

0030c158-0

COLLABORATORS

	<i>TITLE :</i> 0030c158-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	0030c158-0	1
1.1	MuchMore_Anleitung	1
1.2	Was macht MuchMore?	2
1.3	Warum MuchMore?	3
1.4	Starten von MuchMore	3
1.5	Start aus der Shell	3
1.6	Start von der Workbench	5
1.7	ToolTypes von MuchMore	5
1.8	Die Kommandos von MuchMore	6
1.9	Bugs, bekannte Probleme	7
1.10	Leertaste	8
1.11	BackSpace-Taste	8
1.12	Cursor rauf/runter	8
1.13	PgUp und PgDn	8
1.14	T oder Home (NB)	8
1.15	T oder End (NB)	9
1.16	F (Find) oder S (search)	9
1.17	N (Next)	9
1.18	P (Previous)	9
1.19	Shift + <F(n)>	9
1.20	<F(n)>	10
1.21	G (Goto)	10
1.22	Shift + Alt + E (Editor)	10
1.23	Shift + Alt + O (printOut)	10
1.24	W (Write)	10
1.25	C (Copy)	11
1.26	L (Load)	11
1.27	<Help> oder H	11
1.28	ESC, Q, oder 5 (Nummernblock)	11
1.29	Änderungen zu früheren Versionen	12

Chapter 1

0030c158-0

1.1 MuchMore_Anleitung

M u c h M o r e V 4.6

=====

INHALT

Was macht MuchMore?

Weshalb MuchMore und nicht MORE, BLITZ oder LESS?

Starten vom MuchMore

Die ToolTypes von MuchMore

Die Kommandos von MuchMore

Bugs, bekannte Probleme

Änderungen zu früheren Versionen

Disclaimer

Es wird keine Haftung für Schäden übernommen, die durch dieses Programm verursacht werden. Die Benutzung des Programms geschieht auf eigene Gefahr.

Copyright

MuchMore darf auf Public-Domain und Shareware Disketten vertrieben und verbreitet werden, solange diese Disketten zum Selbstkostenpreis verkauft werden. Es darf dort zum Anzeigen von ReadMe- und Dokumentationsdateien benutzt werden, allerdings muß mein Name im Programm (HELP) enthalten bleiben.

MuchMore darf nicht zusammen mit kommerziellen Produkten ohne meine ausdrückliche schriftliche Genehmigung verbreitet werden. Selbstverständlich ist eine solche Genehmigung auch bei jeder

anderen Art von kommerzieller Nutzung nötig.

Bei Änderungen im Programm darf sich der Autor der Änderungen in die UpDate-Liste eintragen, darf meinen Namen jedoch nicht löschen. Außerdem bitte ich darum, daß jemand, der Änderungen (Verbesserungen) macht oder Vorschläge für solche hat, sie mir schickt.

Die Dateien

MuchMore	Das ausführbare Programm
MuchMore.dok	Dieser Text
MuchMore.doc	Englischsprachige Anleitung
Install_MuchMore	Installationsscript für den C= Installer
msrc (dir)	Dieses Directory enthält den Quellcode:
MuchMore.mod	Der Oberon-2 Quelltext
catalogs (dir)	Dieses Directory enthält die Kataloge

Compilieren

MuchMore wurde in OBERON-2 mit dem Amiga Oberon Compiler V3.10 entwickelt und compiliert. Eine Demoversion des Compilers und eines SourceLevel-RunTime-Debuggers ist auf AMOK#75 erschienen. Sie kann über jeden guten PD-Versand bezogen werden.

--- Christian

1.2 Was macht MuchMore?

Was macht MuchMore?

MuchMore ist ein Programm, das ähnlich wie MORE, LESS, etc. ASCII-Files auf dem Bildschirm ausgibt. MuchMore öffnet im Gegensatz zu den anderen Programmen jedoch einen eigenen Screen und gibt den Text dort aus. Der Text wird von unten weich in den Screen gescrollt. Wer will kann mit MuchMore jedoch auch unheimlich schnell scrollen. Zusätzlich gibt es noch eine Such- und eine Print-Funktion. MuchMore paßt sich an die Grafikauflösung des Amigas an, auf dem es gestartet wurde. So läuft MuchMore auf NTSC, PAL, Overscan und wohl auch auf zukünftigen AMIGAs!

MuchMore kann nun Texte anzeigen die mit XPK gepackt oder verschlüsselt sind, falls sich die XPK-Libraries im LIBS: Verzeichnis befinden. Wenn zusätzlich die powerpacker.library (© Nico François) installiert ist, können Sie auch powerpacker-gepackte Texte anzeigen.

MuchMore kann mit dem AmigaDOS Befehl 'Resident' resident im Speicher gehalten werden.

MuchMore ist nun auch pipefähig, so daß man z.B. mit "dir c:|MuchMore" das C: Verzeichnis mit MuchMore auflisten kann, wenn die Shell namenlose Pipes ("|") unterstützt (wie z.B. 'csh').

MuchMore benötigt AmigaOS ab 2.04.

1.3 Warum MuchMore?

Weshalb MuchMore und nicht MORE, BLITZ oder LESS?

Weil MuchMore einfach besser ist. Außerdem ist MuchMore sehr viel kürzer als LESS. BLITZ und ähnliche Programme, die schnell und weich scrollen benutzen meist nur 2 Farben und können keine unterschiedlichen Schriftstile anzeigen. MuchMore hat dagegen 4 Farben und kann Schrift fett, kursiv, unterstrichen und invers anzeigen. Anders als viele andere Programme kann MuchMore einen beliebigen nicht-proportionalen Zeichensatz zum Anzeigen des Textes benutzen.

MuchMore ist ideal, um lange Texte anzuzeigen, da der Text automatisch, während er angezeigt wird, geladen wird. Nachdem man die erste Seite gelesen hat, befindet sich also meist der gesamte Text schon längst im Speicher und man kann ohne Verzögerung durch ihn fahren.

1.4 Starten von MuchMore

Starten vom MuchMore

MuchMore kann auf zwei Weisen gestartet werden:

Aus der Shell

Von der Workbench

1.5 Start aus der Shell

Aus der Shell

Schablone:

B=PLANES/N/K, C=PALETTE/K, D=DISPMODEREQ/S, E=EDITOR/K, F=FONT/K,
I=INTERLEAVED/S, O=OLDSTYLE/S, P=TOOLPRI/N/K, Q=FASTQUIT/S,
S=SCROLLMODE/N/K, T=TABWIDTH/N/K, U=PUBSCREEN/K, X=EXTRASPACE/N/K,
W=WINDOW/S, WL=LEFTEDGE/N/K, WT=TOPEDGE/N/K, WW=WIDTH/N/K, WH=HEIGHT/N/K,

Beispiel:

FW=FRWIDTH/N/K,FH=FRHEIGHT/N/K,FILE:

'FILE' ist der ASCII-Text, der ausgegeben werden soll. Wenn der Name Leerzeichen enthält, muß er mit Anführungszeichen umschlossen werden. ' MuchMore ?' gibt die Usage aus. Wenn MuchMore ohne Argument gestartet fragt MuchMore mit einem FileRequester nach dem Dateinamen.

Die Option 'B=PLANES' legt fest, wieviel Bitplanes~MuchMore verwenden soll. Das Maximum ist zwei Planes, d.h. vier Farben.

Mit der Option 'C=PALETTE' können die Farben von MuchMore's Screen gesetzt werden. Beispiel: C=AAA,000,f00,0f0

Wenn die Option 'D=DISPMODEREQ' angegeben wird, erscheint nach dem Start ein Requester, in dem man den Displaymodus des Screens auswählen kann. Wenn man auf das Gadget 'Save' klickt, wird der Modus in eine Datei namens 'MuchMore.prefs' in ENV: und ENVARC: gespeichert.

Die Option 'X=EXTRASPACE' erlaubt es, den Abstand zwischen den Textzeilen zu vergrößern, was die Lesbarkeit erhöht, besonders wenn man den Font topaz/8 verwendet.

Mit der Option 'F=FONT' kann der Zeichensatz gewählt werden, den ~MuchMore verwenden soll. So wird beim Aufruf mit ' MuchMore ' F=topaz/11 der Zeichensatz Topaz mit der Größe 11 verwendet. Es sind nur nicht-proportionale Zeichensätze erlaubt.

Die Option 'I=INTERLEAVED' bestimmt, ob die BitMap des Screens verschachtelt ist.

Mit der Option 'E=EDITOR' kann man den Editor angeben, mit dem der Text editiert wird wenn man Shift-Alt-E drückt. Die Vorgabe ist "C:Ed". Falls nicht vorhanden wird "%s" angefügt. Das %s wird dann durch den Filenamen des gerade angezeigten Texts ersetzt.

Wenn die Option 'O=OLDSTYLE' gesetzt ist, funktionieren PageUp und PageDown wieder so wie in älteren Versionen von MuchMore, also mit Scrolling.

Wenn die Option 'Q=FASTQUIT' gesetzt ist, wird MuchMore beim Drücken von X, Q und NK5 sofort verlassen.

Mit der Option 'P=TOOLPRI' kann man MuchMore's Taskpriorität bestimmen.

Die Option 'S=SCROLLMODE' bestimmt, welche Betriebssystemfunktion MuchMore zum Scrollen verwenden soll:

0: MakeScreen/MrgCop/WaitTOF	(Nicht 100% systemkonform)
1: ScrollVPort/WaitTOF	(Default)
2: MakeScreen/RethinkDisplay	
3: ScrollRaster	(Für nicht ziehbare Screens und externe Grafikkarten)
4: WaitTOF/ScrollVPort	(wie 1, aber zuerst WaitTOF)
5: ScrollVPort/WaitBOVP	(Busy waiting)

Die Option hat keine Auswirkung, wenn MuchMore in einem Fenster läuft, hier muss mit ScrollRaster gerollt werden.

Mit der Option 'T=TABWIDTH' koennen Sie die Tabulatorbreite setzen. Die Vorgabe ist 8.

Wenn die Option 'W=WINDOW' gesetzt ist, öffnet MuchMore keinen eigenen Screen, sondern benutzt ein Fenster auf der Workbench zum Anzeigen des Texts. Die Position und Größe des Fenster können mit den Optionen LEFTEDGE, TOPEDGE, WIDTH und HEIGHT bestimmt werden.

Mit der Option 'U=PUBSCREEN' geben Sie den Namen des PublicScreens an, auf dem MuchMore sein Fenster öffnen soll.

Mit den Optionen 'FW=FRWIDTH' und 'FH=FRHEIGHT' legen Sie die Größe des Dateiauswahlfensters fest.

Die

ToolTypes

in MuchMore's Icon werden auch beim CLI-Start ausgewertet. Wenn MuchMore resident geladen wurde, wird das Icon im C: Verzeichnis gesucht.

1.6 Start von der Workbench

Von der Workbench

Von der Workbench muß lediglich der auszugebende Text angeklickt und dann MuchMore Shift-Doppelgeklickt werden.

Texte können MuchMore als Default-Tool haben: z.B. ":c/MuchMore", wenn sich MuchMore auf der Diskette im :c-Directory befindet. Die Texte werden dann einfach durch Doppelklick gezeigt.

1.7 ToolTypes von MuchMore

MuchMore unterstützt folgende Tooltypes:

PALETTE=#{, #} entspricht

CLI Option C	EXTRASPACE=#	entspricht
CLI Option X	FONT=name/size	entspricht
CLI Option F	EDITOR=name	entspricht
CLI Option E	OLDSTYLE=TRUE FALSE	entspricht
CLI Option O	FASTQUIT=TRUE FALSE	entspricht
CLI Option Q	PLANES=#	entspricht
CLI Option B	SCROLLMODE=#	entspricht
CLI Option S	TABWIDTH=#	entspricht
CLI Option T	TOOLPRI=#	entspricht
CLI Option P	INTERLEAVED=TRUE FALSE	entspricht
CLI Option I	PUBSCREEN=name	entspricht
CLI Option U	WINDOW=TRUE FALSE	entspricht
CLI Option W	TOPEdge=#	entspricht
CLI Option WT	WIDTH=#	entspricht
CLI Option WW	HEIGHT=#	entspricht
CLI Option WH	FRWIDTH=#	entspricht
CLI Option FW	FRHEIGHT=#	entspricht
CLI Option FH		

Die Tooltypes des anzuzeigenden Textes haben dabei Vorrang vor MuchMore's Tooltypes.

Wenn der Text verschlüsselt ist, erscheint vor dem Laden des Textes ein Eingebefeld indem Sie das Passwort eintragen müssen.

1.8 Die Kommandos von MuchMore

Die Kommandos von MuchMore

Die Funktionen werden durch Drücken der entsprechenden Tasten aufgerufen:

LeerTaste oder linke Maustaste

BackSpace oder rechte Maustaste

Hoch/Runter oder 2/8 des Zehnerblocks

PgUp / PgDn (Zehnerblock)

T (top) oder Home (Zehnerblock)

B (bottom) oder End

F (find) oder S (search)

N (next)

P (previous)

Shift + Fn

Fn

G (goto)

Shift + Alt + E (Edit)

Shift + Alt + O (printout)

W (write)

C (copy)

L (load)

HELP oder H

ESC, Q, X oder 5 des Nummernblocks

1.9 Bugs, bekannte Probleme

Bugs, bekannte Probleme:

Wenn man Programme wie AutoPoint laufen hat, die immer das Fenster über dem sich der Mauszeiger befindet aktivieren, kommt es zu Problemen mit dem Eingabefenster für den Find/Goto Befehl, da es durch inaktivieren wieder geschlossen wird.

Programme wie PowerSnap schnappen an falscher Stelle, nachdem man gescrollt hat. Abhilfe: Vor dem Snappen zweimal die Helptaste betätigen, dadurch wird RasInfo.dyOffset wieder auf Null gesetzt.

Kursive Buchstaben können den Windowrahmen übermalen.

Wenn man MuchMore mit dem APipe-Handler verwendet werden die ersten vier Buchstaben verschluckt.

1.10 Leertaste

LeerTaste oder linke Maustaste

Hält das Scrollen an bzw. startet es wieder. Ist das Scrollen angehalten, wird unten im Screen eine Zeile mit Informationen wie TextLänge, Zeilennummer etc. eingeblendet.

Beim Drücken der Leertaste am Ende des Textes wird MuchMore verlassen.

1.11 BackSpace-Taste

BackSpace oder rechte Maustaste

Scrollt den Text rückwärts. Beim zweiten Drücken wird wie bei der Leertaste angehalten.

1.12 Cursor rauf/runter

Hoch/Runter oder 2/8 des Zehnerblocks

Mit den Cursortasten kann man eine Zeile nach oben bzw. unten fahren.

Die Tasten zusammen mit der Shift-Taste starten bzw. stoppen schnelles hoch & runterscrollen (wie bei Space und BackSpace).

Control plus diese Tasten erhöht die Scrollgeschwindigkeit um den Faktor 2 bis 3.

Zusammen mit Alt wird eine Seite hoch bzw. runter gescrollt.

1.13 PgUp und PgDn

PgUp / PgDn (Zehnerblock)

Scrollt eine Seite hoch bzw. runter.

1.14 T oder Home (NB)

T (top) oder Home (Nummernblock)

Springt zur ersten Seite des Texts.

1.15 T oder End (NB)

B (bottom) oder End (Nummernblock)

Springt zur letzten Seite des Textes.

1.16 F (Find) oder S (search)

F (find) oder S (search)

Liest eine Zeichenkette ein und sucht diese ausgehend von der auf dem Bildschirm obersten Zeile im Text. Wird die Taste zusammen mit der Shift-Taste gedrückt, wird mit Berücksichtigung von Klein und Großbuchstaben gesucht, sonst ohne. War die Suche erfolgreich, wird der Text mit der hervorgehobenen gefundenen Zeichenkette angezeigt. War die Suche nicht erfolgreich, blitzt der Bildschirm kurz auf.

1.17 N (Next)

N (next)

Sucht die nächste Zeile, die den bei 'F' eingegebenen String enthält (s.o.).

1.18 P (Previous)

P (previous)

Sucht rückwärts nach der nächsten Zeile, die den bei 'F' eingegebenen String enthält. Bei Erfolg wird die gefundene Zeile wie bei 'F' ausgegeben, sonst blinkt der Bildschirm kurz.

1.19 Shift + <F(n)>

Shift + F(n)

Shift zusammen mit einer Funktionstaste markiert die aktuelle Textposition. Da es 10 Funktionstasten gibt, können bis zu 10 Marken gesetzt werden.

1.20 <F(n)>

F (n)

Durch Drücken einer Funktionstaste kann man zu einer der zuvor mit Shift-Funktionstaste markierten Positionen springen. Ist Marke Nummer n noch nicht gesetzt, wird diese auf die aktuelle Position gesetzt (wie bei Shift&Fn).

1.21 G (Goto)

G (goto)

Es wird ein Fenster mit einem Integer-Gadget geöffnet, in das eine Zeilennummer eingegeben werden kann, ab der dann der Text angezeigt wird.

1.22 Shift + Alt + E (Editor)

Shift + Alt + E (Edit)

Hiermit kann man den gerade angezeigten Text editieren. Dazu wird der Befehl "C:ED %s" ausgeführt, das '%s' wird durch den Filenamen ersetzt. Mit der Option EDITOR (siehe oben) kann man den Befehl so ändern, daß nicht ED sondern ein beliebiger Editor aufgerufen wird. Achtung: Wenn der Text gepackt war, wird der entpackte im T: Verzeichnis befindliche Text editiert.

1.23 Shift + Alt + O (printOut)

Shift + Alt + O (printout)

Druckt den Text aus. Der Text wird mit "TYPE text TO PRT:" ausgedruckt. Es dürfte also bei keinem Drucker, für den die Preferences richtig eingestellt sind, Probleme geben. Die Befehle TYPE und RUN müssen im C: Directory sein. Während dem Ausdruck muß man NICHT warten, sondern kann mit MuchMore weiter durch den Text scrollen.

1.24 W (Write)

W (write)

Speichert den Text, der zwischen den beiden Textmarkern, die mit F1 und F2 gesetzt wurden in eine Datei oder druckt ihn aus, wenn man bei der Eingabe des Dateinamens einfach RETURN drückt. Es wird der Text von der obersten Zeile des mit F1 markierten Seite bis zur untersten Zeile der mit F2markierten Seite gespeichert. Um die aktuelle Seite auszudrucken reicht es also Shift+F1, Shift+F2, w und RETURN zu drücken. Klar? Nein, dann ausprobieren.

1.25 C (Copy)

C (copy)

Kopiert den Text zwischen den Textmarkern, die mit F1 und F2 gesetzt wurden, ins Clipboard. Dieser Befehl benötigt die `iffparse.library` der Workbench 2.04+.

1.26 L (Load)

L (load)

Öffnet den ASL-FileRequester und lädt dann einen neuen Text.

1.27 <Help> oder H

HELP oder H

Zeigt alle Kommandos von MuchMore an, so daß auch jeder, der diese Dokumentation nicht besitzt die Befehle kennt. Nach dem Drücken einer beliebigen Taste springt MuchMore zurück in den normalen Text.

1.28 ESC, Q, oder 5 (Nummernblock)

ESC, Q, X oder 5 des Numernblocks

Verläßt MuchMore . Um ein versehentliches Verlassen des des Programms zu vermeiden, wird beim ersten Betätigen einer dieser Tasten erst die Infozeile angezeigt und erst beim zweiten Drücken MuchMore verlassen. Mit der ESC Taste kann das Programm jedoch wie bisher sofort verlassen werden.

1.29 Änderungen zu früheren Versionen

Änderungen zu früheren Versionen:

v1.1:

- Erste veröffentlichte Version

v1.2:

- Wird das Scrollen mit <Space> angehalten, werden Filename, Filelänge etc. angezeigt.
- Anstatt Space und Backspace kann ab sofort auch die linke und rechte Maustaste genommen werden.
- Steuerzeichen (unterstrichene, farbige,...) werden nicht mehr ausgegeben, sondern überlesen.

v1.3:

- Starke Geschwindigkeitssteigerung beim Scrollen. Jetzt QText anstatt WarpText (Sorry, Bill). MuchMore ist jetzt trotz größerem Screen genauso schnell wie Blitz! (Video-Sync an).
- Unterstützung der Nummernblocktasten (Home, End, PgUp etc.)
- Screen wird jetzt so groß wie gfxbase^.normalDisplayXXX. Dadurch läuft MuchMore auf PAL, NTSC, Amigas mit Overscanworkbench (wie meiner), mit zukünftigen Auflösungen etc. zusammen. Meine normalerweise 88 Zeichen breiten Texte werden jetzt wenigstens bei mir ohne Zeilenumbrüche ausgegeben.

v1.4:

- Textausgabe und Textladen geschieht jetzt asynchron in 2 Tasks. Dadurch entfällt Wartezeit beim Laden und das Scrolling wird einheitlicher
- Die Möglichkeit, mehrere Texte gleichzeitig zu lesen, habe ich entfernt, da ich sie nie verwendet habe und sie das Programm nur unnötig verkompliziert hat.
- Fehler in Filelänge und End-Funktion verbessert
- Neue Commandos: Find, Next, Previous zum Suchen nach Strings
- Scrolling hält an, wenn das MuchMore-Fenster inaktiviert ist

v1.5:

- Neues Commando: Text ausdrucken (Shift+Alt+O)
- Alle Importe (bis auf Arts) entfernt. Dadurch ähnlicher Effekt wie bei einem Smartlinker.
- Zusätzliche Taste "5" des Nummernblocks für Quit.

v1.6:

- Ab dieser Version können Texte in verschiedenen Zeichensätzen (fett, kursiv, invers) und in verschiedenen Farben angezeigt werden!
-

v1.7:

- Nun ist MuchMore wieder genauso schnell wie bei v1.5, trotz der zweiten Bitplane und den verschiedenen Zeichensätzen.
- Print (shift+alt+o) stürzt nun nicht mehr ab, wenn man MuchMore verläßt, bevor das Drucken beendet ist.

v1.8:

- Fehler, der beim Suchen nach einem String auftrat, nachdem man einen längeren String gesucht hatte, behoben.
- Mit F, N oder P gefundene Zeichenketten werden nun hervorgehoben.
- NK 0 zum Anzeigen von Filelänge etc. hinzugefügt.

v2.0:

- MuchMore nach OBERON umgesetzt
- Fehler, der dafür sorgte das der rechte Mausknopf auf manchen Amigas nicht funktionierte, behoben
- ARP - FileRequester eingebaut
- Neues Kommando: L = neuen Text laden

v2.1:

- Schreib-/Lese-Fehler führen nicht mehr zum gnadenlosen Absturz
- Steuersequenzen wurden rückwärts ausgewertet, was in manchen Fällen zu Fehlern führte. Nun behoben.
- Ping-/Pong- Befehle zum Markieren von bis zu zehn Textstellen (Shift & F1..F10) um dort schnell wieder hinzuspringen (mit F1..F10)
- Neuer Befehl: 'G' für goto

v2.2:

- MuchMore verwendet jetzt keine hartcodierte Tastaturbelegung mehr, sondern die des console.device. Nun dürfte es auch auf japanischen Tastaturen keine Probleme geben.

v2.3:

- Neues Commando: schlafen ("s") und pop-up Funktion mit linke Alt & ESC. "MuchMore-s" geht sofort in den schlafenden Zustand.
- Fehler bei der Ausführung von <CSI>[m behoben. Wird nun wie <CSI>[m behandelt.

v2.4:

- Neues Commando: 'w' für Block speichern / ausdrucken.
- Argumentparser neu geschrieben. Verarbeitet nun auch Anführungszeichen.

v2.5:

- Nimmt nun den mit evt. SetFont gesetzten 8x8 Font anstatt Topaz 8.
- Alloziert nicht mehr für jede Zeile extra Speicher, sondern nur noch 4K große Blöcke.

v2.6:

- asynchrones, schnelles Scrollen mit Control.
- MuchMore ist nun reentrant (pure)
- MuchMore öffnet bei so eingestellten Preferences einen Interlace-Screen.
- Bei Speichermangel wird versucht ein Screen mit nur einer Plane zu öffnen.

v2.7:

- Option '-l' um Interlace zu toggeln
- ASL-FileRequester wird nun statt ARP verwendet, wenn vorhanden
- benutzt RethinkDisplay() nicht mehr 50x die Sekunde, da das unter 2.0 viel zu lahm ist.

v2.8:

- Bei Interlace-Screen wird zwischen den Textzeilen nun zwei Zeilen Platz gelassen, damit der Text leichter lesbar ist.

v3.0:

- Sleep ('-s') wird nicht mehr unterstützt.
- Es wird nun jeder nicht-proportionale Zeichensatz unterstützt
- Neue Optionen beim ~Start vom CLI: '-f' und '-s<size>' zum wählen des Zeichensatzes.

v3.1:

- Benutzt den Screenmode des Workbench Screens

v3.2:

- Unterstützung für nicht scrollbare Screens
- Kein Scrolling mehr bei kompletten Neuaufbau

--- Fridtjof.

v3.2.1:

- XPK Support
- Neue Syntax für Option -f. Beispiel: -ftopaz/11
- Neue Option -p für Password
- Neue Option -e für Extra Spacing
- Neue Option -c für Screen Colors. Beispiel: -c0,abc,f00,0e6
- Tool-Types

v3.2.2:

- Bei einigen Overscan-Auflösungen zeigte die Vorgängerversion die beiden ersten Zeichen einer Zeile am Ende der Zeile an.
-

v3.2.3:

- Der nicht-scrollable Modus funktioniert nun auch unter KS2.04 bei einem 8x8 Font.
- Das Console-Device wurde bisher nicht wieder geschlossen.
- Displaymoderequester
- Neue Option -s für Scrollmode
- Busypointer
- ToolTypes werden auch beim CLI-Start ausgewertet (ab AmigaOS 2.04)
- Beim WB-Start werden MuchMore's ToolTypes nun auch gefunden, wenn MM als DefaultTool des Textes nicht direkt, sondern über den SuchPfad gefunden wurde.

v3.2.4:

- Einige Bugs in GetString() beseitigt
- MM ignoriert nun Tasten/Maus-Drücke während er Busy ist

v3.2.5:

- Asynchrones Scrollen geändert. MM scrollt nun unabhängig von der Maschine beim Halten der Control-Taste ca. 2-3 mal schneller.
- Dos.ReadArgs
- Neue Option -t bzw. "T=TOOLPRI" für Taskpriority

v3.2.6:

- Scrollmode 3 scrollt nun auch weich
- Scrollmode 1 ist nun Default ab 2.04
- Öffnet seinen Screen nun mit voller Textoverscan-Breite
- Neue Option -a bzw. "A=TABWIDTH" für Tabulatorbreite
- Neue Option -o bzw. "O=ONEPLANE"
- Neue Option "N=NOOSCAN"
- Benutzt QText nicht mehr (Sorry, musste sein)

v3.2.7:

- MM poked nun nicht mehr direkt in den Bitplanes herum, sondern benutzt dafür die Funktionen der graphics.library

v3.2.8:

- Option O=ONEPLANE entfernt
- neue Option B=PLANES

v3.2.9:

- V36 ANSI Codes implementiert (<CSI>22-28m)

v3.2.10:

- Locale Support
- Displaymode-Requester nun Font-sensitiv

v3.3:

- Die Tasten Q, X and NK5 zeigen nun erst die Infozeile an, bevor das Programm verlassen wird, um versehentliches Verlassen zu vermeiden

v3.4:

- Neuer Befehl "C", um den Bereich zwischen Marke 1 und 2 ins Clipboard zu kopieren

v3.5:

- Suchen ist nun sowohl mit als auch ohne Beachtung der Groß/Klein- schreibung möglich.
- MM verwendet nun den Boyer-Moore Algorithmus zum Suchen, dadurch ergibt sich eine Beschleunigung um circa den Faktor vier bei einem aus 10 Buchstaben bestehenden Suchtext. Nachteil: Das Suchen eines 1 Zeichen langen Suchtexts benötigt etwas länger als der bisherige Algorithmus.

v3.6:

- Auf Wunsch eines einzelnen Users kann bei PageUp/Down optional wieder gescrollt werden (Option O=OLDSTYLE)
- Bildschirm blitzt, wenn bei Dateianfang (-ende) Up (Down) gedrückt wird

v3.7:

- Neue Option "E=EDITOR"
- MuchMore war nicht 100% pure, da der BusyPointer in MM's Hunks lag -> Probleme mit WShell
- Text wurde nicht gedruckt wenn Spaces im Dateinamen waren
- Option PASSWORD entfernt. Password muss nun im Stringgadget eingegeben werden
- Optionen NOOSCAN und SCROLLMODE entfernt
- Aufgrund eines Bugs in Kickstart 1.3 wurde bei der Darstellung von Kursivschrift manchmal wild im Chipmem herumgepoket was nach kurzer Zeit zum Crash führte. Nun wird unter Kick 1.3 nicht mehr der RastPort des Screens direkt benutzt.

v4.0:

- Kick 1.3 wird nun nicht mehr unterstützt
- MM kann nun auch in einem WB Fenster laufen

v4.1:

- Setzt nun rp.mask bzw. MaxPen (v39), so dass nur noch zwei oder eine (mit Option PLANES=1) Bitplane gescrollt werden, falls MM in einem Fenster läuft.
 - Verwendet unter v39 nun den Prefs BusyPointer.
 - Pipefähig
 - Neue Option: PUBSCREEN
-

- Option SCROLLMODE wiedereingeführt
- Neue Option I=INTERLEAVED

v4.2:

- Zoomgadget
- Benutzt nun eine bessere Methode um auf eine Pipe zu testen, deshalb erscheint bei "run MuchMore" wieder der FileRequester
- Bug fix: Das Icon wird nun auch gefunden, wenn man MM im CLI mit Pfadangabe startet
- Bugs in MuchMore's Tasksynchronisation entfernt
- Neue Option Q=FASTQUIT

v4.3:

- Neue Optionen FRWIDTH, FRHEIGHT
- Passwort wird nicht mehr bei Eingeben angezeigt

v4.4:

- Bug, der unter bestimmten Umständen zum Deadlock führte, beseitigt
- Verwendet nun den ASL ScreenMode-Requester, dadurch ist MM 1480 Bytes kuerzer geworden

v4.5:

- Bugfix: MM hat bisher beim CLOSE den Font nicht wieder auf den ursprünglichen Font zurückgesetzt, wodurch er zweimal geschlossen wurde

v4.6:

- Auf vielfachen Wunsch ist die "Find" Funktion auch über die Taste "s" erreichbar
 - Filtert ASCIIIs 128 bis 160 nicht mehr aus, so daß auch bei Verwendung eines IBM-Fonts Umlaute korrekt dargestellt werden
-