

MyMenu

Ralf Stecher

COLLABORATORS

	<i>TITLE :</i> MyMenu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ralf Stecher	August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MyMenu	1
1.1	main	1
1.2	standart	4
1.3	copyright	4
1.4	haftung	5
1.5	author	5
1.6	schlusswort	6
1.7	mehr-vom-author	7
1.8	vorwort	8
1.9	source-code	9
1.10	installation	9
1.11	starten	10
1.12	konfiguration	11
1.13	zukunft	12
1.14	fehler	12
1.15	danke	13
1.16	author1	13

Chapter 1

MyMenu

1.1 main

Deutsch

Wichtig

Vorwort

Source Code

Installation

Starten

Konfigurationsdatei

Zukünftige Pläne und hoffnungslose Träume

Einschränkungen, Fehler und andere verrückte Dinge

Danksagungen

Autor

English

MyMenu is a program to allow you to create your own menus in the WorkBench to run your own commands. This can save the hassle of opening up lots of drawers to get to the command you want. MyMenu will allow you to execute both CLI and WorkBench programs, and is configured with a normal text file. Hopefully, this program will be obsolete when WorkBench 1.4 is released, since I hear this capability may be added.

Building:

There is a supplied Makefile, which works with Manx 3.6, and a version of make from a Fish disk (I don't remember which one). The Makefile should work with most make programs. I haven't tried to get this

to work with 16-bit integers, or to work with Lattice (although, I would like this).

There are some defines in MyMenu.h that can be customized. DO_WB determines if support for WorkBench programs is included. undefining this can save a lot of space if you don't need this option. DO_PATH if defined will compile in code for path searching for commands. Currently, it is undefined.

Installation:

Copy MyMenu-Handler to the L: directory. MyMenu will also look for this file in the current directory if it does not exist in the L: directory.

Create the file S:MyMenu.conf. The format of this file is described later.

Running:

Step one is to start up the Workbench!

The command "MyMenu" will load and start up the handler process. Menus should now appear in the WorkBench menu strip.

Running MyMenu while the handler is already loaded will cause it to re-parse the configuration file and rebuild the menus.

The command "MyMenu quit" will terminate and unload the handler process.

Configuration file:

The configuration file is read in when you run MyMenu. It is looked for in the S: directory, and in the current directory in that order.

The configuration file defines the menus you want, and what commands they will run. You may put carriage returns between keywords, but you cannot put a carriage return in the middle of a command. Comments begin with a #, and continue until the end of the line. Upper and lowercase does not make a difference.

COLOR n

This will set the foreground pen color for new menus. You can change this as often as you want. The argument is number that is the pen number to use. The default is 2 (black).

MENU [<command-char>] menu-name item-name [sub-item-name] | command-def

Defines a new menu. Each menu definition must have a menu-name and an item-name. A sub-item-name is optional. If any of these names contain whitespace, enclose the name in double quotes, or precede the whitespace character with a backslash (\). A command character may be defined for the menu item by putting the character after the MENU keyword and surround it with <>'s.

Separate the menu definition and the command definition with a vertical bar (|). After the bar, you can have the keywords CLI or WB, to specify that the command is a CLI or WorkBench command. Everything after WB until the end of the line is taken to be the command. Everything after CLI until a space or end of the line is taken to be the command. Everything else will be passed as arguments to the CLI command. Do not include redirection in a CLI command.

Examples:

```
menu Games Tetrix | WB dh0:games/tetrix
```

This will define a menu "Games", with a menu item "Tetrix". If this is selected, the command "dh0:games/tetrix" will be run as a WorkBench program.

```
menu Games "Black Box" | WB df0:Black Box
```

This will define a menu item "Black Box" under the menu "Games". The menu Games will already exist from the previous example, and will now contain "Tetrix" and "Black Box" as items. The command to be run is "df0:Black Box" (Box is not an argument).

```
color 3
```

```
menu Utilities DMouse "Start DMouse" | c:DMouse
```

```
menu Utilities DMouse "Quit DMouse" | c:DMouse quit
```

This will define two sub menu items (drawn in yellow). The second command has an argument.

Future plans and hopeless dreams:

- AREXX support.

- Fix up path searching.

- A better parser.

- Support for graphical menu items.

Limitations, bugs, and other wierd things:

- Path searching does not work. Do not compile this in.

- I would appreciate it if someone can find the problem, since I am stuck. Details are in DoRun.c.

- New preferences causes menus to disappear (but handler is not unloaded). The menus can be rebuilt by running MyMenu again.

- Redirection can not be done for CLI programs, since the code already does redirection to and from NIL:.

- Line limit of 256 characters in configuration file.

Acknowledgements:

I have borrowed ideas and code snipits from other public domain programs. I have also received help directly from some people.

I would like to thank those people (whose names I remember):

Matt Dillon - Ideas for menu building and how to load in the

handler were taken from DME and DMouse.
Peter da Silva - Ideas taken from wblaunch program.
Rob Peck - Some ideas and help.
Davide Cervone - LOTS of ideas taken from MonIDCMP.

Signature:

Darin Johnson
darin@laic.UUCP
leadsv!laic!darin@pyramid.pyramid.com
(415) 368-0972

1.2 standart

Bitte beachtet auch die Copyrightbestimmungen
die diesem Verzeichnis unterliegen.

Fals Euch durch Teile aus diesem Verzeichniss
irgendwelche schäden jedweder Art entstanden
sind, solltet ihr mal hier nachschauen. Aber
ihr solltet doch schon jezt lesen was auf Euch
Zukommt wenn etwas Passiert was von mir nicht
beabsichtigt ist. also, am besten sofort Lesen.

Hinweise zum Author dieses Verzeichnisses, an
anderer Stelle auch Paket genannt können in
diesem abschnitt nachgelesen werden.

Was ich sonst noch alles Produziert habe könnt Ihr
auf der folgenden Seite nachlesen. Vielleicht ist
das eine oder andere dabei das Ihr auch gebrauchen
könnt. Reinschauen lohnt sich immer.

Nun bitte ich Euch nur noch das Schlusswort zu
Lesen in dem etwas in eigener sache steht und
in dem ich mich schon mal im vorraus bei Euch
entschuldigen möchte.

Zurück zur Hauptseite.

1.3 copyright

Dieses Verzeichnis ist Copyright © 1995 by Ralf Stecher, Braunschweig,
Deutschland. Alle Rechte vorbehalten. Dieses Verzeichnis darf
weitergegeben werden, wenn dies völlig kostenlos ausschließlich durch
nicht gewerblich ausgerichtete Institutionen und Vollständig geschieht,
oder wenn eine schriftliche Genehmigung des Herausgebers vorliegt.
Eine Weitergabe durch das »Aminet« oder der AMIGA Times (DiskMag.)
und dessen WEB Seiten ist daher ausdrücklich gestattet.
Ansonsten ist eine kostenlose Weitergabe dieses Dokumentes jederzeit
erlaubt und auch erwünscht. Sogenannte »Kopiergebühren« sind unzulässig.

Wenn Du Fehler in diesen Hypertexten findest, so melde dies doch bitte dem Herausgeber, der sich dann bemühen wird, diese Fehler zu beheben.

```

=====
####
####   ###   #       AMIGA online Magazin ( German )
#### #   #   #####
####   #   #       Ralf Stecher, Abt. Helpline
####   #####   #
#### #   #   #       web:   http://www.amigaworld.com/amigatimes/index.htm
#### #   #   #   #   email: amigatimes@amigaworld.com
####   #####   ##
####
####   amiga times   Testberichte, Storys, Software- Hardwareprobleme....
####
#####
#####
=====

```

Zurück zu den Hinweisen.

1.4 haftung

Ralf Stecher kann in keinem Fall für irgendwelche mittel- oder unmittelbaren Schäden haftbar gemacht werden (dazu gehören ohne Einschränkung Betriebsstörung, oder andere finanzielle Verluste), die durch den Gebrauch oder Nichtgebrauch der Files dieses Verzeichnisses entstehen. Dies gilt auch für den Fall, daß der Herausgeber von der Möglichkeit solcher Schäden in Kenntnis gesetzt wurde.

Bitte beachte vor allem, daß die Möglichkeit besteht, daß Teile dieses Paketes fehlerhaft sind.

Jegliche Nutzung der Beilagen dieses Verzeichnisses geschieht deshalb auf eigene Gefahr.

Zurück zu den Hinweisen.

1.5 author

Herausgeber dieses Hypertext-Dokumentes:

Ralf Stecher
 Cheruskerstr. 19
 38112 Braunschweig
 e-Mail Ralf.Stecher@t-online.de

Die Konvertierung des Textes ins Hypertext-Format wurde komplett von dem Herausgeber durchgeführt. Wenn Du mich ermuntern möchtest, auch in Zukunft etwas zu entwickeln bzw. ins Hypertext-Format zu wandeln und dieses zu veröffentlichen, so schicke mir ein Paar Anregungen.

Wenn ich dann das Gefühl habe, daß meine Arbeit allgemein erwünscht ist, werde ich alle brauchbaren, interessanten Ideen aufnehmen, eventuelle Fehler beseitigen, eventuelle Install Scripts Schreiben, Hypertexte entwickeln, mich um Veröffentlichung bemühen usw.

Wenn Du über Ideen, Programme, DOS-Skripte oder anderes interessantes verfügst, das Du gerne Veröffentlichen hättest, so kannst Du mich gerne wie unten beschrieben kontaktieren.

Selbstverständlich wird jeder der mir etwas brauchbares zusendet auch namentlich genannt, außer eine namentliche Nennung wird nicht erwünscht. Im letzteren Fall wird zwar das zugesandte Material verwendet, ansonsten aber alles vernichtet und eine namentliche Nennung besteht aus UNBEKANNT (schließlich möchte ich mich nicht mit fremden Federn schmücken).

Angenommen wird alles was sich irgendwie Lesen läßt, also AMIGA Diskette, PC Diskette, Handschriftliches, mit Maschine geschrieben oder e-Mails. Nur verwendet keine Sonderformate wenn Ihr mir eine Diskette zuschickt.

Zurück zu den Hinweisen.

1.6 schlusswort

Zum Schluß nur noch eine kleine Bitte:

Falls ihr Probleme habt mit dem was ich geschrieben habe, dann meldet Euch doch bitte bei mir, ich werde sehen was ich für Euch tun kann.

Genauso könnt Ihr mich kontaktieren wenn Ihr andere Probleme mit dem AMIGA DOS habt, irgendwelche Skripte schreiben möchtet oder eine Automatisierung einer Funktion auf Scriptbasis erreichen möchtet.

Alle e-Mails werden garantiert beantwortet solange mein Rechner nicht kaputt geht oder der Modem nicht Versagt.

Ist übrigens kein Schreibfehler:
Der Mo (dulattor) dem (odulator).

Da wir gerade bei diesem leidigen Thema sind:
Für Irgendwelche fehlenden Satzzeichen falsch geschriebene Wörter oder überflüssige Satzzeichen bin ich selber voll Verantwortlich.

Ich bekenne mich Schuldig, verurteilt mich zu zwei Jahre Zwangsschreiben auf einer alten Mechanischen Schreibmaschine, aber manchmal wollen Geist und Finger nicht zusammenarbeiten.

Ralf Stecher
Cheruskerstr. 19
38112 Braunschweig
e-Mail: Ralf.Stecher@t-online.de

Zurück zu den Hinweisen.

1.7 mehr-vom-author

Hier nur noch eine kleine Frage die ich beantworten möchte: Was habe ich sonst noch geschrieben ?

Berechnungsformelsammlung

Ist eine Sammlung von Mathematischen Berechnungsformeln für alles mögliche von der Volumenberechnung über Dichteverhältnisse und Geschwindigkeitsberechnungen bis zu Schall und Chemische Elemente sowie noch mehr für alle in verständlicher Form geschrieben so das alle Schüler etwas lernen können und alle anderen ihre Berufsschulkenntnisse aufarbeiten können.

ED-Menü

ED ? Was ist das ?

Hab ich schon öfter gehört, aber ED liegt der Original Workbench bei, ist ein ASCII Editor, also Ideal zum schreiben von Briefen, Programmen oder Script's und hat in der Originalversion nur eine dürftige englische Menüführung. Dem habe ich abgeholfen. ED hat von mir einen Aufsatz bekommen um dieses Menü künftig in Deutscher Sprache zur Verfügung zu stellen. Dabei wird das Original-Programm nicht verändert !!! Inclusive Installer.

Prüfe_Backup

Ist ein kleines Script um Euch regelmäßig an ein Backup zu erinnern. Ich weiß, gibt es schon im PD Bereich, aber dieses Script belegt keinen Speicher im RAM: läuft nicht ständig im Hintergrund, belagert also nicht die CPU und ist ohne Probleme an Eure Bedürfnisse anzupassen. Inclusive installer.

RequestChoice-Script

RequestCoice haben alle 3.1 User auf ihrer Workbench, aber wie geht man damit um ? Ich zeige es Euch.

ESC-Sequenzen

ESC-Sequenzen sind eigentlich Befehle um die Druckerausgabe zu verändern, aber man kann damit auch Texte die in der Shell erscheinen Verändern. Fast alles ist möglich, von der Farblichen Gestaltung bis zu Zeilensprüngen bleiben nicht viel Wünsche offen, mit Beispielen und ausführlicher Beschreibung.

MyMenu

Das wohl Beste Programm das aus dem PD Bereich kommt, allerdings habe nicht ich es geschrieben und eine Deutsche .guide ist auch noch nicht fertig. Dafür aber ein Installer in deutsch. MyMenu erweitert Eure Menüleiste nach rechts und Ihr könnt diese selbst Belegen. Nie wieder eine Schublade aufmachen, ist doch herlich oder ? Läuft unter allen Kichstart Versionen Problemlos. Kann mir jemand den englischen doc. Übersetzen? Das Programm ist aber so einfach das jeder es auch ohne deutsche Beschreibung nutzen kann. Sollte jeder haben.

Alles was ich hier aufgelistet habe könnt Ihr direkt bei mir Beziehen, eine kleine e-Mail mit Euren Wünschen genügt. Beachtet aber bitte das eine Kommerzielle Vermarktung strengstens Verboten ist und das die Verzeichnisse nur komplett weitergegeben werden dürfen. Einzige Ausnahme hiervon ist das DiscMag AMIGATimes welches auch Teile Veröffendlichen darf.

Ralf Stecher
Cheruskerstr.19
38112 Braunschweig
e-Mail: Ralf.Stecher@t-online.de

[Zurück zu den Hinweisen.](#)

1.8 vorwort

Ein kleines Vorwort...

MyMenu ist ein Programm, dass es erlaubt, seine eigenen Workbenchmenüs für Programme zu erstellen.

Es kann dem Benutzer die Mühe ersparen, viele Schubladen öffnen zu müssen, bis man das gewünschte Programm starten kann.

MyMenu erlaubt, CLI- und Workbenchprogramme auszuführen und wird mittels einer einfachen Textdatei konfiguriert.

Zugeben muß ich an dieser stelle allerdings das das Programm nicht von mir geschrieben wurde und auch nicht mehr das neueste ist. Dennoch läuft und läuft es auf bisher allen AMIGA's und unter allen Kickstartversionen die ich bisher in den Fingern hatte. Nun aber genug gerede, weiter geht's.

<---- Zurück zur Hauptseite.

1.9 source-code

Zum Source Code gibt es auch etwas zu sagen:

Dafür gibt es ein mitgeliefertes Makefile, welches mit Manx 3.6 arbeitet, und eine Version von einer Fish Disk (ich weiss nicht mehr von welcher). Das Makefile sollte mit den meisten Make-Programmen arbeiten. Ich habe nicht versucht, es mit 16-bit Integers oder Lattice zum Laufen zu bringen (obwohl ich das gut fände).

Es gibt da einige Defines in MyMenu.h, die selbst bestimmt werden können. DO_WB entscheidet, ob die Unterstützung von Workbenchprogrammen eingebunden wird. Es kann eine Menge Platz sparen, dies zu deaktivieren, wenn man diese Option nicht braucht. Wenn DO_PATH definiert ist, wird Kode einkompiliert, mit dem man den Pfad durchsuchen lassen kann. Momentan ist es undefiniert.

Folgendes zur erklärung:

Da dieses Programm nicht von mir ist und zum zweiten auch schon ziemlich alt ist und ich drittens nicht unbedingt die große ahnung habe :-)) kann ich zum oben genannten auch nicht viel sagen. Den Source Code habe ich dennoch beigelegt für alle die sich daran (hoffendlich erfolgreich) Versuchen wollen. Wenn es jemand von Euch schafft das Programm zu Verbessern wäre es nett wenn ich eine Version zugeschickt bekomme. Schreibt mir dann auch was Ihr Verändert habt und tut der AMIGA-Gemeinde den gefallen und macht keine eingeschränkte Shareware-Version 'draus. Dieses ist nicht gerade das was der AMIGA-User sucht.

<---- Zurück zur Hauptseite.

1.10 installation

Installation:

Man kopiert den MyMenu-Handler in das L: Verzeichnis.
MyMenu selbst wird nach C: geschaufelt.

Dann erstellt man die Datei S:MyMenu.conf. Das Format der Datei wird noch genauer beschrieben.

Soweit der Wortlaut der originalen Übersetzung.

Da ich der meinung binn das es gut so ist das der User immer mehr Verwöhnt wird habe ich ein Script geschrieben das den Installer anspricht.

Ihr braucht dann nur noch den Hinweisen zu folgen.

<---- Zurück zur Hauptseite.

1.11 starten

Starten:

Schritt eins ist, die Workbench zu starten!

Der Befehl "MyMenu" lädt und startet den Handler-Prozess.
Nun sollten oben in der Menüleiste Menüs erscheinen.

Erneutes Starten von MyMenu, wenn der Handler schon geladen ist, veranlasst das Programm, die Konfigurationsdatei erneut zu lesen und die Menüs neu aufzubauen.

Der Befehl "MyMenu quit" beendet das Programm und den Handler.

Auch hier wieder der Hinweis das oben stehendes der Original aus dem Englischen übersetze wortlaut ist.
Dazu habe ich aber noch einiges zuzufügen:

- 1.) Mit MyMenu hat man die Möglichkeit eine Menüleiste an die vorhandenne der Workbench anzuhängen. Dabei ist es nicht möglich MyMenu aus der WB-Startup heraus zu Starten.
MyMenu wird nach LoadWB in der Startup-Sequence aufgerufen.
- 2.) Vor dem Aufruf des Programms ist ein Wait 3 Aufzurufen.
Sollte eigentlich genügen, wenn nicht, setzt den Wert höher.

Wem dieses zu Umständlich ist, der kann nach dem Hochfahren des Rechners auch über den Menüpunkt "Befehl ausführen" der Workbench auch durch einfaches eintippen des Programmnamens MyMenu Starten.

Bei Schwierigkeiten mit diesem Programm, wendet Euch ruhig an mich .

<---- Zurück zur Hauptseite.

1.12 konfiguration

Konfigurationsdatei:

Die Konfigurationsdatei wird beim Start von MyMenu eingelesen.
Zunächst wird danach in S: gesucht, dann im aktuellen Verzeichnis.

In der Konfigurationsdatei legt man die gewünschten Menüs, und welche Befehle sie ausführen, fest. Man darf Carriage Returns zwischen den Argumenten haben, aber nicht innerhalb eines Befehles.
Kommentare beginnen mit einem # und gehen bis zum Ende der Zeile.
Gross- und Kleinschreibung machen keinen Unterschied.

COLOR n

Setzt die Vordergrundfarbe für die neuen Menüs. COLOR kann beliebig oft verwendet werden. Das Argument n legt die Stiftfarbe fest.
Die Voreinstellung ist 2 (schwarz).

MENU [<befehlszeichen>] menü-name menüpunkt-name [untermenüpunkt-name] | ↔
befehlsdefinition

Definiert ein Menü. Jede Menüdefinition muss aus einem Menünamen und einem Menüeintrag bestehen. Ein Untermenüpunkt ist optional. Wenn ein Name ein Leerzeichen enthält, setzt man ihn in Anführungszeichen.
Ein Befehlszeichen für einen bestimmten Menüeintrag kann man festlegen, indem man es hinter dem Schlüsselwort MENU in eckige Klammern < > setzt.

Man trennt die Menüdefinition und die Befehlsdefinition mit einem vertikalen Strich | .

Nach dem Strich kommt das Argument CLI oder WB , um festzulegen, ob es sich um ein CLI- oder WB-Programm handelt.
Alles nach WB bis zum Ende der Zeile wird als Befehl aufgefasst.
Alles nach CLI bis zu einem Leerzeichen oder bis zum Ende der Zeile wird als Befehl aufgefasst. Alles andere wird als Argument an den CLI-Befehl übergeben. Man darf in einem CLI-Befehl keine Umleitungen verwenden.

Wenn das aufzurufende Programm kein Icon besitzt, so ist auf jeden fall CLI zu Verwenden.

MyMenu kann fast alle Programme aufrufen, zumindest hatte ich bei Zwei oder Drei Programmen keinen Weg gefunden diese mittels MyMenu aufzurufen. Ich sag ja, das Prog. ist schon Älter...

Beispiele:

```
menu Privat "Voreinsteller " "Zeit " | WB SYS:Prefs/Time
```

Dies definiert ein Menü "Voreinsteller" mit dem Eintrag "Zeit".
Wenn man diesen Menüeintrag anwählt, wird der Befehl
"SYS:Prefs/Time" als Workbenchprogramm ausgeführt.

```
menu Privat "Black Box" | WB df0:Black_Box
```

Dies definiert einen Menüpunkt "Black Box" im Menü "Privat". Das Menü "Privat" wird schon durch das vorherige Beispiel existieren,

und wird nun "Zeit" und "Black Box" als Einträge enthalten. Der auszuführende Befehl ist "df0:Black_Box" (Box ist kein Argument).

```
color 3
menu Utilities DMouse "Start DMouse" | cli c:DMouse
menu Utilities DMouse "Quit DMouse" | cli c:DMouse quit
```

Dies definiert zwei Menu-Unterpunkte (in gelb). Der zweite Befehl hat ein Argument.

```
menu Extra "Shell" | cli c:Newshell
```

Und so könnt ihr eine Standart-Shell über mit MyMenu öffnen.
Kein auslagern mehr und auch kein öffnen der Schubladen nötig.

<---- Zurück zur Hauptseite.

1.13 zukunft

Zukünftige Pläne und hoffnungslose Träume:

- AREXX Unterstützung
- Pfadsuche reparieren
- ein besserer Parser
- Unterstützung von grafischen Menüeinträgen

Soviel also zur Original Übersetzung.
Schade das ich bis heute nichts Vergleichbares gefunden habe.
Das mit der ARExx Unterstützung halte ich für durchaus machbar (wurde mir auch bestätigt).
Ein besserer Parser ist wünschenswert.

<---- Zurück zur Hauptseite.

1.14 fehler

Einschränkungen, Fehler und andere verrückte Dinge:

Die Pfadsuche funktioniert nicht. Bitte nicht einkompilieren.
Ich würde es zu schätzen wissen, wenn jemand das Problem findet,
da ich dort festhänge. Details sind in DoRun.c zu finden.

Neue Voreinstellungen lassen das Menü verschwinden (aber der Handler wird nicht beendet). Die Menüs kommen wieder, wenn man MyMenu erneut startet.

CLI-Programme lassen sich nicht umleiten, da der Kode schon nach und aus NIL: umleitet.

Zeilengrenze von 256 Zeichen in der Konfigdatei.

<---- Zurück zur Hauptseite.

1.15 danke

Danksagungen:

Ich habe Ideen und Kode-Stückchen aus anderen PD-Programmen verwendet.
Ich habe auch direkte Hilfe von einigen Leuten bekommen. Ich möchte
diesen Leuten danken (an deren Namen ich mich erinnere):

Matt Dillon	- Ideen für den Menüaufbau, und wie man den Handler lädt habe ich von DME und DMouse bekommen.
Peter da Silva	- Ideen von einem WB-Start-Programm bekommen.
Rob Peck	- Einige Ideen und Hilfe.
Davide Cervone	- VIELE Ideen von MonIDCMP bekommen.

Dem habe ich auch noch etwas hinzuzufügen:

Wolfram Riedel	- Für die Deutsche Übersetzung des original.doc
Darin Johnson	- Für dieses Programm

<---- Zurück zur Hauptseite.

1.16 author1

Author des Programmes:

Darin Johnson
darin@laic.UUCP
leadsv!laic!darin@pyramid.pyramid.com
(415) 368-0972

Author der MyMenu.guide

Ralf Stecher
Cheruskerstr.19
38112 Braunschweig
e-Mail: Ralf.Stecher@t-online.de

<---- Zurück zur Hauptseite.
