

**Default**

**COLLABORATORS**

	<i>TITLE :</i> Default		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 8, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Default</b>	<b>1</b>
1.1	Einleitung . . . . .	1
1.2	Vorwort . . . . .	2
1.3	Installation . . . . .	3
1.4	Konfigurieren . . . . .	3
1.5	Features . . . . .	4
1.6	Programme . . . . .	5
1.7	Dank . . . . .	6
1.8	Shareware . . . . .	6
1.9	Allgemein . . . . .	7
1.10	Probleme . . . . .	8
1.11	History . . . . .	8
1.12	Projekt Guidebuilder . . . . .	10
1.13	Projekt ArexxManager . . . . .	10
1.14	Projekt DosManager . . . . .	11
1.15	Projekt Editor . . . . .	11
1.16	RexxStart . . . . .	13
1.17	RexxSpell . . . . .	13
1.18	Grundlegendes . . . . .	14
1.19	Bedienung . . . . .	16
1.20	Fenster Guidemanager . . . . .	19
1.21	Fenster Linkmanager . . . . .	20
1.22	Appwindow . . . . .	22
1.23	Guide-Editor-Menü . . . . .	23
1.24	Guide-Kurs . . . . .	23
1.25	Guidekonzept . . . . .	24
1.26	Guide Auslesen . . . . .	25
1.27	Texte Auslesen . . . . .	26
1.28	Verknüpfen Mit . . . . .	28
1.29	Layout . . . . .	28
1.30	Colour . . . . .	29
1.31	Markierung . . . . .	29
1.32	Head Erneuern . . . . .	30

---

# Chapter 1

## Default

### 1.1 Einleitung

SkriptManager V\_1.4 22.05.99  
Copyright by Heiko Strohmeier  
Kleiststrasse 10  
50389 Wesseling  
Germany  
Email: HStrohmeier@T-Online.de

Eigentlich schade, daß das allgemeine Feedback für mein Programm bisher recht mager ausgefallen ist. Schlecht ist es ja nun wirklich nicht. Vielleicht liegt es doch tatsächlich an der fast völlig uneingeschränkten Version meiner Releases. Bisher dachte ich immer, daß es fairer wäre alle Funktionen bereitzustellen, da der Spaß beim kennen lernen größer ist. Ich werde also in Zukunft hier etwas umdenken müssen. ...Leider!

Vorwort  
Dies und das

Installation  
Einrichten des Skriptmanagers

Konfigurieren  
Preferenceprogramm vorher starten!

Features  
Was ist alles drin?

Programme  
Hauptprogramme und alle Funktionen

Dank  
Danksagungen, denn man schafft nie alles allein!

Shareware  
LegalStuff

---

Allgemein  
Bedingungen, Haftung, Zukünftiges

Probleme  
Keine Proportional-Zeichensätze (times, ect.) verwenden!

History  
Geschichte halt!

## 1.2 Vorwort

Hallo zusammen!

Die Idee einen GuiderBuilder zu schreiben entstand, als ich damals mein erstes Projekt "Runner\_V1.0" fertig stellte. Es fiel mir schwer das passende Programm zu finden, da es kaum eins gab; und das mit dem ich herumbastelte, verabschiedete sich recht gerne mit abschließender Gurumeldung. (Ich habe die Anleitung für Runner dann per Hand geschrieben.)

Die Handhabung meines Programms sollte auf jeden Fall viel effizienter und flexibler sein; schon gar nicht abstürzen, sollte man mal von der üblichen Verfahrensweise abweichen.

Die Übersicht beim Erstellen eines AmigaGuide Dokuments, sowie das Abändern von Passagen im Quelltext, ist für den Schreiber meistens etwas erschwerlich. Deshalb habe ich beim Erstellen eines Guides eine Datenerhebung integriert. Der Guiderbuilder verfügt über eine Datenorganisation, mit der bequem alle Bestandteile (die vorher eingegeben werden müssen-loggo) zusammen gefügt werden. Dieses Verfahren schließt die meisten standardmäßigen Datenbankvorgänge wie, Projekt laden, Add, löschen, nach oben rücken, nach unten rücken usw. ein.

Es können quasi definierte Links über einen bequemen Klick im Listing des Linkmanagers

Bedienung  
in den Quelltext eingefügt

werden.

Externe Texte, Bilder, Programmstarts, lha-lzx Archive werden über Menüpunkte eingebunden. Texte können nachträglich ohne bereits vorhandenen Quellformattext, im Guidebuilder-Editor, mit AmigaGuide Formatstrings versehen werden. Hierfür existieren Menüpunkte,

GuiderbuilderMenü  
die diese Arbeit imens

erleichtern.

Die Arexx Funktionen im Editor kamen dazu, als ich endlich dank dem tollen Arexx-Kurs von Heiko Kuschel (mehr dazu:

Projekt ArexxManager  
) , diese

Sprache ohne große Schwierigkeiten lernte. Ich schreibe heute alle meine Skripte mit diesem Editor.

Und da es hin und wieder auch lohnt Dosskripte anzufertigen, habe ich eine

---

Startfunktion dafür erstellt.

Und nun-----Viel Spaß mit Skriptmanager

## 1.3 Installation

Die Schublade Skriptmanager irgendwo auf eine Partition (oder ←  
Diskette)

kopieren.

Anschließend die libraries per Installskript (je nach Prozessortyp) ausführen.  
(die im Verzeichnis Skriptmanager/Libs/ befindlichen library's werden nach  
libs: kopiert.)

Zum Schluß das Preferenceprogramm  
Konfigurieren  
starten.

Das Preferenceprogramm (EditPrefs) ist unbedingt vor dem Start des  
Hauptprogramms (Editor) zu öffnen, ansonsten wird der Skriptmanager auf  
keinen Fall starten. Dies gilt auch für das Update auf Version 1.3

Inhalt:

Skriptmanager/Editor  
Skriptmanager/EditPrefs  
Skriptmanager/Libs/Cando.library  
Skriptmanager/Libs/Powerpacker00.library  
Skriptmanager/Libs/Powerpacker20.library  
Skriptmanager/install-libs68000  
Skriptmanager/install-libs68020  
Skriptmanager/Skriptmanager.guide  
Skriptmanager/Cookie.txt  
Skriptmanager/Spell.DAT

Systemvoraussetzungen:

Amiga mit 1MB FastRam und mindestens OS 2.0.

Testrechner:

Amiga1200/ Blizzard1260 / 32MBFastram.

## 1.4 Konfigurieren

Konfiguriert wird Skriptmanager mittels des Preferenceprogramms "Editprefs".  
(Menüpunkt: User/Preference).

Das Preferenceprogramm sollte sich im gleichen Verzeichnis wie der  
Skriptmanager befinden.

Zeichensatz:

Der Zeichensatz, mit dem Skriptmanager arbeiten soll.

Höhe:

Zeichensatzhöhe

Multiview:

---

Der Programmpfad plus Programmname. (Multiview, AmigaGuide, als auch ein gleichwertiges Anzeigeprogramm, welches kompatibel zu den beiden Workbenchprogrammen ist.)

RexxMast:

Der Programmpfad plus Programmname.

Rexx Com.:

Der Programmpfad für die Arexx Hilfsprogramme. Installationsgemäß befinden die sich in: SYS:Rexxc

RexxSkripte:

Sollte eine Schublade für Arexxprojekte bestehen, dann läßt sich ein jeweiliges Projekt schnell laden, indem hier der Projektpfad eingegeben wurde.

Cookie:

Pfad plus Dateinamen eintragen für die Cookietextdatei. Siehe auch Projekt Editor unter UserMenü: "Email"

Projekt:

Eignet sich hervorragend für AmigaGuide-Projekte, die samt Datendatei sofort ermittelt werden können. (Vorausgesetzt der Projektname besitzt die Endung ".guide". Ansonsten wird das Projekt als normale Editierfile behandelt) siehe unter ---Menüpunkt User / Projekt--- Projekt Editor

Skripte:

Hier in den 5 Feldern unter Skripte: werden die im Editor direkt ladbaren Dateien eingetragen.

## 1.5 Features

GuideBuilder:

Der Skriptmanager verfügt über mehrere Funktionen.

Die wesentlichste Funktion ist die Erstellung eines AmigaGuide oder Multiview-Dokuments.

Fremde nicht mit Skriptmanager generierte Guides können ausgelesen werden, und durch die sehr flexible Skriptmanager-Funktionsweise, einer Datenbankmethodik, beliebig verändert werden.

Reine Textdateien können ebenfalls ganz leicht zu einem Guide kreiert werden.

Die Editorfunktion erlaubt das Nachträgliche Editieren aber auch Neuerstellen von Texten, die per Querverweis - Look - im Amigaguide Dokument ihre Dienste verrichten sollen.

Projekt Guidebuilder

E-Mail:

Der Editor selbst kann problemlos für jegliche andere Schreibarbeit herangezogen werden.

So zum Beispiel, schreibe ich alle meine Mails damit. Der Editor erkennt automatisch von wo aus er gestartet wurde, vorausgesetzt man konfiguriert das entsprechende Programm (von wo aus der Editor gestartet wurde), korrekt. Im Falle der Nutzung in einem Emailprogramm, gibt es eine Randomfunktion, die dafür sorgt, das stets ein Cookie Spruch bereit steht. Projekt Editor

ArexxManager:

---

Mit Hilfe des ArexxManagers können ArexxSkripte geschrieben, ausprobiert sowie auf Fehler hin, überprüft werden. Es gibt eine ArexxPortShow Funktion und mit etwas Glück, bald sogar eine direkte OnlineHilfe für Arexx-Befehle und Funktionen.

Projekt ArexxManager

DosSkript:

Auch geschriebene DosSkripte können aus dem Editor heraus gestartet und ausprobiert werden.

Projekt DosManager

## 1.6 Programme

Enthaltene Programme und Funktionen:

Eins vorweg. Die Guidebuilderfunktion des Skriptmanagers ist ab Version 1.3 stark überarbeitet, was sich in der Guide-Erzeuge-Bedienung widerspiegelt. War es vorher nötig ALLE Links im Falle der Unterlinkdefinition mit einzuschließen, um die Nodes auch im generierten Quelltext vorzufinden, ist es Dank eines neuen Erzeuge-Algorithmus nicht mehr erforderlich. Neu hinzugekommen ist, eine Definition (Eingabe eines Zahlenwertes, der absolut erforderlich bei jedem Skriptmanager konzipiertem Guide ist->nicht bei ausgelesenen) der Links, die Bestandteil der Hauptseite (Main-Node) werden sollen. Dieses neue Konzept bringt speziell ausgelesenen Guides 100% Kompatibilität und sorgt dafür, daß diese Guides vollen Datenbank Support bekommen.

Die Neuerungen sind im Guidebuilderabschnitt dieses Guides umfangreich dokumentiert.

Noch ein Wort an alle User, bei denen trotz der Anleitung eventuelle Unklarheiten aufkommen.

Da spezifische Anwendungen im Guidebuilder auch intensives Lesen des Guides voraussetzen (..grins), und ich eventuell dennoch einige Sachen nicht deutlich genug angesprochen habe, ist es wirklich kein Problem, wenn man mir schreibt und ich somit auf Fragen eingehen kann!

Projekt Guidebuilder

Bedienung - Funktionen - Generieren eines Guides

Projekt ArexxManager

ArexxSkripte produzieren - Alle Funktionen!

Projekt DosManager

Handelt eigentlich nur um Skript Edit&Start

Projekt Editor

Editor im Skriptmanager

## 1.7 Dank

An dieser Stelle möchte ich allen Danken, die mit Tips und Rat zur Seite standen.

Besonders danke ich der CanDo Hotline, welche durch Christian Effenberger verkörpert wird, dem amigaOS Redakteur Heiko Kuschel, welcher imens wichtige, zum Teil auch kritische Hinweise gab, und natürlich Hartmut Schuhmacher vom AmigaPlus Magazin, der so freundlich war mein Programm in die PD-News (APlus 3.99 + 6.99) zu bringen, und im Grunde genommen meine Begeisterung für die Programmierung, durch seinen klasse CanDo Programmierkurs (12.94), weckte.

Noch einen Dank an die amigaOS (Kurzvorstellung 4.99), und speziell an den Chefredakteur Thomas Raukamp, der die Copyright Texte zum Arexxspeller zur Verfügung stellte.

## 1.8 Shareware

Shareware:

Das Programm Skriptmanager ist Shareware. Das bedeutet, das Ihr dieses Programm eine gewisse Zeit (bis 30 Tage) kostenlos nutzen dürft. Nach diesem Zeitraum ist eine Registrierung notwendig.

Die Höhe der Registrierungsgebühr für Skriptmanager ist 20 DM. Wenn mit einem Scheck bezahlt wird, der von einer ausländischen Bank stammt oder nicht in DM ausgestellt ist, sind nochmals 5,- DM oder 3\$ zusätzlich hinzu zu fügen, da es hier sehr teuer ist, solche Schecks einzulösen.

Die für die Registrierung gängigen Methoden könnt Ihr weiter unten nachlesen!

Das Programm Skriptmanager und die zugehörigen Dateien sind kopierrechtlich geschützt. (c) Heiko Strohmeier .

Sollte Skriptmanager auf einer Diskette kopiert werden, darf die Gebühr höchstens 4 DM für Disketten betragen.  
(Außnahme: AmigaPlus-Magazin, Amiga Spezial, oder andere Amiga-Printmagazine)

Skriptmanager darf in Public-Domain-Sammlungen aufgenommen werden einschließlich CD-ROMs.

Jegliche weitere kommerzielle Verbreitung oder Nutzung von Skriptmanager und den dazugehörigen Dateien ist ohne meine Genehmigung hiermit untersagt.

Adresse:

Heiko Strohmeier  
Kleiststrasse 10  
50389 Wesseling  
Germany

Allgemein wird es gerne gesehen, wenn man mir eine Email, einen Brief schreibt, in dem etwas über das Programm steht (kritisches oder positives).

---

Ich versende Disketten, wenn mir ein ausreichend frankierter Briefumschlag plus vorformatierter Diskette, zugeschickt wird.

Außerdem ist eine Registrierung, die Skriptmanager freischaltet (den Nervrequester also auch die Einschränkungen beseitigt) nun per Email erhältlich!

email:HStrohmeier@T-Online.de

## 1.9 Allgemein

Haftung und Bedingungen:

Das Programm darf zur Probe 30 Tage benutzt werden.

Danach sollte eine Registrierung in Form der erforderlichen Registriergebühr (20 DM) erfolgen.

Die Benutzung des Programms geschieht ausschließlich auf eigenes Risiko, so übernehme ich keinerlei Haftung für irgendwelche Schäden, die bei der Arbeit mit dem Programm Skriptmanager auftreten.

Skriptmanager ist in der vorliegenden Version nun etwas eingeschränkt. Leider war es sehr erforderlich, da ich wohl ein wenig zu leichtgläubig bin.

-Skriptmanager hat eine max. Laufzeit von 1ner Stunde.

-Der Arexxspeller funktioniert nur für max. 3 Abfragen; das Fenster schließt automatisch nach 3 min. und speert das komplette Programm 10sek. lang.

-Die Speicherfunktion für den Guidebuilderteil ist gesperrt.

-Die Appwindow Option funktioniert in der unregistrierten Version nur für einmaliges Drag&Drop.

-Der Nervrequester erscheint alle 5 min. und deaktiviert Skriptmanager für eine Dauer von 10 sek!

Und nun noch etwas erfreuliches:

```
*****
*   AUCH DIESE NEUE VERSION VON SKRIPT- *
*   MANAGER IST FÜR REGISTRIERTE USER  *
*   FREI BENUTZBAR UND KOSTENLOS!!!    *
*****
```

Zukünftiges:

HTML Export.

AppwindowFunktion verbessern.

Werkzeugboxen.

Arexxspeller Datenfile überarbeiten.

Dos-Speller integrieren.

Der Editor wird weiter ergänzt und verbessert.

Englischversion.

Wenn mir die Zeit bleibt und in Abhängigkeit mit dem künftigen Interesse am Skriptmanager starte ich weitere Projekte, die dem Konzept des Guidebuilders ähneln werden.

Weitere Projekte sind :Installationsmanager und WebManager.

---

## 1.10 Probleme

Probleme sollten eigentlich nicht auftreten.  
Sollte es dennoch mal zu einem Bug kommen, schreibt mir bitte und fügt eine Schilderung über den Vorfall bei.

Es gibt jede Menge Hinweis-Requester für eventuelle Nachfragen, die sich aus der Handhabung, besonders bei einer abgeratener Handhabung des Guidebuilders

Bedienung  
, ergeben.

Proportionalfont wie z.B times sollten nicht als Eingabe-Ausgabe Fonts des Dokumenten-Object voreingestellt werden. Courier oder Helvetica (FixedWidth-Fonts) sind unbedenklich.  
Generell werden, die voreingestellten WB-Fonts genommen.

Es kommt zwar seltener vor, aber trotzdem sollten keine Textdateien geladen werden, deren Zeilenlänge über 255 Zeichen (pro Zeile) betragen.

GuideAuslesen Funktion funktioniert nicht, wenn das eingeladene Guide keine Dateinamenendung .guide besitzt, oder wenn im Dateinamen ein weiterer Punkt (z.b. Heiko.1.guide -> funktioniert nicht! >>> Heikol.guide wäre besser) enthalten ist.

## 1.11 History

Geschichte des Programms:  
Erste Veröffentlichung der Version 1.0

Version 1.1

Alle schon bestehenden Datenfiles sind kompatibel zur neuen Version!  
-Menuepunkt: "MakeGuide/Guidekonzept/Neues Projekt" hatte einen leichten Funktionsfehler!  
-Unterlinkdefinition hatte einen Bug  
-Linkmanagerbedienung durch Refreshfunktion erweitert  
-Seperates Editieren des Headers  
-direkte Speicherfunktion  
-Wenn Programme ins Editorfenster gezogen werden, wird nun der Programmpfad zum Text dazu schreiben, anstatt wie vorher komplett gelöscht zu werden  
-Wurde die Edit-Funktion betätigt, obwohl im InsertText Feld kein Eintrag vorhanden war, beendete der Editor seine Arbeit!  
-Menüpunkt -Projekt laden- identifiziert das Projekt nebst Namen  
-Abschnitt Guidebuilder im Skriptmanager.guide stark verbessert

Version 1.2

---

- Auslesen fremder schon erstellter AmigaGuides.
- lade/App. Funktion von AmigaGuidedateien implementiert
- Oberflächenanpassung überarbeitet
- windowto front bei erneuter Aktivierung
- DelL und InsertLink korrigiert

#### Version 1.3

Der Guidebuilder wurde gehörig umgearbeitet und ist nun allgemeinverständlicher.

Die überarbeitete Generierungsfunktion läßt mehr Freiheit im Linkmanager sowie beim Anpassen ausgelesener Guides.  
(Links müssen nun nicht mehr komplett Unterlink definiert sein, damit die Nodes beim Generieren erfasst werden)

#### Guidebuilder:

- jeder einzelne Querverweis kann im Linkmanager definiert werden
- Guidebuilderintuition (Gui) optimiert
- nachtägliches Setzen von weiteren Links in ausgelesene Guides
- Ausgelesene fremde Guides können nun 100% kompatibel zum Skriptmanager umgestaltet werden.
- Linkmanager verbessert: Unterlinkdefinition nicht mehr dringend erforderlich. Maindefinition integriert
- einfache Prüfoption für "Guides auslesen" eingebaut (da in Version 1.2 die Gefahr besteht, daß beim Auslesen fremder Guides eine Endlosschleife abläuft)
- Texte Auslesen (reine Texte, deren Überschriften gleichzeitig als Verweis & Node in die Datenmaske übernommen werden)
- direkte Ansicht einer Node (Aufruf per Test), die sich im Editor befindet.

#### Editor:

- Skripte können vordefiniert, und aus dem Menü heraus geladen werden
- Neues Fenster - Option
- Menü-Umgestaltung
- Skriptmanager Guide umgeschrieben und ergänzt.

#### Version 1.4

- Ausgelesene Texte werden erkannt, und können direkt nach dem Laden bearbeitet werden.
- kleinere Unstimmigkeiten zwischen Guidebuilderfunktionen im Zusammenspiel behoben
- Fremde Guides-Auslesen Funktion überarbeitet. (Nun sollten alle möglichen Guide-Erzeuge-Standarts erfasst werden)
- Del-Funktion hatte Probleme mit dem letzten Datensatz von Datenfiles
- Datenerfassung im Datenbank-Guidebuilder korrigiert!
- Kleinere Fehler behoben.

#### NEUE FEATURES:

- Verknüpfenfunktion. Zwei Guides zu einem verbinden!
  - lade/AppWindow Funktion im Guidebuilder zum automatischen Eintragen sämtlicher Linknamen-LinkID-Linktexte, (komfortable Datenerfassung innerhalb weniger Sekunden ohne gängiger Dateneditierung möglich) als auch schnellstmögliche Katalogisierung von Verzeichnissen. (CD oder Festplatte)
  - Arexx Spell-Datenbank mit Funktions und Befehlswort Erkennung aus dem Skript heraus.
-

## 1.12 Projekt Guidebuilder

Umfangreiche Erklärungen über die Bedienung des GuideBuilder-  
Abschnittes,  
sowie der Generierung eines AmigaGuides durch Skriptmanager:

Das Erstellen eines AmigaGuides mittels Skriptmanager erfolgt über die 3 Komponenten:

1. Editor - Anzeigen von Texte+Bildern, Entpacken von lzx+lha Archive, Programmstarts, Markieren, Schriftdesign, Text farblich gestalten, Texte editieren.
2. Guidebuilder - zuständig für Head-Main-Node-Querverweis-Daten Verwaltung.
3. Linkmanager - Seitenaufbau und Datenverwaltung.

Grundlegendes  
Funktion, kurze Beschreibung aller wichtigen Elemente

Bedienung  
Übersicht, Erstellen & Auslesen eines Guides

Fenster Guidemanager  
Genauere Bedienung des Guidebuilders-Datenmaske

Fenster Linkmanager  
Sortieren, Definieren, Setzen von Links

Appwindow  
Datenerfassung durch Drag&Drop

Guide-Editor-Menü  
Menüpunkte des Guide-Editors im Skriptmanager

Guide-Kurs  
AmigaGuide Kurs im AmigaOS-Printmagazin

## 1.13 Projekt ArexxManager

ArexxManager

Noch schnell eins vorweg:

Für Interessierte, die immer schon wissen wollten, wie man mal eben schnell Arexx lernt:

Ich möchte an dieser Stelle auf den aktuellen ersten Arexx-Kurs von Heiko Kuschel im neuen Printmagazin "amigaOS" verweisen (Heft 12/98). Dieser Kurs führt jeden interessierten User nach und nach in diese Programmiersprache ein. (...na ja, Ausnahmen bestätigen die Regel, ...sorry!)

Heiko Kuschel erklärt das grundlegende Basiswissen dieser Sprache leicht, verständlich und effizient. Es sind keinerlei Vorkenntnisse im Bereich Programmierung von Nöten.

Wer nicht weiß, wie er ans Magazin kommt, der kann sich bei mir auch

trotz einer Desinteresse "VON WEGEN REGISTRIERUNG" melden. Wenn der Postweg gewählt wurde, legt bitte einen frankierten und adressierten Umschlag bei.

Eine Angelegenheit in eigener Sache:

Ein großer Anteil der Inhalte des Arexxspellers wurde mit freundlicher Unterstützung von Heiko Kuschel erstellt, da viele Erklärungstexte vom Ihm stammen.

Das Copyright dieser Texte liegt allerdings beim Falke-Verlag und wurde mir dank Thomas Raukamp, Chefredakteur der amigaOS, zur Verfügung gestellt.

Somit verkürzte sich die Entwicklungszeit doch immens und die Hauptarbeit lag beim Programmieren und Anpassen.

Danke allen Beteiligten!

RexxStart  
Menüpunkte und deren Erklärung!

RexxSpell  
Die AREXX - Datenbank!!!

## 1.14 Projekt DosManager

Erstellte Dos Skripte können direkt aus dem Editor heraus gestartet und ausprobiert werden.

Dabei ist es nicht nötig Modifikationen vor dem Ausprobieren zu sichern. Es wird automatisch ermittelt, ob sich der Inhalt geändert hat.

Bedienung:

Menüpunkt Skripte/Dos Starten oder F5 betätigen. (Natürlich sollte vorher ein entsprechendes Skript erstellt worden sein.)

## 1.15 Projekt Editor

Das Öffnen oder Einladen von Dateien in den Editor erfolgt auf  zwei

Arten: Entweder man zieht das Piktogramm ins Fenster, oder es wird die Möglichkeit des Menüpunktes (oder Knopf) laden gewählt.

Änderungen an einer Datei kann man mit dem Menüpunkt (oder Knopf) Speichern dauerhaft sichern.

Speichern Als ermöglicht die Sicherung einer Datei unter einem anderen Namen.

Das Einfügen von Dateien ins aktuelle Dokument ermöglicht Menüpunkt (oder Knopf) DateiE.

Mit Hilfe des Menüpunkt (oder Knopf) Neu wird die aktuelle Datei geschlossen und der Text im Dokument gelöscht.

Der komplette Text wird linksbündig formatiert, wenn der Menüpunkt Text Format angewählt wurde. Die Spaltenlänge wird mit Hilfe des Schiebereglers, der sich neben dem Knopf Block befindet, gesetzt.

---

Der Knopf Block formatiert einen Absatz (bis zur nächsten Leerzeile quasi)

Die Knöpfe Ausschneiden(kopieren/löschen) und Einfügen setzen einen vorher markierten Textbereich wieder ein.

Funktionsweise:

Zeilen oder Worte werden kopiert bei Doppelklick, Maustaste wieder los lassen, und ziehen mit der gedrückten linken Maustaste über einen Textbereich.

Klickt man nach diesem Markierungsverfahren auf den Knopf Ausschneiden ist der markierte Text zwar nicht mehr zu sehen, dennoch ist er nicht gelöscht, sondern im Puffer, und kann wieder mittels Knopf Einfügen erneut gesetzt werden. Außerdem wird jedesmal bei Markierung eines Textbereiches, dieser automatisch ins Clipboard kopiert.

Der Menüpunkt Clipboard-Clip Speichern, kopiert den gesamten Text, der sich im Editor befindet in das Clipboard. Dieser Text kann danach in anderen Programmen, die ebenfalls über ein Clipboard verfügen, problemlos weiter verarbeitet werden.

Der Menüpunkt Clipboard-Clip laden, arbeitet genau anders herum. Diese Funktion lädt Text aus einem Clipboard.

Die Buttons "KZ-LZ-EZ" stehen für Löschen Kopieren Einfügen von ganzen Zeilen.

Mit der Scrolling Funktion kann der Text nach oben oder unten gerollt werden, indem man den Mauszeiger auf die Obere oder Untere Leiste des Dokument Fensters führt. (Menüpunkt Scroll ein-aus)

---Menüpunkt USER:----

Da einige Menüpunkte in "User/" konfiguriert werden müssen, damit sie korrekt arbeiten, solltet Ihr auch unter  
Konfigurieren  
mal sehen!

GB:xy.guide : Öffnet die zuvor preferierte Datei. Dient gleichermaßen als GuideBuilder Startdatei, wenn die Endung des Dateinamens auf .guide lautet.

Email : Wählt einen zufälligen Spruch aus einer TextDatei (Cookie.txt). Diese Textdatei kann beliebig weiter ausgebaut werden. Wichtig hierbei ist lediglich die Einhaltung des bestehenden Formates, welches vorsieht, mindestens eine Leerzeile als Trennung zwischen jedem Spruch zu lassen.

Icon on : Erzeugt beim Speichern ein Icon , falls der Menüpunkt aktiviert wird.

Preference : Startet das Preferenceprogramm.

---Menüpunkt Skripte/Skripte---

Hier werden die zuvor im Konfigurationsprog. eingetragenen Dateien angezeigt, und können sofort zwecks Bearbeitung eingeladen werden.

Alles weitere:

Guide-Editor-Menü

Projekt ArexxManager

---

Projekt DosManager

## 1.16 RexxStart

Menüpunkte des Managers im Editor (Skripte/Arexx Skript)

Simple Go:

Startet ein im Editor befindliches ArexxSkript.

ErrorLine:

Cursor springt in die Zeile, die einen gravierenden Fehler aufweist. Sollte das Skript ausführbar sein, ist der Fehler mittels Trace on zu lokalisieren.

Trace On:

Sollte für eine Fehlersuche aktiviert werden, bei der jede Anweisung im Skript in einem Consolenfenster angezeigt wird. Im Consolenfenster werden die angezeigten Ereignisse mit der Eingabetaste [Return] bestätigt.

Trace Off:

Deaktiviert Trace on.

Trace Con On:

Öffnet ein eigenes Trace-Fenster

Trace Con Off:

schließt das Fenster wieder

Show Ports:

Zeigt die momentan ansprechbaren Ports mit Namen an.

Arexx Spell:

RexxSpell

- Die AREXX-DATENBANK im SKRIPTMANAGER

## 1.17 RexxSpell

Rexxspell ist eine bisher noch nicht dagewesene Datenbank, die nahezu alle Funktions und Befehlswoorte der betriebsinternen Arexx-Programmiersprache des Amigas erklärt, und dazu noch im Kontext. Das heißt, daß Befehlswoorte eines Arexx-Skriptes im Editor des Skriptmanagers, erkannt und ausgewertet werden.

(Es wird das an der Cursorstelle befindliche Wort bei Menüaufruf Skripte/ArexxSkripte/Rexxspell im Datenbestand gesucht und angezeigt.)

Der Benutzer erhält eine ausführliche Erklärung und kann außerdem mit Hilfe der Datenbankfunktionsweise des Rexxspellers übersichtlich verschiedene kategorisch gegliederte Befehle/Funktionen nachschlagen. Wie das im einzelnen aussieht werde ich im Verlauf erklären.

---

Unterteilt wird die Funktionweise des REXXspeller

A. - in den Aktionsknöpfen:

1. Strings
2. Ein/Ausgabe
3. Math-Bit-Kon
4. Andere
5. Oper.

B. - in der erweiterten Bedienung

1. Back
2. Set
3. quit
4. SUCHFELD UNTER DEM BEFEHLSWORTLISTING
5. im grossen Listingfenster durch Interaktion auf eventuell vorhandene Befehlswoorte im vorliegenden Text
6. Durch Interaktion im kleinen Befehlswoortlisting

C. - in der Anzeige

1. Erklärungstext im grossen Listingfenster
2. Befehlswoortlisting

Zu Punkt A:

Hiermit werden Befehle, kategorisch gegliedert, ins Listingfenster geholt, um somit eine bessere Übersicht zu wahren.

Zu Punkt B:

4. Suchfeld unter dem BEFEHLSWORTLISTING  
-hiermit läßt sich die gesamte Datenbank nach einem Ausdruck durchforsten! Einfach die Eingabe machen und Returntaste drücken!
5. im grossen Listingfenster durch Interaktion auf eventuell vorhandene Befehlswoorte  
-Dort wo die Erklärungstexte stehen, kann man durch Klicken auf die eventuell vorhandenen Befehls/Funktionswoorte im Text, deren Bedeutung eben schnell nachlesen--- (Punkt 1 oder 4 bringt euch zum zuletzt gelesenen Text zurück!)
6. Durch Interaktion im kleineren Befehlswoortlisting  
-Per Klick auf einen Eintrag im Listing erhält man die jeweilige zum Wort gehörige Erklärung.
1. Back  
- ermöglicht die Rückkehr zum vorherigen Befehls/Funktionswoort.
2. Set  
- schreibt den Befehls/Funktionsstring an die Cursorstelle ins Skript
3. quit  
- Wat wohl? ...Beendet den REXXspeller ;-)

So! Das war's eigentlich schon! Die Bedienung der Datenbank ist wohl für jedermann verständlich genug gehalten!

## 1.18 Grundlegendes

Ausführliche Beschreibung über das Erstellen eines Guides ist [↔](#)  
unter

Fenster Guidemanager  
nachzulesen!

Unter

Bedienung

befindet sich eine Einführung in Form eines Beispiels,  
das alle Komponenten zum Generieren einschließt.

Ich werde in dieser Node eine genaue Funktionserklärung abgeben, um die  
speziellen Eigenschaften dieser Skriptmanager-Funktion überschaubar zu  
machen; danach gehe ich kurz auf das Erstellen eines AmigaGuides ein,  
ohne alle Eigenschaften anzusprechen.

So, .... ähmm!

Der Guidebuilderteil BAUT/KRIERT/ERSCHAFFT/ERSTELLT/GENERIERT ein  
vollständiges, lauffähiges Amigaguide bzw. einen Quelltext.

Ich wähle mit meinem Programm einen Weg, der eine Datenerfassung  
vorsieht (und pro Datensatz eine Textdatei), um ein konkretes Guide  
zu bauen:

Kurzanleitung (...zum Mitmachen, da es so am einfachsten ist):

Im Editor auf Button GB gehen, oder im Menü MakeGuide/Guidekonzept/  
Guidebuilder anwählen. Den Dateirequester einfach mit OK bestätigen.  
Nun sollte die Datenmaske des Guidebuilder erscheinen.

1. Gadgetnamen unter Bezeichnung eintragen,
2. Kommentar, der neben dem Gadget erscheinen soll, unter Kommentar eintragen.
3. Textdatei (die Node, die einen Text beinhaltet und im Guide angezeigt wird, wenn auf den zuvor unter Bezeichnung eingetragenen Gadgetnamen geklickt wird) unter Insert text eintragen [auch mit Hilfe von Button File möglich]
4. Button Linkman. anklicken und im neuen Fenster, im Feld Head: 1 eingeben.

Wenn die Punkte 1-4 ausgeführt wurden, betätigt Button Erzeuge.  
Sofort dannach erscheint ein fertiges Guide im Editor, daß mit dem  
Button Test angezeigt werden kann.  
Also, schneller gehts wohl nicht mehr!

Für die weitere Bearbeitung der Guidekomponenten (Textdateien, Daten)  
dienen:

- die Funktionen des EditorMenüs (MakeGuide/),
- der Button Edit (Fenster:Guidebuilder), der die Textdatei (unter Insert text eingetragen) zur Bearbeitung in den Editor lädt
- und der Linkmanager (Fenster: Linkmanager), welcher sowohl die Unterteilungen (damit sind die Seiten gemeint, die weitere Querverweise beinhalten), als auch das Setzen von definierten Gadgets (Verweis zur Node) managet.

Fenster Linkmanager

(Fenster: Guidebuilder im Menü [Projekt/Links] oder Button Links ↔  
)

Das Auslesen eines anderen Guides, welches nicht mit Skriptmanager  
erstellt wurde, funktioniert ab Version 1.2!

(Menüpunkt im Editor: MakeGuide/Auslesen)

Guide Auslesen

---

NEU:

Das Erstellen eines AmigaGuides mit Hilfe einer völlig gewöhnlichen Textdatei (Ascii), funktioniert ab der Version 1.3 (Menüpunkt im Editor: MakeGuide/Auslesen)

Texte Auslesen

SUPER NEU:

Verknüpfen Mit

und

Appwindow

Beide Funktionen steigern den Komfort im Umgang mit dem Erstellen ←  
oder

Bearbeiten! (Aber lest erst einmal diesen Link bis zum Ende!)

Hier nochmal eine Ultrakurzbeschreibung aller wichtigen Elemente

Die Editierfelder ergeben (beschrieben von oben nach unten)

- die Gadgetaufschrift
- Kommentar zum Gadget
- LinkID
- der einzubindene Text

Die Button:

- Links - Öffnet den Linkmanager und weist ihn an, einen der definierten Nodes/Links im Listing (per Klick), an die im Editor befindliche Cursorstelle zu setzen.
  - Insert - Hat die Funktion ADD oder Einfügen eines Datensatzes in Datensatznummer xy , wobei alle anderen Datensätze eins nach unten befördert werden.
  - Del - löschen eines einzelnen Datensatzes
  - File - öffnet ein Auswahlrequester, der nach (der unter Edit beschriebenen) Textdatei fragt
  - Head - Die ersten Zeilen des AmigaGuideQuellsources in den Editor laden. Erneutes Aktivieren des Button Head speichert die Änderungen
- ```

/* @database "Default" */
/* @autor "Heiko Strohmeier" */
/* @$VER: Version 1.0 vom 19.12.98 */
/* @INDEX "IndexL" */
/* @font courier.font 13 */
/* @Endnode */

```

Die unten stehenden Buttons:

- Edit - Bearbeiten der zuvor (im Feld: InsertText) definierten TextDatei
- Test - Klar, alles ausprobieren
- Linkman.- Links in Nodes schreiben und Linkanzahl der Hauptseite bestimmen
- Erzeuge - AmigaGuide generieren.  
Der Quellcode nebst aller Komponenten wird nun in den Editor geschrieben.  
Mittels "Test" kann ein Vorabergebnis angezeigt werden!

## 1.19 Bedienung

Direkten Anwendern, die nur schnellst möglich ein Guide bauen ←  
wollen,

rate ich den Abschnitt

Grundlegendes  
zu lesen.

Wem es dann immer noch nicht langt, oder wer mehr wissen will, sollte allerdings hier weiter lesen.

Eine detaillierte Anleitung befindet sich unter

Fenster Guidemanager

Das Bearbeiten fremder Guides (Menüpunkt im Editor: MakeGuide/ ↔  
Guide

Auslesen), die nicht mit Skriptmanager erstellt wurden, ist unter

Guide Auslesen  
erläutert.

Auslesen reiner Textdateien, durch Markieren von Überschriften, ist unter

Texte Auslesen  
beschrieben.

Wenn ein Guide ausschließlich mit Hilfe von Skriptmanager erstellt wird, ist zur Bedienung des GuideBuilders eine gewisse Reihenfolge der Abläufe nicht zwingend, allerdings ratsam, da es vor der möglicherweise aufkommenden Verwirrung schützen kann.

Skriptmanager generiert alle Komponenten, die ein Guide beinhalten kann.

Im Grobkonstrukt sind das folgende Inhalte:

- Head       - Die Kopfzeilen, die die Amigaguide-Identifizierung und andere Anweisungen enthalten.
- Main       - Die Hauptseite, die unmittelbar nach Öffnen des Guide erscheinen soll.
- Node       - Textdatei, mit weiteren Links (Querverweisen) und/oder einfacher Text.
- Links       - ein Querverweis, der eine LinkID, Gadgetnamen, Kommentar enthält.

Jeweilige Bestandteile können mittels Guidebuilder durch die Programm-funktionsweise (des Datenhandlings) beliebig mit einander verbunden, verändert, umgestellt werden, ohne auch nur die kleinsten Vorkenntnisse über Amigaguide-Skripte haben zu müssen.

Jegliche Editierarbeit in einem Editor ist für diesen Part somit Vergangenheit.

Ein Beispiel, das die Übersicht erleichtert:

1. Maskenlinkdefinition:

Fenster Guidemanager

die Erstellung von HauptLinks (Links, die zuerst nach Öffnen ↔  
des

Guides sichtbar sind -Main). Diese sollten zuerst erstellt werden.

2. Weitere Maskendefinition (Unterlinks):

Alle weiteren Unterlinks sollten danach folgen.

(damit sind die Seiten (Nodes) gemeint, die weitere Querverweise beinhalten, jedoch nur von bestimmten Nodes aus geöffnet werden sollen)

3. Prüfen:

Die endgültige Maske (Definition aller Links) erst fertig stellen.

(Ist nur für den Anfang; wenn man das Programm besser kennenlernt ist dieser Punkt aus dem Gedächtnis zu streichen!)

## 4. Linkgliederung und Main Definition:

Fenster Linkmanager

Dann, die Definition (Bekanntmachung) möglicher Unterlinks im Fenster

Linkmanager; außerdem Bekanntmachung der Main-Seite (das ist die Hauptseite, die jene Links enthält, die direkt nach dem Öffnen des Guides angezeigt werden sollen).

Beispielsweise:

Maindefinition im Feld Head: (Linkmanager) = 2

(UL steht für Unterlinkdefinition)

|               |                  |                |
|---------------|------------------|----------------|
| Oberpunkt- 1. | -Feature         | erhält UL:5-7  |
| Oberpunkt- 2. | Vorwort          | erhält UL :3-3 |
|               |                  |                |
| Unterlink- 3. | Autor            | erhält UL:4-4  |
|               |                  |                |
| Unterlink- 4. | Adresse          |                |
|               |                  |                |
| Unterlink- 5. | - Funktionen     |                |
| Unterlink- 6. | - Bedienung      |                |
| Unterlink- 7. | - Unterprogramme |                |

Erklärung:

4.1

Die Hauptseite würde die Links Feature + Vorwort enthalten.

4.2

Dem 1. Oberpunkt Feature werden die Unterlinks 5-7 zugeordnet, was bedeutet: Die Unterlinks mit der Bezeichnung Funktionen Bedienung Unterprogramme liegen hinter dem Oberpunkt Feature, und werden angezeigt, wenn das Gadget Feature angeklickt wird.

4.3

Dem 2. Oberpunkt Vorwort wird der Unterlink Autor zugeordnet, was bedeutet: Der Unterlink mit der Bezeichnung Autor obliegt dem Oberpunkt Vorwort, und wird angezeigt, wenn das Gadget Vorwort angeklickt wird.

4.4

Dem Unterlink 3 mit der Bezeichnung Autor wird wiederum der Unterlink 4 mit der Bezeichnung Adresse zugeordnet. Bei Klick auf das Gadget Autor wird auch hier wieder der jeweilige Inhalt von (in diesem Fall) Autor angezeigt, welcher in unserem Beispiel lediglich ein Gadget mit der Bezeichnung Adresse ist.

(Hierbei ist zu beachten, daß die Bekanntmachung einer nummerischen Folge obliegt. Also, 4-7 oder 8-11 beispielsweise.)

5. Mittels Button Erzeugen das Guide generieren. Button Test zeigt ein Ergebnis.

6. Und ganz zum Schluß könnten Editierarbeiten am kompletten AmigaGuide Quelltext vorgenommen werden, was eigentlich nicht gemacht werden sollte, da alle Änderungen am Quelltext eine Datenerfassung verhindern und alles durch den Guidebuilder realisierbar ist.

Eine Unterlinkdefinition ist ab Version 1.3 nicht mehr dringend erforderlich, im Gegensatz zur Maindefinition, was allgemein wohl keiner weiteren Erklärung bedarf.

## 1.20 Fenster Guidemanager

Das Erstellen eines Guides funktioniert wie folgt:  
Button GB oder Menüpunkt MakeGuide/Guidekonzept/GuideBuilder anfahren, und im Auswahlrequester einen Projektnamen angeben. (Entfällt, wenn vorher in der Preference ein Projekt [Pfad und Dateiname .guide] angegeben wurde.)

Konfigurieren

Es öffnet sich eine Eingabemaske, die aus folgenden Editierfeldern ← besteht:

### 1. Bezeichnung

Der Name, der auf dem Gadget im Guide stehen soll, welcher als Querverweis dient.

### 2. Linkroute

Der interne Querverweisname (LinkID), der als Suchwort im Quelltext dient. (Wird automatisch erzeugt, wenn unter Bezeichnung etwas eingegeben wurde, und die Eingabetaste [Return] betätigt wurde)

### 3. Linkkommentar

Eine kurze Zeile, die ungefähr beschreibt, was sich hinter dem Verweis verbirgt.

### 4. Insert Text

Klick man auf das Gadget File, öffnet sich ein Requester, der einen Pfad plus Dateinamen fordert, um einen zum Verweis entsprechenden Text (Datei-> die Node) zu lokalisieren. War die Auswahl erfolgreich, wird der Pfad/Datei in das Feld über Insert Text geschrieben.

Wurden die Felder komplett ausgefüllt, wendet man sich mit Insert, dem nächsten MaskenLink-Eintrag zu.

Resultat der Eingaben (Vorabtest):

Mit Hilfe von Erzeuge anschließend Test läßt sich ein vorab Ergebnis anzeigen.

Der Guide-Quelltext befindet sich im Editor und kann weiterbearbeitet werden, wovon allerdings abzuraten ist, da der zu erstellende Quelltext von den Bestandteilen/Daten des Guidebuilder-Projektes abhängig ist. (das Ändern der Projektdaten ist die bessere Wahl; diese können ja bequem über die verschiedenen Funktionen des Guidebuilders ebenso umgeschrieben werden.)

Funktion der Buttons:

Del löscht einen kompletten Querverweis Datensatz.

Die Option Edit lädt die aktuelle zum Datensatz gehörende TextDatei in den Editor, die anschließend bearbeitet werden kann.

Bei Klick auf Button File öffnet sich ein Auswahlfenster, das einen Dateinamen+Pfad fordert. Anschließend wird dieser dem Feld InsertText übergeben.

Head ermöglicht das Editieren der Hauptseite und der Kopfzeilen (Kopfzeilen=die ersten Zeilen im Guide, Hauptseite=Main bzw. die ersten Links, die die Hauptseite bilden).  
Der Button färbt sich inversiv, um den Arbeitsschritt anzuzeigen.  
Betätigt man Head erneut, werden die Änderungen aktiv.

Mit Links lassen sich im Übersichtslistings des Fenster Linkmanager Querverweise sehr komfortabel im AmigaGuide Quelltext (oder einer Textdatei, die sich möglicherweise gerade im Editor befindet), durch einfaches Klicken auf einen im Listing befindlichen Linknamen, setzen.

Die PfeilButtons unter dem Button Links rufen den nächsten oder vorherigen Datensatz auf.  
Das Feld zwischen den Pfeilbuttons zeigt die Datensatznummer und dient außerdem zum Aufruf eines Datensatzes durch einfache Eingabe einer Nummer gefolgt von einem Return.

Menüpunkte der Maskenoberfläche:

Projekt laden: laden einer Datenmaskendatei.

Projekt speichern Als: Sichert eine Datenmaske unter einem beliebigen Dateinamen.

Projekt speichern: Sichert eine Datenmaske unter einem angegebenen Dateinamen.

Links Setzen: - hat die gleiche Funktion wie Button Links

Head Erneuern:-  
Head Erneuern

## 1.21 Fenster Linkmanager

Der Linkmanager beinhaltet zwei Funktionen, die bei Betätigen des Button Linkman. oder Button Link aufgerufen werden können.

Funktion 1:

Bei Klick auf Button Links (Menü: Projekt/Links) dient der Linkmanager zum Setzen definierter Links (Querverweise - Verweis zur Node) in eine Textdatei, die sich im Editor befindet, oder erst noch geschrieben werden soll. Ein Klick auf einen Eintrag im Listingfenster fügt den notwendigen Formatstring an die Cursorstelle im Editor.

Funktion 2:

Bei Klick auf Button Linkman. wird der Linkmanager zum Unterteiler.

Unterteilung des Guides: 1. Main 2. Unterlinks

1. Main

---

Die wohl notwendigste Sache beim AmigaGuide ist die Main-Seite. Das ist die Hauptseite, also jene, die direkt nach dem Öffnen des Guides erscheinen soll. Hierfür existiert im Linkmanager eine Einrichtung, die dafür sorgt, mitzuteilen, welche Links (von oben nach unten) auf die Hauptseite sollen. Ein Eintrag im Feld Head: gibt der Erzeugefunktion Auskunft darüber, wieviele Links auf die Hauptseite sollen.

Es werden die ersten Links (von oben nach unten) nur einmal pro Definition (bzw. jedesmal bei Änderung) ans Ende des Head's geschrieben. Der Head (Hauptseite oder Main) kann mit Button Head (im Guidebuilder-Datenmaske) umeditiert oder eingesehen werden.

(Die Main-Links, die in der Datenmaske des Guidebuilders bestimmt werden können im Falle einer Umeditierung mit dem Menüpunkt Head Erneuern aktualisiert werden, sodaß ein Einschreiben in die Main-Node erfolgt, trotzdem der Head: Wert nicht geändert wurde.

```
    Head Erneuern
  )
```

Sicherheitshalber wird bei jedem Neuschreiben über die Funktion Head: eine Kopie der alten Hauptseite angelegt. Im Projektverzeichnis unter dem Dateinamen HEAD.OLD zu finden.

## 2. Unterlinks:

Klickt man auf einen Eintrag im Listing des Linkmanagers, benötigen die beiden Felder über Head: eine Beginn- und End Nummer für diesen Link. Die Beginn und End Nummer bestimmt die Unterteilung; ist quasi die Unterlinkdefinition.

Diese Unterlinks werden beim Generieren des Guide später ans Ende jeder zum Link gehörigen Textdatei (der Node) gesetzt und dienen somit als Einteilung der Seiten des Guides.

Jedem einzelnen Link können Unterlinks zugeteilt werden.

Sortieren von Einträgen im Linkmanager:

Hierzu muß ich gleich zu Anfang einiges erläutern.

Die Reihenfolge im Listing des Linkmanagers ist je nach dem wie die Gliederung letzt endlich aussehen soll, sehr entscheidend.

Beispiel:

(UL hat die Bedeutung "Unterlinkdefinition")

Oberpunkte:

```
Nr: 1 = Einleitung      * erhält die UL: 4-4
Nr: 2 = Installation    * erhält die UL: 5-6
Nr: 3 = Konfigurieren  * erhält die UL: 7-7
```

Unterlinks:

```
Nr: 4 = Adresse        * erhält die UL: -
Nr: 5 = Testsystem     * erhält die UL: -
Nr: 6 = Voraussetzung * erhält die UL: -
Nr: 7 = Fenster        * erhält die UL: -
```

wie man nun unschwer erkennt, hätte eine Änderung der Reihenfolge im Unterlink-Segment, eine Änderung der Unterlinkdefinition der Oberlinkpunkte zur Folge.

Refresh sorgt für eine zusätzliche Übersicht bei nachträglicher Änderung der Reihenfolge.

Del UL löscht eine Unterlinkdefinition. (...Anfang-Ende)  
 L Up rückt einen Linkeintrag nach oben.  
 L Down rückt einen Linkeintrag nach unten.

Unterlinkdefinition im Linkmanager:

Im Fenster links stehen alle bereits erstellten Linkbezeichnungen.  
 Klickt man auf einen dieser Linknamen, der noch nicht definiert wurde,  
 öffnet sich ein Hinweisrequester, der Auskunft über die Handhabung einer  
 Definition gibt.

Mit Bestätigung des Requesters werden die gesperrten Begin/End Felder  
 entsperrt, und erwarten einen Eingabewert, der den Anfang eines Links  
 also auch das Ende kenntzeichnet.

(1. Feld bestimmt den Anfang - 2. Feld das Ende).

Beim Erzeugen (unter

Fenster Guidemanager

beschrieben) des Guides werden

diese Beginn und End Unterlinks, ans Ende jeder zum Link gehörigen Node  
 im Guide eingeschrieben.

Die zum Link gehörige Textdatei bleibt unbeschrieben!

## 1.22 Appwindow

Die AppWindowFunktion im Guidebuilder ist eine wirklich klasse Sache. ←

Hiermit läßt sich allein nur durch das hinein ziehen einer Schublade  
 (oder einer File der Schublade) ins Guidebuilder Fenster, die komplette  
 Datenerfassung für ein Guide realisieren. Die verschiedenen Datensatz-  
 bestandteile werden anhand der Dateinamen im Verzeichnis ermittelt.

Heißt eine Datei (Textdatei des Verzeichnisses) beispielsweise Heiko,  
 wird Heiko als Linkname eingesetzt - Mediumname+Heiko wird zur LinkID  
 - und der InsertText Eintrag lautet Laufwerk:Projektschublade/Heiko.  
 Natürlich sollten im betreffenden Verzeichnis wirklich NUR Textdateien  
 vorhanden sein, ansonsten entfernt bitte manuell (vor der Guidegenerier-  
 ung) alle Datensätze, die nicht auf Textdateien verweisen, mit der Del-  
 Funktion im Guidebuilder.

Eine weiterführende Idee ist, daß außerdem CD's oder Festplattenverzeich-  
 nisse kategorisch erfasst werden können und als Guide mit passenden Texten  
 (die man wahrscheinlich meist selbst verfassen muß, es sei denn es ist  
 so eine Art readme Datei für die jeweiligen Verzeichnisse vorhanden :-))  
 dokumentiert sind.

Mit Hilfe der Funktionsweise des Guidebuilders, speziell des  
 Linkmanagers  
 (für die Unterteilungen) und der  
 Verknüpfen Mit  
 Funktion (verschiedene

Datenfiles verschmelzen) könnte man so einzelne Rubriken ganzer CD-Rom-  
 Serien auf einfachste Weise überschaubar machen.

Um beide Features zu gewährleisten wird der Benutzer über einen Abfrage-  
requester gebeten zu spezifizieren, ob Dateien (Texte, die zu einer log-  
ischen Guidegenerierung verwendet werden) oder Verzeichnisse (im Falle  
einer Dokumentierung, wobei für jedes Verzeichnis gleich eine entsprechende  
Textfile geschrieben wird, die allerdings nur einen Verzeichniszeiger be-  
inhaltet) erfasst werden sollen.

Beispiel einer Dokumentierung:

Das Verzeichnis AmigaPlusCD-4.99:Games/ wurde ins Fenster gezogen.  
Dieses Verz. hat die Unterverzeichnisse SpielA,SpielB,SpielC, dann  
würde jeweils eine Textdatei mit Namen

- SpielA mit dem Inhaltstring->AmigaPlusCD-4.99:Games/SpielA
- SpielB mit dem Inhaltstring->AmigaPlusCD-4.99:Games/SpielB
- SpielC mit dem Inhaltstring->AmigaPlusCD-4.99:Games/SpielC

entstehen.

Diese Textdateien dienen dann als Bestandteil für die Komplementierung  
des Guides, welches dadurch eine Dokumentierung letztendlich erhält.

## 1.23 Guide-Editor-Menü

---GuideBuilder Menüpunkte im Editor (MakeGuide/)---

Guidekonzept

MakeGuide-Hauptfunktionen

Guide Auslesen

Fremde AmigaGuides in Skriptmanager erfassen

Texte Auslesen

Ascii Texte werden per Markieren zum Guide gemacht

Verknüpfen Mit

Datenbanken des Guidebuilders verbinden

Layout

Bestimmung des Schriftlayouts

Colour

Farbliche Darstellung von Texten

Markierung

Rechteckige Rahmen für Markierungen

## 1.24 Guide-Kurs

Für Interessierte, die immer schon wissen wollten, wie man  
AmigaGuide Skripts auch ohne Skriptmanager erstellt:

---

Ich möchte an dieser Stelle auf den kommenden AmigaGuide-Kurs von Heiko Kuschel im neuen Printmagazin "amigaOS" verweisen (schätze Heft 7/99). Wie man aus dem ersten Arexx-Kurs von Heiko Kuschel schließen kann, wird auch dieser neue Kurs mit Sicherheit jeden interessierten User nach und nach in diese Hypertext Sprache einführen.

Wer nicht weiß, wie er ans Magazin kommt, der kann sich bei mir auch trotz einer Desinteresse "VON WEGEN REGISTRIERUNG" melden. Wenn der Postweg gewählt wurde, legt bitte einen frankierten und adressierten Umschlag bei.

## 1.25 Guidekonzept

---GuideBuilder Menüpunkte im Editor (MakeGuide/Guidekonzept)---

Neues Projekt:

Entfernt ein bestehendes GuideProjekt samt der Guidedatendatei.

GuiderBuilder:

siehe unter

Bedienung

Link Kombi:

Fügt ein komplettes Linkpaar ein, bestehend aus Link Gadget und Link Route.

Link Gadget:

Fügt einen Querverweis (nur Gadget) ohne die erforderliche Querverweisroute in den Editor ein. Hinweis: Ohne Querverweisroute besitzt der Querverweis lediglich einen optischen Wert.

Link Route:

Fügt eine Querverweisroute (ohne Gadget) in den Editor ein.

Hinweis: Ohne Querverweis besitzt die Querverweisroute keinen nützlichen Wert.

Lokal:

"Lokal" überläßt dem User, ob das zu erstellende Guide für die Allgemeinheit sein soll, oder lediglich für den Gebrauch auf dem eigenen Rechner.

Die Pfade einzubindener Elemente sind entscheidend dafür, ob sie ansprechbar sind. Ist die Lokal-Option gesetzt, wird die individuelle Zusammensetzung nur auf einem Computer richtig angezeigt.

Ist hingegen die Lokal-Option inaktiv (Voreinstellung), kann man sein Guide der Allgemeinheit zugänglich machen, unter der Voraussetzung, daß alle Elemente (Externer Text, Image, Archive, zu startene Prog.) im gleichen Verzeichnis liegen, und zusätzlich der Start des Guides aus diesem Verzeichnis heraus erfolgt. (entweder durch Doppelklick auf das Icon, oder per Shell bzw. Execute-Skript, wobei eine Zuweisung über den Dos Befehl \*cd\* gesetzt werden muß)

Externe Elemente:

Die Vorgehensweise der weiteren Optionen sieht vor, lediglich eine Auswahl der im Dateirequester gelisteten Dateien, zu treffen.

Text Extern:

Diese Option ermöglicht fremde Textdateien über das aktuelle ProjektGuide

zu steuern, und zwar ohne daß der Text im Quelltext eingefügt bzw. in der LinkMaske definiert worden sein muß. Auf diese Weise kann man beispielsweise zwei oder mehrere AmigaGuide Dateien verknüpfen.

Die Abfrage bei Benutzung dieser Option, ob der externe Text bearbeitet werden soll, ist nur von belang, sofern dieser noch keine AmigaGuide spezifischen Formatstrings besitzt.

Also im Falle eines externen AmigaGuides, das vom aktiven Projekt aus gesteuert werden soll, ist die Option "Weiter" im Auswahlrequester zu wählen.

Im Falle einer automatischen Modifizierung durch den GuideBuilder ist zu beachten, daß die angesteuerte Datei ihren Verzeichnispfad in Zukunft behält.

(Nicht vergessen; Die Option Lokal bestimmt, in wie weit das Projekt genutzt werden kann/soll)

Image Extern:

Bindet ein Bild (in Form eines Gadgets) in den Quelltext ein, um es aus dem AmigaGuide heraus betrachten zu können.

Selbstverständlich sollte ein geeignetes Datatype, dem Bildformat entsprechend, vorhanden sein.

Prog. Run:

Startet ein Programm (in Form eines Gadgets) aus dem Quelltext heraus.

Lha 2 Ram:

Entpackt ein lha-Archiv (in Form eines Gadgets) aus dem Quelltext heraus nach Ram: .

Lzx 2 Ram:

Entpackt ein lzx-Archiv (in Form eines Gadgets) aus dem Quelltext heraus nach Ram: .

## 1.26 Guide Auslesen

Bitte vorher den Abschnitt  
Bedienung  
lesen.

Voraussetzung, damit die Funktion nutzbar ist!

Die eingeladene AmigaGuideFile MUß die Endung .guide haben, wobei die Groß/Kleinschreibung uninteressant ist UND DARF keinen weiteren . (auf Deutsch: "Punkt") im Dateinamen haben!

Das Auslesen und das damit verbundene bequeme Bearbeiten und Editieren eines fremden AmigaGuides ist nun mit Veröffentlichung der Version 1.2 möglich.

So funktioniert's:

Das Piktogramm eines AmigaGuides über die Workbench ins Fenster des Editors ziehen und loslassen. (Mittles Laden ebenfalls möglich!) Anschließend den Menüpunkt des Editors - MakeGuide/Auslesen anfahren und alles weitere wie in diesem Guide unter den verschiedenen Links beschrieben!

Projekt Guidebuilder  
Am Rande bemerkt: nur für die Guides, die mit Skriptmanager  
konzipiert

wurden, sind auch alle Funktionen dieses Programmabschnittes nutzbar.

Diese Einschränkungen gehören mit Veröffentlichung der Version  
1.3 der Vergangenheit an!

Verständlicherweise existieren die normalerweise im Linkmanager  
definierten Nodes/Links, beim Auslesen der fremden AmigaGuidefile in  
den automatisch konzipierten Textdateien, welche die Datei Benennung  
der einzelnen LinkID's haben.

z.b.

Unter Bezeichnung steht: Heiko

Unter Linkroute (LinkID) steht dann: HeikoL

demzufolge heißt die konzipierte TextDatei HeikoL.txt

Das führt dazu, daß der Linkmanager nur noch für nachträglich einzusetzende  
Links zu gebrauchen ist (Menüpunkt im Guidermanager: Projekt/Links setzen)  
und daß das Feld Kommentar gesperrt bleibt.

(wofür auch, da alles im AmigaGuide Quellcode bzw. in den erfaßten  
ausgelesenen TextDateien vorliegt.)

Neu:

Will man nun eine völlig neue Gestaltung des Guides, welche bsw.  
eine total andere Hauptseite vorsieht, eine Unterlinkdefinition  
beinhalten soll, oder Links neu anordnen, ist mit Hilfe des Buttons  
Linkman. nun der freie Zugriff auf die Funktionen des Guidebuilders  
gewährleistet.

Sollten die Links im Head (der Hauptseite oder Main) ausdrücklich nicht  
ans Ende der Headzeilen gesetzt werden, oder soll die Hauptseite original  
erhalten bleiben, muß der im Linkmanager befindliche Headeintrag (Feld  
Head:) 0 sein.

(Mehr dazu ist unter

Head Erneuern  
noch einmal erläutert.)

Da der Head nur jeweils einmal pro Eintragsänderung erneuert wird, könnte  
man die Headzeilen auch nachträglich anpassen.

Die Funktion des Button Head (Guidebuilder-Datenmaske) sieht die Änderung  
der Hauptseite vor.

## 1.27 Texte Auslesen

Bitte vorher den Abschnitt  
Bedienung  
lesen.

Voraussetzung, damit die Funktion nutzbar ist!

Der Text muß im Ascii-Format vorliegen, also muß eine  
reine Textdatei sein.

Das Auslesen und das damit verbundene bequeme Bearbeiten und  
Editieren einer schnöden Textdatei ist nun mit Veröffentlichung

der Version 1.3 möglich.

So funktioniert's:

Das Piktogramm einer Textdatei über die Wordbench ins Fenster des Editors ziehen und loslassen. (Mittles Laden ebenfalls möglich!) Anschließend den Menüpunkt des Editors - MakeGuide/Text Auslesen - anfahren.

Damit die neue Funktion ihre Arbeit korrekt verrichtet, müssen vorher einige Markierungen im Text, der zum Guide werden soll, vorgenommen werden.

Über einer Überschrift, welche beispielsweise einen sinngemäßen Inhalt beschreibt oder einleitet, MUß die folgende Markierung gesetzt werden: `;;;B`

`;;;B` beschreibt den Anfang einer Node, welche später automatisch in die Datenmaske mit samt Link und Dateipfad/Dateinamen eingetragen wird.

Diese Anfangsmarkierung MUß in einer Leerzeile über der Überschrift im Text stehen!

Das Ende dieser Node MUß mit folgender Markierung an das ENDE des Abschnitts im Ascii-Text gesetzt werden, und zwar auf eine Leerzeile:

`;;;E`

Beispiel:

```

;;;B           (Markierungsanfang in einer Leerzeile)- (F7)
C++ für Einsteiger (soll die Überschrift sein)
Die Programmierung in C-C++ setzt eine Menge Arbeitswillen
und Motivation voraus.....bla bla
;;;E           (Markierungsende in einer Leerzeile) (F8)

```

Also würde das Gadget (der Link) im AmigaGuide C++ für Einsteiger heißen, der Inhalt dieses Links (die node) würde alles zwischen

`;;;B`

und

`;;;E`

anzeigen und der dazugehörige Dateipfad, (die Textdatei welche aus dem Gesamttext gezogen wird, und schließlich für das Erstellen der Node von größter Wichtigkeit ist) plus Dateiname würde - Dateipfad der AsciiTextDatei (: oder /) C++ für EinsteigerL.txt-lauten.

Ist im Text alles relevante auf diese Art und Weise markiert worden, werden mittels dem Menüpunkt MakeGuide/Text Auslesen alle benötigten Daten, die ein lauffähiges AmigaGuide erstellen, in die Datenmaske übertragen und als Projektfile abgespeichert.

Das Generieren (Erzeugen) des Guides erfolgt wie unter

Grundlegendes

Bedienung

Fenster Guidemanager  
beschrieben.

Übrigens sollte man zum Markieren die Funktionstasten F7 (`;;;B`)

und F8 (;;;E) benutzen, da es je nach dem wie groß der Text ist, doch schon recht nervig sein kann immer das selbe zu tippern.

## 1.28 Verknüpfen Mit

Diese Funktion im Skriptmanager-Menü bezieht sich auf das `↔`  
Verbinden  
zweier Guidebuilder-Datenbanken. Es können sowohl ausgelesene als auch Skriptmanager konzipierte Datenfiles zu einer verknüpft werden.

Stellt Euch vor, Ihr kreiert ein Guide und nach einer Weile fällt euch auf, das eine node doch besser mehrere Unterteilungen haben sollte. Bisher wäre die Realisierung nur mit Hilfe einer Textdateisplittung vonstatten gegangen, die in gewohnter Weise, der DatenMaske im Guidebuilder bekannt gegeben wird.

Genau hier setzt diese Funktion an. Mit Hilfe der  
Texte Auslesen  
Option  
kann man, wie der fleißige Leser bestimmt schon weiß, auf sehr einfache Weise Datenfiles für komplette Guides erschaffen. Diese Eigenschaft im Skriptmanager gab mir Grund zum Anlaß einer Denkpause. Nach Beendigung der Denkpause ergab sich folgendes:

Die Datenfiles lassen sich mit der Funktion Verknüpfen mit zu einer Datenfile verbinden.

Das Verknüpfen sieht vor, daß alle Dateneinträge des momentanen aktiven Projektes, ans Ende einer, per Auswahlrequester bestimmten Guidebuilder Datenfile (mit Endung guide.DAT), angehängt wird. Die letztere Datenfile, also die Zieldatei, wird automatisch nach Funktionsausführung abgespeichert und sollte anschließend die nötigen Datenbankergänzungen im  
Fenster Linkmanager  
erhalten.

## 1.29 Layout

---GuideBuilder Menüpunkte im Editor (MakeGuide/Layout)---

Normal:

Setzt alle anderen gesetzten Formatbestimmungen, ab der im Text befindlichen Cursorstelle, auf normal.

Nützlich bei mehrfach gesetztem Formatlayout (beispielsweise: Kursiv und Unterstrichen zusammen aktiviert). Nach Aktivierung von Normal ist die Schriftanzeige normal (...was sonst!).

---

**Fett:**

Nach Aktivierung ist die Schriftanzeige fett. Bei nochmaligem Betätigen ist die Schriftanzeige fett deaktiviert. (... bezogen auf die im Text befindliche Cursorstelle)

**Kursiv:**

Nach Aktivierung ist die Schriftanzeige kursiv. Bei nochmaligem Betätigen ist die Schriftanzeige kursiv deaktiviert. (... bezogen auf die im Text befindliche Cursorstelle)

**Unterstrichen:**

Nach Aktivierung ist die Schriftanzeige unterstrichen. Bei nochmaligem Betätigen ist die Schriftanzeige unterstrichen deaktiviert. (... bezogen auf die im Text befindliche Cursorstelle)

## 1.30 Colour

---GuideBuilder Menüpunkte im Editor (MakeGuide/Colour)---

Farbliche Kennzeichnung von Textstellen oder dem kompletten Text läßt sich durch die vordefinierten Farbstiften (schwarz,weiß,rot,blau) im Menü MakeGuide/Colour leicht realisieren. Der Formatstring "{APEN 3}" z.b. ermöglicht die Darstellung von Texten(Textstellen) in rot. Um weitere Farben (mehr als die vier vordefinierten) benutzen zu können, ist lediglich die Zahlziffer im String zu ändern.

## 1.31 Markierung

---GuideBuilder Menüpunkte im Editor (MakeGuide/Marker)---

Farbliche Markierungskennzeichnung von Textstellen oder dem kompletten Text läßt sich durch die vordefinierten Markierungsstifte (neutral,weiß,rot,blau) im Menü MakeGuide/Marker leicht realisieren. Der Formatstring "{BPEN 3}" z.b. ermöglicht die Darstellung von Texten (Textstellen) in rot. Um weitere Farben (mehr als die vier vordefinierten) benutzen zu können, ist lediglich die Zahlziffer im String zu ändern.

## 1.32 Head Erneuern

Wenn die Main-Link Eintragungen (die Links auf der Hauptseite) in der Datenmaske des Guidebuilders geändert werden, ist eine Änderung im später generierten Quelltext nicht vorhanden.

Die neuen Eingaben der Felder im Guidebuilder-Datenmaske für die Main-Links werden deshalb nicht berücksichtigt, weil sie neu in den Quelltext geschrieben würden, was bedeutet, daß die Main-Links komplett an den Anfang des ersten gefundenen Linkstring auf der Hauptseite eingeschrieben werden, was einen möglicherweise hübsch gestalteten Textes auf der Hauptseite, im Format (der Einteilung) verändern würde.

Beim Gestalten der Hauptseite wollte ich mehr Freiheiten haben, und das permanente neue Einschreiben der Links in die Main-Node, jedesmal wenn das Guide erzeugt würde, hätte dieser Gestaltungform gehörig an Reiz genommen, da Links jedesmal neu in die Node eingefügt werden, und dazu noch das Layout der Seite Skriptmanagertypisch umgeschrieben wird. Der Text hinter dem letzten Link der Hauptseite wird einfach verschluckt.

Natürlich wird dieser Vorgang daher erst dann in Bewegung gesetzt, wenn der Eintrag im Feld Head: des Linkmanager geändert , oder Head Erneuern aktiviert wird.

Für Ausgelesene Guides, die meist recht individuelle Hauptseiten besitzen, wäre diese Linkmanager-Funktion also auch die Head Erneuern-Funktion weniger geeignet, es sei denn man möchte das Guide mit Hilfe des Linkmanagers, der strukturierten Verfahrensweise des Skriptmanager-Guidebuilder übergeben. Dieses Verfahren würde eine hohe Kompatibilität des Skriptmanagerstil bei Generierung gewährleisten.

Da allerdings kein großartiger Vorteil im Umgang mit der Bedienung des Programms durch diese 100% Kompatibilität erreicht wird, schlage ich vor: Änderungen der Hauptseite bei ausgelesenen Guides, nachträglich mit Hilfe des Buttons Head und dem dann aktuell zur Verfügung stehenden Datenhandling von Skriptmanager, anzupassen.

Nun, dennoch könnten einige Anwender die Skriptmanager konforme Gestaltungsweise bevorzugen, oder Skriptmanager konzipierte Guides bearbeiten, ohne großen Wert auf die eigne Einteilung der Hauptseite zu legen, und hierfür würde das Programm ohne diese Head Erneuern-Funktion keine direkte Möglichkeit bieten.

Sicherheitshalber wird bei jedem Neuschreiben über die Funktionen Head Erneuern und Head: eine Kopie der alten Hauptseite angelegt. Im Projektverzeichnis unter dem Dateinamen HEAD.OLD zu finden.