

FinalExecute

Matthias Faust

COLLABORATORS

	<i>TITLE :</i> FinalExecute		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Matthias Faust	March 26, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	FinalExecute	1
1.1	Inhalt	1
1.2	Allgemeine Nutzerwarnung	1
1.3	Was ist FinalExecute ?	2
1.4	Was kann FinalExecute ?	2
1.5	Was benötigt FinalExecute ?	2
1.6	Wie installiere ich FinalExecute ?	3
1.7	Wie bediene ich FinalExecute ?	3
1.8	Der Start-Vorgang	3
1.9	Beendigungs-Vorgang	4
1.10	Tooltypes	4
1.11	Oberfläche	4
1.12	Tastatur	7
1.13	ARexx-Port	8
1.14	Tab-Expanding	8
1.15	Rechtliches	9
1.16	Zukunft	10
1.17	Bekannte Probleme/Bugs/Tips	10
1.18	Autor	10
1.19	Programmgeschichte	11
1.20	Danksagungen	11

Chapter 1

FinalExecute

1.1 Inhalt

Willkommen zu FinalExecute !

Themen:

Allgemeine Nutzerwarnung

Was ist es ?

Was kann es ?

Was benötigt es ?

Wie installiere ich es ?

Wie bediene ich es ?

Rechtliches

Zukunft

Bekannte Probleme/Bugs/Tips

Autor

Danksagungen

Programmgeschichte

Nervendes Mädchik-Kuhki:

Wenn ich den See seh, brauch ich kein Meer mehr (Matthias "DrMabuse" Lühr)

1.2 Allgemeine Nutzerwarnung

WARNUNG ! Dieses Programm benötigt MUI, sollte das ein Anstoß sein, löschen Sie es ←
ruhig, mit folgenden Worten:

"Ich hasse es, es ist widerlich !"

Wenn Sie es aber gleichgültig, was zu verachten ist, hinnehmen, dann sagen Sie ←
bitte nicht, daß ich Sie
nicht gewarnt habe.

Mit freundlichen Grüßen

Matthias Faust

1.3 Was ist FinalExecute ?

Es gibt da einen Menüpunkt im Workbench-Menü: "Befehl ausführen ...", wird er aufgerufen, erscheint ein spartanisches Fenster, mit einem Cursor der zur Eingabe auf →
fordern soll. Ja, dann beginnt die Grübelei, wie hieß das Verzeichnis mit meinem →
Bild,
meinem Text, dem Befehl gleich... Und oft fällt er einem nicht ein, man greift in
seiner Not zu diversen "Strohhalmen", man ruft den Filebrowser auf, freut sich →
über
das Dock am Bildschirmrand, eine gute Shell und beginnt nach und nach diesen
Menüpunkt zu verachten. Dabei hat er es gar nicht verdient, kein mir bekanntes OS
besitzt ein Feature auf die schnelle mal einen Einzeiler zu starten, ohne dabei →
gleich
alles was Rang und Namen im OS hat als Eingabehilfe nutzen zu müssen.

Doch er ist zu verbessern ! Denn seit jetzt gibt es die exzessive Form für
den "Befehl ausführen"- bzw. "Execute command"-Requester:

FinalExecute

Das heißt:

1. Noch weniger Mausfahrt (Schont die Maus)
2. Noch weniger Tastaturbenutzung (Schont die Tastatur)
3. Noch weniger Arbeit (Schont das Hirn)

1.4 Was kann FinalExecute ?

Ganz kurz: Die Fähigkeiten:

- Befehlsキャッシング (zur Befehlswiederholung)
- wahlweise automatische Shell-Erzeugung
- Namensvervollständigung durch Tab-Expanding mit MultiPfad-Scan
- Drag'n'Drop
- Filerequester
- Möglichkeiten zur Einstellung des derzeitigen Verzeichnisses,
Ausgabe-Umlenkung, Stack-Größe, Priorität, des Shell-Namens und
wahlweises Ausschließen der .info-Files
- Dateityp-Erkennung (benötigt Datatypes.library V45 oder höher von R.Mainz)
- ARexx-Port

1.5 Was benötigt FinalExecute ?

Es benötigt:

- OS 3.0 oder höher, da ich einige neue Funktionen nutze

- muimaster.library ab Version 11, d.h. MUI3.0 oder höher
- mindestens 1 MB Speicher
- mind. MC68020-CPU
- ca. 64 KByte freien Speicherplatz auf der Festplatte
- datatypes.library V45 oder höher, von Roland Mainz, erhältlich im Aminet

1.6 Wie installiere ich FinalExecute ?

Es ist ein Installer-Skript mitgeliefert, einfach Doppelklick auf das Icon, dann läuft alles automatisch ab.

1.7 Wie bediene ich FinalExecute ?

Hier ist ein Überblick über die Haupt-Punkte der Bedienung:

- Start-Vorgang
- Beendigungs-Vorgang
- Tooltypes
- Oberfläche
- Tastatur
- ARexx-Port

1.8 Der Start-Vorgang

Man startet FinalExecute am besten per Doppelklick auf das Icon des Programms man kann dieses natürlich auch per CLI starten...

Am besten ist das Programm in der WBStartup-Schublade aufgehoben, da es dann sofort beim LoadWB-Befehl gestartet wird.

Durch die Tooltypes kann die maximale Anzahl der Befehle im Cache geregelt werden. ↔

Dank MUI ist es als Commodity eingebunden, d.h. die meisten Einstellungen sind durch MUI geregelt. Also bitte die Programm-Einstellungen, was u.a. auch die Aufruftasten betrifft, hierüber regeln. ↔

Zu Beginn wird erscheint ein verhältnismäßig kleines Fenster in der Mitte, des Schirms, dieses kan man nach Wunsch zurechtschieben und vergrößern, und es dann fixieren per MUI-Knopf in der oberen Fensterleiste.

1.9 Beendigungs-Vorgang

Da MUI jedes Programm als Commodity in das System einbindet, wenn man es will, kann es durch das Commodore(RIP)-Tool "Exchange" beendet werden.

Oder man wählt im Projekt-Menü des Programm-Fensters nach dem Aufruf "Beenden ↔".

1.10 Tooltypes

MAXLASTINPUTS - Maximale Menge von Einträgen des Befehls-cache,
Standard: 32 Einträge, jeder Eintrag belegt maximal 1 KByte ↔
Speicher

1.11 Oberfläche

Eingabe-String-Gadget

Dieses bietet Platz für 1 KByte Eingabetext, das müßte eigentlich für 95 % ↔
aller

Eingaben ausreichen, aber es kann ja sich noch etwas ändern.

Es kann in ihm per Cursor-Tasten die letzten Eingaben aufgerufen werden, bzw.
mit der Tab-Taste die Tab-Expanding-Funktion aufgerufen
werden. Außerdem können Icons darüber gezogen werden, deren Dateinamen sofort
eingefügt werden.

Rechts daneben befindet sich ein kleines Popup-Gadget, damit kann ein System- ↔
Filerequester
aufgerufen werden und mehrere Files ausgewählt werden, deren Dateiname wird ↔
danach ebenfalls
sofort eingefügt.

Lupe:

- dieses Gadget erlaubt den freien Zugriff auf den Befehls-cache,
es erscheint eine Liste in welcher auch per Cursor-Tasten
auf- und abgewandert werden kann.
Auf weiterem Druck auf das Gadget verschwindet die Liste wieder.

NewShell:

- bei Druck auf dieses Gadget wird eine neue Shell erzeugt

Ok
--

- führt zur sofortigen Befehls-Ausführung

Abbrechen

- Schließt das Fenster, ohne den Befehl auszuführen.

Außerdem bietet das Programm noch Fenster-Menüs an:

Projekt-Menü

- Über

Ein Requester mit Informationen über dieses Programm erscheint und kann auf per Druck auf Gadget oder Return-Taste beendet werden.

- Über MUI

Service für unregistrierte Benutzer von MUI: man kann sich hier ↔ registrieren lassen. :)

- Verstecken

Schließt das Fenster. Beendet aber das Programm nicht.

- Beenden

Beendet das Programm.

Options-Menü

- Assignliste durchsuchen:

Tab-Expanding durchsucht auch die Assignliste (auch Mehrfach-Assigns).

- WB-Pfade durchsuchen:

Tab-Expanding durchsucht auch die Pfade der Workbench

- Datentyp erkennen:

Wenn ein File per Tab-Expanding, Filerequester oder Drag'n'Drop in das vorher leere Stringgadget eingefügt wurde, dann wird versucht den ↔ Dateityp zu erkennen und ggf. das Hilfsprogramm oder bei mehreren das ausgewählte davor ↔ eingefügt.

- .info-Dateien einschließen:

Falls .info(Icon-)-Dateien mit in alle Operationen eingeschlossen werden ↵
sollen,
muß dieser Punkt ein Häkchen bekommen.

- Optionen editieren:

Das Options-Fenster wird aktiviert, in der Liste befinden sich mehrere ↵
String-Gadgets,
deren Funktion darin besteht, verschiedene Optionen für die Ausführung ↵
eines Programms
für den Benutzer offen zu lassen:

- Ausgabe:

- hier kann die Ausgabe umgeleitet werden, dieses geschieht meistens ↵
über ein CON:-
oder RAW:-Fenster

- Aktuelles Verzeichnis:

- wie der Name schon sagt, kann hier ein Verzeichnis-Name angegeben ↵
werden,
vor der Ausführung der Eingabe wechselt das System dann vorher in ↵
dieses Verzeichnis.

- Stack:

- hier kann die Größe des Stack gewählt werden, er muß mindestens ↵
4096 Bytes
groß sein, es kann sein, daß wenn das Feld leer bleibt, das System ↵
4000 Bytes
einstellt, was ich persönlich frustrierend finde, denn Commodore(↵
RIP) selbst riet
zu einem Stack von 4096 Bytes Größe, also ein Widerspruch in sich ↵
:(

- Priorität:

- hier kann die Priorität des zu erzeugenden Prozesses gewählt ↵
werden,
standardmäßig ist sie auf 0 eingestellt, es kann ein Wert von -128 ↵
bis 127
gewählt werden, aber man sollte nie außerhalb von -5 bis 5 gehen, ↵
da System-
Tasks sonst außer Kontrolle geraten, bzw. das System extrem ↵
langsam wird.

- Shell-Namen:

- falls gerade eine Shell offen ist, kann man sie ja zur Ausgabe ↵
nutzen,

der Name kann hier eingegeben werden.

- Speichern

Speichert die aktuellen Optionen in der
Programm-Prefs-Datei ab.

- Zurückladen

Lädt die in der Programm-Prefs-Datei ein.

1.12 Tastatur

Tabulator-Taste - erfüllt im Eingabe-String-Gadget die Funktion
des Tab-Expandings.

Cursor hoch/runter - erfüllt im Eingabe-String-Gadget die Funktion
der Befehlswiederholung, wie in einer Shell

Return/Enter - erfüllt im Eingabe-String-Gadget die Funktion
der sofortigen Befehlsausführung nach dem Schließen
des Fensters,
in der Options-Liste wird
das jeweils nachfolgende Gadget aktiviert, d.h. es
wird von einem String-Gadget zum nächsten gesprungen.

O - wie Return/Enter bei inaktiven Eingabe-String-Gadget

N - schließt das Fenster und führt den Befehl aber nicht aus.

Rechte Amiga + ? - Öffnet einen Requester mit Informationen über das Programm,
kann per Gadgetdruck oder Return quittiert werden.

Rechte Amiga + ! - Öffnet einen Informations- und Registrierungs-Requester von ←
MUI.

Rechte Amiga + H - Schließt das Fenster ohne das Programm zu beenden.

Rechte Amiga + Q - Schließt das Fenster und beendet das Programm.

Rechte Amiga + A - Durchsuchen der Assignliste an/aus

Rechte Amiga + W - Durchsuchen der WB-Pfade an/aus

Rechte Amiga + I - Ein/Aus-Schließen von ".info"-Dateien

Rechte Amiga + T - Erkennung des Dateityps an/aus

Rechte Amiga + O - Befehlsoptionen editieren

Rechte Amiga + S - Speichert aktuelle Optionen in die MUI-Programm-Prefs-Datei ←
ab.

Rechte Amiga + R - Lädt Programm-Einstellungen wieder ein.

1.13 ARexx-Port

Der ARexx-Port heißt "FINALEXECUTE".

Nun ja, ist zwar nur ein Befehl, aber besser als keiner:

- CALLREQ - öffnet das Programm-Fenster, gedacht falls man in beliebigen Programmen die ARexx unterstützen einen besseren "Execute DOS-Command"-Requester haben will.

Parameter: INPUTSTRING/K - Ein String der das Eingabe-String-Gadget initialisiert.
Beispiel: CALLREQ INPUTSTRING="'dir opt a' "

PUBLICSCREEN/K - Ein String, der den Namen des Publicscreens definiert, auf dem das Fenster sich öffnen soll.
Beispiel: CALLREQ PUBLICSCREEN="Workbench"

1.14 Tab-Expanding

Was ist Tab-Expanding ?

Eine besondere Fähigkeit die ich in vielen Unix-Tools wie XEmacs besonders schätze ↔

ist die Fähigkeit, nach Druck auf die Tabulator-Taste, den Dateinamen zu vervollständigen.

Das wäre eigentlich schon das ganze Geheimnis.

Dasselbe tut auch FinalExecute.

Doch wie nutze ich diese Fähigkeit ?

Gibt man den Anfang bzw. das Fragment eines Namens ein, versucht es (das Programm) das Fragment in Pfad- und Datei-Teil zu zerteilen. Ist ein Pfad vorhanden wird in diesem nach dem Namensmuster, der Datei gesucht. Ist dieser aber nicht vorhanden so versucht er im aktuellen-Verzeichnis das Muster zu finden, ist ebenfalls nicht möglich, so sucht er die Device-, Assign- bzw. Laufwerksliste durch, ist es nicht zu finden, tut er nichts. Findet er aber eine Übereinstimmung, so setzt er sie ein, findet er mehrere öffnet er ein Fenster, worin man sich dann die passende Datei oder das passende Verzeichnis aussuchen kann. Steht der Cursor auf einer Leerstelle oder am Anfang eines Namens, so öffnet er einen Filerequester worin man sich dann den/die Namen herauspicken kann die man suchte.

Man kann wenn man will DOS-Muster einsetzen.

Also wie man sieht, eine Wohltat für Leute die am liebsten sowieso nur einen Finger benutzen um Befehle einzugeben ! :)

1.15 Rechtliches

Disclaimer

I, MATTHIAS FAUST, AM NOT RESPONSIBLE FOR ANY DAMAGES AT YOUR CURRENT
OR/AND FUTURE SYSTEM ENVIRONMENT, NEITHER AT YOUR HARDWARE NOR AT YOUR SOFTWARE, ←
CAUSED BY USE OR MISUSE OF THIS PROGRAM.
THIS PROGRAM/INFORMATION IS PROVIDED "AS-IS". NO WARRANTIES ARE MADE.
USE THIS PROGRAM ON YOUR OWN RISK !

Vertrieb

Ich habe mich entschlossen das Programm unter dem Freeware-Konzept zu vertreiben, ←
obwohl ich mich damit selbst degradiere. Immerhin stecken in diesem Programm
sieben Monate Arbeit drin. Ich will somit indirekt sagen, daß mich über jeden Dank, ←
Trost (für was auch immer) und Geld \$-) freu.

Lizenz

Alle Rechte liegen beim Autor bis 75 Jahre nach seinem Tode.
Der Programm-Quellcode und alle ideellen Details sind Eigentum
des Autors. Jegliche Veränderung des ausführbaren Programm-Codes ist verboten.
Die Benutzung ist nur für non-kommerzielle Zwecke erlaubt.
Veröffentlichungen auf CDROM's, PD-Serien, BBS-Systemen, in LAN's,
in WAN's oder im Internet, sind gestattet wenn folgende Regeln eingehalten werden: ←

- Vertrieb ist non-kommerziell
- Massenmedian-Anbieter schreiben mir eine Nachricht
- Vertreiber und Vertrieb sind frei von politischen Charakteren und frei von antisemitischen und antichristlichen Gedankengut
- Vertrieb erfolgt im ganzen Paket

Aminet-Betreiber dürfen das Paket auch ohne Benachrichtigung anbieten.

Nichtbeachtung wird mit aller Härte des Gesetzes verfolgt und bestraft.

Lamp.mcc/Lamp.mcp and all according files are Copyright © 1997 by Maik Schreiber/IQ Computing. ←

Updates are always available at http://home.pages.de/~bZ/projekte/mcc_lamp/

Datatypes.library ist Copyright ©1997 by Roland Mainz.

1.16 Zukunft

Was könnte in Zukunft besser werden ?

- Such-Option
- Tooltype-Option für mehr String-Speicher ?

Sinnlose Optionen, wie mehrspurige Gleise, Tempo-30-Zone oder ein Hyperlink auf die Homepage von Argue, werden niemals integriert. ↔

Hm, mehr fällt mir nicht ein.

1.17 Bekannte Probleme/Bugs/Tips

- keine Bugs bekannt

Probleme:

Die datatypes.library V45 erstellt Icons zu den internen Klassen (z.B. movie, exe etc.), in Ihrem Datatypes-Verzeichnis. Dies kann äußerst lästig werden wenn es darum geht den Typen u.a. Programm zu erkennen, es dauert dann ewig. ↔

Für Programme ist deshalb eine eigene Erkennung integriert.

1.18 Autor

Schneckenpost:

Während der Semesterferien:

Matthias Faust
Burgker Straße 6
07907 Möschlitz
Deutschland

Ansonsten:

Matthias Faust
Max-Planck-Ring 4, I108
98693 Ilmenau
Deutschland

EMail:

faust@rz.tu-ilmenau.de (sehr empfohlen)

oder:

faust@prakinf.tu-ilmenau.de (nur Notfall)

1.19 Programmgeschichte

Fehlergrade:

- 0 = Verbesserung, Neuerung oder Entfernung der Funktion
- 1 = visueller Art, leichte Störung des Programmablaufs, unnötige Vermehrung von Handgriffen bzw. Zeitverlust
- 2 = Blockierung anderer Systemclients, Mangel an Funktion, fehlerhafte Ausführung, starke visuelle Fehler
- 3 = Datenverluste, illegale Speicherzugriffe, Zwang zum vorzeitigen Neustart ↔
,
starker Verlust von Rechenzeit, starker Speicherplatzschwund, anormales Beenden des Programms, Zerstörung der Nutzeroberfläche

Version	Änderungen	Grad des Fehlers
1.0	« unveröffentlichte Version »	---
1.1	« Erstveröffentlichung »	---
1.11	- Fehler im ARexx-Teil beseitigt: Unterfenster öffneten nur auf der WB	1

1.20 Danksagungen

Danksagungen gehen an:

- Ralph Babel
gemacht hat, illegale Sachen
- für seine Hilfe und dafür, daß er mir Mut ↔
rauszulassen, auch wenn sie das Leben noch ↔
angenehmer gemacht hätten,
außerdem, für seine Ratschläge
PS: Ein GUI mußte aber sein :-)
- Michael "mlelstv" van Elst
erklären,
- für seine Geduld mir 3 Stunden lang zu ↔
wie man das Problem der WB-Verzeichnis- ↔
Abfrage,
systemkonform lösen könnte. Was aber leider ↔
nicht 100%
möglich ist, also lassen wir es am besten ↔
sein.
- Oliver "olk" Knorr
Patch,
- für seine krassen Bemerkungen zum Thema ↔

okay, es geht auch ohne :-)

- Tino "cdTv" Wildenhain
Wunsch sei mir Befehl :) - für seinen Tip mit dem Prefs-Menü (Ja, der ↔
- Arno "tarbos" Völker - für seine geduldigen Betatest.
- Thorsten "Flavour" Stocksmeier
MS Windows® - für sein Argue, dem wohl besten Tool neben ↔
- an alle anderen Leute im
#amigager
Patch-Diskussionen - für die wahrscheinlich sehr anstrengenden ↔
- Dipl.-Inf.
Matthias "DrMabuse" Lühr
Anfang zu nutzen. - für die Erlaubnis das "geniale" Zitat am ↔
- meinen Vater - als Direkt-Betatester :)=
- Gateway 2000 - als würdigen Nachfolger Escoms ;->
- Amiga "International" Technologies
Blauäugigkeit, - für die Spezial-Firma für ungemaine ↔
und den zähesten Fans die es je gegeben hat ↔
:)
Erstaunlich: Gestern pleite, heute ein ↔
multinationaler Technologie-Konzern ;) ↔
Es wäre schön, nicht nur Verlautbarungen, ↔
Audienzen und Konzilien zu geben, ↔
sondern auch mal einen konkreten Plan zur ↔
Hardwarezukunft vorzulegen ;->
- Commodore (RIP)-Entwickler
zu schreiben... - dafür daß sie mir die Chance gaben, soetwas ↔
- Leningrad Cowboys - für den besten Polkka-Punk aller Zeiten...
- Mich
dieser Anleitung - für das Schreiben der letzten Zeile in ↔
-