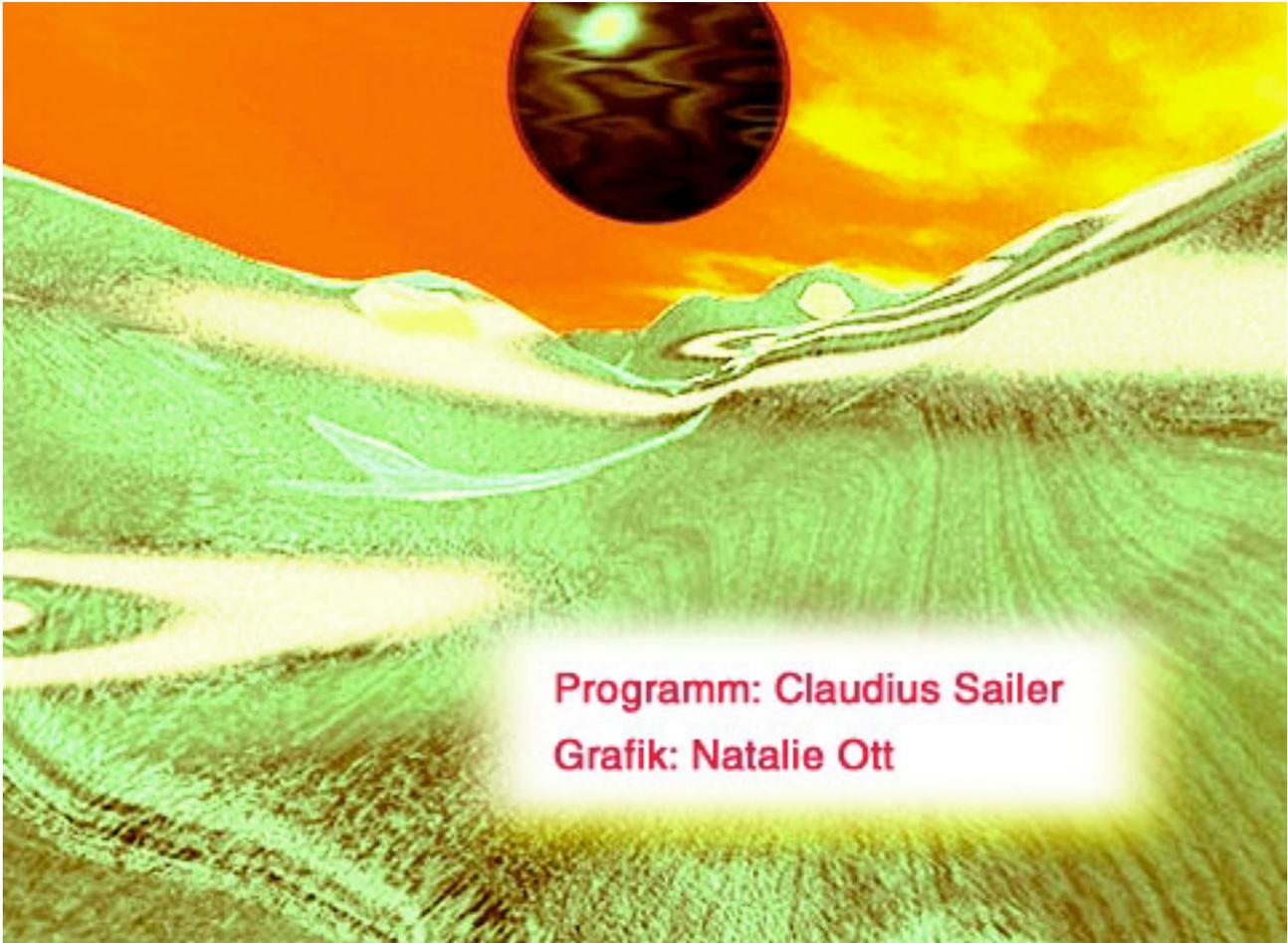


# Application Manager



Programm: Claudius Sailer  
Grafik: Natalie Ott

## Vorwort

Nachdem ich mit der Arbeit an „Application Manager“ angefangen habe und sich die ersten Interessenten meldeten um sich registrieren zu lassen, kam ich immer wieder in die Situation, dass ich die User, ihre Registrierungs codes und deren Informationen, Bugs und Requests irgendwie verwalten musste. Nach anfänglichen Versuchen dies in Textverwaltungen oder Tabellenkalkulationen zu machen, habe ich mich entschieden ein Programm hierfür zu schreiben. Das Ergebnis ist „Application Manager“.

## **Schnellstart**

Wenn Sie das Programm auf einer Festplatte mit HFS+ installiert haben, indem sie einfach den Ordner „Application Manager“ an die gewünschte Stelle kopieren, starten Sie einfach das Programm mit einem Doppelklick auf das Programmicon. Danach brauchen Sie nur über das Menü „Neu“, am Besten im Ordner „Daten“, eine neue Datenbank anlegen. Gehen darauf gleich auf den Menüpunkt Importieren und wählen im Ordner Daten die Datei „Beispieldatei“ und füllen damit die Datenbank. Danach können Sie die von Ihnen gewünschten Einstellungen vornehmen und mit der Arbeit/Testen beginnen.

## **Programmbeschreibung**

„Application Manager“ besteht aus folgenden Bestandteilen:

### **Voreinstellungen**

Diese sollen Ihnen helfen das Programm nach Ihrem Geschmack einzurichten.

### **Arbeitsdatenbanken**

Hier können Sie Ihre Daten bzw. die weiterer Personen ablegen. Die Arbeitsdateien sollten sich immer im Verzeichnis „Daten“ in dem Ordner befinden in dem sich die Programmdatei befindet.

### **BackUp**

Im Ordner Backup im Programmordner legt „Application Manager“ seine automatisch erstellten BackUp-Dateien an.

### **Das Programm**

Das Programm „Application Manager“ ist natürlich der zentrale Punkt von dem aus alle anderen Aufgaben erledigt werden.

## Voreinstellungen

Unter den Voreinstellungen kann jeder User „Application Manager“ nach seinen eigenen Bedürfnissen anpassen. Sämtliche hier vorgenommenen Einstellungen werden in der Datei „Voreinstellungen“ im Programmordner abgelegt. In dieser Datei werden auch der RegistrierungsCode und der Registrierungsname für „Application Manager“ in der Resource Fork abgelegt.



### Letzte geöffnete Datei beim nächsten Programmstart automatisch öffnen

„Application Manager“ sucht beim Programmstart die Datenbank mit der zuletzt gearbeitet wurde und versucht diese wieder zu öffnen.

### Position und Grösse des Fensters ‚Application Manager‘ merken

„Application Manager“ merkt sich die Grösse und Position der Hauptfensters von „Application Manager“ und öffnet das Fenster genau mit diesen gespeicherten Daten.

### Dateiname vor die Fensternamen setzen

Der Name der gerade aktiven Datenbank wird beim Hauptfenster vor den Fensternamen gestellt.

### Schrift in ListBoxes

Die Schrift sowie die Schriftgrösse für Listboxen kann hier festgelegt werden

## **eMail-Programm unterstützt html RETURN-Code**

Aus „Application Manager“ kann man auch eMails über das voreingestellte eMail-Programm verschicken. Hier kann es vorkommen, dass das eMail-Programm Probleme mit dem Code für RETURN hat. Ist diese Voreinstellung nicht aktiviert, dann ersetzt „Application Manager“ alle RETURNS durch Leerzeichen (Hier muss man einfach testen, welche Einstellung für einen die beste Wahl ist).

## **Backupdatei der Arbeitsdatei erstellen**

Hier kann man festlegen ob „Application Manager“ von der Datenbank Backups erstellen soll. Vorgeben muss man nur noch in welchem Zeitintervall solch ein BackUp erfolgen soll. Ich selber habe hier 1 Tag eingestellt.

## **Backup als Textdatei erstellen**

Neben einer reinen Kopie der Datenbank ist auch das BackUp als Textdatei möglich. Die Textdatei ist kleiner als die Datenbankkopie, benötigt aber zur Erstellung etwas mehr Zeit. Als Empfehlung würde ich hier trotzdem ein Backup als Textdatei vorziehen.

## **Zur RegistrierungsCode-Generierung ein RBScript verwenden**

Um RegistrierungsCodes generieren zu können, wurde in „Application Manager“ eine RegCode-Generator integriert. Da hierfür der SourceCode nicht frei zugänglich ist, macht es wenig Sinn diesen zu verwenden. Aus diesem Grund verfügt „Application Manager“ über eine Schnittstelle zu einem RBScript (RealBasic-Script) Sobald „Application Manager“ im Programmordner eine Datei „ApMaRBScript“ findet, wird der darin befindliche Code zur RegistrierungsCode-Generierung herangezogen.

Näher darauf eingegangen wird zu einem späteren Zeitpunkt in diesem Dokument.

## **Mehrspaltige Tabellen/Listen abwechselnd farblich hinterlegen**

Hier kann man definieren, dass in sämtlichen mehrspaltigen Tabellen/Listen jede 2. Zeile andersfarbig hinterlegt wird. Dadurch wird die Lesbarkeit erhöht.

# User

Name	Vorname	Reg	Bug/Req
Bebel	Marc	0	4
Flamm	Franco	1	0
Fritz	Reinhard	1	2
Gestern	Olaf	1	0
Hauser	Klausl	1	2
Kazatakis	Dionysos	2	0
Keusgen	Ralf	0	3
Konradi	Konrad	1	0
Kreuzer	Roul	1	0
Mansfeld	Sibylle	1	0
Quasina	André	1	0
Rader	Steve	1	2
Reusch	Peter W.	1	0
Sailer	Claudius	4	3
Schmidt	Christiane	1	0
Siebel	Wolf	1	0
Sommerfeld	Bernd	1	0
Testuser		1	0
Toom	Rainer	1	0
von Gestern	Markus	1	1
Wasti	Raffael	1	0
Wendler	Thomas	2	0
Westermann	Klaus-Dieter	1	0
Wini	Stefan	0	0
Wühler	Alfred	1	2
Zyska	Tim	1	0

Vorname:

Name:

Anrede:

eMail-Adresse:

Strasse:

Hausnummer:

PLZ:

Ort:

Telefonnummer Eins:

Telefonnummer Zwei:

Geburtsdag:

Ersterfassung:

Letzte Änderung:

Kommentar:

User | Programme | Release | Registrierungen | Bugs/Requests

Hier kann man alle „Kunden“ anlegen die irgendein Programm registriert haben, registrieren möchten oder einfach nur Bugs und Requests gemeldet haben. In der ListBox kann man dann sehen, dass z.B. Bebel Marc bisher 4 Bug/Requests gestellt hat, aber noch kein Programm registriert hat.

# Programme

Programmname	Reg	Upd
Application Manager	0	0
CodeSaver	3	0
JobTimeControl	0	0
MoneyControl	18	0

Programmname: Application Manager

Codifier Eins: 28yy=P§ERkvr3i42MDP\_

Codifier Zwei: (mX9SreV7s5fz4PrYBk=

Ersterfassung: 08.03.2002

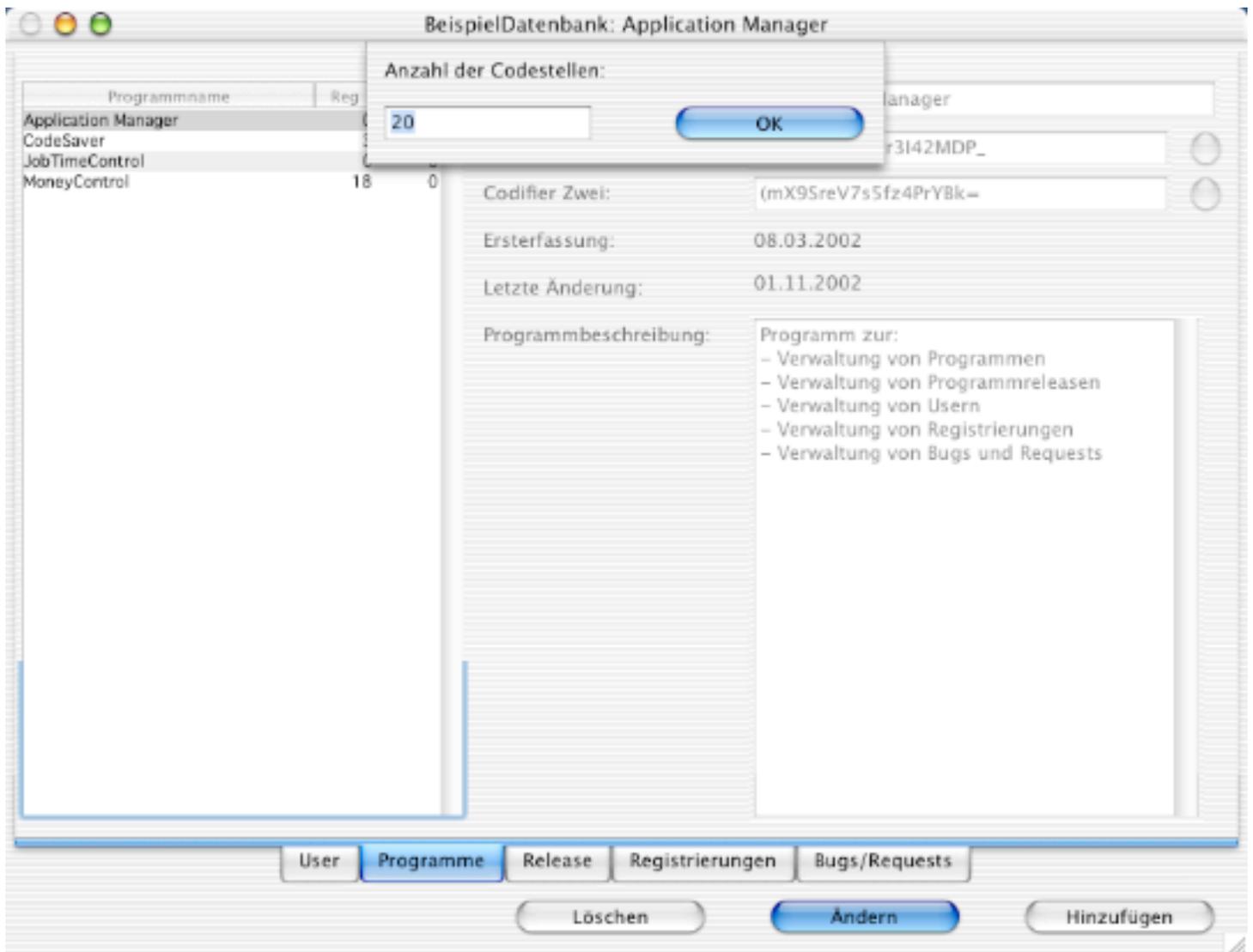
Letzte Änderung: 01.11.2002

Programmbeschreibung: Programm zur:  
- Verwaltung von Programmen  
- Verwaltung von Programmreleasen  
- Verwaltung von Usern  
- Verwaltung von Registrierungen  
- Verwaltung von Bugs und Requests

User Programme Release Registrierungen Bugs/Requests

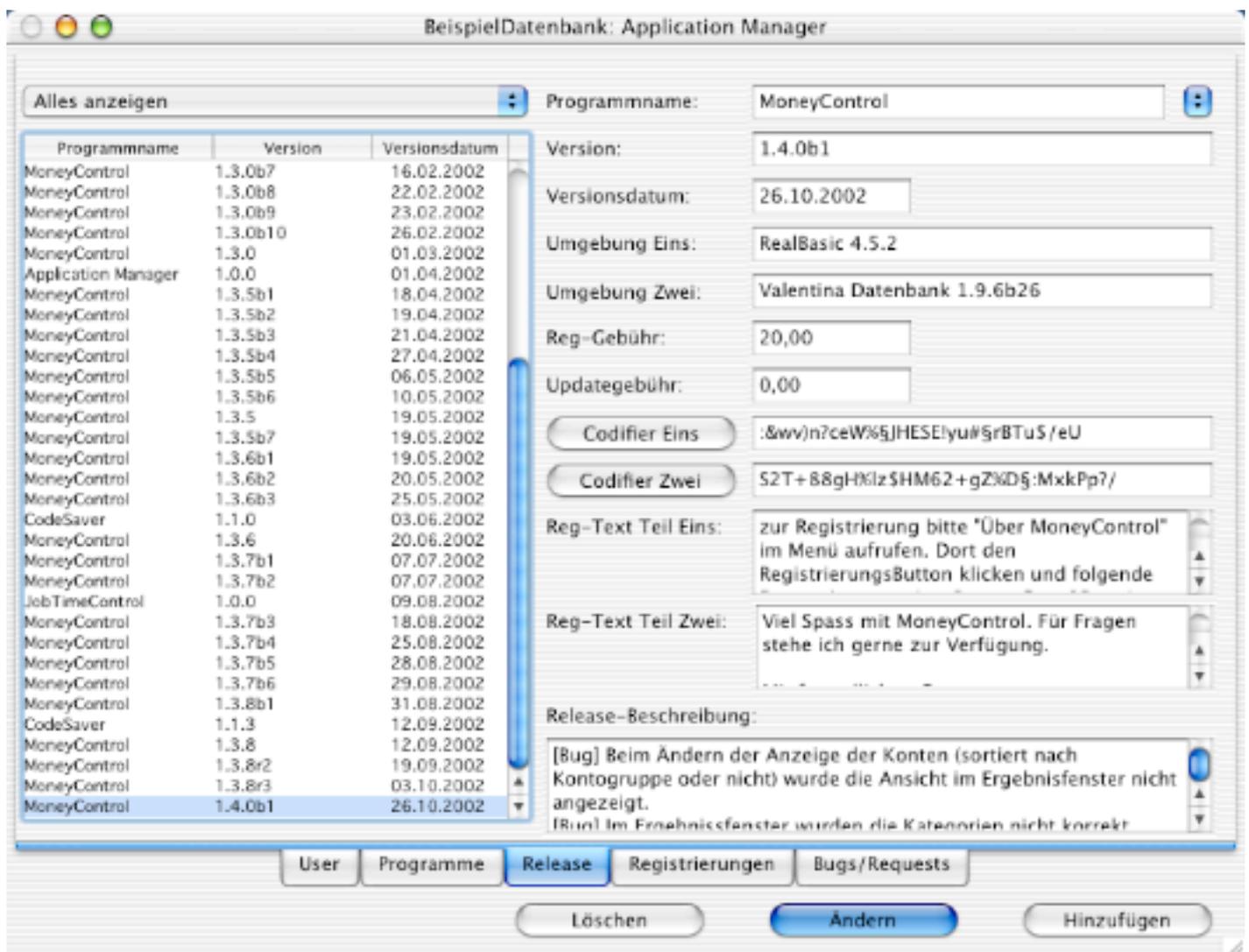
Löschen **Ändern** Hinzufügen

Hier werden die verschiedenen „Programme“ angelegt. Darüber hinaus werden hier auch die Codifier angelegt, die zur Generierung eines RegistrierungsCodes notwendig sind.



Drückt man auf einen der Buttons hinter den Codifiern, kann man sich auch die Codifier per Zufall generieren lassen. Dabei muss man nur die Anzahl an gewünschten Stellen angeben und den Rest erledigt „Application Manager“.

# Release



Hier werden die einzelnen Programmrelease verwaltet. Verschiedene Zusatzinformationen erleichtern die späteren Arbeiten. Die Codifier können hier nicht mehr eingegeben werden, aber per Klick auf den Button vom Fenster „Programme“ geladen werden. Damit kann man verschiedenen Versionen unterschiedliche RegistrierungsCodes mitgeben ohne alte RegistrierungsCodes nicht wieder reproduzieren zu können.

Um für ein neues Release also einen neuen RegistrierungsCode zu erzwingen, gibt man einfach unter „Programme“ neue Codifier an. Danach legt man ein neues Release an, lädt die Codifier durch drücken der Buttons und schon wird für jeden User, der einen neuen RegistrierungsCode wünscht für diese Version ein neuer RegistrierungsCode erzeugt.

Unter „Reg-Text“ kann man definieren, wie in einer eMail der Mailtext aufgebaut sein soll (siehe Registrierungen). Die Mail wird nachher so aussehen:

```
Betreff: RegistrierungsCode fuer ‚Programmname‘  
MailText: Sehr geehrter Vorname Nachname [RETURN][RETURN]  
MailText: RegText Teil Eins [RETURN][RETURN]  
MailText: RegName: RegistrierungsName [RETURN]  
MailText: RegCode: RegistrierungsCode [RETURN][RETURN]  
MailText: RegText Teil Zwei
```

# Registrierungen

Programmname	Name, Vorname
Application Manager	Sailer, Claudius
CodeSaver	Kazatakis, Dionysos
CodeSaver	Konradi, Konrad
CodeSaver	Sailer, Claudius
CodeSaver	Wendler, Thomas
JobTimeControl	Sailer, Claudius
MoneyControl	Flamm, Franco
MoneyControl	Fritz, Reinhard
MoneyControl	Gestern, Olaf
MoneyControl	Hauser, Klaus
MoneyControl	Kazatakis, Dionysos
MoneyControl	Kreuzer, Roul
MoneyControl	Mansfeld, Sibylle
MoneyControl	Quasina, André
MoneyControl	Rader, Steve
MoneyControl	Reusch, Peter W.
MoneyControl	Sailer, Claudius
MoneyControl	Schmidt, Christiane
MoneyControl	Siebel, Wolf
MoneyControl	Sommerfeld, Bernd
MoneyControl	Testuser,
MoneyControl	Toom, Rainer
MoneyControl	von Gestern, Markus
MoneyControl	Wasti, Raffael
MoneyControl	Wendler, Thomas
MoneyControl	Westermann, Klaus-Dieter
MoneyControl	Wühler, Alfred
MoneyControl	Zyska, Tim

BeispielDatenbank: Application Manager

Alles anzeigen

Programmname: MoneyControl

Releaseversion: 1.3.0

Update

Name, Vorname: Wasti, Raffael

Registrierungsname: Raffael Wasti

Registrierungscode: 5WZN-F90H-9CEH-6F5B

Registrierungsdatum: 11.02.2002

Reg-Gebühr: 15,00

Registrierung bezahlt

Bezahlt am: 14.02.2002

Reg-Code zugeschickt

Mail verschicken

Zugeschickt am: 12.02.2002

Kommentar:

User Programme Release **Registrierungen** Bugs/Requests

Löschen Ändern Hinzufügen

Hier kann man für jedes Programm und für jeden User speichern, wann er was registriert hat, welcher Registrierungsname und welcher Registrierungscode dabei verwendet wurde, und viele weitere kleine nützliche Informationen.

Mit dem Button hinter dem Eingabefeld „Registrierungsname“ wird aus dem „Name, Vorname“ automatisch ein „Registrierungsname“ erstellt. Diese Funktionalität muss jedoch nicht verwendet werden. Mit dem Button hinter dem Eingabefeld „Registrierungscode“ wird ein Registrierungscode aus den Informationen „Programmname“, „CodifierEins“ und „CodifierZwei“ der ReleaseVersion und dem „Registrierungsname“ erstellt. Auch hier kann man eine manuelle Eingabe vornehmen. Ansonsten wird hier das RBScript (siehe weiter unten) zur RegistrierungsCodeermittlung herangezogen. Mit dem Button hinter „Reg-Gebühr“ wird die Sharewaregebühr, die hinter der Releaseversion hinterlegt ist aus der Datenbank ausgelesen und angezeigt.

Mit dem Button „Mail verschicken“ wird eine Mail an den User geschickt mit den relevanten Registrierungsdaten und den Registrierungstexten, die man unter Release dem jeweiligen Release voreingestellt hat. Als Mail-Medium wird die voreingestellte Mail-Applikation in MacOS X verwendet.

## Bugs/Requests

BeispielDatenbank: Application Manager

Alles anzeigen

Nr.	Kurzbezeichnung	Programmname
2	Sortierung der Konten/Kategorien	MoneyControl
4	Export für Steuerfuchs	MoneyControl
5	Passwortabfrage für MoneyControl	MoneyControl
6	Zusätzliche graphische Auswertungen	MoneyControl
1	Gruppen von Konten	MoneyControl
3	Konten Schnellanlage	MoneyControl
7	Export Abbrechen	MoneyControl
8	OnlineAnbieter für Wertpapierkurse	MoneyControl
9	Fehlengabe Kommentare Buchungen	MoneyControl
10	IconLeiste für Konten/Depot	MoneyControl
11	SofortBuchungen mit Vorschau	MoneyControl
13	Anlegedatum zu Beginn markieren	MoneyControl
12	Masterkonto einrichten	MoneyControl
14	Kategoriegruppe im Buchungsfenster a...	MoneyControl
15	Summezeile in Analysen	MoneyControl
16	Druckfunktionalität	MoneyControl
17	Journalansicht Buchungen	MoneyControl
18	Journalansicht Kategorien	MoneyControl
19	Kontenjahresübersicht	MoneyControl

Bug-Nummer: 11

Kurzbezeichnung: SofortBuchungen mit Vorschau

Programmname: MoneyControl

Releaseversion: 1.3.0

Name, Vorname: Fritz, Reinhard

Art: Feature Request

Erfassungsdatum: 05.05.2002

Priorität: mittel

Aktueller Zustand: erledigt

Gefixed

Fixing Datum: 20.05.2002

Gefixed in Version: 1.3.6b1

Beschreibung: Wie kann ich das Programm dazu bringen, meine regelmäßigen Buchungen z.B. bis Ende des Jahres automatisch einzutragen?

Letzte Änderung: 22.05.2002

User Programme Release Registrierungen **Bugs/Requests**

Natürlich bleibt es nicht aus, dass Fehlermeldungen oder Wünsche von „Endanwendern“ gemeldet werden. Diese können dann hier verwaltet und abgearbeitet werden. Mit dem Button „Mail verschicken“ kann an den Bug/Request-Melder eine Mail verschickt werden.

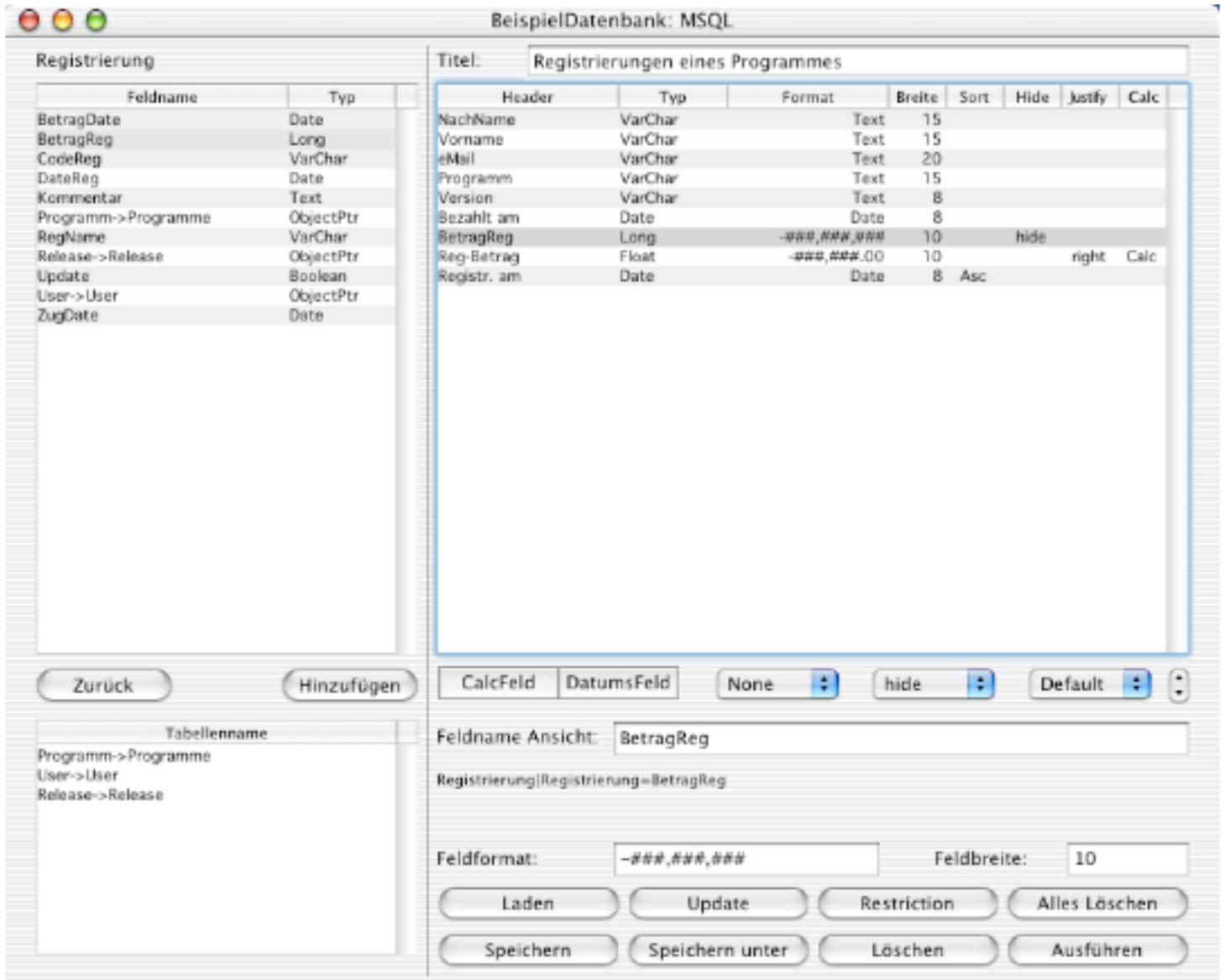
## Statistik

Statistik	
User:	26
Programme:	4
Release:	46
Registrierungen:	28
Bugs/Requests:	19
Version Realbasic:	4.5.2
Valentina Version:	1.9.6
Version Datenbank:	1

Da man hin und wieder auch wissen möchte was so alles an Daten sich in der Datenbank befinden, wird hierüber im Menüpunkt „Statistik“ ausführlich Auskunft gegeben.

# MSQL

Bei MSQL handelt es sich um ein Report-Generator-Tool, das helfen soll, ohne große SQL-Kenntnisse eigene Reports für „Application Manager“ zusammen zu stellen.



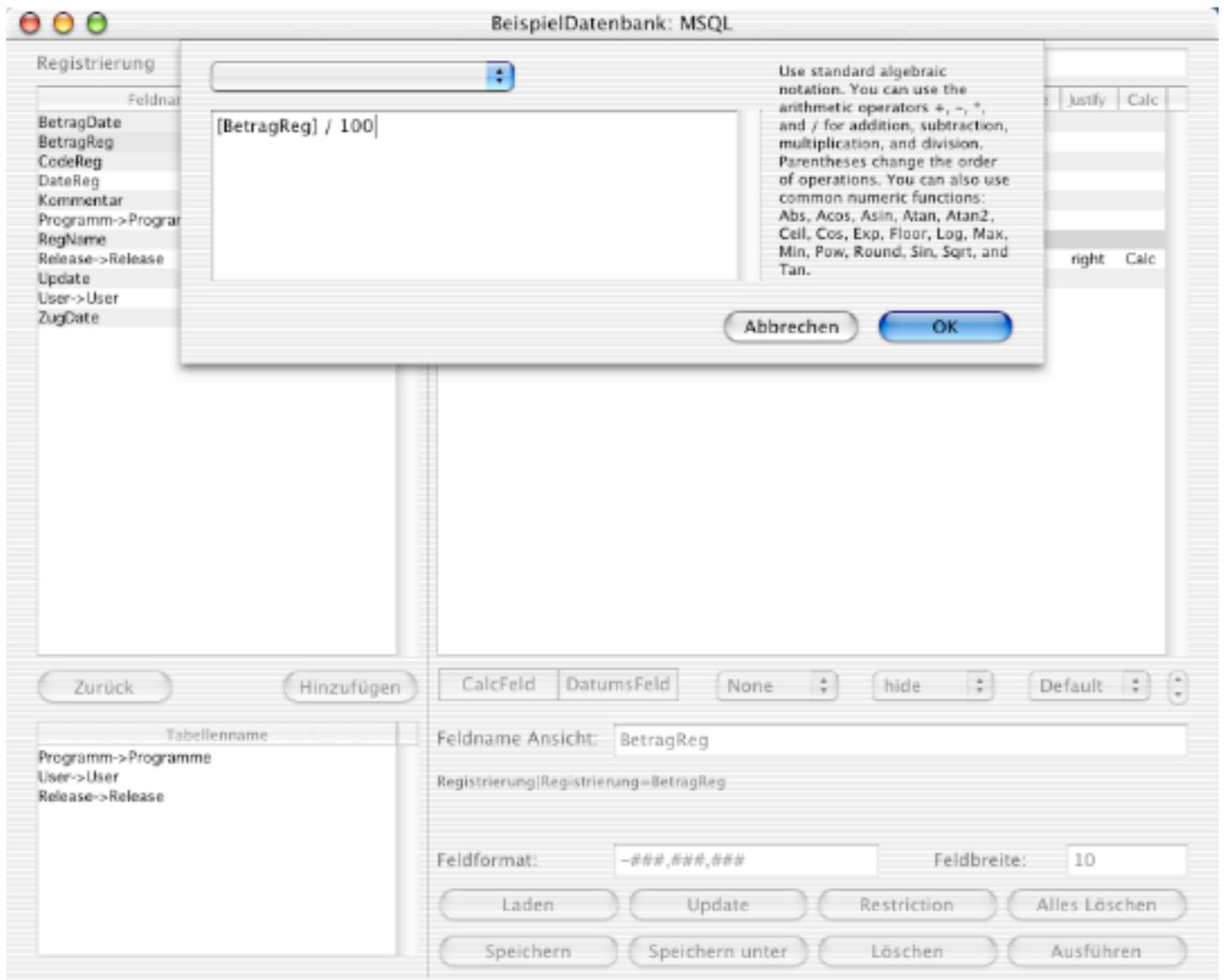
Im Basisfenster kann man sich seinen Report nach eigenen Vorstellungen zusammenstellen. Berechnungsfelder (CalcFeld) und/oder Datumsfelder (DatumsFeld) erweitern die Möglichkeiten und geben weitere Funktionalitäten. Darüber hinaus lassen sich mit Restrictions die Abfragen einschränken. Über den Button „Ausführen“ kann dann die Reporterstellung gestartet werden.

Das Fenster „MSQL“ wird geprägt durch 3 Listboxen. Ganz links unten befindet sich die ListBox für die zur Verfügung stehenden Tabellen von "Application Manager". Dabei werden entweder alle zur Verfügung stehenden Tabellen angezeigt (wenn man sich auf der untersten Tabellenhierarchie befindet) oder die Tabellen, die in der aktivierten Tabelle als Verlinkungen zur Verfügung stehen. Sobald man eine Tabelle auswählt, werden in der ListBox links oben die Felder der Tabelle angezeigt. Wenn ein Tabellenfeld eine Referenz zu einer anderen Tabelle beinhaltet, dann wird die referenzierte Tabelle in der ListBox für die Tabelle (links unten) angezeigt. Mit dem Button "Einfügen" kann man ausgewählte Tabellenfelder in den Report übernehmen, mit einem Klick auf eine Tabelle in der Listbox für Tabellen kann man sich eine Hierarchieebene tiefer in die

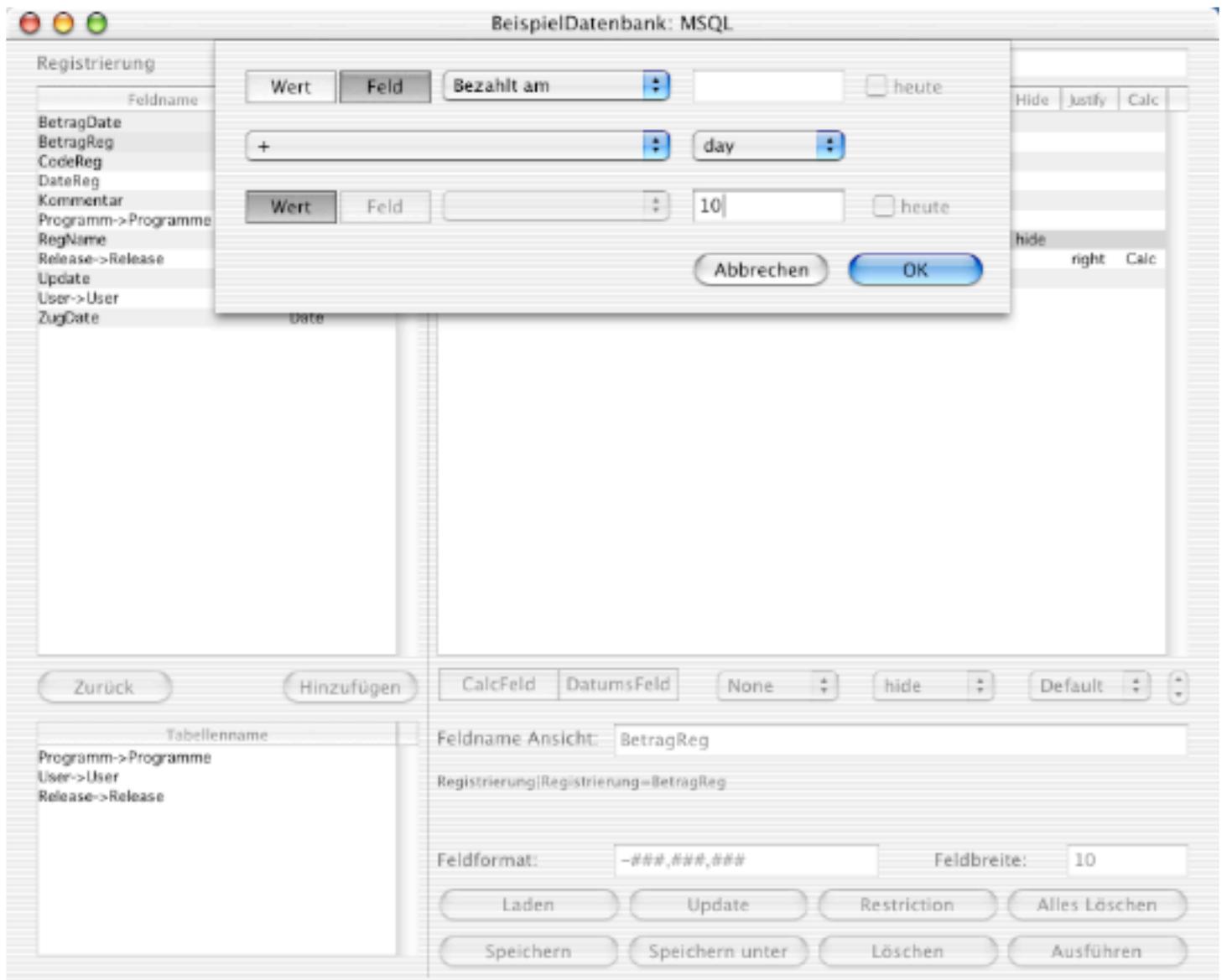
Datenbankstruktur bewegen. Mit dem Button "Zurück" geht es den Hierarchieweg rückwärts, bis man sich wieder an der Ausgangsposition befindet. Durch das Navigieren in der Datenbankstruktur und das gezielte Auswählen von Tabellenfeldern, kann man sich seine individuellen Reports zusammen stellen.

Wenn man in der größten ListBox, die die Reportbestandteile anzeigt, eine Zeile auswählen, dann werden änderbare Daten unterhalb der Listbox angezeigt und auch der Datenbankpfad (bei CalcFeldern und DatumsFeldern die Berechnungsformel), der zu diesem Feld führt. Gleichzeitig wird in den Listboxen für die Tabellen und Tabellenfelder genau dieses ausgewählte Feld und der dazugehörige Pfad nachgebildet und angezeigt (Nicht bei CalcFeld und DatumsFeld). Alle änderbaren Informationen können dann angepasst und mit dem Button "Update" für den Report aktiviert werden. Mit dem Button "Alles Löschen" lässt sich der gesamte Report aus dem Arbeitsspeicher entfernen und mit dem Button "Löschen" die jeweils ausgewählte ReportZeile. Sollte auf diese zu löschende Zeile jedoch eine Referenz zu einer Restriction oder zu einem CalcFeld/DatumsFeld bestehen, dann erscheint eine Warnmeldung und die Löschung wird nicht durchgeführt. Über den Button "Speichern" erscheint ein Speicherndialog, über den man den Report in seinem aktuellen Zustand speichern kann. Mit dem Button "Laden" kann man einmal abgespeicherte Reports auch wieder in MSQL laden. Dabei macht MSQL einen kleinen Test der Datei um abzufragen, ob es sich bei der Datei, die versucht wird zu laden, auch tatsächlich um eine MSQL-Datei handelt.

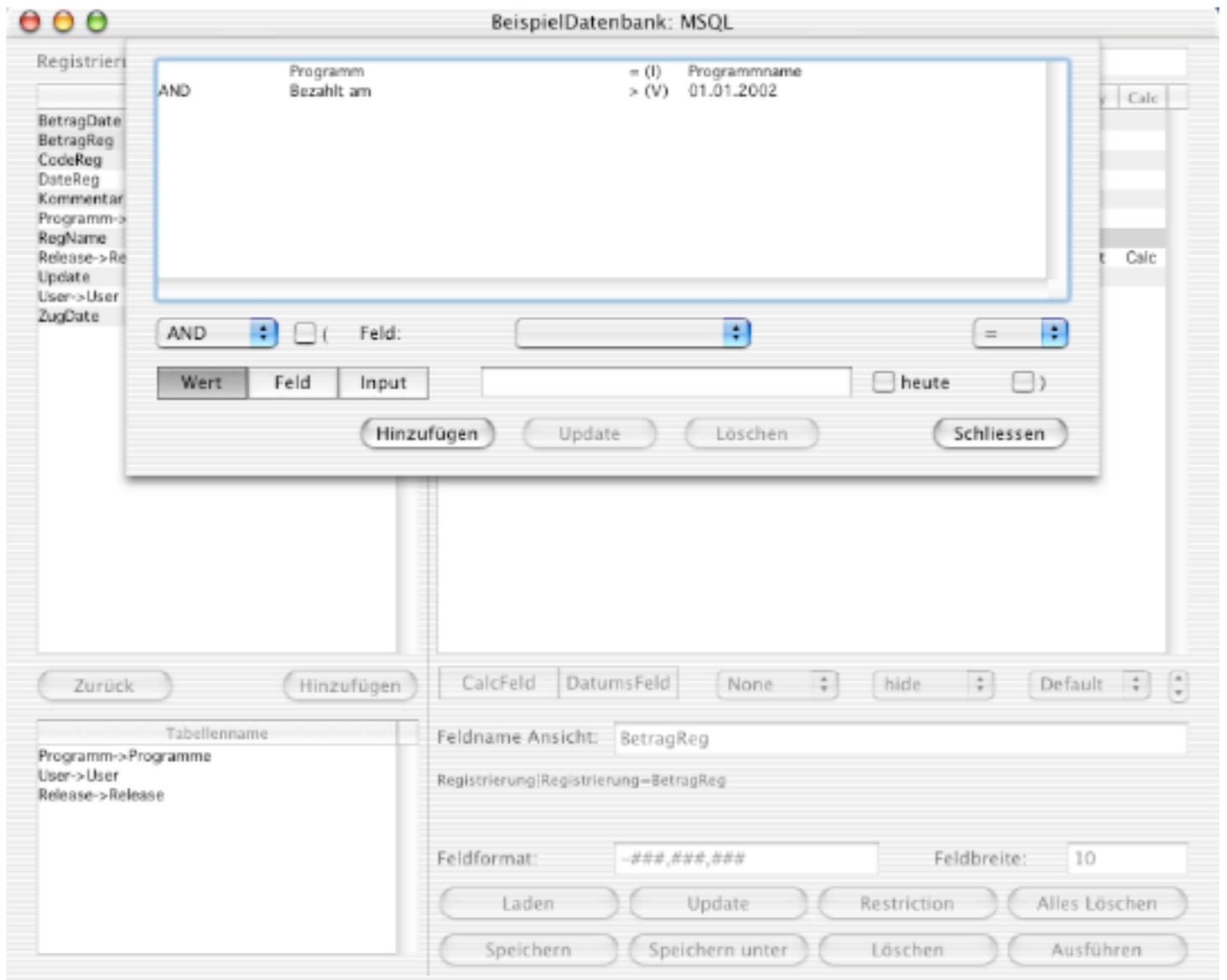
Wenn man den Button "Ausführen" drückt, dann wird der Report mit seinen Bestandteilen erst zusammengebaut und danach erstellt. Dabei kann, sobald das Fenster „Result“ angezeigt wird, im Fenster „MSQL“ weitergearbeitet werden, ohne dadurch die Ergebnisse des Reports zu verändern. Erst bei nochmaligem drücken des "Ausführen"-Buttons werden die Änderungen des Reports an das Fenster „Result“ übertragen. Somit kann im Fenster „Result“ der Report beliebig oft wiederholt werden, auch wenn im Fenster „MSQL“ die Reportmerkmal bereits massiv verändert wurden oder ein anderer Report über den Button "Laden" geladen wurde.



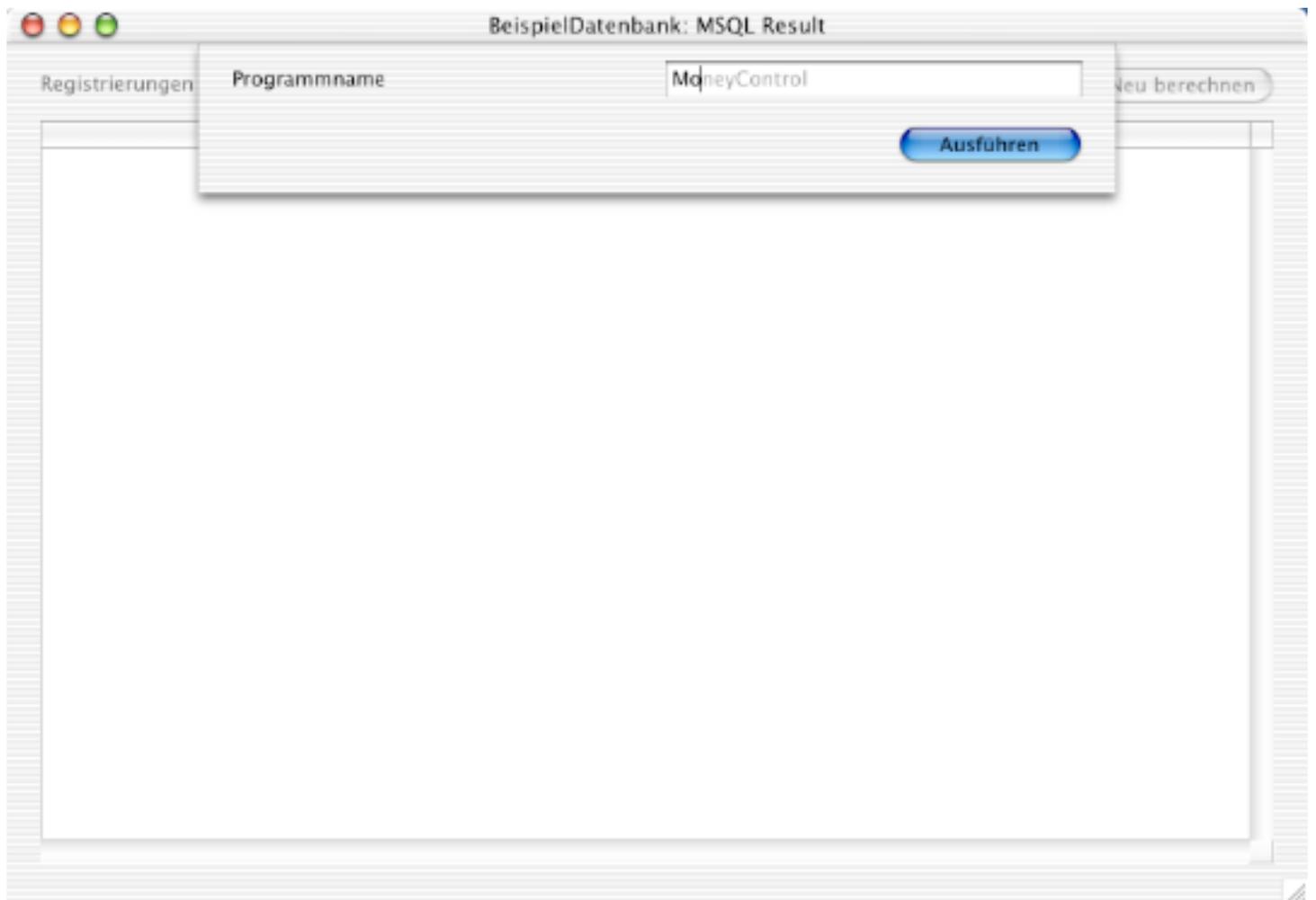
CalcFelder geben die Möglichkeit Berechnungen in MSQL zu integrieren. Da alle Zahlen in der "Application Manager"-Datenbank als Ganzzahlen abgelegt sind, muss man, um z.B. die korrekten Beträge für die Registrierungsgebühren zu bekommen, den Betrag durch 100 teilen.



Mit Datumsfeldern kann man verschiedene Berechnungen mit Datumsfeldern erstellen.



Bei Restrictions werden dem Report Einschränkungen hinzugefügt. Dabei können statische wie dynamische Einschränkungen verwendet werden. Alle statischen Einschränkungen werden beim „Ausführen“ des Reports direkt in SQL-Code umgesetzt. Alle dynamischen Einschränkungen werden nach dem Drücken des Buttons „Ausführen“ angezeigt und abgefragt.



Hier sieht man z.B. eine dynamische Einschränkung, wobei abgefragt wird welchen „Programmnamen“ man sich anzeigen lassen möchte. Sollte bei den dynamischen Einschränkungen falsche Werte eingegeben werden, dann kann auch der Report keine vernünftigen Ergebnisse ausgeben.

BeispielDatenbank: MSQL Result

Registrierungen eines Programmes Neu berechnen

NachName	Vorname	eMail	Programm	Version	Bezahlt am	Reg-Betrag	R
Wasti	Raffael	wasti@mac.com	MoneyControl	1.3.0	14.02.2002	15,00	
Wühler	Alfred	Alf@t-online.de	MoneyControl	1.3.0	25.02.2002	15,00	
von Gestern	Markus	Markus@gestern.ch	MoneyControl	1.3.0	05.03.2002	15,00	
Kazatakis	Dionysos	dk@t-online.de	MoneyControl	1.3.0	06.03.2002	15,00	
Flamm	Franco	franco@tester.de	MoneyControl	1.3.0	07.03.2002	15,00	
Fritz	Reinhard	family@gool.de	MoneyControl	1.3.0	08.03.2002	15,00	
Quasina	André	A.Quasina@gmx.de	MoneyControl	1.3.5	24.05.2002	15,00	
Sommerfeld	Bernd	Arzt@t-online.de	MoneyControl	1.3.6	25.06.2002	15,00	
Reusch	Peter W.	p.reusch@xxx.de	MoneyControl	1.3.6	04.07.2002	15,00	
Toom	Rainer	rtoom@gmx.de	MoneyControl	1.3.6	01.08.2002	15,00	
Rader	Steve	steve@trallala.au	MoneyControl	1.3.7b4	28.08.2002	15,00	
Siebel	Wolf	wolf@t-online.de	MoneyControl	1.3.8	16.09.2002	15,00	
Gestern	Olaf	Olaf@Gestern.com	MoneyControl	1.3.8	17.09.2002	15,00	
Westermann	Klaus-Dieter	Westermann-Gabbeck@t-online.de	MoneyControl	1.3.8	17.09.2002	15,00	
Mansfeld	Sibylle	siby@gmz.de	MoneyControl	1.3.8	17.09.2002	15,00	
Zyska	Tim	tzyska@uni-bonn.de	MoneyControl	1.3.8r3	07.10.2002	15,00	
Hauser	Klaus	khauser@woonline.de	MoneyControl	1.3.0b1	02.01.2002	15,00	

17 Datensätze

Das „Result-Fenster“ gibt das Ergebnis des Reports aus.

FensterSQL

```

SELECT t2.NachName, t2.Vorname, t2.eMail, t3.PName, t4.Version, t1.BetragDate, t1.BetragReg, t1.DateReg
FROM Registrierung t1, User t2, Programme t3, Release t4
WHERE t1.User=t2.RecID
AND t1.Programm=t3.RecID
AND t1.Release=t4.RecID
AND ( t3.PName='MoneyControl'
AND t1.BetragDate>'01.01.2002'
)
ORDER BY t1.DateReg ASC

```

Schliessen

Über ein Contextual-Menü, das man im „Result-Fenster“ aktivieren kann, kann man sich den Report-SQL-Code anzeigen lassen. Man kann auch die Input-Einschränkung sehen  
 „AND (t3.PName='MoneyControl'“

## Anmerkungen/Einschränkungen

Trotz der Möglichkeiten, die MSQL bietet, gibt es leider auch Einschränkungen, die MSQL mit sich bringt und die zum Einen an der Entwicklungsumgebung RealBasic liegen oder zum Anderen an der darunter liegenden Valentina Datenbank.

### RealBasic

- Da RealBasic intern nur Ganzzahlen mit einfacher Länge (4 Byte) verarbeiten kann, kann ein MSQL-Report nur aus maximal 32 Zeilen bestehen. Ob überhaupt je diese Grenze mit einem Report erreicht wird, kann ich nicht sagen, halte ich derzeit jedoch für sehr unwahrscheinlich und somit eher für die Ausnahme. Sollte es in RealBasic irgendwann auch Ganzzahlen mit 8 Byte geben, würde sich die maximale Zeilenzahl auf 64 erhöhen.

### Valentina

- Valentina kann derzeit keine Berechnungen im SQL-Statement (im SELECT und WHERE) ausführen. Von daher werden sämtliche CalcFelder und DatumsFelder ausserhalb des SQL-Statements ermittelt und dargestellt. Das führt zu einer spürbaren Verlangsamung der Performance, so dass man mit der Verwendung von CalcFelder und DatumsFelder sparsam umgehen sollte. Die Berechnung der CalcFelder wird mit RBScript durchgeführt, das bei einer fehlerhaften Formel ERROR als Ausgabe zurückgibt. Bei den Datumsberechnungen wird ebenfalls bei einem Fehler ERROR ausgegeben. Die Berechnung hierfür wird in RealBasic direkt durchgeführt.
- Der gesamte Prozess zur Ermittlung und Darstellung des Reports wurde in einem Thread ausgelagert, so dass mit MSQL und "Application Manager" weiter gearbeitet werden kann.
- Aus dem oben aufgeführten Grund, dass Valentina derzeit keine Berechnungen im SQL-Statement ausführen kann, können für CalcFelder und DatumsFelder keine Restrictions erfaßt werden und bei CalcFelder und DatumsFelder auch nicht auf CalcFelder oder DatumsFelder als Berechnungsgrundlage/Referenz zugegriffen werden.
- Wenn 2 Tabellen über Tabellenfelder miteinander in den Restrictions verknüpft werden sollen, dann unterstützt Valentina nur einen Istgleich-Vergleich. (z.B. Tabelle1.Nachname = Tabelle5.Name)
- Restrictions lassen auch LIKE-Bedingungen zu (Gilt für sämtliche Textvergleiche jedoch nicht für Zahlen oder Datum). Um eine Restriction als LIKE-Restriction erkennen zu können, muss der Bedingungsteil wie folgt aussehen:
  - %LikeText% = Vor "LikeText" und nach "LikeText" ist beliebiger Text möglich
  - %LikeText = Der Suchtext endet mit "LikeText"
  - LikeText% = Der Suchtext beginnt mit "LikeText"

# RBScript

Im Ordner RBScript finden Sie Beispiele für einen möglichen Code zur Erzeugung von RegistrierungsCodes. Aus „Application Manager“ können immer folgende Werte übergeben werden:

- ◇ `UserName=Input("UserName")` --> das entspricht ‚Registrierungsname‘
- ◇ `Programm=Input("Programm")` --> das entspricht ‚Programmname‘
- ◇ `CD_Zwei=Input("CD_Zwei")` --> das entspricht ‚CodifierEins‘
- ◇ `CD_Drei=Input("CD_Drei")` --> das entspricht ‚CodifierZwei‘

Damit „Application Manager“ ein RBScript-Programm zur CodeGenerierung verwendet, muss die Datei mit dem RBScript-Code im Programmordner liegen und den Namen „ApMaRBScript“ haben.

## RBScript Sprachumfang

### Hinweise

Die RBScript-Sprache ist eine Implementierung der REALbasic-Sprache, die es dem Anwender erlaubt, REALbasic-Code innerhalb einer compilierten Anwendung zu schreiben und auszuführen. Scripte werden in Maschinensprache compiliert und laufen nicht durch einen Interpreter.

Wie oben schon erwähnt können 4 Werte übergeben werden. Ob Sie diese auch in Ihrem RBScript verwenden und/oder abfragen wollen bleibt Ihnen überlassen. Um am Ende einen RegistrierungsCode zurückgeben zu können müssen Sie.

```
Print ‚String‘
```

verwenden. Wobei String für den neu generierten RegistrierungsCode steht. Sollte kein vernünftiger Wert ermittelt werden können, empfehle ich einen LeerString zurückzugeben.

### Fehlerbehandlung im Ihrem Code

Wenn ein Fehler auftritt, wenn das RBScript Ihren Code compiliert, wird der CompilerError-Event des RBScript-Objekts aufgerufen und liefert die passende Fehler-Information, so daß Sie entscheiden können, wie der Fehler behandelt werden soll. Wenn der Fehler auftritt, während der Code am Laufen ist, wird der RuntimeError-Event des RBScript-Objekts aufgerufen und liefert die passende Fehler-Information. Sie können dann entscheiden, wie auf den Fehler reagiert werden soll.

### Variablen und Konstanten

Alle numerischen und alphanumerischen Operatoren werden unterstützt:

`+`, `-`, `*`, `/`, `\`, `Mod`, `<`, `=`, `>`, `<=`, `>=`, `<>`

Logische Operatoren: `And`, `Not`, `Or`.

Alle Formen von Kommentaren werden unterstützt: `'`, `//` und `REM`.

### Daten-Typen

RB Script unterstützt die folgenden Daten-Typen:

- Integer
- Single
- Double
- Boolean
- String
- Variant

Arrays können jeden dieser Typen verwenden.

## Kontrollstrukturen

Alle Kontrollstrukturen, die in dieser *Sprachreferenz* festgelegt sind, werden unterstützt. Dies beinhaltet:

- ◇ For ... Next
- ◇ Do ... Loop
- ◇ If ... Then ... End If
- ◇ Select Case ... End Select
- ◇ While...Wend
- ◇ Return Befehl

## Funktion Beschreibung

Input(Prompt als String) Bekommt die Eingabe vom Anwender. (UserName=Input("UserName"), Input("Programm"), Input("CD\_Zwei"), Input("CD\_Drei"))

Print(String) Sendet den ‚String‘ an „Application Manager“ zurück

- Abs(Double) As Double
- Acos(Double) As Double
- Asc(String) As Integer
- AscB(String) As Integer
- Asin(Double) As Double
- Atan(Double) As Double
- Atan2(Double, Double) As Double
- BitwiseAnd(Integer, Integer) As Integer
- BitwiseOr(Integer, Integer) As Integer
- BitwiseXor(Integer, Integer) As Integer
- CDbI(String) As Double In jedem Fall zu Val() identisch
- Ceil(Double) As Double
- Chr(Double) As String
- ChrB(Double) As String
- Const Name = Wert
- Cos(Double) As Double
- CountFields(String, String) As Integer
- CStr(Double) as String
- Dim Erlaubt es eher, mehrere Variablen von verschiedenem Typ zu deklarieren, als nur mehrere Variablen des gleichen Typs. Leere Arrays können durch Angabe von -1 Elementen erzeugt werden.
- Do...Loop
- Exit
- Exp(Double) As Double

- False
- Floor(Double) As Double
- For...Next Erlaubt Fließkomma-Schleifenzähler und Schrittwerte
- Format(Double, String) As String
- Goto Schlüsselwort
- Hex(Integer) As String
- If...Then...Else
- InStr(Integer, String, String) As Integer
- InStrB(Integer, String, String) As Integer
- Left(String, Integer) As String
- LeftB(String, Integer) As String
- Len(String) As Integer
- LenB(String) As Integer
- Log(Double) As Double
- Lowercase(String) As String
- LTrim(String) As String
- Max(Double, Double) As Double
- Microseconds As Double
- Mid(String, Integer, Integer) As String
- MidB(String, Integer, Integer) As String
- Min(Double, Double) As Double
- NthField(String, String, Integer) As String
- Oct(Integer) As String
- Pow(Double, Double) As Double
- Redim
- Rem
- Replace(String, String, String) As String
- ReplaceAll(String, String, String) As String
- Right(String, Integer) As String Arbeitet wie RightB.
- RightB(String, Integer) As String
- Rnd As Double
- Round(Double) As Double
- RTrim(String) As String
- Select Case
- Sin(Double) As Double
- Sqrt(Double) As Double
- Str(Double) As String
- StrComp(String, String, Integer) As Integer
- Tan(Double) As Double
- Ticks as Integer
- Titlecase(String) As String
- Trim(String) As String
- True
- Ubound(Array) As Integer
- Uppercase(String) As String
- Val(String) As Double Wissenschaftliches Format wird nicht erkannt.
- While...Wend

## **Fehlernummer Beschreibung**

- 1 Syntax-Fehler.
- 2 Falscher Typ.

- 3 Select Case unterstützt diese Ausdrucks-Typ nicht.
- 4 Der Compiler ist nicht implementiert (überholt).
- 5 Der Stack des Parsers ist übergelaufen.
- 6 Zu viele Parameter für diese Funktion.
- 7 Nicht genug Parameter für diese Funktion.
- 8 Die Anzahl der übergebenen Parameter für diesen Funktionsaufruf stimmt nicht.
- 9 Parameter sind inkompatibel mit dieser Version.
- 10 Zuweisung eines inkompatiblen Datentyps.
- 11 Nicht definierter Identifier.
- 12 Nicht definierter Operator.
- 13 Logische Operationen benötigen boole'sche Operanden.
- 14 Array-Grenzen müssen Integer sein.
- 15 Eine Nicht-Funktion kann nicht aufgerufen werden.
- 16 Ein Element kann nicht von etwas angefordert werden, das kein Array ist.
- 17 Nicht genug Indizes für die Dimension dieses Arrays.
- 18 Zu viele Indizes für die Dimension dieses Arrays.
- 19 Sie können einem Wert nicht auf einmal ein ganzes Array zuweisen.
- 20 Ein ganzes Array kann nicht auf einmal in einem Ausdruck verwendet werden.
- 21 Ein Ausdruck kann nicht als ein ByRef-Parameter übergeben werden.
- 22 Doppelter Identifier.
- 23 Der Back-End-Code-Generator versagte.
- 24 Zweideutiger Aufruf einer überladenen Methode.
- 25 Mehrfache Vererbung ist nicht erlaubt.
- 26 Kann keine Instance von einem Interface erzeugen.
- 27 Kann eine Klasse nicht implementieren als wäre es ein Interface.
- 28 Vererbung funktioniert nur bei Klassen.
- 29 Diese Klasse implementiert das angegebene Interface nicht vollständig.
- 30 Event-Handler können sich nicht außerhalb einer Klasse befinden.
- 31 Das Ergebnis eines Funktionsaufrufs darf nicht ignoriert werden.
- 32 Das „Self“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
- 33 Das „Me“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
- 34 Kann aus einer „Sub“-Routine keinen Wert zurückliefern.
- 35 Hier wird ein Exception-Objekt benötigt.
- 36-39 Überholt
- 40 Destruktoren dürfen keine Parameter haben
- 41 Das „Super“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
- 42 Das „Super“-Schlüsselwort darf nicht in einer Klasse verwendet werden, die keine Eltern hat.

Einige der hier aufgeführten FehlerCodes sind für „Application Manager“ irrelevant. Aus Gründen der Vollständigkeit habe ich sie aber trotzdem aufgeführt.

## Sonstige Funktionalitäten

- Da das Programm auf einer Datenbank von Valentina aufbaut, gibt es die Möglichkeit alle Datenbankbestandteile in eine Textdatei zu exportieren und auch wieder zu importieren. Beim Import aus einer Textdatei werden alle Datenbankeinträge zuerst gelöscht und dann mit den zu importierenden Daten gefüllt.
- In der Datei „Voreinstellungen“ werden die Voreinstellungen gespeichert, die Größe und die Position des Hauptfensters und die zuletzt geöffnete Datenbank.

- Wurde die BackUp-Funktionalität aktiviert, dann legt das Programm automatisch einen Unterordner „BackUp“ an und speichert dort die BackUp-Dateien ab.

## **Sharewaregebühren**

Sollte Ihnen „Application Manager“ zusagen, dann wenden Sie sich bitte bezüglich eines RegistrierungsCodes und der Sharewaregebühren von EUR 15,- an folgende eMail-Adressen: [ApplicationManager@sailer-online.de](mailto:ApplicationManager@sailer-online.de)

## **Haftung**

Als Autor und Verfasser von „Application Manager“ übernehme ich für etwaige Schäden oder Fehlkalkulationen sowie anderweitige Probleme keine Haftung. Die Benutzung von „Application Manager“ findet unter dem Risiko eines jeden Nutzers selbst statt.

## **Schlussbemerkung**

„Application Manager“ steht in einer Version für MacOS X zur Verfügung (läuft derzeit nicht unter UFS). Bei Fragen oder Anregungen wenden Sie sich bitte an [ApplicationManager@sailer-online.de](mailto:ApplicationManager@sailer-online.de). Das Programm darf nur als Komplettpaket (Programm, Dokumentationen, Beispieldatei und Startdatei) weitergegeben werden.

## **Systemvoraussetzungen**

PPC mit MacOS X, mindestens 20MB Platz auf der Festplatte (HFS+) und eine Monitorauflösung von mindestens 800x600Punkten.