wird. Weil wir ja aber ordnungsliebende Entwickler sind, suchen wir einen geeigneten Platz. Ein Dialog fordert uns auf einen Namen für die eben erstellte Komponente zu erfinden. Nennen Sie es `io_PicName`.

Jetzt muß nur noch die Anweisung erfolgen, zum richtigen Zeitpunkt tatsächlich den Inhalt des Eingabefelds zu speichern. Hierzu erzeugen wir einen Link für die dafür vorgesehene Schaltfläche `btn_Save`. Der Link soll bei dem Ereignis **Linke Maustaste losgelassen** die Nachricht **Drag&Drop** verschicken. Die Datenübertragung soll von `edt_Selection` nach `io_PicName` stattfinden. Das war's schon.



Ein ehemaliger Entwickler von MINDMAP hat einmal vorgeschlagen, man solle diesen Vorgang komplizierter gestalten, weil er sonst nicht genügend gewürdigt würde. Ein ehemaliger Entwickler. (Die letzten beiden Sätze sind vom Autor frei erfunden, der ehemalige Entwickler nannte ihn einen Lügner.)

Bitte probieren Sie diese Funktion sofort aus! Wählen Sie einige Dateinamen aus der Listbox und kopieren Sie sie in das dafür vorgesehene Eingabefeld. Drücken Sie dann die Schaltfläche `Save`. Ein Dialog fragt nach Dateinamen und Speicherort. Bitte entscheiden Sie sich für einen Pfad, an den Sie sich später auch noch erinnern können. Dann suchen Sie doch einmal die fragliche Datei mit dem Dateimanager auf der Festplatte. Mal ehrlich. Ist das nicht schön? Ist das nicht etwas ganz anderes als das Speichern irgendeines Dokumentes mit irgendeiner Textverarbeitung? Oder einer Tabellenkalkulation?

Spätestens jetzt dürfen Sie sich eine ausgedehnte Pause gönnen. Sie haben es sich verdient.

Jetzt setzen wir noch einen drauf. Vielleicht werden wir in unserer schwierigen Arbeit als Art-Director des öfteren

gestört. Kann ja sein. Für diesen Fall möchten wir unter Umständen eine begonnene Auswahl an Bildern zu einem späteren Zeitpunkt wieder fortsetzen. Dazu wäre es erforderlich, die einmal abgelegte Datei wieder zu laden. Wählen Sie die Schaltfläche `btn_Save` an und erzeugen Sie mit `F2` eine weitere Schaltfläche. Nennen Sie die Schaltfläche `btn_Load` und beschriften Sie diese mit *Load*. Der Link wurde ja beim Duplizieren mitkopiert. Wir müssen ihn nur noch so ändern, daß die Richtung der Datenübertragung beim Drag&Drop von `io_PicNames` nach `edt_Selection`. Mein Gott, lesen sie sich den letzten Satz noch einmal langsam durch, damit Sie nichts falsch machen. Aber das ist wirklich alles!

Probieren Sie die Anwendung doch einmal aus. Schalten Sie dazu den Link wieder an, der auf der Liste `lst_Pictures` liegt, und den wir abgeschaltet hatten, damit die Tests der anderen Links etwas flüssiger vonstatten gehen, sofern Sie das nicht längst getan haben (s. Anwendung **TUT_12.MM**).

Weil alles so gut geklappt hat, und weil Sie wirklich immer am Ball geblieben sind, folgen hier drei Aufgaben, die zu lösen, Sie nun im Stande sein müßten.

Übungsaufgabe 1

Bitte installieren Sie eine weitere Schaltfläche und eine weitere Input/Output Komponente, um die Daten der Komponente `edt_Selection` in die Zwischenablage (Clipboard) zu übernehmen. Testen Sie dann bitte, ob Sie den in einem Texteditor ablegen können.

Übungsaufgabe 2

Wenn die Applikation gestartet wird, dann soll der Inhalt des Eingabefeldes `edt_Selection` leer sein, damit eine neue Auswahl getroffen werden kann. Das Stichwort ist **Wertzuweisung** und der Wert `""` (also zwei Anführungszeichen mit nichts dazwischen). Das Ereignis dürfte auch klar sein.

Übungsaufgabe 3

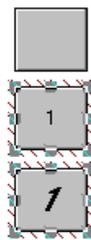
Bitte fügen Sie noch eine Schaltfläche hinzu, welche die Einträge in `edt_Selection` komplett löscht, damit schnell eine neue Auswahl erzeugt werden kann. Dieses Zubehör ist ein sehr naher Verwandter der Aufgabe 2.

Ein "Taschenrechner"

Der Umgang mit Variablen

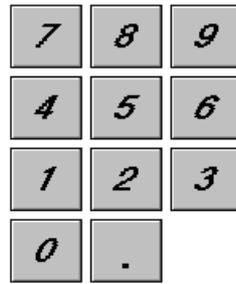
In der folgenden Übung kommen wieder einige neue, und für eingefleischte Programmierer andere Erkenntnisse hinzu. Auch wenn es Ihnen trivial erscheint, und auch wenn Sie bereits über leistungsfähigere Versionen dieser Spezies verfügen, lassen Sie uns gemeinsam einen Taschenrechner "bauen". Um etwas Spannung zu erzeugen, wird das auch ein etwas anderer Rechner als die, welche Sie vielleicht aus dem Zubehör von MS-Windows kennen.

Öffnen Sie bitte eine neue Anwendung in MINDMAP. Wie Sie ja wissen, besteht ein Rechner vor allem aus Tasten. Diese Tasten tragen Nummern und unsere Komponenten sollen ja auch Namen tragen. Hier machen wir uns die Eigenschaft von MINDMAP zunutze, Komponenten automatisch zu benennen.



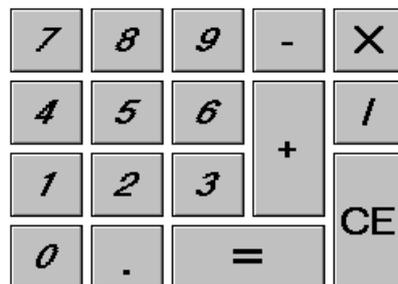
Erzeugen Sie eine quadratische Schaltfläche, etwa mit der Kantenlänge von einem halben Zoll. Da es die erste Schaltfläche in dieser Anwendung ist, schlägt MINDMAP als Komponentenamen einen Namen vor, in dem die Ziffer 1 vorkommt. Bevor wir nun diese Schaltfläche kopieren, betiteln wir sie mit 1 und passen Schriftart und -größe dergestalt an, daß Sie die Aufschrift komfortabel lesen

können. Wenn Sie nun mit dieser einen Schaltfläche fertig sind, verwenden Sie die Taste F2, um die Taste zehnmal zu duplizieren. Beachten Sie die für Rechner typische Anordnung, wenn Sie die Schaltflächen nun deren Namen entsprechend ausrichten.



Benennen Sie nun die Schaltfläche btn10 in btn0 um, die Schaltfläche btn11 in btnPoint.

Erzeugen Sie Schaltflächen für die vier Grundrechenarten und beschriften Sie diese entsprechend. Benennen Sie die Tasten, damit Sie diese später auch identifizieren können (z.B. btnMutiply, btnPlus...). Fügen Sie noch eine Taste für das Ergebnis ein und beschriften Sie diese mit "="; der Name könnte btnEqual sein. Desweiteren erzeugen Sie eine Schaltfläche für die Taste CE. Richten Sie auch diese Tasten Ihrem werten Geschmack entsprechend aus. Wenn Sie ihr Geschmack bezüglich Taschenrechnerdesign im Stich läßt, können Sie auch dieser Empfehlung folgen:



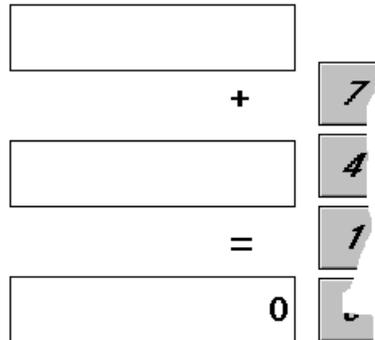
Jetzt benötigen wir noch derer drei Eingabefelder, welche die Operanden und das Ergebnis abbilden werden. Die Namensvorschläge sind edtOp1, edtOp2 und edtResult. Diese Felder sollen zwar Zahlen aufnehmen, da wir aber deren Inhalt aus einzelnen Zahlen verketteten werden, soll das Format „String“ sein. Das Ergebnisfeld allerdings kann als **Klasse Numerisch**, **Typ Gleitkommawerte** und mit dem **Format *0.*** formatiert werden. Bestimmen Sie jetzt noch über die objekt-spezifischen Attribute der Eingabefelder, daß **Eingabe Ausrichtung** und **Anzeige Ausrichtung** jeweils auf *rechts* stehen. Unterbinden Sie bitte auch die Möglichkeit der Eingabe in das Ergebnisfeld durch den Benutzer (Option **editierbar** deaktivieren).



Wie Sie vielleicht bemerkt haben, wurden im Laufe des Tutorials Schritt für Schritt die Hilfen in Form von Bildern zurückgenommen. Der Grund ist der, den Sie bereits vermuten: Börsartigkeit. Wir bei MINDMAP amüsieren

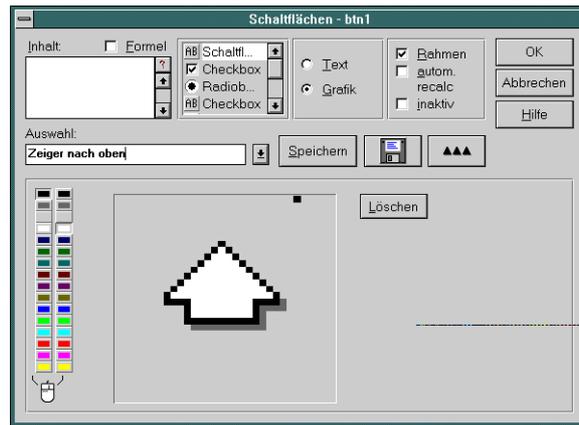
uns köstlich bei der Vorstellung, wie Sie verzweifelt versuchen den entsprechenden Dialog zu den Angaben im Tutorial zu finden.

Zwischen die beiden Eingabefelder für die Operanden setzen Sie nun bitte ein Textfeld mit dem Namen txOp und eines über das Ergebnisfeld mit dem Namen txEqual. Beschriften Sie dieses Textfeld mit dem Equal-Zeichen. Richten Sie alles hübsch aus, damit es vielleicht so aussieht:

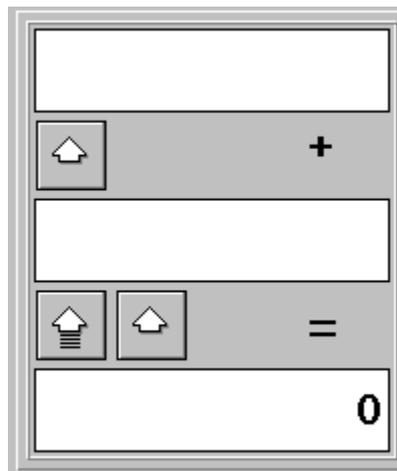


Der Anwender dieses Rechners soll nun eine Zahl in das erste Eingabefeld eingeben können, dann die gewünschte Rechenart wählen, den zweiten Wert in das nächste Eingabefeld eingeben und in dem dritten Feld das Ergebnis erhalten. Um Zwischenergebnisse verarbeiten zu können, soll es außerdem möglich sein, den zweiten Operanden in das Feld des ersten zu kopieren und den Inhalt des Ergebnisfeldes in das Feld des ersten oder zweiten Operanden zu übertragen.

Hierzu erzeugen wir eine kleine Schaltfläche, die zwischen die Felder paßt. Diese Taste soll nun keinen Text als Aufschrift erhalten, sondern ein kleines Piktogramm. Dies können wir über die Einstellungen bei den objekt-spezifischen Attributen der Schaltfläche realisieren. Dieser Dialog sieht wie folgt aus:



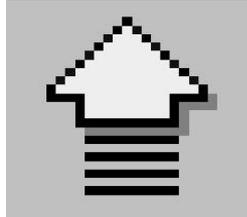
In dem Dialog, in dem Sie bisher die Beschriftung der Schaltflächen durchgeführt haben, wählen Sie bei der mit tleren Option statt **Text** bitte **Grafik** aus. Suchen Sie nun in der **Auswahl**-Liste der angebotenen Bitmaps das mit dem Namen **Zeiger nach oben**. In dem Editor erscheint nun ein kleiner Pfeil. Bestätigen Sie diesen Dialog mit OK und duplizieren Sie die Schaltfläche zweimal. Ordnen Sie die Schaltflächen etwa so an, wie in folgender Abbildung ersichtlich.



Zwei der Schaltflächen haben nun ein korrektes Piktogramm. Sie sollen den Wert des in Frage kommenden Eingabefeldes um eins weiter hoch kopieren. Eine der unteren Schaltflächen soll allerdings den Wert um zwei Stationen hoch kopieren. Deshalb werden wir das Piktogramm verändern.

Öffnen Sie bitte hierzu den Dialog zur Einstellung der objekt-spezifischen Attribute dieser Schaltfläche. Verwenden Sie die Taste mit den drei Pfeilen nach unten, um den Editor erneut aufzurufen. Sie finden den Pfeil vor, den wir nun ein wenig verändern. Bitte beachten Sie die Farbauswahl auf der linken Seite. Die linke Spalte bezieht sich auf die Farbe, die Sie mit der linken Maustaste verwenden, die rechte stellt diejenige Farbe dar, die auf der rechten Maustaste liegt. Zum Löschen eines versehentlich gemalten Pixels verwenden Sie einfach die Ursprungsfarbe der Stelle, die Sie korrigieren möchten.

Ziehen Sie nun in der Farbe Ihrer Wahl einige Linien unter dem Pfeil, vielleicht so:



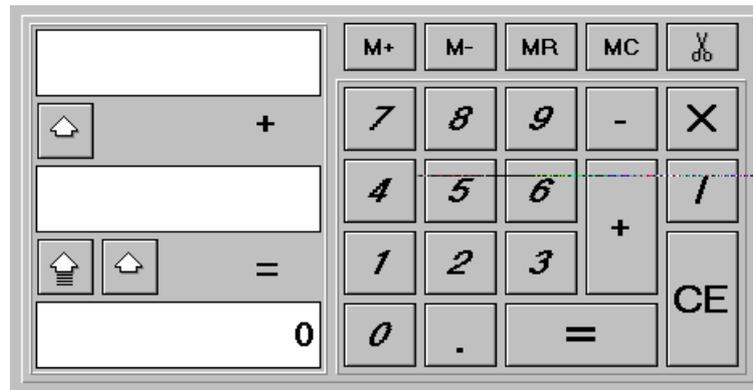
Durch diese Linien soll der "Express-Character" dieser Schaltfläche verdeutlicht werden. Benennen Sie bitte die normalen Schaltflächen mit btnUp1, bzw. btnUp2 und die andere mit btnExpressUp.



Auch hier, wie eigentlich überall, gelten unsere Vorschläge als durchaus bevormundend. Wenn Sie Ihre kreativen Auswüchse nun hemmungslos austoben... Wir wissen nichts davon. Schämen Sie sich!

Wie Sie sicher bemerkt haben, gibt es bereits eine Vielzahl an vorgefertigten Piktogrammen in der entsprechenden Liste. Wenn nun eine Ihrer Kreationen besonders gelungen ist, oder aus anderen, uns unbekanntem Gründen aufgehoben werden soll, dann können Sie diese der Auswahlliste hinzufügen. Wenn Sie also nun den Pfeil mit den dynamisierenden Linien vor sich haben, dann geben Sie bitte in das Feld unter dem Label **Auswahl** (treffen Sie nicht die Aufklapptaste der Liste) einen Namen ein, beispielsweise *Fast Up* oder *Up to Top*. Drücken Sie dann auf die Schaltfläche *Speichern*. Nun steht dieses Piktogramm auch für andere Schaltflächen zur Verfügung.

Auf folgender Abbildung sehen Sie wie der Rechner aussehen könnte.



Bitte erzeugen Sie die noch fehlenden Schaltflächen. Verwenden Sie auch Rechtecke als Rahmen, um verschiedene Bedienelemente voneinander abzugrenzen. Außerdem benötigen wir wenigstens zwei Rechtecke für weitere Zwecke. Bedienen Sie sich der Schriftfarben, um die Zahlentasten von den Funktionstasten besser unterscheidbar zu machen. Gönnen Sie den Betrachtern einen leicht lesbaren Font für die jeweilige Anzeige.

Gewissermaßen zum Warmwerden, beginnen wir nun, die Schaltflächen mit Leben zu füllen. Nehmen wir uns zunächst die Rechenoperatoren vor. Diese sollten bei Druck auf das entsprechende Symbol das Textfeld `txOp` mittels **Wertzuweisung** mit dem angewählten Operator beschriften. Das können Sie schon einmal ausprobieren. Bei Druck auf eines der Rechensymbole ändert sich das Textfeld entsprechend. Klappt's? Der Link, der auf den Operator-Tasten liegt, sieht so aus:

↑☞ Linke Maustaste losgelassen: Wertzuweisung: txOp << "+"

respektive "/", "+", "-".

Die Schaltfläche "=" soll nun in das Ergebnisfeld das Ergebnis aus den beiden Operanden und dem Operator abbilden. Die Links hierzu enthalten eine **Bedingung** und sehen folgendermaßen aus.

Wenn: txOp = "x"



Linke Maustaste losgelassen: Wertzuweisung:
edtResult << val(edtOp1)*val(edtOp2)

Die Links für die anderen Rechenarten sehen analog aus. Auf der Taste liegen also vier Links, die sich eigentlich nur durch den entsprechenden Operator unterscheiden.

Jetzt benötigen die Schaltflächen im Zahlenblock noch Links, die bei Betätigen auch einen Eintrag in das entsprechende Eingabefeld machen. Aber in welches Feld? Wir haben durch unsere eigenwillige Konstruktion ja zwei mögliche Eingabefelder. Hierzu gibt es verschiedene Lösungsansätze, fast wie überall in der Programmiererei. Man könnte beispielsweise das Feld in irgendeiner Form vor der Eingabe markieren. Oder man könnte die Eingabe davon abhängig machen, ob bereits ein Operator gedrückt wurde, oder ob bereits die Equal Taste verwendet worden ist. Wir verfolgen in diesem Beispiel einen anderen Ansatz. Unsere Maus hat mindestens zwei Tasten. Die linke Taste wird die Werte in das obere Eingabefeld transportieren, und die rechte wird das gleiche für das untere Eingabefeld tun.

Erzeugen Sie also zunächst einen Link für die Taste 1. Legen Sie fest, daß wenn die linke Maustaste gedrückt wird eine Wertzuweisung an das Feld edtOp1 stattfindet. Die Formel hierfür ist :

edtOp1 + "1"

Also, das Eingabefeld edtOp1 soll den Inhalt seiner selbst, sowie eine angehängte "1" bekommen (z. B. "1" + "1" = "11").

Das ist nun keineswegs ein Witz, sondern dem Umstand zu verdanken, daß wir das Eingabefeld als String formatiert haben, und es insofern keine mathematische Relevanz hat.

Vervielfältigen Sie den Link, indem Sie im Dialog mit der Linkliste diesen Link markieren und die Schaltflächen *Kopieren* und *Einfügen* betätigen. Ändern Sie den zweiten Link bitte so, daß er erstens auf das Ereignis rechte Maustaste anspricht und zweitens die Eingabe in das Feld edtOp2 vorgenommen wird. Bitte denken Sie auch daran, die Formel entsprechend anzupassen.

Im Übersichtsdialog der Links markieren Sie nun bitte beide Links unter Zuhilfenahme der *Shift*- oder *Ctrl*-Taste, und kopieren Sie beide Links. Fügen Sie nun diese Links bei jeder der Zahlen-Schaltflächen und bei der Komma-Schaltfläche ein und verändern Sie sie entsprechend. Danach benötigen Sie dringend eine Pause.

Wo wir schon bei den Schaltflächen auf der rechten Seite sind, erzeugen Sie doch bitte Links für die Taste *CE*, die die Felder *edtOp1* und *edtOp2* löscht, indem ein Leerstring ("") eingefügt wird. Das Ergebnisfeld *edtResult* soll gelöscht werden, indem die Ziffer *0* eingetragen wird.

Nun werden die Pfeiltasten auf der linken Seite mit Leben gefüllt. Die obere Pfeiltaste soll auf Druck den Wert des Felds *edtOp2* in das Feld *edtOp1* schreiben und seinen eigenen Inhalt löschen. Die untere Taste mit dem gleichen Piktogramm soll analog dasselbe für die Felder *edtResult* und *edtOp2* machen. Die *Expresstaste* schließlich soll den Wert im Feld *edtResult* in das oberste Feld kopieren und seinen eigenen Wert löschen.

Sie erreichen das durch Wertzuweisungen. Bei der Übertragung des Inhaltes des Ergebnisfeldes nimmt *MINDMAP* selbsttätig die Umwandlung von einer Zahl in einen Text vor.

Der Rechner sollte bei einem Testlauf in der Zwischenzeit zu folgendem in der Lage sein:

- ▶ Mit der linken Maustaste Werte in dem oberen, und mit der rechten Maustaste Werte in dem unteren Feld abzulegen.
- ▶ Bei Druck auf eine der Operator-Tasten den entsprechenden Operator in dem dafür vorgesehenen Textfeld abzubilden.
- ▶ Die *Equal*-Taste kann das Resultat aus der Berechnung von Operanden und Operator im Ergebnisfeld anzeigen.
- ▶ Die *CE*-Taste löscht alle Einträge.
- ▶ Die "Transporttasten" bewegen die Einträge in den Feldern jeweils nach oben und löschen die ursprüngliche Position.

Wenn das Bild Ihres Rechners in etwa unserem Layout entspricht, dann gibt es da weitere Schaltflächen, die noch ohne Funktion sind. Es handelt sich hier um die, für Taschenrechner typischen Memory-Tasten, *M+* und *M-*, um

dem Speicher den Betrag im Ergebnisfeld hinzuzufügen, *MR* für Memory-Recall, um den Wert im Speicher in eines der beiden Operanden-Felder einzufügen und *MC*, um den Speicher zu löschen.

Wohin allerdings mit den jeweiligen Zahlen? Um diese Werte zu speichern, benötigen wir einen geeigneten Ort. Die Programmierer unter Ihnen werden jetzt selbstverständlich sofort an Variablen denken. Variablen sind Bereiche im RAM des Computers, die einen bestimmten Namen tragen. Diese Variablen nehmen dann Werte verschiedenster Art auf, und sind über den vergebenen Namen referenzierbar. Da es sich bei MINDMAP ja nicht um eine Programmiersprache handelt, fehlt den Programmierern nun die Möglichkeit, die entsprechenden Zeilen einzugeben, die üblicherweise zur Deklaration von Variablen erforderlich sind.

Wir verwenden als "Container" für Werte in MINDMAP einfach eine geeignete Komponente. Es existiert keine spezielle "Variablenkomponente". Es besteht die Möglichkeit, spezielle Komponenten für die Aufnahme von Werten anzulegen, oder Bestandteile der entworfenen Oberfläche zu verwenden.

Auch wenn Sie bereits plazierte Komponenten zu diesem Zwecke nutzen können, so raten wir Ihnen dringend, dies nicht zu tun. Aus Gründen von Lesbarkeit, Wart- und Pflögarkeit einer Anwendung ist es besser, eigene Komponenten für Variablen anzulegen, und sie mit einem "sprechenden" Namen zu versehen. Ganz besonders dankbar wäre der Kollege, der Ihre Anwendung später warten muß, wenn Sie an dieser Komponente die Möglichkeit genutzt hätten, eine Beschreibung anzuhängen. Das Attribut **Beschreibung** existiert für sämtlichen Komponenten. Machen Sie nicht zu sparsam davon Gebrauch! Vielleicht müssen Sie Ihre Anwendung später selber warten. Oder gehören Sie zu den Glücklichen, die sich noch Jahre später an jeden Link erinnern, den Sie einmal plazierte haben?

Wir entscheiden uns also dafür, eine Komponente zu plazieren, die zur Zwischenspeicherung von Werten dienen soll, welches wir aber zur Laufzeit nicht auf dem Bildschirm sehen wollen, und deshalb später unsichtbar gemacht wird. Wir haben die Wahl zwischen Komponenten, die immer ihren Wert anzeigen (z. B. Eingabefeld oder Text Komponente), und Komponenten, die zwar einen Wert annehmen können, ihn aber nicht anzeigen, wie beispielsweise Rechtecke.

Ende der langen Rede. Platzieren Sie nun bitte ein Rechteck, und benennen Sie es **Memory**. Selbstverständlich ist hier auch ein anderer Name möglich, aber bitte keine Leerschritte oder Sonderzeichen. Damit wir es nicht vergessen, gehen wir gleich zum Attribut *Effekt* dieses Rechtecks und stellen dieses so ein, daß es zur Laufzeit der Anwendung unsichtbar ist.

Diese "Variable" mit dem Namen **Memory** muß nun noch initialisiert werden. Bitte erzeugen Sie für diese Komponente einen Link, der ihr selbst beim Programmstart die Zahl *0* zuweist. Also, beim **Starten der Anwendung** soll eine **Wertzuweisung** mit dem Wert *0* an die Komponente selbst stattfinden. Auf diese Weise wird festgelegt, daß diese "Variable" Zahlen entgegennimmt. Außerdem können wir nun sicher sein, daß diese Variable beim Programmstart immer den Wert *0* hat.

Die Links für die entsprechenden Schaltflächen können Sie nun alleine schaffen. So hat die Schaltfläche *M+* den Link:

↑	Linke Maustaste losgelassen: Wertzuweisung: Memory << Memory+edtResult
---	--

Belegen Sie bitte analog hierzu die Schaltfläche *M-*.

Die *MR*-Schaltfläche soll sich wieder ähnlich den Zahlentasten verhalten. Die linke Maustaste soll dem oberen Eingabefeld den Wert von **Memory** zuweisen, und die rechte Maustaste soll dies für das untere Eingabefeld tun. Die *MC*-Schaltfläche setzt den Wert der Komponente **Memory** wieder auf *0* zurück.

Schließlich muß nun noch die geheimnisvolle Schaltfläche mit der Schere aktiviert werden. Natürlich gilt auch hier die Gestaltung der Schaltfläche wieder als zwingender Vorschlag. Sie soll dazu dienen, den Wert des Ergebnisfeldes in die MS-Windows Zwischenablage zu befördern. Platzieren Sie hierzu zunächst eine Input/Output Komponente in Ihrer Anwendung. Wählen Sie das Symbol für das **Clipboard** und **Text** in der darauf erscheinenden Auswahl. Nennen Sie die Komponente *io_cli* und erzeugen Sie einen Link, der diese Komponente bei Programmstart unsichtbar macht. Dies geschieht über die Nachricht **Attributänderung**.

Erzeugen Sie nun einen Link für die Schaltfläche mit der Schere als Symbol, mittels dessen ein **Drag&Drop** vom Er-

gebnisfeld auf die I/O- Komponente stattfindet. Jetzt können Sie das Ergebnis Ihrer Berechnungen in die Zwischenablage übernehmen, und in jeder anderen Windows-Anwendung weiterverarbeiten.

Gehen Sie jetzt bitte in den Run Mode und testen Sie Ihren neuen, etwas eigenwilligen Taschenrechner. Na, das ist doch super! Auch wenn Sie diesen Taschenrechner nicht dauernd benutzen, was wir uns überhaupt nicht vorstellen können, dann haben Sie doch sicher viel dabei gelernt.

Diese Anwendung liegt Ihnen unter dem Namen **Taschenrechner.mm** vor

Ein Video-Recorder

In diesem Kapitel kommen fast nur Dinge vor, die Sie schon kennengelernt haben. Einige der bisher nicht erwähnten Komponenten oder Links können jedoch noch von Interesse sein, und schließlich haben Sie nach diesem Kapitel auch einen feinen "Fernseh apparat", mit dem Sie komfortabel Video-Clips ansehen können.

Der Bildschirm beherbergt eine Vielzahl von Komponenten mit einer Menge Links. Wahrscheinlich werden Sie alle Komponenten leicht erkennen und nachbilden können. Ein erklärendes Bild hilft Ihnen später dabei.

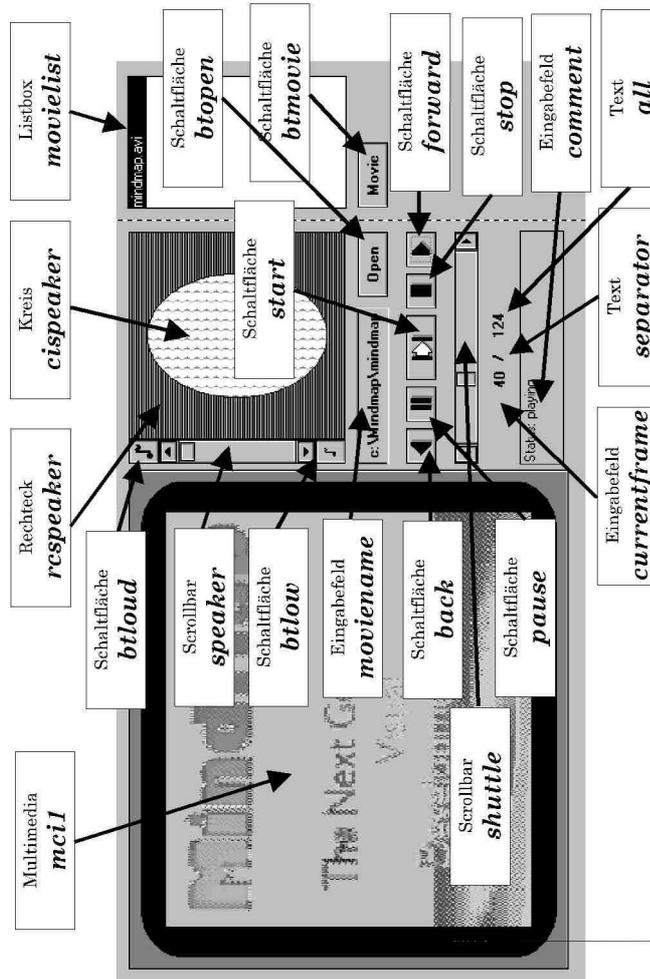


Zunächst allerdings installieren wir eine Multimedia Komponente. Werden Sie nach einer Datei gefragt, für die diese Komponente da sein soll, dann beantworten Sie den Dialog mit *Abbrechen*. Wir werden später eine Einrichtung schaffen, die uns die Auswahl für die angebotenen Video-Clips ermöglicht.

Auf dem folgenden Bild läßt sich erkennen, welcher Art die Komponenten sind und wie sie heißen. Auf den nach folgenden Seiten finden Sie zu den angegebenen Namen die relevanten Attribute und die Links. Diese Bauanleitung ist also eine andere als die bisherigen, aber Sie haben im Laufe der letzten Kapitel so entscheidend viel Leistung gezeigt, daß man Ihnen diese verdichtete Form eines Bau plans ohne weiteres zumuten kann.

In der Grafik sind nur die Elemente mit einer Legende versehen, die auch eine Funktion besitzen. Sie stellen fest, daß diverse Rahmen existieren, die der Kosmetik dienen. Da wir Ihnen hier keinerlei Vorschriften machen möchten, auch nicht, was die Aufteilung ihres Bildschirms angeht, wird auf diese Komponenten kein Bezug genommen. Lassen Sie also ruhig Ihrer Kreativität ungezügelt Lauf.

Übersicht



Listbox "movielist"



Listbox; Dateiliste: Dateinamen



	Anwendung gestartet: movielist: Verzeichnis "c:\\MINDMAP*.avi"
	Mausbewegung in das Objekt: Wertzuweisung: comment << "Wählen Sie eine Video-Datei und bestätigen Sie mit OK."
	Mausbewegung aus dem Objekt: Wertzuweisung: comment << "Status: "+mciGetMode (mci1)
	Cursorwechsel : Drag&Drop movielist -> movienamen
	Doppelclick : btmovie: Effekt ändern in ■
	Doppelclick : btopen: Effekt ändern in ✗
	Doppelclick : movielist: Effekt ändern in ✗
	Doppelclick: mci1 -> open new file movienamen



Doppelclick: Wertzuweisung: all << mciGetLength(mci1)

Kommentar

Der erste Link, der auf **Anwendung gestartet** reagiert, muß an das Verzeichnis angepaßt werden, in dem sich Ihre AVI-Video-Clips befinden. In Ihrem MINDMAP-Verzeichnis müßte sich mindestens eine AVI-Datei befinden.

Die Links mit den Ereignissen **Mausbewegung in/ aus dem Objekt** sind für uns neu. Sie reagieren darauf, wenn der Mauszeiger sich über die betreffende Komponente bewegt, oder sie wieder verläßt. In unserem Beispiel verwenden wir diese praktische Funktion, um eine Erläuterung zu dieser Komponente in das Eingabefeld comment zu schreiben. Der zweite dieser beiden Links verwendet eine Funktion, eine sog. Parser-Funktion, mit welcher der Status der im Klammern - paar genannten Multimedia Komponente abgefragt werden kann. Diese Funktion liefert einen Wert wie "playing" und "stopped" zurück.

Der Link mit der Nachricht Drag&Drop weist dem angegebenen Eingabefeld moviename, den Inhalt der ausgewählten Zeile der Listbox movielist zu. Eine Wertzuweisung würde übrigens an dieser Stelle nicht befriedigend funktionieren, da Sie lediglich den Dateinamen, nicht jedoch den gesamten Pfad überträgt.

Diese Listbox liegt eigentlich direkt auf unserem stilisierten Lautsprecher. Sie wird erst sichtbar, wenn die Schaltfläche btmovie gedrückt wird. Aus Gründen der Übersichtlichkeit befindet sie sich im Bild neben dem Lautsprecher (der übrigens zu rein garnichts gut ist...). Die drei Links, die die Effekte steuern, schalten jeweils sichtbare Komponenten aus, wenn dies sinnvoll ist und umgekehrt.

In dem Link, der ein Kommando an die Multimedia Komponente sendet, muß in der Liste mit den zur Verfügung stehenden Kommandos *Open New File* angewählt werden. Der Inhalt des Formelfeldes wird ohne Anführungsstriche angegeben. Der Dateiname resultiert aus der Selektion in der Listbox moviename.

Der letzte Link weist dem Textfeld all die Anzahl Frames zu, aus denen das jeweils angewählte Video besteht. In all wird also die Länge des ausgewählten Videos angezeigt.

Schaltfläche "btmovie"



MS Sans Serif, 10 point, Bold



	Anwendung gestartet: movielist: Effekt ändern in ✗
	Anwendung gestartet: btopen: Effekt ändern in ✗
	Anwendung gestartet: btmovie: Effekt ändern in ■
	Linke Maustaste losgelassen: movielist: Effekt ändern in ■
	Linke Maustaste losgelassen: btopen: Effekt ändern in ■
	Linke Maustaste losgelassen: btmovie: Effekt ändern in ✗
	Mausbewegung in das Objekt: Wertzuweisung: comment << "Öffnen die Video-Liste"

 Mausbewegung aus dem Objekt: Wertzuweisung: comment
<< "Status: "+mciGetMode (mci1)

Kommentar

Diese Schaltfläche dient eigentlich nur dazu, Dinge sichtbar und unsichtbar zu machen. Die durch Betätigen dieser Schaltfläche erscheinenden Komponenten `movielist` und `btopen` haben dann "richtige" Funktionalität. Die beiden Links am Ende dienen wieder einer Hilfsfunktion, die im Kommentar zur Listbox bereits beschrieben wurde. Diese Schaltfläche befindet sich im eigentlichen Leben genau über der Schaltfläche `btopen`. Hier wurde sie, wie auch die Listbox, aus Gründen der Übersichtlichkeit zur Seite geschoben.

Schaltfläche "btopen"



MS Sans Serif, 10 point, Bold



 Linke Maustaste losgelassen: btopen: Effekt ändern in ✗

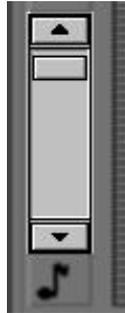
 Linke Maustaste losgelassen: movielist: Effekt ändern in ✗

	Linke Maustaste losgelassen: btmovie: Effekt ändern in ■
	Linke Maustaste losgelassen: mci1 -> open new file moviename
	Linke Maustaste losgelassen: Wertzuweisung: all << mciGetLength(mci1)
	Mausbewegung in das Objekt : Wertzuweisung: comment << "Öffnen des ausgewählten Videos"
	Mausbewegung aus dem Objekt : Wertzuweisung: comment << "Status: "+mciGetMode (mci1)

Kommentar

Diese Schaltfläche erscheint alternierend mit der Schaltfläche Movie. Sie erledigt die gleiche Funktion wie der Doppelklick auf einen Eintrag in der Listbox movielist. Die Doppelklick-Funktion der Listbox ist allerdings nur für die ganz schnellen unter uns gedacht. Diese Schaltfläche führt die Hauptfunktion etwas gemächlicher aus, und muß schon aus ergonomischen Gründen vorhanden sein. Die einzelnen Links sind bereits im Kommentar zur Listbox erklärt. Deshalb verzichten wir hier auf wiederholende Äußerungen.

Scrollbar "speaker"



- ▶ **Scrollbar Typ:** *Vertikal mit Pfeilen*
- ▶ **Unterer Wert:** *0*
- ▶ **Oberer Wert:** *1000*
- ▶ **Zeilenvorschub:** *1*
- ▶ **Seitenvorschub:** *100*



Scrollbar Änderung: mci1 -> set volume to speaker

Kommentar

Hier wird im ersten Link wieder ein Kommando an die Multi-media Komponente gesendet. Es setzt den Lautstärkewert für das Video auf den mit der Scrollbar eingestellten Wert. In den Attributen haben wir festgelegt, daß Werte zwischen 1 und 1000 zur Verfügung stehen. Auf die beiden letzten Links brauchen wir nun nicht mehr näher eingehen.

Schaltfläche "btlow" und "btloud"



Einstellung **Bitmap**; Zeichnen des Symbols Note im Bitmap-Editor

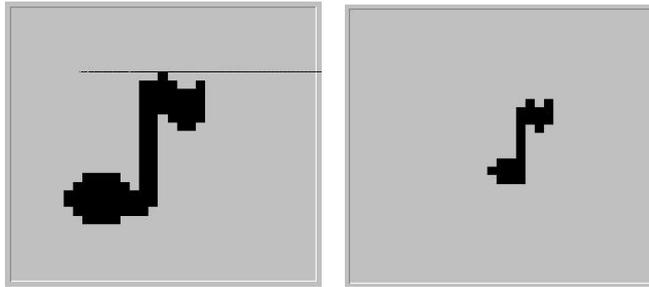


Keine

Kommentar

Diese beiden Schaltflächen haben keine Funktion. Sie sollen hier lediglich mit der kleinen und der großen Note anzeigen, in welche Richtung man den Schieber der Scrollbar bewegen muß. Zwar wurde ja versprochen, daß wir Komponenten, die keine Funktion haben, nicht besprechen. Komponenten aber, in deren Natur es liegt, eine Funktion zu haben und dennoch ohne eine solche bleiben, sind allerdings verdächtig, und werden deshalb hier trotzdem erwähnt.

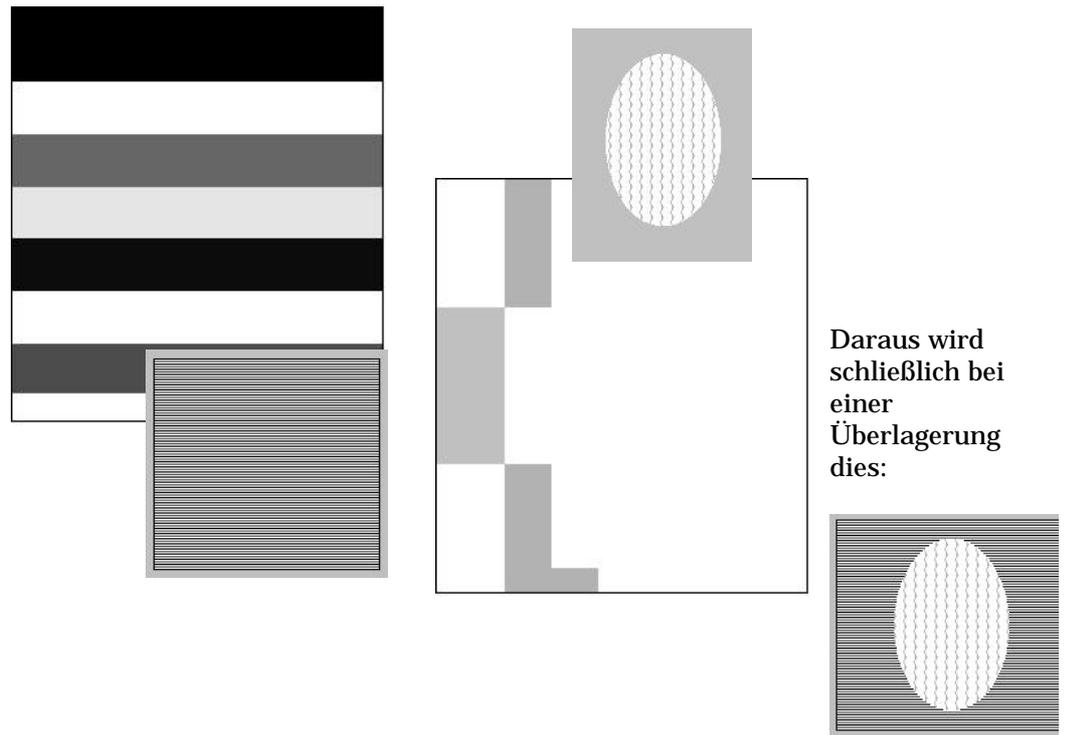
Diese Schaltflächen stellen eine prächtige Übung im Umgang mit dem Bitmap-Editor der Befehlsschaltflächen dar. Diesen finden Sie bei den objekt-spezifischen Attributen, wenn dort **Bitmap** eingestellt ist. In unserem Falle sehen sie so aus:



Bitte verstehen Sie richtig: wir finden diese Grafiken auch nicht sehr gelungen, aber wir sind ja auch bemüht, Sie nicht zu entmutigen.

Rechteck "rcspeaker" und Kreis "cisppeaker"

Für das Rechteck wurde ein Füllmuster kreiert, das hart an das Design der 50er Jahre erinnert. Kennen Sie noch die stoffbespannten Radios mit den Holzwangen und den elfenbeinfarbenen Tasten? Und den sogenannten magischen Augen, die die Wärme der Röhren und den Empfangspegel anzeigten? Verwenden Sie doch im Füllmuster-Generator einzelne Streifen in den wildesten Farben, sicher kommt ein ähnliches geschmackloses Muster heraus, wie in unserem Beispiel. Leider können die Farben hier nicht in ihrer ganzen Grausamkeit wiedergegeben werden. Gehen Sie bitte ähnlich wie oben beschrieben mit dem sich auf dem Rechteck befindlichen Oval vor.



Eingabefeld "moviename"



Farbe: grau



MS Sans Serif, 10 Points, Bold



Nicht **editierbar**, Rahmen **Intended**



Keine

Kommentar

Dieses Feld erhält über einen Link mit Nachricht **Drag&Drop** Daten aus der Listbox. Bei diesen Daten handelt es sich um Pfad und Dateinamen der Video-Datei, welche der Ladefunktion der Schaltfläche `btopen` zur Verfügung gestellt werden. Links existieren keine. Wie bereits bei der Beschreibung der Listbox erwähnt, reicht eine Wertzuweisung an dieser Stelle nicht aus, um den vollständigen Pfad zu übertragen. Die tatsächliche Information in der Listbox beschränkt sich nicht alleine auf den sichtbaren Eintrag, sondern enthält vielmehr auch Laufwerks- und Pfadbezeichnungen.

Schaltfläche "back"



Objekt-spezifische Attribute: **Bitmap**, Bitmap: **Pfeil nach links**



Linke Maustaste losgelassen: mci1 -> play reverse



Mausbewegung in das Objekt: Wertzuweisung: comment
<< "Rückwärts abspielen des Videos"



Mausbewegung aus dem Objekt: Wertzuweisung: comment
<< "Status: "+mciGetMode (mci1)

Kommentar

Diese Schaltfläche sendet einen MCI-String an die Multimedia Komponente, welcher veranlaßt, daß das Video von der momentanen Stelle aus rückwärts gespielt wird. Wie erfolgreich dieses Unterfangen schließlich ist, hängt ein wenig von dem Kompressionsverfahren ab, mit dem das Video bearbeitet wurde. Meist funktioniert der Ton rückwärts ganz passabel, die Bilder springen aber sehr. Das hängt damit zusammen, daß bei den hier üblichen Kompressionsverfahren nur in bestimmten Intervallen ganze Bilder abgespeichert werden. Die Frames zwischen diesen kompletten Bildern bestehen

dann lediglich aus den jeweiligen Veränderungen zum Bild vorher. Auf diese Weise spart man natürlich Platz, und vorwärts gespielt entsteht auch das richtige Resultat. Aber rückwärts gespielt machen die Bilder vor dem Ganzbild keinen Sinn mehr, denn das Bild, auf das sie aufbauen, ist noch nicht bekannt. Wenn das Kompressionsverfahren nur Vollbilder, oder wenigstens viele davon aufweist, dann wird auch rückwärts sauber abgespielt.

Schaltfläche "pause"



Objekt-spezifische Attribute: **Bitmap**, Bitmap: selbst designed



↑	Linke Maustaste losgelassen: mci1 -> pause
☞	Mausbewegung in das Objekt: Wertzuweisung: comment << "Anhalten des Videos. Drücken Sie Start, um fortzufahren."
☞	Mausbewegung aus dem Objekt: Wertzuweisung: comment << "Status: "+mciGetMode (mci1)

Kommentar

Die Pause-Schaltfläche soll den Videoclip anhalten. Dies wird erreicht über den MCI-Befehl *pause playing*. Ansonsten unterscheidet sie sich von der vorangegangenen Schaltfläche nur durch den Hilfetext und das Bitmap. Ein besonderer Vorzug gegenüber Ihrem Video-Recorder ist der Umstand, daß bei Druck auf die Pause-Schaltfläche die Kopftrommeln und das Band nicht leiden.

Schaltfläche "start"



Objekt-spezifische Attribute: **Bitmap**, Bitmap: selbst designed

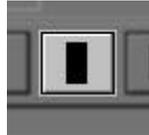


	Linke Maustaste losgelassen: mci1 -> play from start
	Mausbewegung in das Objekt: Wertzuweisung: comment << "Starten des Videos vom Anfang"
	Mausbewegung aus dem Objekt: Wertzuweisung: comment << "Status: "+mciGetMode (mci1)

Kommentar

Diese Taste hat vielleicht nicht das klassische Piktogramm; sie hat auch keine klassische Funktion. Sie spult das Video an den Anfang zurück, und spielt es dann ab. Eine Rewind-Funktion ist also enthalten. Das hat Ihr Video Recorder sicher nicht!

Schaltfläche "stop"



Objekt-spezifische Attribute: **Bitmap**, Bitmap: selbst designed



↑	Linke Maustaste losgelassen: mci1 -> stop
↑	Linke Maustaste losgelassen: mci1 -> seek to 1
↖	Mausbewegung in des Objekt: Wertzuweisung: comment << "Stoppen und Zurückspulen des Videos"
↘	Mausbewegung aus dem Objekt: Wertzuweisung: comment << "Status: "+mciGetMode (mci1)

Kommentar

Die Stop-Schaltfläche stoppt über ein MCI-Kommando zu - nächst das Video und sucht danach den Frame 1, um an den Anfang des Videos zu gelangen. Auch hier ist somit eine Re - wind-Funktion enthalten. Die Pause-Schaltfläche läßt das Video stehen.

Schaltfläche "forward"



Objekt-spezifische Attribute: **Bitmap**, Bitmap: **Arrow right**



	Linke Maustaste losgelassen: mci1 -> play
	Mausbewegung in das Objekt: Wertzuweisung: comment << „Abspieles des Videos von der momentanen Position“
	Mausbewegung aus dem Objekt: Wertzuweisung: comment << "Status: "+mciGetMode (mci1)

Kommentar

Der MCI-Befehl *play from current position* spielt das Video einfach von der gegenwärtigen Stelle aus weiter ab.

Scrollbar "shuttle"



- ▶ **Scrollbar Typ:** *Horizontal mit Pfeilen*
- ▶ **Unterer Wert:** *0*
- ▶ **Oberer Wert:** *all*
- ▶ **Zeilenvorschub:** *1*
- ▶ **Seitenvorschub:** *all/10*



Scrollbar Änderung : Wertzuweisung: currentframe << shuttle



Scrollbar Änderung : mci1 -> seek to shuttle

Kommentar

Diese Scrollbar hat die Aufgabe, das Video zu steuern. Vielleicht hat Ihr Video Recorder zuhause ein Jog-Shuttle. Es

taucht meistens in Form eines runden Knopfes auf, der es ermöglicht, das Video im Einzelbildmodus mehr oder weniger schnell vor- und zurück zufahren. Diese Scrollbar hier hat einen wirksamen Bereich von 1 (Frame 1 des Videos) bis all. Mit all ist hier die Komponente mit dem Namen all gemeint. Sie ist das Textfeld, welches bei Druck auf die Schaltfläche Open, den Wert der maximalen Anzahl an Frames eines Clips bekommt. Bitte vergleichen Sie dazu den Link, der auf der Listbox, sowie auf der Schaltfläche Open liegt. Damit Sie nicht blättern müssen, bilden wir ihn hier noch einmal ab:

↑ Linke Maustaste losgelassen : Wertzuweisung: all
<<mciGetLength(mci1)

Die Funktion `mciGetLength(mci1)` ermittelt also die Länge des Clips, hier in Frames, und übergibt sie an die Komponente all. Deshalb können wir als Maximalwert für die Scrollbar auch den Wert von all verwenden. Mit dem **Seitenvorschub** gehen wir sogar noch einen Schritt weiter. Da jede Seite 10% des Maximalwertes betragen soll, teilen wir einfach all durch 10. Fertig!

Aber jetzt kommt das Schönste! Programmierer aufgepaßt. Unsere Scrollbar verhält sich ohne weiteres Zutun dynamisch. Und das ist keineswegs selbstverständlich. Normalerweise würde nämlich das Programm beim Aufbau der Bildschirmseite irgendwann an die Aufgabe gelangen, die Scrollbar zu zeichnen. Hierzu benötigt es natürlich die Werte **Unterer Wert, Oberer Wert, Zeilenvorschub** und **Seitenvorschub**. Und dabei bleibt es, bis der Wert willentlich und durch Programmierung geändert wird. Da der Wert der Variablen all auf 0 stehen sollte, weil ja am Anfang der Sitzung gar kein Video aktiv ist, wäre die Scrollbar absolut unnützlich. MINDMAP reagiert jedoch auf die Veränderung der Variablen insofern, als es die davon abhängigen Komponenten ebenfalls aktualisiert. Deshalb wird die Scrollbar, je nach geladenem Clip, immer als Maximum den Wert der der Variablen all darstellen. Und der wird durch oben genannte Funktion gesteuert.

Eingabefeld "current-frame", Text "separator", Text "all"



MS Sans Serif, 10 points, bold



Link für Text Komponente all



Kommentar

Diese Anzeige besteht aus drei Feldern. Das Eingabefeld mit der Bezeichnung `currentframe` wird gespeist durch die Komponenten `shuttle` und `mci1`. Es dient dazu, den aktuellen Frame des Clips anzuzeigen. Dieser Anzeige soll nicht nur reagieren, wenn die Scrollbar `shuttle` verändert wird, sondern auch wenn das Video abspielt.

Die Text Komponente `separator` hat nur kosmetische Funktion. Sie soll mit ihrem Schrägstrich die beiden anderen Felder optisch trennen.

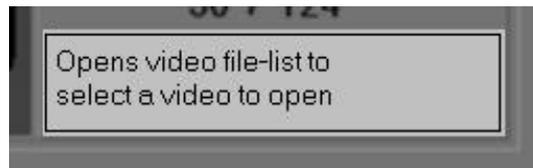
Die Text Komponente `all` enthält immer den Wert der maximalen Anzahl an Frames des geladenen Videos. Sie wird ver-

sorgt, wenn ein Clip über Doppelklick aus der Listbox, oder wenn ein zuvor angewähltes Video über die Schaltfläche btopen geladen wird.

 Linke Maustaste losgelassen: Wertzuweisung: all
<<mciGetLength(mci1)

Die Felder wurden hier so angelegt, daß keine Rahmen sichtbar sind.

Eingabefeld "comment"



Farbe: grau



MS Sans Serif, 8 points, Standard



Nicht **editierbar**, Rahmen Standard, Mehrzeilig

Kommentar

Alle Links, die mit den Ereignissen  oder  eingeleitet werden, beziehen sich auf dieses Eingabefeld. Außer dem enthält die Multimedia Komponente mci1 einen Link, der sich

ebenfalls auf dieses Feld bezieht. Comment dient sozusagen als Kurzhilfezeile. Die Einträge werden über **Wertzuweisungen** vorgenommen. Hierbei ist darauf zu achten, daß die Zeilenschaltungen im Formelfeld von Wertzuweisungen auch denen im Eingabefeld entsprechen. Wenn Sie das Geflackere bei heftigen Mausbewegungen nervös machen sollte, sind Sie mit Ihrem Wissen längst in der Lage, eine Schaltfläche einzubauen, mit der diese Kurzhilfe ein- und ausgeblendet werden kann. Hier würde sich eine Checkbox geradezu anbieten.

Multimedia Komponente "mci1"



Alle Optionen "aus", Dateiauswahl: keine



Position changed : Wertzuweisung: shuttle <<
mciGetPosition(mci1)



Position changed : Wertzuweisung: currentframe <<
mciGetPosition(mci1)



Mode changed : Wertzuweisung: comment << "Status:
"+mciGetMode(mci1)

Kommentar

Das Kernstück unseres Fernsehers enthält herzlich wenig Attribute und Links, wie sooft im Leben. Die meisten Funktionen beherrscht diese Komponente schon von alleine. Wir steuern eigentlich nur, welcher Clip geladen wird, die Lautstärke, den Frame und die Laufrichtung. Alle anderen Informationen bezieht die Komponente aus den Header-Informationen des jeweiligen Clips.

Der erste der drei Links dient dazu, die Scrollbar shuttle, analog zu den abgespielten Frames zu bewegen. Der zweite Link aktualisiert die Anzeige currentframes. Durch den dritten Link wird das Kurzhilfefeld comment mit der aktuellen Modusinformation versorgt.

Tip: Es kann bei der "Programmierung" unseres Fernsehers vorkommen, daß unter bestimmten Umständen die MCI Komponente "vergißt", wozu sie da ist. Man kann dem meist damit abhelfen, indem man über die objekt-spezifischen Attribute in dem erscheinenden Dialog irgendein Video über Datei lädt.

So, die Funktionalität unseres Programmes stimmt. Teilweise haben wir ja auch schon kosmetische Veränderungen vorgenommen. Jetzt sind Sie dran. Stellen Sie alles in Frage. Entfernen Sie ihren eigenen Viewer.

Die ganz Mutigen unter Ihnen können sich auch daran versuchen, unseren Picture-Viewer mit dem Video-Recorder zu kombinieren. Das Prinzip ist das gleiche.

Zum Vergleich liegt Ihnen diese Anwendung unter dem Namen **vidplay.mm** vor.

Viel Vergnügen!

Datenbank

Allgemeines

Wahrscheinlich die Hauptanwendung aller kybernetischen Bemühungen ist der Umgang mit Datenbanken. Zwar scheint dies eine Domäne der Großrechner und mittleren DV zu sein, aber die PCs haben sich im Laufe der Zeit gemausert und bieten für kleinere Anwendungen schon ganz passable Leistungsmerkmale. Selbst bei firmenweiten Anwendungen und bei großen Datenmengen stehen die PCs ihren Mann, wenn es sich um eine Client-Server Architektur handelt. Bei dieser Idee werden die Aufgaben gewissermaßen verteilt. Die leistungsfähige Maschine im Hintergrund, der Server, übernimmt die intensive Arbeit mit den Daten, und der PC repräsentiert sie auf dem Bildschirm. Auf diesem Wege kann sich jede Komponente ganz auf seine Stärken konzentrieren.

Waren noch vor einigen Jahren Datenhaltung und Benutzeroberfläche ein und dasselbe Programm, so finden wir heute einen Frontend - Backend Split vor. Die Datenbank-Engine, das Arbeitstier, liefert wie eine Blackbox Daten in unterschiedlicher Performance hinsichtlich Tempo und Sicherheit. Diese Daten werden einer Schnittstelle nach außen zur Verfügung gestellt, und es hängt von der Kommunikation der Programme über diese Schnittstelle ab, mit welchem Wirkungsgrad die Daten vorne verarbeitet werden können.

Der Transportmechanismus findet auf verschiedenen Ebenen statt. Ein Transportprotokoll von mittlerweile exorbitanter Wichtigkeit trägt den Namen TCP/IP. Es findet vor allem im Datenverkehr mit UNIX-Rechnern Einsatz. Dieses Protokoll trägt Informationen in Paketform von einem Rechner auf den anderen. Welcher Art diese Pakete aber sind, also welchen direkten und welchen indirekten Informationsgehalt sie haben, darauf hat TCP/IP keinen Einfluß.

Wie man ja weiß, gibt es eine Vielzahl von Datenbankanbietern, deren Namen wir hier nicht auflisten müssen. Diese Datenbankanbieter liefern in der Regel ein Backend, also die

Datenbank-Engine aus. Mit der Zeit ist es selbst so hoch schwebenden Kreisen zu Ohren gekommen, daß es dem Anwender nicht reicht zu wissen, daß seine Daten gut aufgehoben sind. Er möchte auch einen entsprechenden Zugriff. Deshalb hat man auf der Seite der Hersteller mehr oder aber auch weniger dafür getan, daß die Daten den Rechner in Richtung Anwender verlassen können. Die einen haben lediglich ein Protokoll zur Verfügung gestellt und Drittanbieter motiviert, adäquate Programme für den Client zu programmieren. Andere haben gleich einen ganzen Satz Werkzeuge zur Verfügung gestellt, um Programmierern den Weg zu bereiten, Anwendungen für das Backend zu schreiben. Aber natürlich haben alle ihr eigenes Süppchen gekocht. Das ist zunächst einmal kein besonderer Nachteil, denn diese Werkzeuge sind in besonderem Maße auf die jeweiligen Datenbanken und deren Leistung abgestimmt. Der Entwickler solcher Programme mußte sich aber auch binden. Kunden, die Ihre Datenbank auf ein anderes Produkt umstellen, benötigen insofern auch eine andere Frontend-Applikation. Systemhäuser können nur den Markt für diese eine Datenbank abdecken. Natürlich gab es bald Produkte unabhängiger Firmen, die eine ganze Reihe von Datenbanken unterstützten. Die ihrerseits haben ihre liebe Not mit den Entwicklungszyklen der Datenbankhersteller. Die Treiber müssen angepaßt werden.

Im Zuge der Standardisierung ging man dann vor einiger Zeit noch einen Schritt weiter. Ein Konsortium legte die Kriterien für eine offene und unabhängige Schnittstelle fest, deren Name ODBC ist, der Abkürzung für Open DataBase Connectivity. Man empfahl also den Datenbankherstellern eine Schnittstelle zur Verfügung zu stellen, die die Datenbank nach außen hin von den Konkurrenzprodukten nicht unterscheidbar macht! Das darf man sich ruhig einmal auf der Zunge zergehen lassen. Aber im Ernst: so wichtig die Unterscheidungsmerkmale der verschiedenen Hersteller für eine Firma und deren Administratoren auch sein mag, dem Anwender selbst dürfte es vollkommen egal sein, ob die Daten, die er vor sich sieht, aus der Datenbank des Herstellers A oder aus der des Herstellers B kommen. Und dem Entwickler des Frontends sollte das auch egal sein.

Verreisen Sie hin und wieder ins Ausland? Nehmen Sie Ihren Föhn, den Rasierapparat oder gar Ihre Mikrowelle mit? Sind Sie schon einmal daran gescheitert, den Stecker Ihres billig in

Kuala Lumpur erstandenen elektrischen Fußnagelhobels in eine Steckdose in Connemara zu stopfen? Ist deshalb eine Verabredung geplatzt? Seitdem besitzen Sie diesen Universalstecker, der zwar gräßlich aussieht, aber seine Dienste verrichtet. Nein, wir versuchen nicht, den elektrischen Strom mit einem Datenstrom zu vergleichen. Wir haben nur nach einem praktischen Beispiel gesucht. Und es ist uns nicht gelungen!

Nun hört sich diese Entwicklung ja ganz vorteilhaft an, dennoch hat sie einen Haken. Durch den Umstand, daß eine einheitliche Schnittstelle definiert wurde, kann auch auf die spezifischen Merkmale der Hersteller keine Rücksicht genommen werden. Weiterhin ist es durch aus so, daß sich diese Standardisierung in der Entwicklung befindet und wahrscheinlich immer befinden wird. Damit man auf ein schlagkräftiges Endprodukt nicht zu lange warten muß, beschloß man, den Anwendern und Entwicklern kleinere Portionen in Form von Levels zuzuwerfen. Jeder der Level soll weitere allgemeine Features der Datenbanken unterstützen.

Keiner weiß, wann Sie diese Zeilen lesen. Keiner weiß, wie weit man dann mit ODBC sein wird. Wir gehen deshalb einfach pragmatisch ans Werk und *benutzen ODBC* für unsere Zwecke.

Verfügen Sie denn überhaupt über eine Datenbank? Entschuldigung, diese Frage ist berechtigt. Wenn Sie den ODBC-Treiber bereits auf Ihre Datenbank konfiguriert haben, dann können Sie für die folgenden Übungen Ihre Datenbank benutzen, egal ob sie sich auf dem PC oder irgendwo befindet. Auf jeden Fall aber können Sie den Treiber verwenden, der von MINDMAP zur Verfügung gestellt wird, und Daten im dBase-Format ablegt. Keiner mag behaupten, daß diese Engine über besondere Merkmale verfügt, aber zur Übung und für kleinere Dinge ist sie brauchbar. Die Konfiguration des Treibers erledigen Sie bitte mit dem entsprechenden Programm.



ODBC Admin

Dieses Icon finden Sie sowohl in der MINDMAP Programmgruppe als auch in der Systemsteuerung von MS-Windows. Mit diesem ODBC-Administrationsprogramm können Sie neue ODBC-Treiber installieren, und auch neue Datenquellen einrichten.

Wenn Sie bereits Erfahrung im Umgang mit Datenbanken haben, können Sie sich das folgende Kapitel eigentlich sparen.

Hier soll in komprimierter Form dargelegt werden, wie eine Datenbank Daten organisiert. Die klaren Sieger unter den verschiedenen Datenbankformen sind die *relationalen Datenbanken*. Das führte so weit, daß im Laufe der Zeit fast jeder Hersteller behauptete, relationale Datenbanken anzubieten. Dies ist aber häufig nicht wahr. Wir wollen hier keine Definition vorlegen, die beschreibt, was eine relationale Datenbank ist. Die Regeln jedoch, die ein gewisser Mr. Codd, sozusagen einer der Papas der relationalen Idee definiert hat, halten lange nicht alle die ein, die behaupten hier mitreden zu können. Aber was soll's. Solange sich die Datenbanken brauchbar verhalten, solange kann es uns egal sein, ob sie irgendwelchen akademischen Hürden gerecht werden.

Ein weiteres Zauberwort ist *SQL*. Das steht für Structured Query Language und stellt einen Standard dar, mit dem Datenbanken erzeugt, modifiziert und nach deren Inhalt gefragt werden kann. *SQL* besteht aus einer Reihe von natürlich-sprachlichen Elementen, von denen man nur etwa 20 kennen muß, um mühelos eine florierende Firma zu verwalten. Die Sprache *SQL* ist indes nicht schwer. Schwerer ist die relationale Algebra zu verstehen, die dahinterliegt. MINDMAP nimmt Ihnen den Umgang mit *SQL* ab. Hier wird am Anfang nur selten richtiger *SQL*-Code eingegeben, und man kommt schon ganz schön weit mit dem, was sich interaktiv generieren läßt. Falls Sie mehr über *SQL* wissen möchten, finden Sie am Ende dieses Tutorials ein **Kleines Datenbank-Brevier**, welches die wichtigsten Elemente dieser Sprache auflistet.

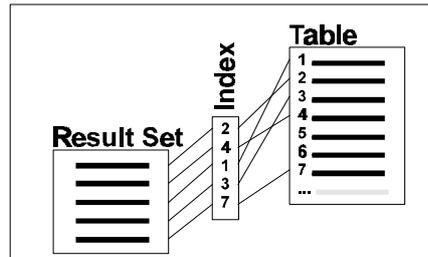
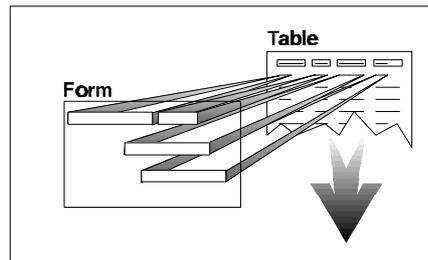
Fast alle der angebotenen Datenbanken, insbesondere wenn sie sich relational nennen, arbeiten nach einem bestimmten Prinzip. Der größte Behälter, vergleichbar mit einem Aktenschrank (ein stark strapaziertes Beispiel) ist die Datenbank! Hier sei unterschieden zwischen dem Datenbankprogramm und einer mit einem bestimmten Namen versehenen Datenbank, die mit diesem Programm erstellt worden ist. Diese Datenbank beherbergt diverse Tabellen, die in unserem Vergleich Aktenordnern gleichkommen. Diese Tabellen können, müssen aber nicht, in gewisser Relation zueinander stehen (was eben dazu führt, daß sich jeder, der diesen Trick beherrscht, *relational* nennt). Diese Relation erfolgt über Einträge gleichen Inhaltes in den Tabellen. Hierzu wird die Tabelle in Spalten und Zeilen aufgeteilt, was ja ohnehin der Natur einer Tabelle entspricht.

Hier verlassen wir die Aktenordner-Idee. Denn dort stellt normalerweise jedes Blatt eine Einheit an Informationen dar. Aber auf diese Sicht auf die Daten kommen wir noch zu sprechen. Also, eine Einheit an zusammengehörender Information entspricht einer Zeile in der Tabelle. Zeilen zweier Tabellen, die wenigstens eine Spalte mit dem gleichen Eintrag aufweisen, können (müssen nicht) miteinander verbunden werden. Diese Bindung hat keinen statischen Charakter und wird auch nirgendwo niedergeschrieben. Ein solcher Join existiert streng genommen nur in der Sekunde, in der er gebraucht wird. Er entsteht mit seiner Definition während einer Abfrage und zerfällt zu nichts, wenn die Abfrage beendet ist. Insofern sind die Tabellen untereinander vollkommen unabhängig. Um das Aktenordner-Beispiel weiter auszubauen, stellen Sie sich bitte vor, ein Kunde reklamiert eine Bestellung. Die Bestellungen werden aber nach Kundennummer abgeheftet. Um die Kundennummer zu ermitteln, wird also die Kundenkartei zur Rate gezogen. Mit der Nummer nimmt man sich einen Ordner vor, der die Bestellungen enthält. Dort findet man tatsächlich die reklamierte Bestellung. Jetzt kann man den Ordner mit den Bestellungen getrost in den Müll werfen. Der Inhalt der Kundenkartei bleibt davon unberührt. Auch ist es möglich die Kundenkartei zu verbrennen, ohne daß die Bestellungen an sich leiden. Ob man jetzt noch besonders viel mit den verwaisten Informationen anfangen kann, steht auf einem ganz anderen Blatt.

Wie man ohne weiteres feststellen kann, ist die Kundennummer das Bindeglied zwischen Kartei und Ordner. Aber eben nur in dem Augenblick, in dem wir sie verwenden. Da nach schlummern beide weiter den Schlaf der Gerechten.

Nun noch die Sicht auf die Daten. Grundsätzlich legt die Datenbank die Daten in beliebiger Reihenfolge in jeweils einer Zeile ab. Jede Zeile ist unterteilt in mehrere Spalten. Die Spalten tragen Namen wie *Name* oder *Gewicht* oder *Preis*. Manche Spalten tragen leider auch den Namen *Limit*. Die Daten, die in beliebiger Reihenfolge von der Datenbank abgelegt werden, können in einer vom Fachmann festgelegten Reihenfolge angezeigt werden. Die Sortierungen, die so vorgenommen wird, ist nur eine Illusion einer anderen Anordnung. Diese Anordnung betrifft die Zeilen als solche. Was aber mit der Anordnung der Spalten? Schließlich ist auf einer Kartei -

karte ja auch alles hübsch aufgeräumt. Und jeder Datensatz hat dort seine eigene Karte.



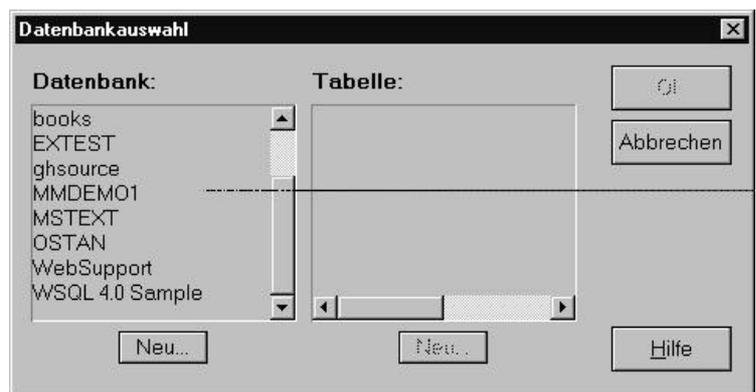
Datenbank Manager Komponente

Um den Umgang mit Datenbanken in MINDMAP richtig zu verstehen, sollten wir schrittweise vorgehen. Und da ja bekanntlich der erste Schritt einer Reise schon die Hälfte des Weges ist, zögern wir keine Sekunde mehr. Rufen Sie also MINDMAP auf, und beginnen Sie eine neue Anwendung.

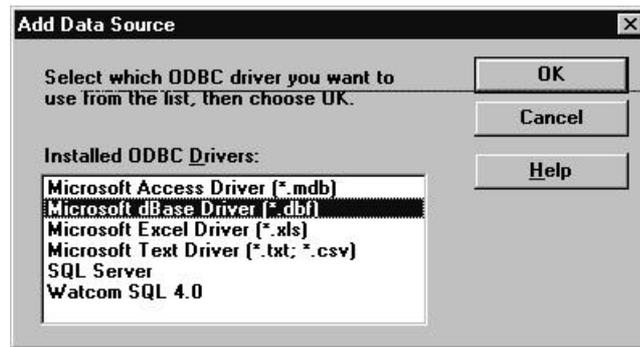
Für den Anfang erzeugen wir eine Datenbank mit nur einer Tabelle, die nur wenige Felder hat. An dieser werden wir dann einige Eigenschaften und Funktionen von MINDMAP ausprobieren. Wählen Sie hierzu das Datenbank-Werkzeug an.



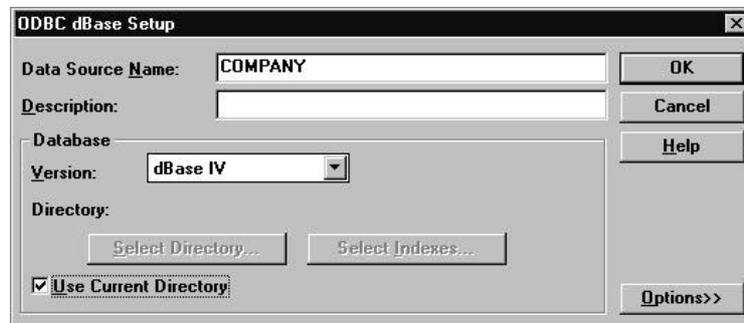
Sobald Sie dieses angeklickt haben, erscheint der Dialog **Datenbankauswahl**:



Auf der linken Seite sind die bereits installierten Treiber und Datenquellen aufgeführt. Da wir eine neue Datenquelle einrichten wollen, betätigen wir die Schaltfläche *Neu* auf der linken Seite. Es wird der folgende Dialog sichtbar:



Erfahrenere Anwender werden diesen Dialog kennen. Dieser und den nächsten Dialog sieht man auch, wenn man außerhalb von MINDMAP eine ODBC-Datenquelle installiert (Systemsteuerung). Hier wird nach dem zu verwendenden Treiber gefragt. Für unser kleines Beispiel wählen Sie bitte *Microsoft dBase Driver (*.dbf)* aus, und bestätigen Sie mit *OK*. Dies bringt Sie zu folgendem Dialog:

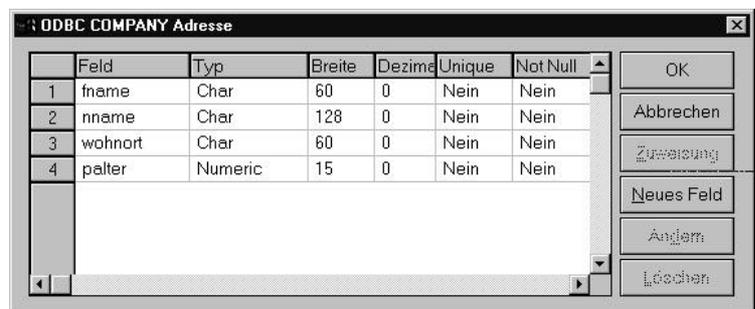


Tragen Sie hier bitte den Namen der Datenquelle (**Data Source**) ein. Diesen Namen können Sie frei wählen. Er muß nicht mit dem Namen der Datenbank identisch sein. Über den Namen, den Sie hier vergeben, werden Sie in MINDMAP die Datenbank ansprechen. Für das Beispiel schlagen wir *Company* vor. Bitte geben Sie auch das Verzeichnis an, in welchem sich die Datenbank befinden soll, oder wählen Sie einfach die Option *Use Current Directory*. Bestätigen Sie den Dialog mit *OK*.

Nun befinden Sie sich wieder im Dialog **Datenbankauswahl**, der jetzt die soeben erzeugte Datenbank *Company* enthält. Selektieren Sie bitte diesen Eintrag. Sobald man in diesem

Dialog links eine Datenbank auswählt, erscheinen rechts die Tabellen, die zu dieser Datenbank gehören. Sie können dies ausprobieren, indem Sie die Datenbank *MMDEMO1* auswählen, die von MINDMAP installiert werden, sofern sie dies gewünscht haben. Sehen Sie die verschiedenen Tabellen? Zurück zur Datenbank *Company*. Hier bleibt die Liste der Tabellen natürlich leer. Schließlich müssen wir ja erst welche erzeugen. Dies tun wir, indem wir die Schaltfläche *Neu* betätigen, welche sich auf der rechten Seite befindet. Dies bringt uns zum nächsten Dialog, wo wir den Namen der Tabelle, die wir erzeugen wollen, eintragen müssen. Geben Sie hier bitte als Tabellename *Adresse* ein und bestätigen Sie mit *OK*.

Nun ändert sich der Maus-Cursor in ein Fadenkreuz. Wir sind also jetzt im Zeichen Modus für eine Datenbank Komponente. Klicken Sie nun mit der Maus irgendwo auf die freie Arbeitsfläche, und die Komponente wird plziert. Gleichzeitig wird ein neues Fenster geöffnet, das es Ihnen ermöglicht, die Struktur der Tabelle zu definieren. Das Fenster sieht wie folgt aus, nur daß noch keine Felder definiert sind.



Betätigen Sie bitte die Schaltfläche *Neues Feld*, so daß der folgende Dialog erscheint. MINDMAP benötigt nun für die Definition der Tabelle einige Angaben. Der hierfür vorgesehene Dialog bietet bereits Vorgaben, die der Leistung des ausgewählten Treibers entsprechen. Vergeben Sie bitte Spaltennamen und Datentypen, wie in unserer Abbildung.

Beginnen Sie nun mit den notwendigen Eintragungen für das erste Feld *fname*. Bestätigen Sie dann den Dialog mit *OK*.



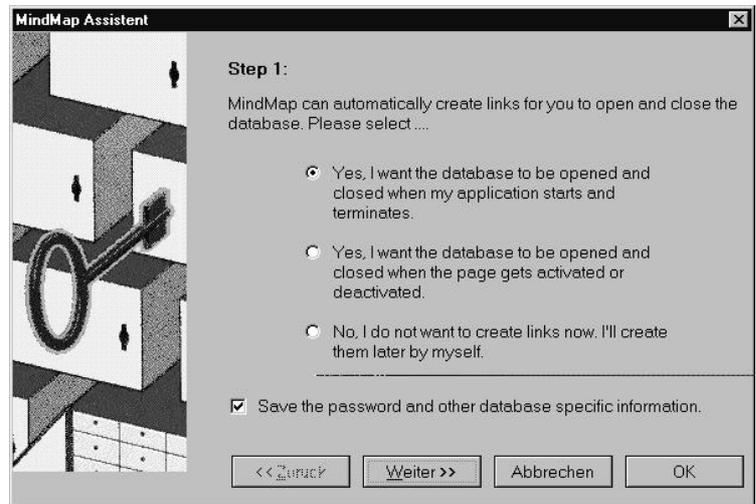
Dieser Dialog enthält Angaben zu verschiedenen Eigenschaften der Tabellenspalten, d. h. zu den Eigenschaften der Datenbankfelder. Der Umfang der Möglichkeiten differiert zwischen den verschiedenen Daten bank Treibern. Im Wesentlichen sind jedoch folgende Ein gabemöglichkeiten vorgesehen:

- ▶ Tragen Sie bitte bei **Name** den Feldnamen ein. Dieser Name wird in aller Regel nicht dem Namen entsprechen, der später in der Anwendung auftaucht. Vermeiden Sie bitte Sonderzeichen oder Leer schritte im Hinblick auf die Portabilität der Anwendung auf andere Betriebssysteme oder Datenbanken.
- ▶ Der **Datentyp** bestimmt die Art der zu erwartenden Daten, welche in dieser Spalte abgelegt werden. Die häufigsten Typen werden *Char* für Character und *numerisch* sein. Verschiedene Datenbanken können verschiedene Feldtypen haben. Überlegen Sie sich, ob ein Feld, das Zahlen auf nehmen soll, auch eine mathematische Relevanz haben wird. Die Telefonnummer beispielsweise nimmt zwar nur Zahlen entgegen. Sie hat aber oft führende Nullen oder Zwischenräume, die die Lesbarkeit verbessern. Rechnen im Sinne von mathematischen Berechnungen muß man mit Telefonnummern nie.
- ▶ Bei **Länge** tragen Sie bitte ein, wie lang die maximale Ein gabe sein darf. Auch hier gibt es Unterschiede zwischen den einzelnen Daten banken. Bitte vergegenwärtigen Sie sich, daß ein Feld mit der Länge 200 immer auch entsprechend viel Platz verbraucht. Egal ob 199 oder 1 Zeichen abgespeichert worden sind.
- ▶ **Dezimal** betrifft die Nachkommastellen eines numerischen Feldes.
- ▶ Durch **NotNull** wird definiert, ob ein Anwender gezwungen sein wird, das Feld unbedingt mit Daten zu versehen. Die Daten bank wird die Annahme eines

Datensatzes verweigern, wenn dieses Feld nicht ausgefüllt ist. Sie kann allerdings nicht darüber entscheiden, welchen Sinn ein eventueller Eintrag hat. Nicht alle Treiber unterstützen diese Eigenschaft. Dann ist diese Option im Dialog von MINDMAP inaktiv.

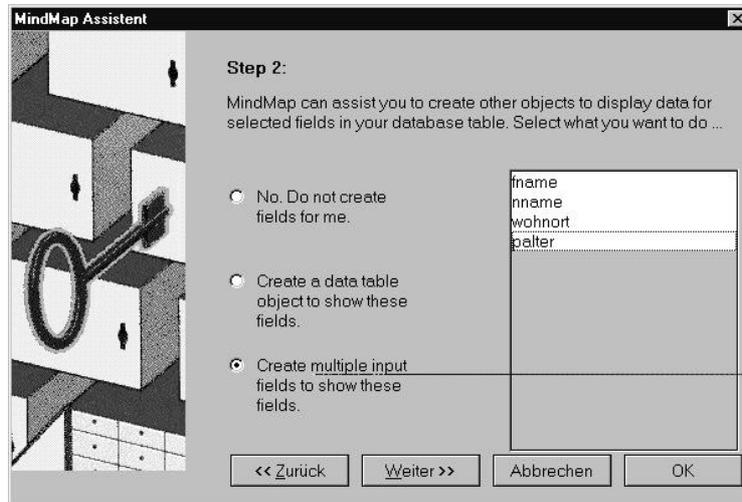
- ▶ **Unique** legt fest, daß ein Eintrag in dieser Spalte genau nur einmal innerhalb dieser Tabelle vorkommen darf. Auch diese Option wird nicht von allen Treibern unterstützt, und deshalb im Dialog von MINDMAP inaktiv.

Wiederholen Sie bitte den Vorgang *Neues Feld* solange, bis Sie sämtliche Felder definiert haben. Danach wird der Dialog **ODBC COMPANY Adresse** mit **OK** geschlossen, was zur Folge hat, daß der **MindMap Datenbank Assistent** gestartet wird.



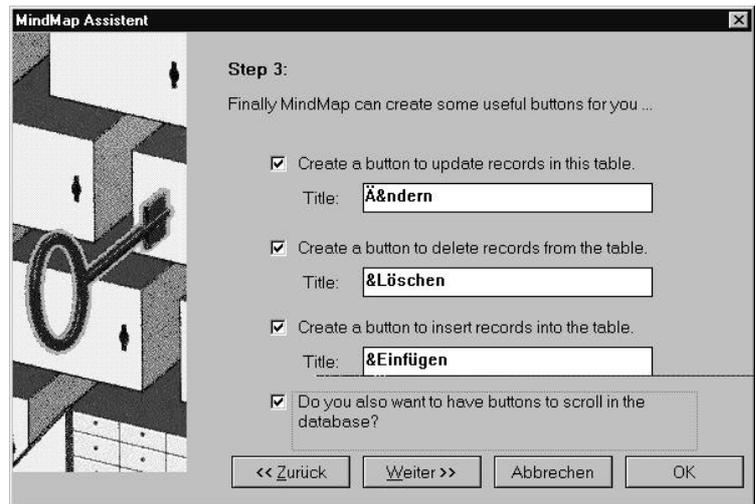
Die erste Frage des Assistenten **Step 1** bezieht sich darauf, ob MINDMAP die Datenbank beim Start des Programmes sofort öffnen soll oder nicht. Dem ist in aller Regel zuzustimmen, wenn sich nicht besonders viele Datenbank Komponenten in Ihrer Applikation tummeln, und das sofortige Öffnen all dieser Kanäle unsinnig erscheint. In einem solchen Falle wäre die zweite Option sinnvoll, die einstellt, daß die Datenbank erst beim Anwählen der jeweiligen Seite geöffnet wird. Vielleicht möchten Sie einmal auch keine der beiden Möglichkeiten in Anspruch nehmen, und die entsprechenden Links später lieber selbst erzeugen. Dann muß hier die dritte Option gewählt

werden. Wir schlagen vor, die erste Option, die bereits defaultmäßig eingestellt ist, unverändert zu lassen, und die Schaltfläche *Weiter* zu betätigen, was Sie zum nächsten Dialog bringt:



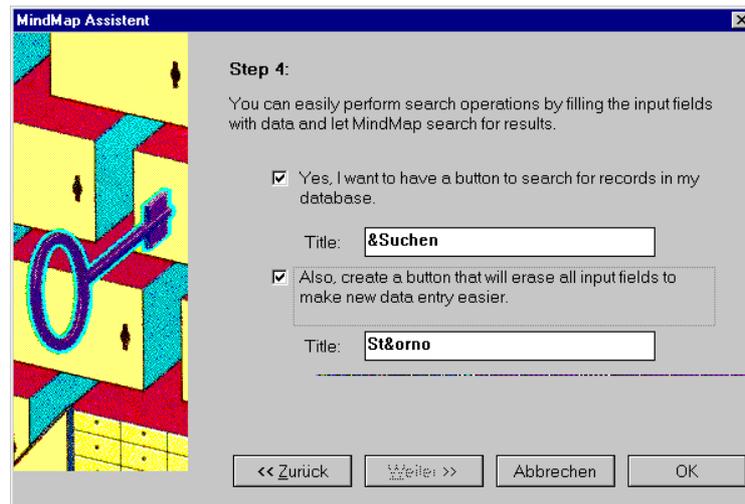
Step 2 des Assistenten fragt danach, ob der Tabellen definition entsprechende Eingabefelder erzeugt werden sollen, oder ob eine MINDMAP Datatable mit bestimmten Feldern angelegt werden soll, oder aber keines von beiden. Wenn Sie Eingabefelder oder Datatable erzeugen lassen wollen, dann dürfen Sie nicht vergessen, die gewünschten Felder in der rechten Box auszuwählen. Für unser kleines Übungsbeispiel wählen wir aus, daß der Assistent Eingabefelder für alle Datenbankfelder generieren soll.

Bestätigen Sie die Dialoge des Assistenten bitte nicht mit *OK* wenn Sie dies nicht ausdrücklich wollen. *OK* beendet den Assistenten. *Weiter>>* bringt Sie zur nächsten Stufe.



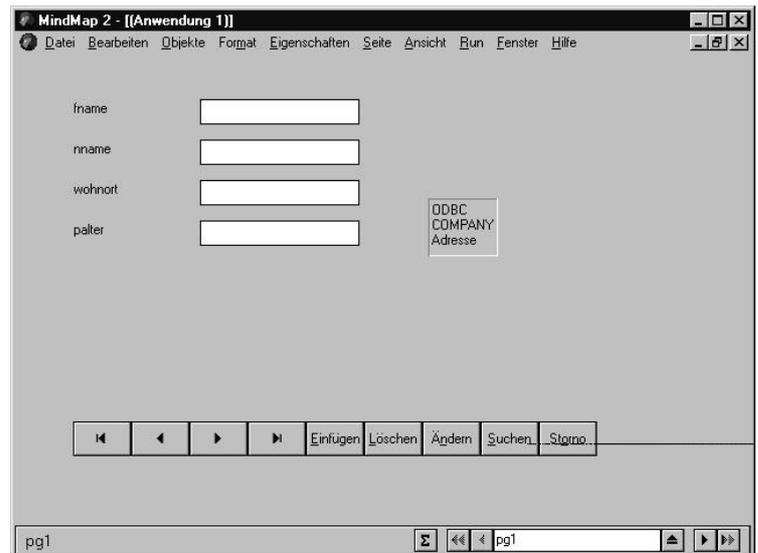
In **Step 3** werden Sie gebeten anzugeben, ob Sie zur Steuerung der Datenbank, Schaltflächen für die drei wichtigsten Datenbankoperationen *Ändern*, *Löschen* und *Einfügen* möchten. Außerdem können Sie angeben, ob Sie Schaltflächen zur Bewegung innerhalb der Datensätze benötigen. Wir wählen hier sämtliche Optionen aus, so daß all die Schaltflächen mit den notwendigen Links automatisch für uns erzeugt werden. Diese Schaltflächen lassen sich später sehr einfach so formatieren, wie es Ihrem bisher schon mehrfach unter Beweis gestellten guten Geschmack entspricht!

Durch *Weiter>>* gelangen wir zu **Step 4**, dem letzten Dialog des MINDMAP Datenbank Assistenten.



Hier kann festgelegt werden, ob es auch Schaltflächen für *Suchen* und *Storno* geben soll. Mit der Schaltfläche *Suchen* wird der Assistent einen Link erzeugen, der es dem Anwender ermöglicht, die Datenbank nach bestimmten Kriterien zu durchsuchen. Diese Suchkriterien richten sich danach, welche Eintragungen ein Anwender in die Eingabefelder gemacht hat. Aus diesen Eintragungen generiert MINDMAP nämlich selbstständig das notwendige SQL-Statement. *Storno* erleichtert hierzu die Arbeit. Es wird der Inhalt aller Eingabefelder gelöscht, was besonders bei der Neu eingabe eines Datensatzes und bei der Eingabe von Suchkriterien vorteilhaft ist. Selbstverständlich werden die Inhalte nicht aus der Datenbank selbst gelöscht, sondern nur aus den Feldern der Anwendung.

Insgesamt ermittelt der Assistent die wichtigsten Angaben zur komfortablen Platzierung von Datenbanken in einer MINDMAP-Anwendung. Der Dialog **Step 4** enthält keine *Weiter>>*-Schaltfläche mehr. Nachdem wir auch hier beiden Optionen ausgewählt haben, schließen wir den Dialog mit *OK*. Nun sehen Sie, wie der Assistent die Datenbank Komponente und alle gewünschten Felder und Schaltflächen in Ihrer Anwendung platziert.



Auf dem Bildschirm befindet sich neben den Eingabefeldern mit Ihren Labels und den Schaltflächen mit den Datenbankfunktionen noch eine weitere Komponente, die eigentliche Datenbank Komponente. Diese Komponente verschwindet zur Laufzeit visuell aus unserer Anwendung und ist nur im Edit Mode sichtbar. Sie verfügt, wie alle anderen Komponenten in MINDMAP, über Attribute, die sich mit der rechten Maustaste aufrufen lassen. Auf diese kommen wir aber später zurück.

Sehen wir uns zunächst einmal an, was sich bereits jetzt mit unserer Datenbankanwendung machen läßt. Gehen Sie in den Run Mode und geben Sie Daten in die Eingabefelder ein. Speichern Sie diese anschließend mit *Einfügen*. Löschen Sie die Eingabefelder mit *Storno*, und geben Sie den nächsten Datensatz ein. Wiederholen Sie diese Schritte bitte mehrfach, so daß die Tabelle über einen Datenbestand von einigen Datensätzen verfügt.

Innerhalb der eingegebenen Sätze kann nun mit den dafür vorgesehenen Schaltflächen geblättert werden. Mit der linken Schaltfläche gelangen Sie zum ersten, mit dem nächsten zum jeweils vorherigen Datensatz. Die nächste Schaltfläche blättert zum jeweils nächsten Datensatz. Der letzte der Blätterknöpfe bringt Sie zum letzten Datensatz. Mit der Schaltfläche *Löschen* kann der momentan angezeigte Datensatz gelöscht

werden; mit *Ändern* wird der momentan angezeigte Datensatz entsprechend der Änderungen in den Feldern modifiziert.

Nachdem Sie so etwa 300 bis 500 Ihrer engsten Freunde als Datensätze erfaßt haben (es dürfen auch wesentlich mehr sein!), verwenden wir einmal die Suchfunktion.

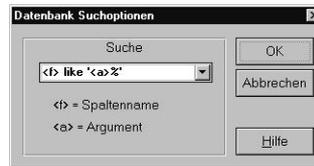
Ein ganz simples Experiment wäre es, wenn Sie einfach auf einem beliebigen Datensatz stehenbleiben und die Schaltfläche *Suchen* drücken. MINDMAP wird dann nach einem Datensatz suchen, der genau den in den Eingabefeldern eingegebenen Kriterien entspricht. Wenn Sie nun die Blätterknöpfe verwenden, wird sich kein anderer Datensatz zeigen. Das bedeutet, daß die Ergebnismenge (Result Set) genau einen Satz umfaßt, und unsere Datenbank nun so tut, als gäbe es tatsächlich nur diesen einen. Drücken Sie nun auf *Storno* und dann noch einmal auf *Suchen*, dann werden wieder alle Datensätze angezeigt.

Sie haben die Möglichkeit, Einträge in verschiedene Eingabefelder zu machen, und nach diesen suchen zu lassen. Es wird dann nach den Datensätzen gesucht, die in all Ihren Felder die gewünschten Einträge enthalten. Kenner wissen, daß es sich hier um eine sog. **UND-Verknüpfung** handelt. Aber dank MINDMAP braucht man das nicht weiter zu interessieren. Trägt man verschiedene Suchbegriffe in ein Eingabefeld ein und trennt diese durch Semikolon ab, so wird nach sämtlichen Datensätzen gesucht, bei denen einer der Suchbegriffe als Eintrag vorkommt. Dies nennen Fachleute eine **ODER-Verknüpfung**.

Die Suche erstreckt sich auf die Einträge, wie sie in den Datenbankfeldern stehen. Allerdings besteht die Möglichkeit in Eingabefeldern nur die Anfangsbuchstaben einzugeben. Die Eingabe von *Jo* bringt also beispielsweise das Ergebnis *Jones* und *Jordan*. Der Eintrag *jO* allerdings, bringt kein Ergebnis. Bei der Suche wird zwischen Groß- und Kleinbuchstaben unterschieden.

Für Eingabefelder können Sie über die Attribute einstellen, ob die Suche genau nach dem eingegebenen Begriff stattfinden soll, oder ob der eingegebene Begriff nur irgendwo in einem Eintrag vorkommen soll. Eine dritte Einstellung ermöglicht es, nur nach Einträgen zu suchen, die mit einem bestimmten Buchstaben oder einer bestimmten Buchstabenkombination beginnen (s. Bsp *Jo*).

Selektieren Sie bitte eines der Eingabefelder, und drücken Sie die rechte Maustaste. Wählen Sie aus dem Attributefenster das Attribute, das durch die Lupe symbolisiert wird. Es erscheint der folgende Dialog:



Hier sind die Einstellungen für die Suchbegriffe festgelegt. Sehen Sie sich diese Einstellungen für die verschiedenen Eingabefelder an. Sie werden feststellen, daß Sie sich unterscheiden, was nicht zuletzt mit den unterschiedlichen Datentypen zusammenhängt. Verändern Sie die Einstellungen und vergleichen Sie die unterschiedlichen Suchergebnisse.

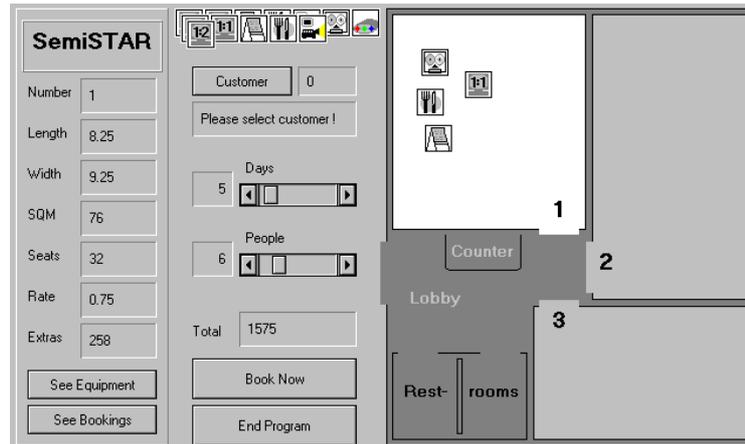
Natürlich kann man auch komplexe SQL-Abfragen machen. Dazu lesen Sie bitte im **Kleine Datenbank-Brevier** am Ende des Tutorials.

Bau einer erweiterten Datenbank- anwendung

In diesem Kapitel beschäftigen wir uns mit dem Bau einer etwas umfangreicheren Anwendung. Ein kleines Unternehmen vermietet drei Seminarräume mit entsprechendem Equipment. Um anrufenden Interessenten schnell den zu erwartenden Preis nennen zu können, hat der EDV-Beauftragte mit MIND-MAP eine Anwendung erstellt, bei der zunächst der in Frage kommende Raum angewählt, und anschließend dann in diesen das gewünschte Equipment gezogen wird. Die Kundendaten werden aus einer Datenbank geholt, oder bei Neukunden werden diese Daten erfaßt. Die Anzahl der gebuchten Tage und die der Teilnehmer werden auf Knopfdruck gebucht. Damit die Anwendung nicht zu umfangreich wird, wurde auf eine kalendergerechte Terminierung verzichtet. Natürlich steht es Ihnen frei, diese nachträglich einzufügen. Das stellt eine hübsche Übung dar. Diese Anwendung ist übrigens leicht an andere Erfordernisse anpaßbar, auch wenn es sich dann um die Vermietung von Kraftfahrzeugen, Fahrrädern oder Studios handelt. Insgesamt existieren hier vier verschiedene Seiten mit diversen Funktionen, die nachfolgend beschrieben werden.

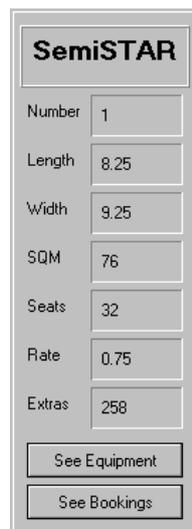
Beschreibung der Seite 1

Die erste Seite der Anwendung sieht wie folgt aus:



Sie ist der Hauptteil der Anwendung und enthält eine Vielzahl an Komponenten. Die Grafiken liegen zum bequemen Einbau vor. Natürlich ist es nicht unbedingt erforderlich, mit derselben Vielfalt an Ausstattungsgegenständen aufzuwarten. Beachten Sie dann aber bitte, die Teile der Anwendung zu modifizieren, die hiervon abhängig sind.

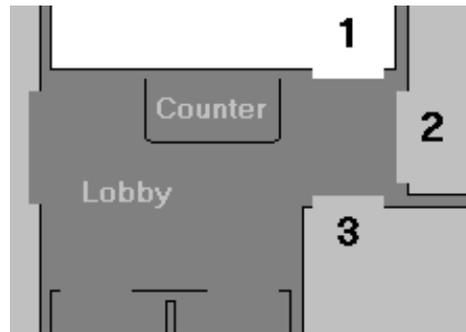
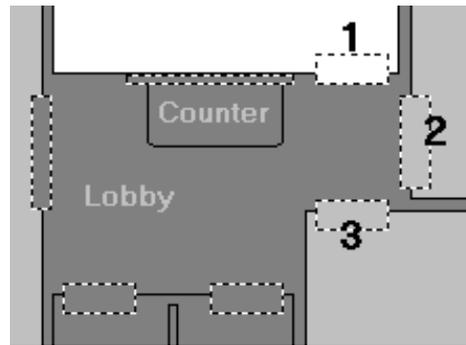
Auf der linken Seite der Anwendung sind die Daten der Räume zu sehen.



Diese kommen aus einer eigens dafür eingerichteten Tabelle in der Datenbank **Seminar** (ODBC Datenquelle). Natürlich er -

scheint eine Datenbank-Tabelle für die Verwaltung von nur drei Räumen etwas übertrieben. Hier wird aber veranschaulicht, wie die Steuerung der angezeigten Datensätze auch über andere Komponenten als eben immer nur über die gewöhnlichen Schaltflächen geschehen kann. Neben den fest in der Tabelle definierten Feldern findet man hier auch solche, die aus den Angaben der Datenbank ein Ergebnis errechnen. Dies ist ein Tip für den Umgang mit Datenbanken im allgemeinen. Sind nämlich Informationen z.B. über die Breite und Länge des jeweiligen Raumes vorhanden, dann ist es nicht notwendig die Datenbank selbst mit den Daten über die Quadratmeterzahl zu belasten. Vielmehr ergibt sich diese Zahl immer wieder aus der Multiplikation der beiden Werte. So würde es sich auch bei Steuern verhalten. Ganz unten sind noch zwei Schaltflächen zu sehen, die auf die Seiten drei und vier der Anwendung verzweigen.

Auf der rechten Seite des Bildschirms sehen Sie grundrißartig die Räume skizziert. Man kann darüber streiten, mit welcher Detailverliebtheit man sich an die Darstellung macht. Auf jeden Fall entsprechen die Proportionen der Räume auf dem Bildschirm den Längenangaben in der Datenbank. Dies kann erreicht werden, indem man die Größenverhältnisse der wirklichen Räume durch eine geeignete Zahl teilt, und bei der Gestaltung der Räume im Rechner auf die Angaben links unten in der Statuszeile achtet. Die Türen in den Räumen wurden einfach durch randlose Rechtecke erzeugt. In den folgenden Abbildungen sehen Sie die Elemente des Grundrisses mit und ohne Umrahmung. Das soll es Ihnen erleichtern, eine ähnliche Optik zu erreichen. Die Räume selbst reagieren mit einer Veränderung der Farbe oder des Füllmusters, wenn Sie an- oder abgewählt werden. So gibt es keine Zweifel, welcher Raum gerade bearbeitet wird.



In der Mitte schließlich findet man am oberen Rand das "Lager" für alle Gegenstände und Leistungen, die zu dem nackten Raum angemietet werden können. Von der Aus-rüstung des Raumes mit PCs für jeden Teilnehmer über Essen reicht hier das Angebot bis hin zu einem Videobeamer.



Die Preise der jeweiligen Gegenstände und Leistungen wurden in dieser Anwendung nicht in einer Datenbank-Tabelle gespeichert, um so möglichst verschiedene Übungsbeispiele zu bekommen. Im richtigen Leben erschiene es sicher sinnvoll, eine eigene Datenbank-Tabelle einzurichten, die zentral gepflegt wird. Die einzelnen Elemente des Equipments können mit der Maus auf den gewünschten Raum gezogen werden und erhöhten auf diese Weise entsprechend die Miete. Die Ausstattung mit PCs und das Essen sind personenbezogen und hängen somit von der Anzahl Teilnehmer ab. Für die Miete eines Diaprojektors kann dieses Verfahren natürlich nicht

angewendet werden. Hier spielt es keine Rolle ob 5 oder 15 Menschen die Bilder betrachten. Für den Anwender dieser Applikation wird es möglich sein, die Grafiksymbole einfach per Doppelklick wieder "aufzuräumen".

Unterhalb des Equipments werden die Kunden daten vermerkt.

Diese Felder sollten sich Eingabeversuchen entziehen, damit vermieden wird, einen Kunden falsch oder einen Neukunden garnicht oder unvollständig einzugeben. Die Schaltfläche *Customer* verzweigt auf die Seite 2 der Anwendung, wo der Kunde angewählt oder erfaßt werden kann. Eine eventuelle Auswahl eines Kunden auf Seite 2 wird dann in diese Felder übertragen.

Die beiden Scrollbars lassen Eingaben zu der Anzahl der Teilnehmer und der Anzahl der zu buchenden Tage zu. Die Anzahl der Tage ist nicht wirklich beschränkt. Bei der Anzahl der Teilnehmer richtet sich die über die Scrollbar auszuwählende Zahl jedoch nach der Maximalbelegung des angewählten Raumes. Diese geht aus dem entsprechenden Feld in der Datenbank-Tabelle *semiroom* hervor.

Das darunterliegende Feld *edttotal* zeigt den aktuellen Berechnungsstand in Abhängigkeit von Raum, Equipment, Teilnehmerzahl und Anzahl der gebuchten Tage an. Die Schalt -

fläche darunter *Book now* schreibt die erforderlichen Daten in eine Zeile der Datenbank-Tabelle *semibook*. Dies allerdings nur, wenn die Angaben komplett sind. Die Schaltfläche darunter, beendet die Anwendung.

Auf dieser Seite existieren zwei Datenbank Komponenten, und zwar für die Tabellen *semibook* und *semiroom*. Die Vorsilbe *semi* soll hier nichts mit "halb" zu tun haben, sondern lediglich die Zugehörigkeit der Tabelle zu der Datenquelle *seminar* und der Anwendung zeigen. Bitte beachten Sie die Einstellungen der Optionen in den Dialogen **Datenbankmodus** und **Feldzuweisungen**.

Die Datenbank Komponenten sind zwar grundsätzlich im Wege, und stehen deshalb Toolbars und ähnlichem in nichts nach, aber zur Laufzeit verschwinden Sie ja wieder.

Bau der Seite 1

Beginnen wir mit der Konstruktion der Seite 1. Erstellen Sie zunächst das, die Räume umgebende Rechteck *rcmain*. Es hat die Maße (320,5) - [315x375] Pixel. Beim Zeichnen der Rechtecke können Sie in der Statuszeile links unten am Bildschirm erkennen, wie groß die Rechtecke jeweils werden. Damit dieses Rechteck nicht versehentlich immer wieder angewählt wird, verwenden Sie bitte den Menübefehl **Eigenschaften | Fixieren**, während das Rechteck angewählt ist. Wie aus der vorhergehenden Beschreibung ersichtlich ist, entsprechen die Maße der Räume proportional den Angaben in der Datenbank.

Plazieren Sie die Räume und versehen Sie sie mit Raumnummern und Türen. Installieren Sie nun noch den Empfang und die sanitären Anlagen.

Die Namen der Komponenten können Sie aus der folgenden Liste ersehen

Großes Rechteck	rcmain
Eingang	doormain
Raum 1	room1
Raum 2	room2

Raum 2	room3
Tür Raum 1	door1
Tür Raum 2	door2
Tür Raum 3	door3
Raumnummer 1	txrnr1
Raumnummer 2	txrnr2
Raumnummer 3	txrnr3
Theke	rrcounter
Abdeckung Theke	rccounter
Beschriftung Theke	txcounter
Beschriftung Lobby	txlobby
Toiletten	rcrestrooms
Trennwand Toiletten	rcrestwall
Toilettentür 1	rcrestdoor1
Toilettentür 2	rcrestdoor2
Beschriftung "Rest"	txrest
Beschriftung "rooms"	txrooms

Am oberen Rand des Rahmens auf der linken Seite befindet sich ein Titel. Die Felder darunter sind Eingabefelder mit entsprechenden Textfeldern als Labels. Am unteren Rand befinden sich zwei Schaltflächen.

Großer Rahmen, links	rcmain1
Text als Titel	txtitel
Label Roomnumber	txroomnum
Eingabefeld Roomnumber	edtrroomnum
Label Length	txlength
Eingabefeld Length	edtlength
Label Width	txwidth
Eingabefeld Width	edtwidth

Label Quadratmeter	txsqm
Eingabefeld Quadratmeter	edtsqm
Label Seats	txseats
Eingabefeld Seats	edtseats
Label Rate	txrate
Eingabefeld Rate	edtrate
Label Extras	txextras
Eingabefeld Extras	edtextras
Schaltfläche Equipment	btnequipment
Schaltfläche Bookings	btnbookings

Am oberen Rand der Bildschirmmitte findet man das "Lager" für Equipment, welches zu den Räumen hinzugemietet werden kann. Die Positionen der einzelnen Grafiken ist vorerst nicht von entscheidender Bedeutung, da wir sowieso für jedes Element einen Link vor gesehen haben, der es bei Doppelklick an seine Ausgangsposition befördert.

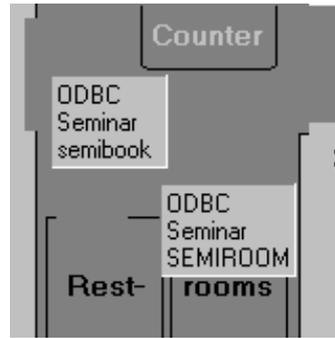
Beamer 1	grfbeam1
Diaprojektor 1	grfdia1
Diaprojektor 2	grfdia2
Flipboard 1	grfflip1
Flipboard 2	grfflip2
Flipboard 3	grfflip3
Lunch 1	grflunch1
Lunch 2	grflunch2
Lunch 3	grflunch3
Tonband 1	grftape1
Tonband 2	grftape2
PC 2:1 1	grfpca1
PC 2:1 2	grfpca2

PC 2:1 3	grfpca3
PC 1:1 1	grfpcb1
PC 1:1 2	grfpcb2
PC 1:1 3	grfpcb3

Der Bereich unter diesem "Lager" enthält Schaltflächen, Scrollbars, Text- und Eingabefelder. Der Scrollbereich der oberen Scrollbar `scrldays` ist von 0 bis 100. Die untere Scrollbar `scrpeople` geht von 0 bis 25.

Schaltfläche Customer	btncustomer
Anzeige Kundennummer	edtcustnum
Anzeige Kundenname	edtcustomer
Label Days	txdays
Anzeige Days	edtdays
Scrollbar Days	scrldays
Label People	txpeople
Anzeige People	edtpeople
Scrollbar People	scrpeople
Label Total	txttotal
Anzeige Total	edttotal
Schaltfläche Book Now	btnbooknow
Schaltfläche End	btnend

Nun benötigen wir noch zwei Datenbank Komponenten mit den Komponenten-Namen `dbroom` für die Datenbank-Tabelle `semiroom` und `dbbook` für die Tabelle `semibook`.



Beide Datenbank Komponenten werden beim Starten der Anwendung geöffnet. Folgende Links bestehen zum Datenbank Komponenten dbroom.

DBROOM	
RNUMBER	edroomnum
RLENGTH	edtlength
RWIDTH	edtwidht
RMAXSEATS	edtseats
RFACTOR	edtrate

Die Datenbank Komponente dbbook wird zur Speicherung der Buchung verwendet.

Bevor wir die Links zwischen den Komponenten auf der Seite 1 herstellen, erzeugen wir die übrigen Seiten, da sich einige Links auf Komponenten dieser Seiten beziehen.

Seite 2 - Kundeninforma- tionen

Customers

C_NUM	<input type="text" value="1"/>	CITY	<input type="text" value="San Francisco"/>
COMPANY	<input type="text" value="CompTrain Inc."/>	ZIP_CODE	<input type="text" value="1234"/>
F_NAME	<input type="text" value="Tony"/>	STATE	<input type="text" value="California"/>
L_NAME	<input type="text" value="Clark"/>	PHONE	<input type="text" value="1773846"/>
STREET	<input type="text" value="122 Madison Av."/>	FAX	<input type="text" value="1775856"/>

Diese Seite enthält eine Datenbank Komponente, sowie Felder und Schaltflächen, um die Daten in der Datenbank zu bearbeiten. Diese Datenbank Komponente wird, wie auch die der folgenden Seiten, erst bei Aktivierung der Seite geöffnet und bei Deaktivierung wieder geschlossen. Es erscheint nicht erforderlich, daß dies beim Start der Applikation geschieht. Diese Einstellung ist über den Datenbank Assistenten leicht einzustellen, kann aber auch nachträglich geändert werden. Ansonsten begegnen Sie auf dieser Seite nichts Neuem. Die Schaltflächen *Carry Back* und *Cancel* verzweigen einfach wieder auf die Seite 1; einmal mit und einmal ohne eine Übernahme der ausgewählten Daten nach Seite 1.

Äußerer Rahmen	rcmain2
Innerer Rahmen	rcinner2
Titel	txtitel2

Datenbank Komponente SEMICUST	dbcust
Schaltfläche erster Satz	btndbfirst2
Schaltfläche nächster Satz	btndbnext2
Schaltfläche vorheriger Satz	btndbback2
Schaltfläche letzter Satz	btndblast2
Schaltfläche Einfügen	btndbinsert2
Schaltfläche Update	btndbupdate2
Schaltfläche Löschen	btndbdelete2
Schaltfläche Suchen	btndbsearch2
Schaltfläche Clear	btndbclear2
Schaltfläche Carry Back	btncarryback
Schaltfläche Cancel	btncancel
Label Kundennummer	txc_num2
Eingabefeld Kundennummer	edtc_num2
Label Firma	txcompany2
Eingabefeld Firma	edtcompany2
Label Vorname	txf_name2
Eingabefeld Vorname	edtf_name2
Label Nachname	txl_name2
Eingabefeld Nachname	edtl_name2
Label Street	txstreet2
Eingabefeld Street	edtstreet2
Label City	txcity2
Eingabefeld City	edtcity2
Label Zipcode	txzipcode2
Eingabefeld Zipcode	edtzipcode2
Label Staat	txstate2
Eingabefeld Staat	edtstate2
Label Phone	txphone2

Eingabefeld Phone	edtphone2
Label Fax	txfax2
Eingabefeld Fax	edtfax2

Die Links zu der Datenbank Komponente sind analog zu den entsprechenden Feldnamen.

Seite 3 - Equipment

Diese Seite sieht wie folgt aus:

Equipment

<input checked="" type="checkbox"/> 1-1	One PC per person	<input type="text" value="0"/>
<input checked="" type="checkbox"/> 1-2	One PC for two persons	<input type="text" value="0"/>
<input type="checkbox"/>	Tape recorder	<input type="text" value="0"/>
<input type="checkbox"/>	Video Beamer	<input type="text" value="0"/>
<input type="checkbox"/>	Lunch	<input type="text" value="0"/>
<input type="checkbox"/>	Flip chart	<input type="text" value="0"/>
<input type="checkbox"/>	Dia projection	<input type="text" value="0"/>

Auf dieser Seite existiert keine Datenbank Komponente. Statt dessen verwenden wir eine andere Funktion von MINDMAP. Es besteht nämlich die Möglichkeit eine *.ini-Datei anzulegen und abzufragen. Natürlich hätten wir dies auch für die lächerlichen drei Räume machen können, aber bitte vergessen Sie nicht, dies alles dient der Übung und dem Kennenlernen von MINDMAP. Viele Wege führen nach Rom und vielleicht ein paar mehr zu einer Anwendung.

Eine INI-Datei ist eine Datei, die vor der Ausführung eines Programms, oder auch während der Ausführung, gelesen wird.

Sie enthält Voreinstellungen, die den Verlauf eines Programmes bestimmen können. Die Endung *INI* steht für Initialisierung. Gequälte Windows-Anwender kennen das: *WIN.INI*, *SYSTEM.INI* und so weiter und so fort. Im richtigen Leben hat man kaum eine Kontrolle darüber, welches installierte Programm, welche vernichtenden Einträge in den INI-Dateien vornimmt. Außerdem gibt es in der Regel keinerlei Informationen darüber, wie man diese wieder entfernt. Jetzt allerdings sind Sie am Zug. Auf diesen Augenblick haben Sie lange warten müssen. MINDMAP macht es möglich! Sie alleine können bestimmen, welche Einträge in Ihre private INI-Datei kommen und welche nicht. Natürlich nur solange Sie das tun, was wir Ihnen vorschreiben! Wir legen also eine INI-Datei an, und lesen beim Programmstart die Werte wieder aus. Und nachdem wir jetzt derart großartige Ankündigungen gemacht habe, verträsten wir Sie aus lauter Sadismus auf später.

Die Berechnungen auf Seite 1 hängen von den Einträgen in den Feldern dieser Seite ab. Für die Aufnahme der Daten wäre auch eine Tabelle geeignet gewesen; wir hätten dann aber nicht die hübsche grafische Entsprechung mit den Symbolen. Die Seite 3 wird aufgerufen, wenn auf Seite 1, links unten, die Schaltfläche *See equipment* gedrückt wird. Mit *Back* kehrt man wieder zur Seite 1 zurück.

Äußerer Rahmen	rcmain3
Symbol Beamer	grfbeamer
Symbol Diaprojektor	grfdia
Symbol Tonband	grftape
Symbol Flipchart	grfflip
Symbol Lunch	grflunch
Symbol PC 1:1	grfpca
Symbol PC 2:1	grfpcb
Innerer Rahmen	rcinner3
Label PC 1:1	txpca3
Label PC 1:2	txpcb3
Label Tonband	txtape

Label Beamer	txbeamer3
Label Lunch	txlunch3
Label Flipchart	txflip3
Label Diaprojektor	txdia3
Eingabefeld PC 1:1	p_pc1
Eingabefeld PC 1:2	p_pc2
Eingabefeld Tonband	p_tape
Eingabefeld Beamer	p_beamer
Eingabefeld Lunch	p_lunch
Eingabefeld Flipchart	p_flip
Eingabefeld Diaprojektor	p_dia
Schaltfläche Back	btnback3
Titel	txtitel3
Schaltfläche Sichern	btnsave

Seite 4 - Booking

Booking

CUSTNUM	<input type="text" value="2"/>	COMPANY	<input type="text" value="Bits and Bytes"/>
ROOMNUM	<input type="text" value="0"/>	F_NAME	<input type="text" value="Linda"/>
BOOKDAYS	<input type="text" value="15"/>	L_NAME	<input type="text" value="Rocco"/>
BOOKPPEL	<input type="text" value="25"/>	PHONE	<input type="text" value="98734098784"/>

Hier finden wir gleich zwei Datenbank Komponenten. Auch diese werden erst bei Aktivierung der Seite geöffnet. Die beiden Datenbank Komponenten sind über einen Join miteinander verbunden. Das soll ermöglichen, daß zu der Kundennummer in der Datenbank-Tabelle *semibook* der Kunde aus der Datenbank-Tabelle *semicust* angezogen wird. Die Links der Datenbank Komponente zu den Feldern verhalten sich wieder analog zu den Feldnamen. Der Link zwischen den Tabellen, also der Join, geschieht über das Feld *custnum*. Bitte beachten Sie die Einstellungen der Optionen in den Dialogen **Datenbankmodus** und **Feldzuweisungen**.

Äußerer Rahmen	rcmain4
Innerer Rahmen	rcinner4
Titel	txtitel4
Schaltfläche Zurück	btnback4
Datenbank Komponente SEMICUST	cust4
Datenbank Komponente SEMIBOOK	booking4

Schaltfläche erster Satz	btndbfirst4
Schaltfläche nächster Satz	btndbnext4
Schaltfläche vorheriger Satz	btndbback4
Schaltfläche letzter Satz	btndblast4
Schaltfläche Satz Einfügen	btndbinsert4
Schaltfläche Satz ändern	btndbupdate4
Schaltfläche Satz löschen	btndbdelete4
Schaltfläche Satz suchen	btndbsearch4
Schaltfläche Clear	btndbclear4
Label Kundennummer	txcustnum4
Eingabefeld Kundennummer	edtcustnum4
Label Raumnummer	txroomnum4
Eingabefeld Raumnummer	edtroomnum4
Label Gebuchte Tage	txbookdays4
Eingabefeld Gebuchte Tage	edtbookdays4
Label Teilnehmer	txbookpple4
Eingabefeld Teilnehmer	edtbookpple4
Label Firma	txcompany4
Eingabefeld Firma	edtcompany4
Label Vorname	txf_name4
Eingabefeld Vorname	edtf_name4
Label Nachname	txl_name4
Eingabefeld Nachname	edtl_name4
Label Telefon	txphone4
Eingabefeld Telefon	edtphone4

Nun sind die Komponenten aller Seiten vorhanden und be - dürfen nur noch der Interaktion untereinander. Wir fügen also im nächsten Teil alle Links ein, die erforderlich sind. Beginnen wir auf der Seite 1.

Links der Seite 1

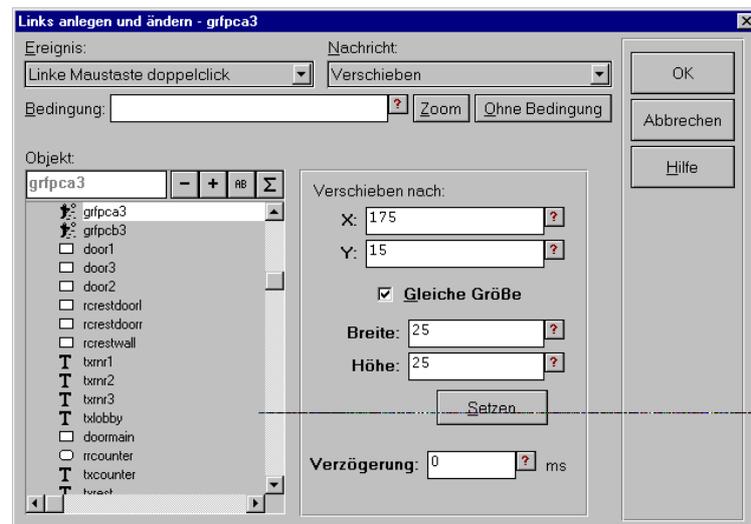
Beginnen wir mit den kleinen Symbolen, die das Equipment darstellen. Diese Komponenten enthalten einen Link, der sie ohne Mühe wieder an ihren ursprünglichen Platz befördert und einen, der für die Neuberechnung des Eingabefeldes *edtextures* verantwortlich ist.

Linke Maustaste losgelassen: Aufruf von edtextures

Dieser Link verursacht eine Verzweigung auf die Komponente edtextures. Bei dieser Komponente wird auf die Nachricht **Verzweigung** mit einer Neuberechnung reagiert. Für die korrekte Neuberechnung sind drei Links erforderlich. Da das Neuberechnen von verschiedenen Stellen aus initiiert werden soll, erscheint es sinnvoll diesen Vorgang an einer zentralen Stelle durchzuführen und durch eine **Verzweigung** lediglich auszulösen.

Der zweite Link sieht folgendermaßen aus:

Linke Maustaste doppelklick: Verschiebe < Komponentename> (pos1,pos2)[size1 x size2]



In der folgenden Tabelle stehen die Namen der grafischen Komponenten und die Positionen, wie sie in dem Beispiel verwendet wurden. Die Angaben über die Größe bleiben bestehen. In dem Rechten Klammerpaar steht immer [25 x 25].

grfbeam1	290, 5	p_beamer
grftape1	265, 5	p_tape
grftape2	270, 10	p_tape
grfdia1	245, 10	p_dia
grfdia2	240, 5	p_dia
grflunch1	215, 5	edtpeople * p_lunch
grflunch2	220, 10	edtpeople * p_lunch
grflunch3	225, 15	edtpeople * p_lunch
grfflip1	190, 5	p_flip
grfflip2	195, 10	p_flip
grfflip3	200, 15	p_flip
grfpca1	165, 5	edtpeople * p_pc1
grfpca2	170, 10	edtpeople * p_pc1
grfpca3	175, 15	edtpeople * p_pc1
grfpcb1	140, 5	edtpeople * p_pc2
grfpcb2	145, 10	edtpeople * p_pc2
grfpcb3	150, 15	edtpeople * p_pc2

Die Angaben in der rechten Spalte dieser Tabelle stellen den Inhalt des Parser-Fensters dar. Bitte tragen Sie für jede der Komponenten den entsprechenden Inhalt in das Parserfeld ein.



Hierzu wird, während die Komponente angewählt ist, das Parserfenster rechts unten am Bildschirm auf geklappt und der Eintrag vorgenommen. Sie brauchen diese Eingaben nicht mit `Enter` zu bestätigen. Wählen Sie nach der Eingabe einfach die nächste Komponente an.



Die Einträge im Parsereditor stellen den Wert der Komponente dar. Dieser Wert wird aus den Angaben zum Equipment auf Seite 3 bezogen. Das "p" im Komponentennamen soll eine Abkürzung für Preis darstellen. Da die Ausstattung mit PCs und das Essen abhängig von der angemeldeten Personenzahl sind, werden sie im Parser mit dem Wert im Feld `edtpeople` multipliziert. Das Feld `edtpeople` wird über die Scrollbar `scrpeople` eingestellt. Da die Parserfunktionen dynamisch sind, spielt es für die Berechnung keine Rolle, ob die Personenzahl vor oder nach der Verwendung des Equipments eingestellt wird.



Stellen Sie bitte sicher, daß das Attribut **Runmode Optionen** für jede der Graphik Komponenten auf *beliebig* eingestellt ist.

Starten Sie doch testweise einmal die Anwendung, und ziehen Sie einige der Graphik Komponenten an eine andere Position. Führen Sie danach einen Doppelklick auf diese Komponenten aus. Wie Sie jetzt sehen müßten, ist das "Aufräumen" über die Anwendung wesentlich einfacher als das tatsächliche Humschleppen der Teile.

Als nächstes nehmen wir uns die Räume an sich vor. Sie sollen folgende Eigenschaften haben:

- ▶ Durch die Auswahl eines Raumes soll der entsprechende Datensatz in der Datenbank-Tabelle *semiroom* angezogen werden, damit in den Eingabefeldern auf der linken Seite die Raumangaben erscheinen, und eine korrekte Neuberechnung der Kosten durchgeführt werden kann.
- ▶ Nachdem die Daten für einen bestimmten Raum aus der Datenbank in die entsprechenden Eingabefelder gefüllt worden sind, wird aus den Angaben Raumlänge `edtlength` und Raumbreite `edtwidth` die Quadratmeterzahl des Raums berechnet.

- ▶ Die Auswahl eines der Räume soll diesen und die dazu gehörige "Tür" weiß einfärben, damit ersichtlich wird, um welchen Raum es sich handelt.
- ▶ Die anderen Räume und Türen sollen wieder ihre ursprüngliche Farbe, hier grau, bekommen, wenn ein Raum angewählt wird.

Sie können statt unterschiedlicher Farben auch andere Füllmuster verwenden. So macht sich ein diagonales Streifenmuster statt dem dunkleren Grau auch sehr schick.

Die Links auf *room1* beispielsweise sehen folgendermaßen aus:

Linke Maustaste losgelassen: dbroom: SQL Select " where rnumber=1"

Linke Maustaste losgelassen: Wertzuweisung: edtsqm« edtlength * edtwidth

Linke Maustaste losgelassen: room1: Farben ändern in <weiß>

Linke Maustaste losgelassen: room2: Farben ändern in <grau>

Linke Maustaste losgelassen: room3: Farben ändern in <grau>

Linke Maustaste losgelassen: door1: Farben ändern in <white>

Linke Maustaste losgelassen: door2: Farben ändern in <grau>

Linke Maustaste losgelassen: door3: Farben ändern in <grau>

Linke Maustaste losgelassen: Aufruf von edtextras

Die Links auf den anderen beiden Räumen unterscheiden sich nur dadurch von diesen, daß im ersten Link der Datensatz des 2. oder 3. Raums aufgerufen wird, und selbstverständlich durch analoge Farbeinstellungen. Bei Auswahl von *room2* wird dieser weiß und die anderen beiden grau, usw.

Auch hier wird eine **Verzweigung** zu edtextras durchgeführt. Es kann ja vorkommen, daß sich das Equipment in einem nicht angewählten Raum befindet. Die Auswahl dieses Raumes muß dann auch zu einer Neuberechnung führen.

Da die Daten der Datenbank-Tabelle *semiroom* ausschließlich dann angezeigt werden sollen, wenn man o. g. Link auf die Datenbank Komponente *dbroom* ausführt, ist es erforderlich, eine bestimmte Einstellung bei den Attributen dieser Datenbank Komponente vorzunehmen. Selektieren Sie bitte hierfür die Komponente *dbroom* und wählen dort das Attribut **Datenbankmodus** aus. Deaktivieren Sie in diesem Dialog die Option *Ersten Datensatz anzeigen nach Datenbank öffnen* .

Nachdem die rechte Seite der Anwendung nun fertig ist, wenden wir uns der linken Seite zu. Wir wählen die Titel Komponente *txtitel* aus, um die Anwendung zu initialisieren. Damit ist gemeint, daß alle irreführenden Wertangaben auf *0* gestellt, und die Räume abgewählt werden. Außerdem wird unsere INI-Datei gelesen, und die Werte in die Felder von Seite 3 eingetragen. Das Zurücksetzen der Werte in den Feldern und die Einfärbung der Räume wären eigentlich nicht unbedingt erforderlich. MINDMAP behält seine letzte Einstellung nur im Edit Mode. Wird die Anwendung weitergegeben, werden die Felder lediglich die Werte haben, die sie beim Erzeugen der EXE-Datei hatten. Deshalb ist es ratsam, darauf zu achten, wie die Anwendung aussieht, bevor man sie lauffähig macht.

Die Links dieser Komponente folgen hier:

Anwendung gestartet: Wertzuweisung: <i>edroomnum</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edlength</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edwidth</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edtsqm</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edtseats</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edextras</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edrate</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edtdays</i> « 0
Anwendung gestartet: Wertzuweisung: <i>edtpeople</i> « 0

Anwendung gestartet: Wertzuweisung: edtcustnum« 0

Anwendung gestartet: Wertzuweisung: edtcustomer « "Please
select customer!"

Anwendung gestartet: room1 : Farben ändern in <grau>

Anwendung gestartet: room2 : Farben ändern in <grau>

Anwendung gestartet: room3 : Farben ändern in <grau>

Anwendung gestartet: door1 : Farben ändern in <grau>

Anwendung gestartet: door2 : Farben ändern in <grau>

Anwendung gestartet: door3 : Farben ändern in <grau>

Anwendung gestartet: Wertzuweisung: p_pc1 «
readprofile("Equipment", "PC_A",
0,GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_pc2 «
readprofile("Equipment", "PC_B", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_tape «
readprofile("Equipment", "Tape", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_beamer «
readprofile("Equipment", "Beamer", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_lunch «
readprofile("Equipment", "Lunch", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_flip «
readprofile("Equipment", "Flip", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Anwendung gestartet: Wertzuweisung: p_dia «
readprofile("Equipment", "Dia", 0,
GetHomeDir("samples\seminar\seminar.ini"))

Auf die Datei **seminar.ini** kommen wir später noch einmal zurück, Wenn es darum geht, diese zu erstellen oder zu verändern. Nachfolgend sehen Sie eine Erklärung der Parser-Funktion `readprofile` und den Aufbau der Datei selbst.

Der Befehl `readprofile` liest eine INI-Datei ein und sucht nach einer Rubrik und darunter nach einem bestimmten Begriff. Rückgabewert ist der Wert, der, durch Gleichheitszeichen abgetrennt, hinter diesem Suchbegriff steht. Wird der gesuchte Begriff nicht gefunden, wird ein Defaultwert ausgegeben, den man selbst bestimmen kann. Die Syntax lautet:

```
READPROFILE(<Rubrik>, <Suchbegriff>,
            <Defaultwert>, <Dateiname>)
```

Als Dateinamen haben wir wiederum eine Parser-Funktion verwendet. Die Funktion `GetHomeDir` gibt das Verzeichnis zurück, aus dem MINDMAP gestartet wurde. Der darauf folgende Eintrag führt in das Unterverzeichnis `samples/seminar` und trägt den Namen **seminar.ini**.

Die Datei **seminar.ini** enthält in diesem Beispiel folgenden Inhalt:

```
[Equipment]
```

```
PC_A=20
PC_B=15
Tape=20
Beamer=80
Lunch=18
Flip=10
Dia=20
```

Natürlich wäre es möglich und vielleicht auch sinnvoll gewesen, die Daten der Räume auf ähnliche Weise und innerhalb derselben Datei abzulegen. Hierbei sei noch einmal an den Übungscharakter dieser Anwendung erinnert.

Eine andere Komponente auf der linken Seite des Bildschirms, das eine Funktionalität aufweist, ist das Eingabefeld `edtexttras`. Hier existieren einige Links, die eine Neuberechnung durchführen, wenn es das **Ziel einer Verzweigung** ist, d. h. wenn man von einer anderen Komponente zu diesem Eingabefeld verzweigt.

Wenn: <code>edroomnum=1 AND edtpeople > 0 AND edtdays > 0</code>
--

Ziel einer Verzweigung: Wertzuweisung: edtextras «
gsum(room1)

Wenn: edtroomnum=2 AND edtpeople > 0 AND edtdays > 0
Ziel einer Verzweigung: Wertzuweisung: edtextras «
gsum(room2)

Wenn: edtroomnum=3 AND edtpeople > 0 AND edtdays > 0
Ziel einer Verzweigung: Wertzuweisung: edtextras «
gsum(room3)

Wenn: edtdays = 0 OR edtpeople = 0
Ziel einer Verzweigung: Wertzuweisung: edtextras « 0

Im Klartext heißt das:

Wenn das Eingabefeld edtroomnum (Die angewählte Raumnummer) den Wert 1 enthält, also Raum 1 angewählt ist, und die Eingabefelder edtpeople und edtdays nicht auf Null stehen, dann wird die grafische Summe der Komponente room1 berechnet und an das Eingabefeld edtextras übertragen. Analoges gilt für die Fälle, in denen Raum 2 oder Raum 3 angewählt worden ist.

Die Syntax der Parser-Funktion zur Berechnung der grafischen Summe lautet `gsum(<Komponentenname>)`. Durch diese Formel wird die Summe der Werte aller Komponenten ermittelt, die sich innerhalb der angegebenen Komponente befinden. Zum Beispiel hat der Beamer seinen Wert aus dem Eingabefeld `p_beamer` auf Seite 3, welches selbst durch den Inhalt in der Datei `seminar.ini` versorgt wird. Ziehen wir den Beamer in den Raum 1, dann wird über `gsum(room1)` der Wert ausgegeben. Ziehen wir eine weitere Komponente in diesen Raum, werden die Werte dieser Komponente addiert. Entfernen wir eine Komponente aus dem Raum, wird die Summe neu berechnet. Diese Funktion kann für sehr viele Applikationen nützlich sein, egal ob es sich um Vermietung, Verleih, Einkauf, Verkauf oder ähnliches handelt. So wäre es auch denkbar, einen Netzwerkkonfigurator als Planungsinstrument zu konstruieren, der es dem Techniker und Einkäufer ermöglicht, die Netzwerkkomponenten in einen Kreis zu ziehen, und gleich die Kosten zu erfahren. Die Werte müssen nicht unbedingt monetär sein. Vielleicht wird beim Beladen eines LKWs das Gesamtgewicht ermittelt, oder bei

Flugzeugen die Spritmenge in Abhängigkeit zu den Gewichten der mitfliegenden Personen und ihrem Gepäck. Oder vielleicht möchte man die Kalorien eines Kuchens aufgrund seiner Bestandteile ausrechnen lassen. Denken Sie an Ihre Diät!

Genug der Anregungen. Der letzte Link in dieser Liste soll das Feld `edtexttras` wieder auf Null zurücksetzen, wenn keine Personen oder Tage eingegeben worden sind.

Jetzt folgen die Links der unteren beiden Schaltflächen in diesem Bereich. Die Schaltfläche *See Equipment* soll auf die Seite 3 der Anwendung verzweigen.

Linke Maustaste losgelassen: Aufruf von Seite 3

Mit der Schaltfläche *See Bookings* verzweigt man auf die Seite 4.:

Linke Maustaste losgelassen: Aufruf von Seite 4

Jetzt fehlen uns nur noch die Links für die Komponenten im mittleren Bereich. Die Schaltfläche *Customer* befördert uns durch

Linke Maustaste losgelassen: Aufruf von Seite 2

zu Seite 2 der Anwendung, in der ein Kunde ausgewählt oder eingegeben werden kann.

Die Eingabefelder `edtdays` und `edtpeople` haben keine Links. Die Werte, die sie anzeigen, werden ihnen von der jeweiligen Scrollbar, `scrldays`, bzw. `scrpeople`, zugewiesen. Aus diesen Eingabefeldern wird auch in die Datenbank abgespeichert.

Das Eingabefeld `edttotal` hat ebenfalls keinen Link. In diesem Eingabefeld wird immer die Gesamtmiete angezeigt, die bei Änderung der Scrollbars neu berechnet wird.

Die Scrollbar `scrldays` hat folgende Links:

Scrollbar Änderung: Wertzuweisung: `edtdays << scrldays`

Scrollbar Änderung: Aufruf von `edtexttras`

Scrollbar Änderung: Wertzuweisung: `edttotal << (edtexttras+(edtrate*edtsqm))*edtdays`

Die Links der Scrollbar *scrpeople* sehen wie folgt aus:

Scrollbar Änderung: Wertzuweisung: `edtpeople << scrpeople`

Scrollbar Änderung: Aufruf von `edtexttras`

Scrollbar Änderung: Wertzuweisung: `edttotal << (edtexttras+(edtrate*edtsqm))*edtdays`

Die Links der Schaltfläche *Book Now!* sehen wie folgt aus:

Wenn: `edtroomnum>0 AND edtpeople>0 AND edtdays>0`
Linke Maustaste losgelassen: `dbbook: SQL Kommando "insert into semibook values (" + edtcustnum + " , "+ edtroomnum + " , "+ edtdays + " , "+ edtpeople + ")"`

Wenn: `edtroomnum>0 AND edtpeople>0 AND edtdays>0`
Linke Maustaste losgelassen: `btnbooknow: Objekt ändern in "Booked !"`

Wenn: `edtroomnum<1 OR edtpeople<1 OR edtdays<1`
Linke Maustaste losgelassen: `btnbooknow: Objekt ändern in "SELECTION UNCOMPLETE!"`

Wenn: `edtroomnum>0 AND edtpeople>0 AND edtdays>0`
Mausbewegung aus dem Objekt: `btnbooknow: Objekt ändern in "Book now"`

Wenn: `edtroomnum<1 OR edtpeople<1 OR edtdays<1`
Mausbewegung in das Objekt: `btnbooknow: Objekt ändern in "Book now!"`

Natürlich ist der erste Link der wichtigste. Er sorgt dafür, daß in die Datenbank-Tabelle *semibook*, die Buchung eingetragen wird. Das erfolgt durch einen SQL-Befehl. Dieser Befehl wird aus Text und Inhalten von Eingabefeldern zusammengesetzt, weshalb erklärlich wird, warum es etwas konstruiert aussieht. Wäre das Statement statisch, würde es so lauten:

```
Insert into semibook values (value1, value2,
                             value3, value 4)
```

Um diesen Datensatz in die Datenbank anzufügen, hätte man natürlich nicht den SQL-Befehl selbst generieren müssen. Selbstverständlich wäre auch das Datenbank-Kommando **Datensatz anfügen** möglich gewesen, welches wir bereits

kennen. MINDMAP hätte dann intern den SQL-Befehl generiert; wir wären nicht weiter damit be helligt worden. Doch wir wollen ja etwas lernen!

Hierbei sind die Kommata als Trennzeichen wichtig. Wir verketten also die Textteile und die Werte der Eingabefelder mit "+"-Zeichen und erhalten dann das beschriebene Statement. Die Bedingung *Wenn* soll dafür sorgen, daß nur Sätze weg geschrieben werden, die auch alle notwendigen Einträge enthalten. Die darauf folgenden Links verändern die Beschriftung der Schaltfläche, abhängig davon, ob die Buchung erfolgreich war oder nicht. Die Beschriftung der Schaltfläche ändert sich, je nachdem ob sich die Maus über die Schaltfläche bewegt oder sich außerhalb befindet.

Die letzte Komponente dieser Seite ist die Schaltfläche *End Program*. Sie führt über einen Link ein Menükommando durch:

Linke Maustaste losgelassen: Befehl Anwendung schließen

Jetzt sind alle Komponenten der Seite 1 beschrieben. Weiter geht es mit der Arbeit an Seite 2

Die Links der Seite zwei

Die Komponenten der Seite zwei tragen folgende Namen:

Äußerer Rahmen	rcmain2
Innerer Rahmen	rcinner2
Titel Customers	txtitel2
Datenbank Komponente SEMICUST	dbcust2
Schaltfläche erster Satz	btndbfirst2
Schaltfläche vorheriger Satz	btndbback2
Schaltfläche nächster Satz	btndbnext2
Schaltfläche letzter Satz	brndblast2
Schaltfläche Satz einfügen	btndbinsert2

Schaltfläche Satz löschen	btndbdelete2
Schaltfläche Satz ändern	btndbupdate2
Schaltfläche Satz suchen	btndbsearch2
Schaltfläche Clear	btndbclear2
Label Customernumber	txc_num2
Eingabefeld Customernumber	edtc_num2
Label Company	txcompany2
Eingabefeld Company	edtcompany2
Label Vorname	txf_name2
Eingabefeld Vorname	edtf_name2
Label Nachname	txl_name2
Eingabefeld Nachname	edtl_name2
Label Straße	txstreet2
Eingabefeld Straße	edtstreet2
Label City	txcity2
Eingabefeld City	edtcity2
Label Zipcode	txzipcode2
Eingabefeld Zipcode	edtzipcode2
Label State	txstate2
Eingabefeld State	edtstate2
Label Phone	txphone2
Eingabefeld Phone	edtphone2
Label Fax	txfax2
Eingabefeld Fax	edtfax2
Schaltfläche Carry Back	btncarryback
Schaltfläche Cancel	btncancel

Auf die Links der datenbank-spezifischen Schaltflächen soll hier nicht näher eingegangen werden. Der MINDMAP Datenbank Assistent erstellt sie auf Wunsch. Über deren Funktionalität wurde schon mehrfach gesprochen. Wichtig sind

eigentlich nur die Schaltflächen *Carry Back* und *Cancel*, deren Links im folgenden aufgeführt sind.

Schaltfläche *Carry back*

Linke Maustaste losgelassen: Wertzuweisung: edtcustnum« edtc_num2
Linke Maustaste losgelassen: Wertzuweisung: edtcustomer« edtcompany2
Linke Maustaste losgelassen: Aufruf von Seite 1

Der erste Link überträgt die Kundennummer des angewählten Kunden in das entsprechende Eingabefeld auf Seite 1. Der zweite macht dies analog für den Namen des Kunden. Durch den dritten Link schließlich wird zu Seite 1 zurückgekehrt.

Schaltfläche *Cancel*

LinkeMaustaste losgelassen: Wertzuweisung: edtcustnum« 0
LinkeMaustaste losgelassen: Wertzuweisung: edtcustomer« "No Customer selected!!"
LinkeMaustaste losgelassen: Aufruf von Seite 1

Hier geschieht ähnliches wie bei der Komponente *Carry Back*. Nur werden nicht die Kundennummer und der Kundename übernommen, sondern gewissermaßen als Warnung der Hinweis darauf, daß kein Kunde ausgewählt worden ist.

Seite 3 - Equipment

Zunächst folgt eine Auflistung der Komponenten:

Äußerer Rahmen	rcmain3
Innerer Rahmen	rcinner3
Titel Equipment	txtitel3
Symbol Beamer	grfbeamer

Symbol Dia	grfdia
Symbol Tape	grftape
Symbol Flip	grfflip
Symbol Lunch	grflunch
Symbol PC_A	grfpca
Symbol PC_B	grfpcb
Text Beamer	txbeamer3
Text Dia	txdia3
Text Tape	txtape3
Text Flip	txflip3
Text Lunch	txlunch3
Text PC_A	txpca3
Text PC_B	txpcb3
Eingabefeld Beamer	p_beamer
Eingabefeld Dia	p_dia
Eingabefeld Tape	p_tape
Eingabefeld Flip	p_flip
Eingabefeld Lunch	p_lunch
Eingabefeld PC_1	p_pc1
Eingabefeld PC_2	p_pc2
Schaltfläche Back	btnback3
Schaltfläche Save	btnsave

Diese Seite hat nur zwei funktionale Schaltflächen. Die Schaltfläche *Back* enthält lediglich eine Verzweigung auf die Seite 1. Die Schaltfläche *Save* jedoch kann mehr. Sie speichert etwaige Veränderungen in den Eingabefeldern, die die Mietpreise angeben, in die Datei mit dem Namen **seminar.ini**. Wir sind auf diese Funktion bereits bei der Beschreibung der Titeldatenkomponente der Seite 1 eingegangen. Dort werden die Inhalte der Datei ausgelesen und an die Eingabefelder übertragen, die wir hier auf dieser Seite sehen. Diese Werte werden von den gra -

phischen Symbolen auf Seite 1 angenommen. Natürlich hätte man diese Werte auch direkt den Symbolen zuweisen können, dann hätte man ihnen auch jede Änderung extra mitteilen müssen. Auf die gewählte Weise ist es möglich, den Preis einer Komponente temporär zu ändern. Da die Parserfunktion immer dynamisch wirkt, wird eine Veränderung der Preise während eines Buchungsvorganges auch die Summe verändern.

Zunächst erzeugen wir jedoch je einen Link für alle Eingabefelder dieser Seite. Er soll dafür sorgen, daß sich die Beschriftung der Schaltfläche *Save* ändert, wenn sich der Inhalt eines Eingabefelds geändert hat. Das soll den Anwender darauf aufmerksam machen, daß der Wert noch nicht gespeichert worden ist.

Tastatureingabe: btncsave: Schriftarten ändern in <Textfarbe rot>

Natürlich können Sie sich auch etwas wirkungsvolleres einfallen lassen. Für diese Übung soll das jedoch reichen.

Die Links der Schaltfläche *Save* sehen Sie nun:

```
Linke Maustaste losgelassen: Wertzuweisung btncsave:«
    writeprofile("Equipment", "PC_A", p_pc1,
        GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "PC_B", p_pc2,
    GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "Tape", p_tape,
    GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "Beamer", p_beamer,
    GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "Lunch", p_lunch,
    GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "Flip", p_flip,
    GetHomeDir("samples\seminar\seminar.ini"))
writeprofile("Equipment", "Dia", p_dia,
    GetHomeDir("samples\seminar\seminar.ini"))
```

Linke Maustaste losgelassen: btnsave: Schriftarten ändern in
<Textfarbe schwarz>

Die Parser-Funktion `writeprofile` verhält sich analog zu dem Befehl `readprofile`, nur daß er eben schreibt statt liest. Die Parameter innerhalb der Klammern sind etwa die gleichen, nur der Defaultwert existiert nicht. Dafür wird der Wert verwendet, der sich in dem Eingabefeld befindet, welches in der Klammer als Parameter aufgeführt ist. Außer dem schreiben wir alle Zeilen auf einmal in das Feld für die Wert-übergabe. Das stellt übrigens einen Trick dar. Wir verwenden an dieser Stelle die Nachricht **Wertzuweisung** und geben als Ziel die Schaltfläche selbst an. Die Schaltfläche kann mit dem zugewiesenen nicht viel anfangen. Dafür wird auf diese Weise die Befehlskette abgearbeitet. Diese Eigenschaft kann auch in vielen anderen Fällen ein nützliches Werkzeug sein. Auf die Form der INI-Datei sind wir bereits eingegangen. Nach dem Erstellen der Anwendung ist die INI-Datei selbstverständlich noch leer. Vergessen Sie deshalb nicht, geeignete Werte einzutragen, und diese zu speichern!

Der letzte Link setzt die Farbe des Labels der Schaltfläche `Save Textes` nach erfolgter Speicherung wieder zurück auf schwarz.

Seite 4 - Bookings

Auf dieser Seite können die Buchungen eingesehen und verändert werden. Hierzu werden zwei Datenbank-Tabellen verwendet. In der Tabelle `semibook` sind die Buchungen gespeichert. Die Tabelle `semicust` enthält die Einträge der Kunden, deren Kundennummer in der Tabelle `semibook` gezeigt wird.

Äußerer Rahmen	<code>rcmain4</code>
Innerer Rahmen	<code>rcinner4</code>
Datenbank Komponente SEMICUST	<code>cust4</code>
Datenbank Komponente SEMIBOOK	<code>booking4</code>

Titel Bookings	txtitel4
Schaltfläche erster Satz	btndbfirst4
Schaltfläche vorheriger Satz	btndbback4
Schaltfläche nächster Satz	btndbnext4
Schaltfläche letzter Satz	btndblast4
Schaltfläche Satz einfügen	btndbinsert4
Schaltfläche Satz löschen	btndbdelete4
Schaltfläche Satz ändern	btndbupdate4
Schaltfläche Satz suchen	btndbsearch4
Schaltfläche Clear	btndbclear4
Label Customernumber	txcustnum4
Eingabefeld Customernumber	edtcustnum4
Label Roomnumber	txroomnum4
Eingabefeld Roomnumber	edtroomnum4
Label Days Booked	txbookdays4
Eingabefeld Days Booked	edtbookdays4
Label People Booked	txbookppel4
Eingabefeld People Booked	edtbookppel4
Label Company	txcompany4
Eingabefeld Company	edtcompany4
Label Vorname	txf_name4
Eingabefeld Vorname	edtf_name4
Label Nachname	txl_name4
Eingabefeld Nachname	edtl_name4
Label Phone	txphone4
Eingabefeld Phone	edtphone4

Die einzelnen Felder sind entsprechend ihrer Namen mit den Feldern der Datenbank-Tabelle verknüpft. Beide Datenbank-Tabellen sind über die Kundennummer verbunden. Die Tabelle *semibook* treibt in diesem Falle die Tabelle *semicust*.

Die Schaltfläche *Back* sorgt dafür, daß man auf Seite 1 zurückkehren kann..

Gratuliere! Sie sind durch. Diese kleine Anwendung kann nun beliebig erweitert werden. Vielleicht ist eine Buchungsnummer erforderlich, oder das Datum der Buchung. Vielleicht wagen Sie sich auch an eine Terminverwaltung? Es sei noch einmal betont, daß diese Anwendung in ihrem jetzigen Zustand lediglich ein Übungsbeispiel darstellt und kein ernsthaftes Programm.

Sollten sich Schwierigkeiten beim Erstellen oder dem Ablauf des Programmes ergeben, verwenden Sie bitte die Menü-Option **View | View System Log**. Der System Log erscheint zur Laufzeit und gibt Auskunft darüber, was genau im Programmablauf geschieht. Weiterhin gibt es die Möglichkeit, Breakpoints zu setzen. Dies geschieht durch Auswählen der entsprechenden Option im Dialog mit der Linkliste, oder durch die Selektion im Menü **Run | Debug Unterbrechung**. Ist ein Breakpoint gesetzt, bleibt das Programm während der Ausführung an der gewünschten Stelle stehen und läßt unter anderem die Möglichkeit zu, im Einzelschrittmodus fortzufahren. Es ist ein wenig Fingerfertigkeit erforderlich, wo man einen Breakpoint setzen soll, wenn man keinen Verdacht hat, an welcher Stelle, oder wenigstens in welcher Gegend das Programm fehlerhaft ist. Notfalls setzen Sie den Breakpoint an eine sehr "frühe" Stelle und klicken sich Schritt für Schritt durch den Programmablauf.

Die Anwendung **seminar.mm** mit allen dazugehörigen Dateien liegt Ihnen vor.

So, jetzt wissen Sie genug über MINDMAP, so daß Sie alleine weitermachen können. Natürlich konnten wir an dieser Stelle nicht alle Features zeigen, die Ihnen MINDMAP zur Verfügung stellt. Doch haben Sie jetzt das nötige Rüstzeug, um erfolgreich mit MINDMAP zu arbeiten.

Viel Vergnügen und bleiben Sie dran!

Kleines Datenbankbrevier

Allgemeines

Wer sich bisher nicht mit **SQL** beschäftigt hat, dies aber in Zukunft möchte, findet in diesem kleinen Brevier einige Einsteiger Tips, die die ersten Schritte etwas erleichtern mögen.

SQL wurde in der Mitte der 70er Jahre von IBM entwickelt. Im Laufe der Zeit übernahmen andere Softwarehäuser diese Idee und entwickelten ihrerseits **SQL**-Konforme Produkte, die auch auf anderen Plattformen liefen. Auf diese Weise kamen verschiedene Dialekte dieser Sprache auf den Markt, die die eine oder andere Besonderheit der jeweiligen Datenbanken hervorheben. Um wenigstens einen bestimmten Kern der Sprache zu verallgemeinern, definierte ein Komitee des **ANSI** (American National Standards Institute) in den frühen 80ern eine Anzahl Anweisungen, die möglichst allen Herstellern gemein waren. Dementsprechend unterscheidet man heute auch zwischen **ANSI-SQL** und dem Native-**SQL** der einzelnen Softwarehäuser. Viele große Firmen halten sich heute strikt an die ANSI-Vorgaben und verzichten auf die Erweiterungen um einen hohen Grad an Unabhängigkeit von einzelnen Herstellern und Portabilität der Anwendungen zu erreichen.

SQL, Structured Query Language, ist eine Sammlung natürlicher-sprachlicher Anweisungen, die dazu geeignet sind entsprechende Datenbanken zu bearbeiten. SQL gliedert sich in Gruppen, die die Definition, die Kontrolle und die Manipulation einer Datenbank ermöglichen und die Sicherheitsaspekte behandeln.

Zu den Anweisungen, die die Definition erledigen, gehören

- ▶ CREATE DATABASE
- ▶ CREATE TABLE
- ▶ CREATE INDEX

► CREATE SYNONYMS

Anweisungen aus dem Bereich Manipulation sind

- ALTER
- INSERT INTO
- DELETE FROM
- UPDATE und auch
- SELECT

Die Sicherheitsaspekte beinhalten Anweisungen wie

- GRANT
- REVOKE

mit denen Zugriffsberechtigungen gewährt und widerrufen werden.

In Erweiterung zu den "gewöhnlichen" SQL-Datenbanken verfügen einige über eine Einrichtung die die Datensicherheit behandelt. Diese Datenbanken nennen sich OLTP-Datenbanken. Das OLTP steht für OnLine Transaktion Processing, dessen Charakteristika auch definiert sind. Auch hier gilt wieder, daß jeder Hersteller seinem Produkt Eigenschaften verleiht, die es vom Wettbewerb abgrenzen sollen. Zu den Fähigkeiten einer OLTP-Datenbank gehört, daß sie in der Lage ist eine Datensicherung auf Band aufzuzeichnen, ohne daß das Tagesgeschäft unterbrochen wird. Sie soll also 7 Tage in der Woche jeweils 24 Stunden ohne Unterbrechung zu Verfügung stehen. Auch diese Datenbanken sind nicht gegen Fehler gefeit, die etwa durch Headcrashes oder andere mechanische Defekte an der Hardware entstehen. Deshalb verfügen die meisten über eine Einrichtung, die den Datenbestand zweifach, gewissermaßen gespiegelt vorhalten. Fällt durch irgendeinen Grund eine Festplatte aus, wird sofort die ständig aktuell gehaltene Kopie verwendet. Der Administrator der Datenbank erkennt den Ausfall an einer Meldung auf seiner Konsole und kann entsprechend reagieren.

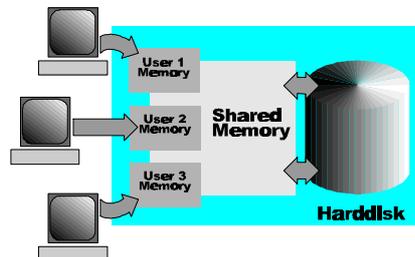
Und selbst wenn Fälle eintreten, gegen die die Pseudo-intelligenz aller Software machtlos ist, besitzen moderne OLTP-Datenbanken Mechanismen, die den Verlust auf ein Minimum begrenzen. Hier wird ständig eine Datei mitgeführt, die die Aktionen der Anwender aufzeichnet. Bei einem Absturz

des Systemes kann mit einem sogenannten "Rollback" an die zuletzt konsistente und widerspruchsfreie Stelle der Datenbank gesprungen werden und mit einem Recovery anhand der LOG-Datei der Schaden auf ein Minimum beschränkt werden. Oft ist hierzu noch nicht einmal ein Administrator notwendig. Vielmehr erkennt das System selbsttätig, ob ein Schaden vorliegt und bemüht sich die notwendigen Schritte einzuleiten. Erst wenn drastische Maßnahmen, wie beispielsweise das Einlegen der letzten Bandsicherung notwendig sind, muß ein Fachmann zugegen sein.

Das Wort Transaction bezieht sich auf eine Besonderheit komplexem Datenmanagements. Eine transaktionsorientierte Datenbank ist darauf abgerichtet, logisch zusammengehörende Anweisungen entweder in ihrer Gesamtheit oder im Zweifel lieber ganz und gar nicht durchzuführen. Das ist in etwa vergleichbar mit den Buchungen in einer Buchhaltung. Das Abbuchen eines Betrages auf einem Konto hat immer auch das Buchen auf das Gegenkonto zur Folge. Eine Buchung auf nur eines dieser Konten würde die Buchhaltung durcheinander bringen. Die transaktionsorientierte Datenbank wird es auch bei einem eventuellen Absturz des Systems zwischen zwei Buchungsvorgängen nicht zulassen, daß eine Buchung ohne die entsprechende Gegenbuchung erfolgt. Deshalb "sammelt" sie zunächst alle zusammengehörende Buchungen innerhalb einer Transaktion, simuliert gewissermaßen den ganzen Vorgang und beschließt ihn wenn alle benötigten Schritte unternommen wurden mit einem Commit. Diese Vorgänge spielen sich allerdings im Sekunden- oder gar Millisekundenbereich ab. Stürzt der Rechner an irgendeiner Stelle der Transaktion ab, dann wird diese nicht durchgeführt und der Datenbestand bleibt konsistent. Fein, nicht wahr? Mit einem weiteren Feature bemühen sich die Hersteller besonders attraktiv zu wirken. Mit dem Optimizer. Ein Optimizer ist ein Programmteil, der dazu dient die Anfragen an die Datenbank so zu optimieren, daß möglichst schnelle Antwortzeiten entstehen. Er merkt sich zum Beispiel häufig verwendete Anfragen oder oft verwendete Tabellen oder Indizes und hält sie wenn möglich im Arbeitsspeicher der Maschine vor. Jedem leuchtet ein, daß der RAM einer Maschine schneller arbeitet als die mechanische Hardware. Auch faßt eine Optimizer ähnliche Anfragen an eine Tabelle, die etwa zur gleichen Zeit eintreffen zusammen und erledigt

sie in einem Rutsch. Optimizer sind eine Wissenschaft für sich und gleichzeitig auch oft Gedeih und Verderb von Datenbankfirmen, weil natürlich die Antwortzeit als Auswahlkriterium bei einem Kauf einer Datenbank ein ganz wichtiges Kriterium ist.

Wo wir doch bereits von RAM gesprochen haben, wenden wir uns nun noch einmal ein wenig ab von den PCs mit den merkwürdigerweise erfolgreichen Betriebssystemvarianten eines bedeutenden Softwarehauses. Sehr viele der Datenbank-Engines, also die dem Anwender abgewandte Seite, laufen auf Multiuser und Multitasking Betriebssystemen. Hierzu gehört beispielsweise UNIX als vielleicht das berühmteste. Sehen wir uns eine Situation an, in der einige Anwender zur gleichen Zeit auf dem System und mit einer bestimmten Datenbank arbeiten. Normalerweise wird ja für jeden Anwender der sich in dem System anmeldet ein bestimmter Bereich des RAM reserviert mit dem er und seine Programme arbeiten können. Die Portionen, die dadurch entstehen, kennen sich gewissermaßen nicht und laufen getrennt voneinander. Lediglich der Speicherplatz auf der Festplatte wird von mehreren Anwendern und Programmen gemeinsam verwendet. Was aber nun, wenn der Optimizer erfahren muß was die einzelnen Anwender als Anfrage auf die Datenbank vorbereiten? Und wie soll er denn zum Zwecke der Beschleunigung Daten im RAM vorhalten, wenn doch jeder seine eigene Portion RAM reserviert? Hierzu wird ein Bereich im RAM des Servers als Shared Memory deklariert. Dieser Bereich steht allen Anwendern der Datenbank gemeinsam zur Verfügung.



Bleiben wir noch einen Augenblick bei UNIX für eine weitere allgemeine Information. Bestimmte Betriebssysteme sind für spezielle Aufgaben optimiert. UNIX allerdings ist ein "allgemeines" Betriebssystem ohne besonderen Fokus. UNIX soll eine Plattform für einen großen Bereich an Software sein, von

denen die Entwickler des Betriebes zum Zeitpunkt seiner Programmierung eventuell noch nicht einmal etwas ahnten. Und tatsächlich wird UNIX für CAD/CAM für Text -verarbeitung, Fakturierung, Prozesssteuerung und Datenbank -anwendungen und vielem mehr verwendet. Die Prozeduren, die UNIX zur Verwaltung von Dateien verwendet, werden deshalb auch für einen Text und für Daten gleichermaßen eingesetzt. Tatsächlich erfüllen diese Prozeduren jedoch im Hinblick auf Tempo nicht unbedingt die Anforderungen einer modernen Datenbank. Deshalb sind einige Datenbankhersteller dazu übergegangen, den Bereich in dem die Daten liegen mit schnelleren Prozeduren zu verwalten. Dieser Bereich steht dann UNIX nicht mehr zur Verfügung sondern wird durch das Datenbankprogramm gesteuert. Hier werden nun Daten nicht mehr in langen sequentiellen Ketten aufbewahrt, sondern in blockähnlichen Strukturen, die ein erhebliches Wachstum an Geschwindigkeit bedeuten können. Der Bereich der Festplatte, den UNIX verwaltet, wird Cooked Device genannt. Der Bereich den die Datenbank für sich in Anspruch nimmt, heißt deshalb Raw Device.

Das war jetzt nur ein kleiner Ausflug in diese edle Welt der Datenbanken. Ordinäre Programme verfügen nicht zwangsweise über diese Fähigkeiten. Und es steht zur Frage ob man bei kleineren Anwendungen tatsächlich mit Kanonen auf Spatzen schießen soll.

Sehen wir uns endlich SQL etwas genauer an.

SQL

Wie bereits erwähnt, besteht diese Sprache aus einzelnen Anweisungen, die in einzelnen Zeilen oder als Zusammenstellung von mehreren Anweisungen an das Datenbankprogramm gesendet werden. Um dem Anwender das Eingeben einzelner Anweisungen zu erleichtern, werden meist hierzu geeignete Programme verwendet. Die Datenbankhersteller bieten in aller Regel solche Programme an, MINDMAP ist auch für diese Zwecke geeignet. Damit besser verstanden wird, was wirklich im Hintergrund passiert, sehen wir uns einige typische Anweisungen einmal genauer an.

Bevor mit den Daten umgegangen wird, muß eine Datenbank erstellt werden. Dies erfordert zunächst ein simples Statement, da eigentlich nur ein Platz für die Aufnahme von Tabellen und deren Daten deklariert und dieser Platz mit einem Namen versehen wird.

In den gezeigten Statements wurde nach folgendem Schema vorgegangen: Wörter in Großbuchstaben stellen die unumstößliche Syntax der Anweisung dar. Spitze Klammern stehen für die Eingabe eines benutzerdefinierten Textes, meist Namen. Eckige Klammern können zur genaueren Spezifizierung der Anweisung eingegeben werden, können aber auch entfallen.

```
CREATE DATABASE <databasename> [Options]
```

ist die Anweisung, um eine Datenbank zu erstellen. Ausgeschrieben könnte die Anweisung folgendermaßen aussehen:

```
CREATE DATABASE agency
```

Dies erzeugt eine Datenbank mit dem Namen *agency*. Der Owner der Datenbank ist der Ersteller.

```
CREATE DATABASE agency WITH LOG MODE ANSI
```

erzeugt eine Datenbank mit dem Namen *agency* und verwendet den ANSI-Standard bei der Verwendung einer LOG-Datei (wenn das Programm eine Wahl zuläßt).

Die Datenbank benötigt nun noch Tabellen für die organisierte Aufnahme von Daten. Nehmen wir einmal an, in der Tabelle sollen die Stammdaten von Kunden gespeichert werden und die Tabelle trägt den Namen *artists*.

```
CREATE TABLE <tablename> (columnname1 <column  
definition1>, <columnname2> ..... )
```

oder besser

```
CREATE TABLE artists (artnumber INTEGER NOT NULL,  
artname CHAR(30), fname CHAR(20), lname CHAR(20) )
```

Diese Anweisung erstellt eine Tabelle mit dem Namen *artists* in der Datenbank *agency*. Die Tabelle enthält vier Felder, nämlich *artnumber*, *artname*, *fname*, *lname*. Das Feld *artnumber* nimmt nur ganzzahlige Werte entgegen (INTEGER) Außerdem akzeptiert es keine Nichteingabe. Der Anwender soll also gezwungen werden, eine Nummer zu vergeben (NOT NULL). Die übrigen drei Felder sind solche, die Buchstaben entgegennehmen (CHAR). Die maximale Länge der Eingaben kann den Wert in den Klammern nicht überschreiten.

CHAR(20) bedeutet also, in dieses Textfeld kann man maximal 20 Buchstaben eingeben. Es bedeutet auch, für dieses Feld wird der Raum von 20 Buchstaben in der Tabelle reserviert. Das heißt, daß bei ständigen Eingaben von nur drei Buchstaben dennoch der Platz von 20 verbraucht wird.

Es soll nun verhindert werden, daß die *artnumber* in der Tabelle zweimal vergeben wird. Hierzu erstellen wir einen Index der dies verhindert.

```
CREATE [option] INDEX <indexname> ON <tablename>
      (columnname)
```

oder

```
CREATE UNIQUE INDEX artistnumber ON artists (artnumber)
```

Dieser Index stellt eine gesonderte Datei dar. Er stellt sicher, daß die *artnumber* nur einmal vergeben wird.

Indizes sind auch für andere Funktionen verwendbar. Ein Index kann für eine schnelle, sortierte Anzeige von Daten sorgen. Außerdem können Indizes für eine effiziente Verknüpfung mehrerer Tabellen herangezogen werden.

Unsere Tabelle soll nun mit Daten versorgt werden. In den meisten Fällen wird ein kompletter Datensatz angelegt. Ist dies nicht der Fall, oder unterscheidet sich die Reihenfolge der Daten von denen der Zielspalten, müssen die auszufüllenden Spaltennamen explizit genannt werden. Der häufigere und einfachere Weg lautet

```
INSERT INTO <tablename> VALUES (<value1>, <value2>, ...)
```

die andere Methode wäre

```
INSERT INTO <tablename> (<columnname1>, <columnname2>,
<...>) VALUES (<value1>, <value2>, <...>)
```

In unserem Falle sieht das Statement etwa so aus

```
INSERT INTO artists VALUES (1000, "Zorro", "Bill", "Wates")
```

Nehmen wir an unser Künstler hat keinen Künstlernamen.
Dann sähe das Statement so aus

```
INSERT INTO artists VALUES (1010, "", "Richard", "Fear")
```

Einfach weglassen gilt also nicht. Übrigens muß leider gesagt werden, daß unsere Datenbanken meist keinen Spaß ver- stehen, wenn Leerschritte, Blanks mit im Spiel sind. "Wates" ist nicht dasselbe wie " Wates"!

Eine Recherche nach Künstlern, deren Nachname mit dem Buchstaben "W" beginnt -einem der häufigsten Wünsche von Auftraggebern- würde also bei der Verwendung von führenden Blanks fruchtlos bleiben.

Die Anweisung

```
INSERT INTO artists VALUES (1020, "Velvet", "Nina",
"Symphone", 1030, "Satin", "Barry", "Black")
```

würde zwei Datensätze, bzw. Zeilen in die Tabelle einfügen.

Es gibt weitere Methoden um Datensätze in die Tabelle einzugeben, die Erklärung von dynamischen Statements und Platzhaltern würde jedoch für eine kurze Übersicht zu weit führen.

Damit unsere Theorie etwas umfassender wird, erstellen wir in kurzen Schritten eine weitere Tabelle:

```
CREATE TABLE events (artnumber INTEGER NOT NULL, what
CHAR(30), when DATE)
```

und füllen diese mit Daten:

```
INSERT INTO artists VALUES (1020, "Festival Montreux",
"11.10.94", 1020, "Dahome Jazz", "08.06.95", 1020, "New
Orleans Hard Rock", "12.12.95", 1030, "Orlando Music
Fair", "02.06.92")
```

Wir halten fest: der Künstler mit der Nummer 1020 war in Europa, Afrika und USA unterwegs, der Künstler mit der Nummer 1030 hat in Florida Kultur verbreitet.

Jetzt begeben wir uns an die erste und zunächst einmal simple Abfrage.

```
SELECT <columnname1>, <columnname2>, <...> FROM
<tablename>
```

In unserem Falle könnte das so aussehen:

```
SELECT lname, fname FROM artists
```

Ist eine bestimmte Reihenfolge erwünscht, dann bedienen wir uns des Schlüsselwortes *ORDER BY*. Eine Sortierung muß das Feld, nach dem sortiert wird, immer beinhalten.

```
SELECT arname, fname, lname FROM artists ORDER BY
arname
```

Da keinerlei konkretisierendes Kriterium verwendet wurde, wird die Anzahl der Datensätze auch nicht eingeschränkt. So werden die Felder *fname* und *lname* aller Datensätze angezeigt. Eine Qualifizierung der Ausgabe würde lauten:

```
SELECT lname FROM artists WHERE arnumber = 1020
```

Hier wird nun der *lname* des Künstlers angezeigt, dessen *arnumber* den Wert 1020 hat.

Statt dem Gleichzeichen (=) können auch andere Vergleichsoperatoren angewandt werden. Diese wären "<" und ">", wie auch "<>", ">=", "<=", und "NOT", was in diesem Falle dem "<>" entspricht.

Wenn die Anzeige aller Felder gewünscht wird, kann auch mit sogenannten Wildcards gearbeitet werden. Die Eingabe eines Sternes (*) anstatt der Spaltennamen wird alle Spalten anzeigen.

```
SELECT * FROM <tablename> [WHERE...]
```

Wenn die gesuchten Datensätze mehr als ein Kriterium erfüllen sollen, dann kann man mit den Schlüsselwörtern *AND* und *OR* arbeiten. Hierbei ist die Bedeutung der beiden

Worte sehr wichtig. *AND* bedeutet eine stärkere Einschränkung und *OR* eher das Gegenteil.

```
SELECT * FROM <tablename> WHERE <columnname1>
      <operator> <value> AND <columnname2> <operator>
      <value>
```

oder

```
SELECT * FROM artists WHERE artnumber < 1050 AND fname
      = "Bob"
```

Diese Abfrage wird alle Zeilen anzeigen, bei denen die *artnumber* kleiner als 1050 ist UND (*AND*) der Vorname "Bob" ist. Beide Kriterien müssen erfüllt werden.

```
SELECT * FROM artists WHERE artnumber < 1050 OR fname =
      "Bob"
```

Diese Abfrage wird alle Datensätze zeigen deren *artnumber* kleiner 1050 ist. Außerdem werden dessen ungeachtet alle Sätze angezeigt, bei denen der Vorname "Bob" ist. Hierbei können also auch Datensätze auftauchen, deren *artnumber* kleiner als 1050 ist, da das zweite Kriterium, nämlich *fname* gleich "Bob", erfüllt wird. Das Problem beim Verständnis dieses Vorganges ist unsere Sprechweise. Wir neigen dazu zu sagen, daß wir alle Datensätze kleiner 1050 haben möchten UND alle, die "Bob" enthalten. Und nicht ODER, oder?

Das Geheimnis ist, wir beziehen in der oben genannten Sprechweise das *AND* auf die zu erwartende Datenmenge und nicht auf die Kriterien.

Wenn wir nun alle Zeilen haben möchten, deren *lname* mit "W" anfängt, dann helfen uns die mathematischen Vergleichsoperatoren herzlich wenig. Hierbei verwenden wir das Schlüsselwort *LIKE* und die Wildcard %.

```
SELECT * FROM artists WHERE lname LIKE "W%"
```

Noch interessanter wird es, wenn wir eine Ergebnismenge (Result Set) haben wollen, in der Inhalte aus der Tabelle *artists* und der Tabelle *events* enthalten sind. Hier begeben wir uns ja in die Welt der Joins, die ansatzweise bereits erklärt wurde.

```
SELECT * FROM artists, events WHERE artists.artname LIKE
"%zz%" AND artists.artnumber = events.artnumber
```

Diese Anweisung selektiert alle Datensätze aus den Tabellen *artists* und *events*, bei denen im Künstlernamen der Spalte *artname* der Tabelle *artists* die Buchstabenkombination "zz" vorkommt und die *artnumber* der Tabelle *artists* mit der *artnumber* der Tabelle *events* übereinstimmt.

Hübsch, nicht?

Wie am Anfang erklärt, existiert der Link zwischen den beiden Tabellen alleine in der Sekunde der Abfrage und kein bißchen länger. Auch bei der Definition der Tabellen haben wir keinen Link erklärt. Wir haben lediglich dafür gesorgt, daß es zwei Spalten mit dem potentiell gleichen Inhalt gibt und haben diese Spalten der Verständlichkeit halber auch gleich benannt. In dem Statement fällt auf, das wir den Spaltennamen den Tabellennamen vorangestellt haben. So kommt nichts durcheinander.

Wenn nun ein Eintrag innerhalb einer Zeile geändert werden soll, dann ist *UPDATE* der richtige Weg.

```
UPDATE <tablename> SET <columnname> = <value> WHERE
<columnname> = <othervalue>
```

Die *WHERE*-Klausel muß sich an dieser Stelle nicht auf die zu verändernde Spalte beziehen. Sie wird dies aber oft tun. Sie spezifiziert den zu verändernden Datensatz. Aber Achtung! Eine *WHERE*-Klausel kann mehrere Zeilen zum Ergebnis haben.

```
UPDATE artists SET artname = "Django" WHERE artname =
"Zorro"
```

würde dasselbe verursachen wie

```
UPDATE artists SET artname = "Django" WHERE artnumber =
1000
```

Leider hat "Django" nun den Vertrag mit unserer Agentur gekündigt. Ehrlich gesagt wahr er aber auch schwer vermittelbar. Wir möchten ihn also deshalb aus unserer Datenbank entfernen.

```
DELETE FROM artists
```

Halt, Stop!! Diese verheerende Zeile würde kritiklos alle Datensätze der Tabelle *artists* löschen! Bitte spezifizieren Sie die zu löschenden Datensätze genau.

```
DELETE FROM artists WHERE arname = "Django"
```

Das war aber knapp. Sie sollten verhindern, daß irgendjemand in Ihrer Datenbank Unfug treibt. Anschauen, so nehmen wir einmal an, darf man jeden Datensatz, wenn man schon einmal einen Netzwerk-Login besitzt; Eingeben möglichst auch. Aber die Tabelle sollte von keinem anderen als von Frau Schulze modifiziert werden, und kein anderer als Herr Mayer soll Datensätze löschen dürfen.

```
GRANT <privileg> TO <grantee>
```

Hier wird davon ausgegangen, daß die gegenwärtig geöffnete Datenbank bearbeitet wird. Bei *GRANT* auf eine andere als die gegenwärtig Datenbank muß diese erst geöffnet werden.

Zudem handelt es sich bei dieser Anweisung um ein Database-Level Privileg. Ein anderes Level betrifft die Tabellen.

```
GRANT connect TO public
```

ermöglicht schon einmal allen Mitarbeitern eine ganze Menge. Sie können nun bei allen Tabellen in der Datenbank *SELECT*, *INSERT*, *UPDATE* und *DELETE* anwenden, vorausgesetzt dies ist auf dem Tabellenlevel nicht untersagt. Außerdem darf nun auch jeder Anwender "seine" Tabellen modifizieren und löschen und seinerseits *GRANTS* auf Tabellenlevel vergeben und entziehen.

Herr Mayer soll zu etwas mehr in der Lage sein. Er bekommt ein

```
GRANT resource TO "mayer"
```

Nun darf er alles, was ein ordentliches "Connect" ausmacht, und außerdem auch neue Objekte erstellen. Herr Mayer steht in diesem Statement in Anführungsstrichen, weil ANSI-Datenbanken den Text sonst in Großbuchstaben abbildet.

Frau Schulze darf ja mehr. Sie bekommt einen größeren *GRANT*, einen mit mehr Wirkung.

```
GRANT dba TO schulze
```

DbA steht an dieser Stelle für Database Administrator. Es ist der Generalschlüssel zur Datenbank.

Jetzt entziehen wir aber allen Usern wieder ein paar Rechte auf Tabellenebene.

```
REVOKE update, delete, alter ON artists, events FROM public
```

Fein, die Allgemeinheit kann nun nichts mehr wesentliches kaputtmachen.

```
GRANT all ON artists, events TO "schulze"
```

```
GRANT update, delete ON artists, events FOR "mayer"
```

Eine Vertretung von Herrn Mayer soll nun in dessen Abwesenheit auch wenigstens *artname* ändern dürfen.

```
GRANT update ON artists (artname) TO "Knickerbocker"
```

Hier wird in der Klammer die Spalte spezifiziert, deren Inhalt geändert werden darf.

So, das dürfte für die ersten Schritte genug sein. Wie Sie sich vorstellen können, kann man mit Büchern über SQL und Datenbanken ganze Bibliotheken füllen. So weit wollten wir hier aber nicht gehen. Schließlich darf der zaghafte Anfänger auch nicht erschreckt werden.

Wer sich tiefer in die Materie einarbeiten will, dem sei geraten, sich mit entsprechender Literatur zu versorgen, Kurse zu besuchen, aber vor allem viel Praxiserfahrung zu sammeln.

Viel Spaß mit SQL!

Anhang: Beispielanwendungen

Die Anwendungen befinden sich in Ihrem MINDMAP-Verzeichnis unter **Tutorial** bzw. **Samples\Seminar**.

Bitte vergessen Sie nicht, für die Semiarverwaltung, welche im Kapitel *Bau einer erweiterten Datenbankanwendung* beschrieben ist, die ODBC Datenquelle mit Namen **Seminar** zu installieren.

Kapitel	Anwendung	Zubehör
Einleitung	keine	keines
Erste Schritte	TUT_01.MM TUT_02.MM	keines
Nachrichten und Links	TUT_03.MM TUT_04.MM TUT_05.MM TUT_06.MM TUT_07.MM	keines
Ein Picture-Viewer	TUT_08.MM TUT_09.MM TUT_10.MM TUT_11.MM TUT_12.MM	keines
Ein "Taschenrechner"	Taschenrechner.MM	keines
Ein Video-Recorder	Vidplay.MM	keines
Datenbank	keine	keines
Bau einer erweiterten Datenbankanwendung	Seminar.MM	semibook.dbf semicust.dbf semiroom.dbf seminar.ini

Kleines Datenbankbrevier	keine	keines
-----------------------------	-------	--------