

**//Help/deutsch/ToolManager**

COLLABORATORS
---------------

	TITLE : //Help/deutsch/ToolManager		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		March 26, 2025	

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>//Help/deutsch/ToolManager</b>	<b>1</b>
1.1	//Help/deutsch/ToolManager.guide	1
1.2	ToolManager.guide/Copyright	1
1.3	ToolManager.guide/GiftWare	3
1.4	ToolManager.guide/Future	3
1.5	ToolManager.guide/Author	4
1.6	ToolManager.guide/Requirements	4
1.7	ToolManager.guide/Installation	5
1.8	ToolManager.guide/Concepts	5
1.9	ToolManager.guide/Exec objects	6
1.10	ToolManager.guide/Image objects	7
1.11	ToolManager.guide/Sound objects	7
1.12	ToolManager.guide/Menu objects	7
1.13	ToolManager.guide/Icon objects	8
1.14	ToolManager.guide/Dock objects	8
1.15	ToolManager.guide/Preferences	9
1.16	ToolManager.guide/MainWindow	10
1.17	ToolManager.guide/ExecWindow	11
1.18	ToolManager.guide/ImageWindow	12
1.19	ToolManager.guide/SoundWindow	12
1.20	ToolManager.guide/MenuWindow	13
1.21	ToolManager.guide/IconWindow	13
1.22	ToolManager.guide/DockWindow	14
1.23	ToolManager.guide/GroupWindow	16
1.24	ToolManager.guide/ClipWindow	16
1.25	ToolManager.guide/GlobalWindow	16
1.26	ToolManager.guide/Hotkeys	17
1.27	ToolManager.guide/InputEvent classes	18
1.28	ToolManager.guide/Qualifiers	18
1.29	ToolManager.guide/Key codes	19

---

1.30	ToolManager.guide/rawkey key codes . . . . .	20
1.31	ToolManager.guide/rawmouse key codes . . . . .	20
1.32	ToolManager.guide/Hotkey examples . . . . .	20
1.33	ToolManager.guide/Questions . . . . .	21
1.34	ToolManager.guide/History . . . . .	23
1.35	ToolManager.guide/Credits . . . . .	27
1.36	ToolManager.guide/MUI . . . . .	28
1.37	ToolManager.guide/Index . . . . .	29

# Chapter 1

## //Help/deutsch/ToolManager

### 1.1 //Help/deutsch/ToolManager.guide

ToolManager 3.0 Dokumentation

\*\*\*\*\*

Wichtige Informationen:

Copyright	Copyright und andere legale Dinge
GiftWare	Wenn Sie ToolManager mögen...
Future	Über die Zukunft von ToolManager
Author	Wohin sendet man Fehlerreports, Kommentare und Spenden?

Benutzung:

Requirements	Was wird benötigt, um ToolManager laufen zu lassen?
Installation	Wie man ToolManager installiert
Concepts	Die Konzepte hinter ToolManager
Preferences	Wie man ToolManager konfiguriert

Anhänge:

Hotkeys	Wie man Tastenbefehle definiert
Questions	Häufig gestellte Fragen
History	Entwicklungsgeschichte des ToolManagers
Credits	Der Autor möchte sich bedanken bei...
MUI	Informationen über MUI
Index	Index für dieses Dokument

### 1.2 ToolManager.guide/Copyright

Copyright und andere legale Dinge

\*\*\*\*\*

Diese Dokumentation darf kopiert und weitergegeben werden solange die Copyright-Notiz und diese Erlaubnis unverändert auf allen Kopien

---

enthalten ist.

#### COPYRIGHT

Copyright (C) 1990-1997 Stefan Becker

Kein Programm, Dokument, Datei oder Quelltext aus diesem Software-Paket darf, weder ganz noch auszugsweise, einem anderen Software-Paket beigelegt werden, außer es liegt eine schriftliche Erlaubnis des Autors vor.

#### KEINE GARANTIE

Es gibt keine Garantie für dieses Software-Paket. Obwohl der Autor versucht hat Fehler zu vermeiden, kann er keine Garantie dafür geben, daß das in dieser Dokumentation beschriebene Software-Paket 100% fehlerfrei ist. Sie benutzen dieses Material also auf eigenes Risiko. Der Autor kann nicht für irgendwelche Schäden haftbar gemacht werden, die durch die Benutzung dieses Software-Paketes entstanden sind.

#### VERTRIEB

Dieses Software-Paket kann frei verteilt werden ("freely distributable"). Es kann auf jedes Medium übertragen werden, das für den Vertrieb von freier Software benutzt wird, wie z.B. Public Domain Diskettensammlungen, CDROMs, FTP Server oder Mailboxsysteme.

Um den einwandfreien Zustand des Software-Paketes sicherzustellen, sollten die Vertreiber nur die Original-Archive verwenden:

ToolManager3\_0Binaries.lha (Dateiname im Aminet: ToolManagerBin.lha)  
ToolManager3\_0Developer.lha (Dateiname im Aminet: ToolManagerDev.lha)  
ToolManager3\_0Extras.lha (Dateiname im Aminet: ToolManagerExt.lha)  
ToolManager3\_0Locale.lha (Dateiname im Aminet: ToolManagerLoc.lha)  
ToolManager3\_0Sources.lha (Dateiname im Aminet: ToolManagerSrc.lha)

Der Autor kann nicht dafür verantwortlich gemacht werden, wenn das Software-Paket unbenutzbar geworden ist aufgrund von Veränderungen an dem Inhalt der Archive oder den Archiven selbst.

Es wird keine obere Grenze für die Vertriebskosten festgesetzt, d.h. für die Kosten von Disketten, Bändern oder CDROMs, oder dem Prozeß der Vervielfältigung. Solche Grenzen haben sich als schädlich für die Idee der frei verteilbaren Software erwiesen, z.B. wurden Software-Pakete entfernt anstatt den Preis für eine Diskette unter die Grenze zu senken.

Obwohl der Autor keine Grenzen für die Vertriebskosten festlegt, möchte er doch seine persönliche Meinung zu diesem Thema ausdrücken:

- \* Dieses Software-Paket sollte für jedermann kostenfrei zugänglich sein wann immer dies möglich ist.
  - \* Wenn Sie dieses Software-Paket unter normalen Bedingungen von einem Public Domain Händler auf einer Diskette bezogen haben und Sie haben mehr als 5 DM oder US \$5 bezahlt, dann haben Sie eindeutig zu viel gezahlt. Bitte unterstützen Sie nicht länger eine solche
-

Geschäftemacherei und suchen Sie sich so bald wie möglich ein günstigere Bezugsquelle.

#### NUTZUNGSBESCHRÄNKUNGEN

Kein Programm, Dokument, Datei oder Quelltext aus diesem Software-Paket darf, weder ganz noch auszugsweise, auf einer Maschine benutzt werden, die

- \* für die Forschung, Entwicklung, Konstruktion, Tests oder Produktion von Waffen oder anderen militärischen Anwendungen benutzt wird. Dies schließt auch solche Maschinen ein, die für die Schulung für die obengenannten Zwecke benutzt wird.
- \* von Menschen benutzt wird, die Gewalt gegen andere Menschen, z.B. Ausländer, akzeptieren, unterstützen oder selbst anwenden.

### 1.3 ToolManager.guide/GiftWare

Wenn Sie ToolManager mögen...

\*\*\*\*\*

ToolManager ist GiftWare, keine ShareWare!

Wenn Sie also dieses Programm mögen und es oft benutzen, dann sollten Sie sich überlegen, ob Sie mir eine kleine Spende zukommen lassen, um die Arbeit zu honorieren, die ich in dieses Programm gesteckt habe. Ich schlage eine Spende in der Größenordnung von US \$10-\$20 oder 10-20 DM vor. Bitte senden Sie keine Schecks oder Geldanweisungen von außerhalb Europas, denn deren Einlösung kostet meist mehr als der Betrag, der auf dem Papier steht.

Wenn Sie es sich nicht leisten können, mir eine Spende zukommen zu lassen, dann müssen Sie sich nicht schuldig fühlen. Sie sollten mir aber trotzdem eine Postkarte oder einen Brief schicken, wenn Sie ToolManager benutzen (Ich liebe es Post zu bekommen :-). Siehe Author.

### 1.4 ToolManager.guide/Future

Über die Zukunft von ToolManager

\*\*\*\*\*

Seit der Freigabe der letzten großen Version 2.1 von ToolManager im Mai 1993 hat der Amiga und seine Fangemeinde eine stürmische Zeit durchgemacht. Zum Zeitpunkt, an dem diese Zeilen geschrieben werden, sieht die Zukunft leider immer noch nicht rosig aus. Trotzdem habe ich mich entschlossen diese neue Version 3.0 des ToolManagers zu entwickeln, weil ich einen enormen Feedback von hunderten zufriedenen Benutzern erhalten habe.

Mein A3000 hat nun ein ehrwürdiges Alter erreicht und in der aktuellen Situation ist es ziemlich ungewiß, welche Richtung der Amiga nehmen wird. Ich kann es mir nicht erlauben, jedes Upgrade oder eine der (hoffentlich) neuen Maschinen zu kaufen, denn dieses Projekt ist ein reines Hobby. Es liegt also an Ihrem Feedback und Ihren Spenden, ob ich an einer zukünftigen Version von ToolManager arbeiten kann.

Dies ist auch ein Aufruf an die Firmen, die an den zukünftigen Amigas arbeiten (Amiga Technologies, Phase5, ProDAD, VisCorp oder wer auch immer momentan involviert ist). Ich bin nur ein FD Autor und kann es mir nicht erlauben jedes kommende System zu kaufen oder die Kosten für die Entwicklermaterialien für jedes System zu tragen. Also benötige ich Ihre Unterstützung, wenn ToolManager auf Ihrem System laufen soll. Bedenken Sie, daß ToolManager eines der meistgenutzten (wenn nicht sogar das meistgenutzte) Programme auf dem Amiga ist und es daher ein Bonus für Ihr System wäre.

Die Zukunft des ToolManagers hängt von IHRER Unterstützung ab!

## 1.5 ToolManager.guide/Author

Wohin sendet man Fehlerreports, Kommentare und Spenden?

\*\*\*\*\*

Der Autor kann unter den folgenden Adressen erreicht werden:

Postadresse:

Stefan Becker  
Bonner Ring 68  
D-50374 Erfstadt  
GERMANY

Electronic Mail:

stefanb@yello.ping.de

Es existiert auch eine ToolManager Homepage im World Wide Web:

<http://www.ping.de/sites/yello/toolmanager.html>

## 1.6 ToolManager.guide/Requirements

Was wird benötigt, um ToolManager laufen zu lassen?

\*\*\*\*\*

ToolManager benötigt mindestens:

AmigaOS 3.0 (V39)

oder besser für Speicher-Pools und den picture.datatype.



**WBStart 2.2**

Dieses macht es möglich, daß ToolManager Workbench-Programme starten kann.

**DOSPath 1.0**

Dieses erledigt die Behandlung von AmigaDOS Kommandopfaden für ToolManager.

Zusätzlich unterstützt er:

**ScreenNotify 1.0**

Dieses Paket ermöglicht es ToolManager seine Dock-Fenster automatisch zu öffnen oder schließen, wenn sich öffentliche Bildschirme öffnen oder schließen.

**picture.datatype V43**

Die Erweiterungen dieses verbesserten picture.datatype werden automatisch unterstützt, wenn er auf Ihrem System installiert ist. Siehe Credits.

Der Voreinsteller benötigt:

**AmigaOS 3.0 (V39)**

oder besser für Speicher-Pools.

**MUI 3.7**

Das objekt-orientierte GUI-System. Siehe MUI.

**Pophotkey, Popport, Popposition**

MUI Custom-Klassen für Popups. Siehe Credits.

## 1.7 ToolManager.guide/Installation

**Wie man ToolManager installiert**

\*\*\*\*\*

Bitte benutzen Sie das beigelegte Installer-Skript, um den ToolManager zu installieren. Dieses Skript benötigt den AmigaOS Installer 43.3. Diese Version finden Sie, z.B. auf dem Aminet.

Stoppen Sie den alten ToolManager bevor Sie die neue Version installieren!

## 1.8 ToolManager.guide/Concepts

**Die Konzepte hinter ToolManager**

\*\*\*\*\*

ToolManager ist ein Program, das es Ihnen ermöglicht Ihre Programme

---

in sehr bequemer Weise zu starten. Sie können Programme per Tastenbefehl starten (siehe Hotkeys), vom Hilfsmittelmenü der Workbench oder durch Anklicken von Piktogrammen entweder auf der Workbench oder in speziellen Dock-Fenstern. Sie können sogar Piktogramme aus Workbench-Schubladen auf diese Piktogramme ziehen, um die Programme mit Dateien zu versorgen. Zusätzlich können Sie auch einen Ton mit jeder dieser Aktionen verbinden.

Alle diese Dinge werden von ToolManager Objekten kontrolliert. Jedes Objekt hat eine spezielle Aufgabe und enthält alle Information, die zur Ausführung dieser Aufgabe nötig sind. Programme werden zum Beispiel durch Programmobjekte repräsentiert und enthalten Informationen über den Programmnamen und die Stackgröße.

Es gibt zwei Arten von Objekten. Programm-, Bild- und Tonobjekte werden als Basisobjekte bezeichnet, weil sie nur Informationen über ein Programm, ein Bild oder einen Ton enthalten.

Menü-, Piktogramm- und Dock-Objekte werden zusammengesetzte Objekte genannt, weil sie mehrere Basisobjekte für ihre Aufgabe benutzen. Ein Piktogramm auf der Workbench wird zum Beispiel durch ein Piktogrammobjekt repräsentiert, das einen Verweis auf ein Bildobjekt für die Bilddaten und einen Verweis auf ein Programmobjekt enthält. Das Programm wird gestartet, wenn man das Piktogramm aktiviert,

Exec objects	Programme
Image objects	Bilder
Sound objects	Töne
Menu objects	Einträge im Hilfsmittelmenü der Workbench
Icon objects	Piktogramme im Workbench-Fenster
Dock objects	Dock-Fenster

## 1.9 ToolManager.guide/Exec objects

Programme  
=====

Ein Programmobjekt enthält Informationen über ein Programm, das gestartet wird, wenn das Objekt aktiviert wird. Das Objekt kann direkt durch einen Tastenbefehl oder durch ein zusammengesetztes Objekt aktiviert werden. Bei der Aktivierung können Dateien angegeben werden, die an das Programm als Startparameter weitergereicht werden. Programmobjekte werden mit Hilfe des Fensters für Programmobjekte konfiguriert.

ToolManager kann verschiedene Arten von Programmen ausführen:

Shell

Das Programm wird so ausgeführt als hätte der Benutzer den Befehl in der Shell eingegeben. Shell-Skripte müssen mit diesem Modus gestartet werden. Sie können die üblichen [] Platzhalter benutzen, um anzugeben wo die Dateinamen in die Kommandozeile eingefügt werden.

---

#### Workbench

Das Programm wird wie von der Workbench gestartet. Alle Dateien werden als Workbench-Argumente übergeben. Programme, die nur in der Shell arbeiten, oder Shell-Skripte können nicht mit diesem Modus gestartet werden.

#### ARexx

Ein ARexx-Skript wird ausgeführt.

#### Dock

Das Kommando spezifiziert ein ToolManager Dock-Objekt, das aktiviert werden soll. Sie können diesen Modus benutzen, um Docks aus anderen Docks heraus zu öffnen.

#### Tastenbefehl

Ein Tastenbefehl wird erzeugt. Dieser Modus mag nützlich sein, um andere Programme mit Hilfe von ToolManager zu kontrollieren.

## 1.10 ToolManager.guide/Image objects

#### Bilder

=====

Ein Bildobjekt enthält Informationen über ein Bild, das von einem zusammengesetzten Objekt benutzt werden kann. ToolManager kann Workbench Piktogrammdateien laden, die von Piktogrammobjekten benutzt werden. Falls ein Dock-Objekt ein Bildobjekt benutzt, dann werden die Bilddaten mit Hilfe des DataTypes System geladen. Daher kann ToolManager jede Bilddatei laden für die Sie einen korrekten DataType auf Ihrem System installiert haben. Bildobjekte werden mit Hilfe des Fensters für Bildobjekte konfiguriert.

## 1.11 ToolManager.guide/Sound objects

#### Töne

=====

Ein Tonobjekt enthält Informationen über einen Ton-Befehl. Dieser Ton-Befehl wird per ARexx an ein externes Ton-Abspielprogramm geschickt. Tonobjekte werden mit Hilfe des Fensters für Tonobjekte konfiguriert.

## 1.12 ToolManager.guide/Menu objects

#### Einträge im Hilfsmittelmenü der Workbench

=====

---

Ein Menüobjekt ist ein zusammengesetztes Objekt, bestehend aus einem Programmobjekt und einem Tonobjekt. Es erzeugt einen Eintrag in dem Hilfsmittelmenü der Workbench. Wenn dieser Eintrag selektiert wird, dann werden das Programm- und das Soundobjekt aktiviert. Alle auf der Workbench angewählten Piktogramme werden als Startparameter an das Programm weitergereicht. Menüobjekte werden mit Hilfe des Fenster für Menüobjekte konfiguriert.

## 1.13 ToolManager.guide/Icon objects

Piktogramme im Workbench-Fenster

=====

Ein Piktogrammobjekt ist ein zusammengesetztes Objekt, bestehend aus einem Programmobjekt, einem Bildobjekt und einem Tonobjekt. Es erzeugt ein Piktogramm im Workbenchfenster mit der Hilfe der Bilddaten aus dem Bildobjekt. Piktogrammobjekte werden mit Hilfe des Fensters für Piktogrammobjekte konfiguriert.

Piktogramme können auf zwei Arten aktiviert werden. Sie können das Piktogramm doppelklicken oder Sie können Piktogramme auf der Workbench anwählen und auf das Piktogramm werfen. Wird das Piktogramm aktiviert, dann werden das Programm- und das Tonobjekt aktiviert. Die Piktogramme, die auf das Piktogramm geworfen wurden, werden als Startparameter an das Programm weitergereicht.

## 1.14 ToolManager.guide/Dock objects

Dock-Fenster

=====

Ein Dock-Objekt ist ein zusammengesetztes Objekt, das ein Fenster mit Reihen aus Knöpfen für den Benutzer erzeugt. Jeder Knopf besteht aus einem Programmobjekt, einem Bildobjekt und einem Tonobjekt. Jeder Knopf kann entweder einen Text, ein Bild oder beides zusammen darstellen. Dabei wird der Name des Programmobjekts als Text benutzt. Das Bildobjekt wird für das Bild benutzt. Dock-Objekte werden mit Hilfe des Fensters für Dock-Objekte konfiguriert.

Jeder Knopf kann auf zwei Arten aktiviert werden. Sie können den Knopf anklicken oder Sie können Piktogramme auf der Workbench anwählen und auf den Knopf werfen. Wird der Knopf aktiviert, dann werden das Programm- und das Tonobjekt aktiviert. Die Piktogramme, die auf den Knopf geworfen wurden, werden als Startparameter an das Programm weitergereicht.

Ein Dock-Objekt kann mit Hilfe eines Tastenbefehls "aktiviert" werden. Wenn das Dock-Fenster geschlossen ist und der Benutzer den Tastenbefehl eingibt, dann wird das Dock-Fenster geöffnet, und umgekehrt.

Falls die `screennotify.library` auf Ihrem System installiert ist, dann kann ToolManager Dock-Fenster automatisch öffnen und schließen. Jedesmal wenn ein öffentlicher Bildschirm geschlossen wird, dann werden zuerst die Dock-Fenster auf diesem Bildschirm geschlossen. Wenn der öffentliche Bildschirm wieder geöffnet wird, dann werden die Dock-Fenster für diesen Bildschirm wieder geöffnet.

## 1.15 ToolManager.guide/Preferences

Wie man ToolManager konfiguriert

\*\*\*\*\*

ToolManager kann mit Hilfe des ToolManager Voreinstellers konfiguriert werden:

MainWindow	Das Hauptfenster
------------	------------------

Fenster für die Konfiguration von Objekten:

ExecWindow	Wie man Programmobjekte konfiguriert
ImageWindow	Wie man Bildobjekte konfiguriert
SoundWindow	Wie man Tonobjekte konfiguriert
MenuWindow	Wie man Menüobjektekonfiguriert
IconWindow	Wie man Piktogrammobjekte konfiguriert
DockWindow	Wie man Dock-Objekte konfiguriert

Andere Fenster:

GroupWindow	Wie man eine Gruppe umbenennt
ClipWindow	Ablage für Objekte
GlobalWindow	Globale ToolManager Einstellungen

Der Voreinsteller versteht die standardmäßigen Workbench Tool Types und Shell Parameter:

FROM (Nur beim Aufruf aus einer Shell)

Gibt den Dateinamen an, aus der der Voreinsteller die Konfiguration laden soll.

EDIT (voreingestellte Aktion)

Die Konfiguration kann verändert werden.

USE

Die neue Konfiguration soll temporär benutzt werden.

SAVE

Die neue Konfiguration soll permanent benutzt werden.

CREATEICONS

Wird eine Konfigurationsdatei gespeichert, dann wird ein Piktogramm erzeugt. Wenn der Voreinsteller von der Workbench gestartet wird, dann ist dies die voreingestellte Verhaltensweise. Beim Aufruf aus

der Shell werden ohne diese Option keine Piktogramme erzeugt.

## 1.16 ToolManager.guide/MainWindow

### Das Hauptfenster

=====

Das Hauptfenster enthält Listen für die Objekte. Sie können den Typ der gerade angezeigten Objekte auswählen, wenn Sie den entsprechenden Typ anwählen. Jede Liste kann mehrere Gruppen enthalten. Jede Gruppe kann mehrere Objekte enthalten.

Ein Doppelklick auf den Gruppennamen öffnet ein Fenster zum Ändern des Gruppennamens. Ein Klick auf das Symbol links neben dem Gruppennamen öffnet oder schließt die Gruppe. Wenn eine Gruppe geöffnet ist, dann sind alle Objekte in dieser Gruppe sichtbar. Ein Doppelklick auf den Namen eines Objektes öffnet das Konfigurationsfenster für das Objekt.

Um Gruppen zu verschieben müssen Sie erst eine auswählen, sie zu der neuen Position verschieben während Sie den linken Mausknopf gedrückt halten und dort den Mausknopf loslassen. Auch Objekte können so zwischen Gruppen verschoben werden.

Jede Liste hat vier Knöpfe:

#### Neue Gruppe

Erzeugt eine neue Gruppe. Ein Fenster mit dem Gruppennamen öffnet sich, so daß Sie den Namen der neuen Gruppe ändern können.

#### Neues Objekt

Erzeugt ein neues Objekt in der angewählten Gruppe. Ein Konfigurationsfenster öffnet sich, so daß Sie die Einstellungen für das Objekt verändern können.

#### Entfernen

Entfernt die angewählte Gruppe oder das angewählte Objekt. Wenn eine Gruppe entfernt wird, dann werden auch alle in ihr enthaltenen Objekte entfernt.

#### Sortieren

Wenn eine Objekt oder eine offene Gruppe angewählt ist, dann wird der Inhalt der Gruppe alphabetisch sortiert. Andernfalls werden die Gruppen alphabetisch sortiert.

Mit den Knöpfen in der untersten Reihe des Fensters können Sie dem Voreinsteller mitteilen, wo die aktuelle Konfiguration gespeichert werden soll. Der Name der Konfigurationsdatei lautet ToolManager.prefs.

#### Speichern

Lege die Konfiguration in dem Verzeichnis ENVARC: ab. Die neue Konfiguration wird automatisch von ToolManager verwendet und sie wird auch den Neustart des Systems überdauern. Nach dem Speichern der Konfiguration beendet sich der Voreinsteller.

#### Benutzen

Lege die Konfiguration in dem Verzeichnis ENV: ab. Die neue Konfiguration wird automatisch von ToolManager verwendet, aber sie wird den Neustart des Systems nicht überdauern. Nach dem Speichern der Konfiguration beendet sich der Voreinsteller.

#### Test

Lege die Konfiguration in dem Verzeichnis ENV: ab. Die neue Konfiguration wird automatisch von ToolManager verwendet, aber sie wird den Neustart des Systems nicht überdauern. Der Voreinsteller beendet sich nicht.

#### Abbrechen

Der Voreinsteller wird verlassen. Alle nicht gespeicherten Änderungen gehen dabei verloren.

## 1.17 ToolManager.guide/ExecWindow

Wie man Programmobjekte konfiguriert

=====

Programmobjekte enthalten Informationen über Programme. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

#### Name

Der Name des Objektes

#### Programmtyp

Der Aufrufmodus für das Programm. Sie können zwischen den Modi Shell, Workbench, ARexx, Dock, Tastenbefehl und Netzwerk wählen. Der Modus Netzwerk wird allerdings zur Zeit nicht unterstützt.

#### Befehl

Der Name des Programmes. Dies ist abhängig vom Programmtyp entweder ein Dateiname, der Name eines Dock-Objektes oder eine Tastenbefehlsbeschreibung.

#### Tastenbefehl

Die Beschreibung für den Tastenbefehl, der dieses Programmobjekt aktiviert.

#### Stack

Die Stackgröße für das Programm. ToolManager erzwingt eine Minimalgröße von 4096 Bytes.

#### Priorität

Die Priorität für das Programm. Normalerweise sollten Sie nur den Wert 0 benutzen.

#### Argumente

Wenn diese Option aktiv ist, dann werden angewählte Dateien als Startparameter an das Programm weitergereicht. Andernfalls werden die Dateien ignoriert.

#### Nach vorne

Wenn diese Option aktiv ist, dann wird der angegebene öffentliche Bildschirm nach vorne geholt bevor das Programm gestartet wird.

#### Aktuelles Verzeichnis

Das Programm wird aus diesem Verzeichnis heraus gestartet.

#### Befehlspfad

Sie können hier eine Liste von Verzeichnissen angeben, getrennt durch Strichpunkte, die von Shell-Programmen benutzt werden, um andere Programme zu finden.

#### Ausgabedatei

Die Ausgabe von Shell-Programmen wird in diese Datei umgelenkt. Wenn Sie hier ein Konsolenfenster angeben, dann erfolgt sowohl die Ausgabe als auch die Eingabe in diesem Fenster.

#### Öffentlicher Bildschirm

Gibt den öffentlichen Bildschirm an, der vor der Ausführung des Programmes nach vorne geholt wird.

## 1.18 ToolManager.guide/ImageWindow

### Wie man Bildobjekte konfiguriert

=====

Bildobjekte enthalten Informationen über Bilder. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

#### Name

Der Name des Objektes.

#### Datei

Der Name der Datei aus der die Bilddaten gelesen werden sollen. Normalerweise müssen Sie die Endung .info entfernen, wenn Sie eine Piktogrammdatei laden wollen.

## 1.19 ToolManager.guide/SoundWindow

### Wie man Tonobjekte konfiguriert

=====

Tonobjekte enthalten Informationen über Töne. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

#### Name

Der Name des Objektes.

#### Befehl



Der ARexx-Befehl, der zu dem externen Ton-Abspielprogramm geschickt werden soll.

#### ARexx-Port

Der Name des Ports für das externe Ton-Abspielprogramm.  
Voreingestellt ist PLAY, der von dem Programm upd benutzt wird.

## 1.20 ToolManager.guide/MenuWindow

Wie man Menüobjekte konfiguriert

=====

Menüobjekte enthalten Informationen über Einträge im Hilfsmittelenü der Workbench. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

#### Name

Der Name des Objektes. Dieser wird auch für den Menüeintrag benutzt.

#### Programmobjekt

Ein Verweis auf das zu benutzende Programmobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf dieses Feld. Sie können das Objekt konfigurieren, wenn Sie auf dieses Feld klicken.

#### Tonobjekt

Ein Verweis auf das zu benutzende Tonobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf dieses Feld. Sie können das Objekt konfigurieren, wenn Sie auf dieses Feld klicken.

## 1.21 ToolManager.guide/IconWindow

Wie man Piktogrammobjekte konfiguriert

=====

Piktogrammobjekte enthalten Informationen über Piktogramme im Workbench-Fenster. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

#### Name

Der Name des Objektes.

#### Programmobjekt

Ein Verweis auf das zu benutzende Programmobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf dieses Feld. Sie können das Objekt konfigurieren, wenn Sie auf dieses Feld klicken.

---

**Bildobjekt**

Ein Verweis auf das zu benutzende Bildobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf dieses Feld. Sie können das Objekt konfigurieren, wenn Sie auf dieses Feld klicken.

**Tonobjekt**

Ein Verweis auf das zu benutzende Tonobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf dieses Feld. Sie können das Objekt konfigurieren, wenn Sie auf dieses Feld klicken.

**Position**

Gibt die Position des Piktogramms in X- und Y-Koordinaten an, z.B. würden Sie für X = 100 und Y = 55 hier 100/55 eingeben. Die Koordinaten sind immer relativ zur linken oberen Ecke des Workbench-Fensters.

**Zeige Namen**

Wenn diese Option aktiv ist, dann wird der Name des Objektes unterhalb des Piktogrammes eingeblendet.

## 1.22 ToolManager.guide/DockWindow

**Wie man Dock-Objekte konfiguriert**

=====

Dock-Objekte enthalten Informationen über Dock-Fenster. In dem Konfigurationsfenster können Sie folgende Dinge einstellen:

**Name**

Der Name des Objektes. Dieser wird auch für den Fenstertitel benutzt.

**Öffentlicher Bildschirm**

Gibt den öffentlichen Bildschirm an, auf dem das Dock-Fenster erscheint. Sie müssen hier den korrekten Namen eines öffentlichen Bildschirms eintragen, wenn Sie das automatische Öffnen und Schließen des Dock-Fensters wünschen.

**Zeichensatz**

Dieser Zeichensatz wird für die Texte der Knöpfe benutzt.

**Spalten**

Die Anzahl der Spalten für die Knopfreihe in dem Dock-Fenster. Die Spalten werden von links nach rechts mit Knöpfen gefüllt. Wenn die letzte Spalte einer Zeile gefüllt ist, dann wird eine weitere Knopfreihe angefügt. Alle Knöpfe haben die gleiche Breite und Höhe.

**Position**

Gibt die Position des Dock-Fensters in X- und Y-Koordinaten an, z.B. würden Sie für X = 150 und Y = 200 hier 150/200 eingeben. Die Koordinaten sind immer relativ zur linken oberen Ecke des

öffentlichen Bildschirms.

#### Einträge

Jeder Eintrag in dieser Liste erzeugt einen Knopf. Die linke Spalte enthält einen Verweis auf das zu benutzende Programmobjekt, die mittlere Spalte enthält einen Verweis auf das zu benutzende Bildobjekt und die rechte Spalte enthält einen Verweis auf das zu benutzende Tonobjekt. Wählen Sie ein Objekt aus dem Hauptfenster oder einer Ablage aus und werfen Sie es auf die Liste. Sie können Objekte konfigurieren, indem Sie den Namen doppelklicken. Die Liste kann per Auswahl und Verschieben von Einträgen sortiert werden. Wenn Sie den Knopf "Entfernen" drücken, dann wird der ausgewählte Eintrag gelöscht. Die in ihm enthaltenen Objekte werden jedoch nicht entfernt.

#### Aktiviert

Das Dock-Fenster wird geöffnet, wenn die Konfiguration geladen wird.

#### Backdrop

Das Dock-Fenster wird nach hinten verlegt, nachdem es geöffnet wurde.

#### Rahmen

Ist diese Option aktiv, dann sieht das Dock-Fenster wie ein normales Fenster aus, mit Rahmen und Fensterknöpfen. Andernfalls hat das Fenster überhaupt keinen Rahmen. Beachten Sie dabei, daß Sie das Dock-Fenster nur dann verschieben können, wenn es einen Rahmen hat.

#### Menü

Das Fenster erhält ein Menü. Das Menü erlaubt es Ihnen das Dock-Fenster zu schließen, den ToolManager Voreinsteller zu starten oder ToolManager zu verlassen.

#### Vorderster

Wenn diese Option aktiv ist, dann wird das Dock-Fenster immer auf dem vordersten öffentlichen Bildschirm geöffnet.

#### Pop Up

Das Dock-Fenster wird automatisch geschlossen, nachdem ein Knopf aktiviert wurde.

#### Zentriert

Das Dock-Fenster öffnet sich immer zentriert um die aktuelle Mausposition.

#### Stationär

Normalerweise erinnert sich ein Dock-Fenster an die Position, wenn es geschlossen wird. Es wird sich auf der gleichen Position wieder öffnen. Wenn diese Option aktiv ist, dann benutzt das Dock-Fenster immer die eingestellte Position.

#### Bilder

In den Knöpfen des Dock-Fensters werden Bilder dargestellt. Beachten Sie, daß Sie dafür Bildobjekte für die Einträge auswählen müssen.

---

#### Text

In den Knöpfen des Dock-Fensters wird der Name des zugehörigen Programmobjektes dargestellt. Beachten Sie, daß Sie dafür Programmobjekte für die Einträge auswählen müssen.

## 1.23 ToolManager.guide/GroupWindow

Wie man eine Gruppe umbenennt

=====

Sie können den Namen der Gruppe in dem Eingabfeld ändern.

## 1.24 ToolManager.guide/ClipWindow

Ablage für Objekte

=====

Dieses Fenster enthält eine Liste mit Verweisen auf Programmobjekte, Bildobjekte und Tonobjekte. Sie können Objekte aus dieser Liste ziehen und sie auf Konfigurationsfenster werfen. Sie können ein Objekt konfigurieren, indem Sie seinen Namen doppelklicken. Wenn Sie den Knopf "Entfernen" drücken, dann wird das ausgewählte Objekt von der Liste entfernen. Das Objekt selber wird nicht gelöscht.

Die Ablage kann vom Menü des Hauptfensters aus geöffnet werden. Es können mehrere Ablagefenster gleichzeitig geöffnet sein.

## 1.25 ToolManager.guide/GlobalWindow

Globale ToolManager Einstellungen

=====

Mit diesem Fenster können sie die globalen Einstellungen für ToolManager verändern. Es wird vom Menü des Hauptfensters aus geöffnet. Sie können die folgenden Einstellungen vornehmen:

#### Aktuelles Verzeichnis

Setzt das aktuelle Verzeichnis für den ToolManager Prozess. Alle Dateien ohne einen absoluten Pfadnamen werden relativ zu diesem Verzeichnis aus geöffnet. Voreingestellt ist die Partition, von der aus das System gestartet wurde.

#### Voreinstellungsprogramm

Pfadname für die Programmdatei des ToolManager Voreinstellers. Voreingestellt ist SYS:Prefs/ToolManager.

**Aktiviere Netzwerk**

Diese Option wird momentan nicht unterstützt.

**Aktiviere Remap**

Diese Option aktiviert den Farbanpassung im picture.datatype.  
Deaktivieren Sie diese Option nur, wenn die Bilder in  
Dock-Fenstern mit den falschen Farben dargestellt werden.

**Remap-Präzision**

Gibt die Genauigkeit der Farbanpassung an. Falls die Auswahl der  
Farben durch die Anpassung auf ihrem System nicht  
zufriedenstellend ist, dann können Sie versuchen diesen Wert zu  
ändern.

## 1.26 ToolManager.guide/Hotkeys

**Wie man Tastenbefehle definiert**

\*\*\*\*\*

Diese Kapitel beschreibt wie man einen Tastenbefehl als einen Input  
Description String definiert, der dann von Commodities ausgewertet  
werden kann. Jedes Mal, wenn ein Tastenbefehl ausgeführt wird, erzeugt  
Commodities eine Ereignis, das dann von ToolManager dazu benutzt wird  
Programmobjekte zu aktivieren oder Dock-Objekte umzuschalten. Ein  
Description String hat die folgende Syntax:

```
[<Klasse>] {[<Qualifier>]} [-][upstroke] [<Tastencode>]
```

Alle Befehlsworte können groß oder klein geschrieben werden.

Klasse beschreibt die InputEvent-Klasse. Dieser Parameter ist  
optional und falls er weggelassen wird, dann wird die Vorgabe rawkey  
benutzt. Siehe InputEvent classes.

Qualifier sind "Signale", die gesetzt oder nicht gesetzt sein  
müssen zu dem Zeitpunkt, an dem der Tastenbefehl ausgeführt wird,  
sonst wird kein Ereignis erzeugt. Für jeden Qualifier, der gesetzt  
sein soll, müssen Sie das Befehlswort angeben. Alle anderen Qualifier  
müssen dann nicht gesetzt sein. Falls Sie einen Qualifier ignorieren  
wollen, dann setzen sie ein - vor sein Befehlswort. Siehe Qualifiers.

Normalerweise wird ein Ereignis erzeugt, wenn eine Taste gedrückt  
wird. Falls das Ereignis generiert werden soll wenn die Taste  
losgelassen wird, dann müssen Sie das Befehlswort upstroke angeben.  
Wenn sowohl beim Drücken als auch beim Loslassen der Taste ein  
Ereignis erzeugt werden soll, dann müssen sie das Befehlswort  
-upstroke angeben.

Der Tastencode ist abhängig von der InputEvent-Klasse. Siehe  
Key codes.

InputEvent classes  
Qualifiers

Tastenkodes

Beispiele für Tastenbefehle

Achtung: Wählen Sie ihre Tastenbefehle sorgfältig, denn Commodities hat eine hohe Priorität in der InputEvent-Handlerkette, d.h. vorgegebene Definitionen werden übergangen.

## 1.27 ToolManager.guide/InputEvent classes

InputEvent classes

=====

Commodities unterstützt die meisten der InputEvent-Klassen, die von dem input.device erzeugt werden. Diese Sektion beschreibt die Klassen, die nützlich für ToolManager sind.

rawkey

Dies ist die vorgegebene Klasse. Sie beschreibt alle Ereignisse, die durch die Tastatur erzeugt werden können. Zum Beispiel erzeugt rawkey a oder a jedesmal ein Ereignis, wenn die Taste "a" gedrückt wird. Sie müssen einen Tastenkode für diese Klasse angeben. Siehe rawkey.

rawmouse

Diese Klasse beschreibt alle Ereignisse, die durch die Maus erzeugt werden können. Sie müssen einen Tastenkode für diese Klasse angeben. Siehe rawmouse.

diskinserted

Ereignisse dieser Klasse werden generiert, wenn eine Diskette in ein Laufwerk gelegt wird. Diese Klasse besitzt keine Tastenkodes.

diskremoved

Ereignisse dieser Klasse werden generiert, wenn eine Diskette aus einem Laufwerk genommen wird. Diese Klasse besitzt keine Tastenkodes.

## 1.28 ToolManager.guide/Qualifiers

Qualifiers

=====

Commodities unterstützt die folgenden Qualifiers:

lshift, left\_shift

Linke Shift-Taste

rshift, right\_shift

Rechte Shift-Taste

shift  
Irgendeine Shift-Taste

capslock, caps\_lock  
Caps-Lock-Taste

caps  
Irgendeine Shift-Taste oder die Caps-Lock-Taste

control, ctrl  
Control-Taste

lalt, left\_alt  
Linke Alt-Taste

ralt, right\_alt  
Rechte Alt-Taste

alt  
Irgendeine Alt-Taste

lcommand, lamiga, left\_amiga, left\_command  
Linke Amiga-/Kommando-Taste

rcommand, ramiga, right\_amiga, right\_command  
Rechte Amiga-/Kommando-Taste

numericpad, numpad, num\_pad, numeric\_pad  
Dieses Befehlswort muß angegeben werden, wenn eine Taste von der Zehnertastatur benutzt wird.

leftbutton, lbutton, left\_button  
Linke Maustaste

midbutton, mbutton, middlebutton, middle\_button  
Mittlere Maustaste

rbutton, rightbutton, right\_button  
Rechte Maustaste

repeat  
Dieser Qualifier ist gesetzt, wenn die Tastenwiederholung aktiv ist. Dies ist nur sinnvoll für die InputEvent-Klasse rawkey.

## 1.29 ToolManager.guide/Key codes

Tastenkodes  
=====

Jede InputEvent-Klasse besitzt ihre eigenen Tastenkodes:

rawkey  
rawmouse

---

## 1.30 ToolManager.guide/rawkey key codes

Tastenkodes für die InputEvent-Klasse rawkey

---

a-z, 0-9, ...  
ASCII-Zeichen

f1, f2, ..., f10, f11, f12  
Funktionstasten

up, cursor\_up, down, cursor\_down  
left, cursor\_left, right, cursor\_right  
Cursor-Tasten

esc, escape, backspace, del, help  
tab, comma, return, space, spacebar  
Spezialtasten

enter, insert, delete  
page\_up, page\_down, home, end  
Tasten der Zehnertastatur. Diese Tastenkodes müssen mit dem  
Qualifier numericpad benutzt werden!

## 1.31 ToolManager.guide/rawmouse key codes

Tastenkodes für die InputEvent-Klasse rawmouse

---

mouse\_leftpress  
Drücke die linke Maustaste.

mouse\_middlepress  
Drücke die mittlere Maustaste.

mouse\_rightpress  
Drücke die rechte Maustaste.

Achtung: Um einen dieser Tastenkodes zu benutzen, müssen sie auch  
das entsprechende Qualifier-Befehlswort angeben, z.B.

rawmouse leftbutton mouse\_leftpress

## 1.32 ToolManager.guide/Hotkey examples



## Beispiele für Tastenbefehle

=====

```
ralt t
    Rechte Alt-Taste festhalten und "t" drücken.

ralt lalt t
    Rechte und linke Alt-Taste festhalten und "t" drücken.

alt t
    Irgendeine Alt-Taste festhalten und "t" drücken.

rcommand f2
    Rechte Amiga-Taste festhalten und die zweite Funktionstaste
    drücken.

numericpad enter
    Enter-Taste auf der Zehnertastatur drücken.

rawmouse midbutton leftbutton mouse_leftpress
    Mittlere Maustaste festhalten und die linke Maustaste drücken.

diskinserted
    Eine Diskette in ein Laufwerk einlegen.
```

## 1.33 ToolManager.guide/Questions

## Häufig gestellte Fragen

\*\*\*\*\*

Hier finden Sie die Antworten zu den meistgestellten Fragen über ToolManager:

- Wenn ich den ToolManager Voreinsteller starte, dann erscheint nur ein Hinweisfenster mit dem Text "Program initialization failed". Was ist passiert?

Der Voreinsteller überprüft einige grundlegende Voraussetzungen, bevor er das erste Fenster öffnet. Bitte überprüfen Sie, daß diese Voraussetzungen erfüllt sind! Es könnte auch zu wenig Speicher auf Ihrem System vorhanden sein. Sie müssen dann zuerst einige andere Programme beenden, bevor sie den ToolManager Voreinsteller starten können. Weiterhin kann auch nur ein Voreinsteller zur gleichen Zeit aktiv sein.

- Wie kann ich den ToolManager Voreinsteller auf einem anderen öffentlichen Bildschirm als der Workbench laufen lassen?

Wählen Sie den Eintrag MUI... im Optionen-Menü des Hauptfensters aus. Nun selektieren Sie die Seite System und geben den Namen des öffentlichen Bildschirms in das Eingabefeld ein. Für weitere Details lesen Sie bitte die Dokumentation zu MUI.

- Nachdem ich meine alte ToolManager 2.x Konfiguration konvertiert habe, fehlen einige Bilder in den Dock-Fenstern und einige Dock-Fenster erscheinen garnicht mehr!

ToolManager 3.0 unterstützt in Dock-Fenster nur Bilddateien, die mit Hilfe des picture.datatypes geladen werden können. In Ihrer alten Konfiguration haben Sie den Dock-Objekten einige Bildobjekte zugeordnet, die auf Piktogrammdateien verweisen. Wenn Sie diese Dateien weiterverwenden wollen, dann müssen Sie einen Datatype für Piktogramme auf Ihrem System installieren. Sie können einen solchen Datatype z.B. auf dem Aminet finden. Sie können natürlich auch die Piktogramme in IFF Bilddateien umwandeln.

- Nachdem ich meine alte ToolManager 2.x Konfiguration konvertiert habe, fehlen einige Piktogramme im Workbench-Fenster!

ToolManager 3.0 unterstützt nur Piktogrammdateien für Piktogrammobjekte. In Ihrer alten Konfiguration haben Sie den Piktogrammobjekten einige Bildobjekte zugeordnet, die auf IFF Bilddateien verweisen. Wenn Sie diese Dateien weiterverwenden wollen, dann müssen Sie diese in Piktogrammdateien umwandeln.

- Wenn ich Piktogrammdateien in Dock-Fenstern benutze, dann erscheint zusätzlicher Text in den Bildern oder die Bilder haben einen dicken Rahmen. Was ist passiert?

Sie haben einen icon.datatype auf Ihrem System installiert, der zusätzliche Informationen aus dem Piktogramm in dem Bild anzeigt. Bitte lesen Sie in der Dokumentation des icon.datatype nach, wie man diese Zusatzinformationen abschalten kann. Wenn Sie den dicken Rahmen nicht mögen, dann müssen Sie den icon.datatype so konfigurieren, daß er den normalen Piktogrammrahmen nicht erzeugt.

- Warum kann ToolManager nicht mehrere "Hilfsmittel" Menüs oder Untermenüs erzeugen?

Mehrfache Menüs oder Untermenüs werden zur Zeit noch nicht vom Betriebssystem unterstützt. Um sie zu erzeugen, muß man sie in das System patchen, was zu einem instabilen System führen kann. Daher werde ich so etwas nicht in ToolManager einbauen.

- Wie kann ich ein horizontales Dock-Fenster erzeugen?

Setzen Sie dazu einfach die Spaltenzahl gleich der Anzahl der Einträge in dem Dock-Objekt.

- Wie kann ich ein Ausgabefenster für Shell-Programme erzeugen?

Ausgabefenster können mit Hilfe des CON: Geräts erzeugt werden. Benutzen Sie den folgenden Dateinamen, um ein sich automatisch öffnendes Fenster mit einem Close-Gadget zu erzeugen, das sich nach dem Ende des Programmes nicht automatisch schliesst:

CON:10/10/640/100/Ausgabefenster/AUTO/CLOSE/WAIT

Das Gerät CON: hat viele Optionen, die Sie dem AmigaDOS Handbuch entnehmen können.

---

- Wie kann ich die Argumente in die Mitte einer Shell/ARexx Kommandozeile einfügen?

Normalerweise werden alle Argumente an die Kommandozeile angehängt. Um sie jedoch irgendwo in der Kommandozeile einzufügen, benutzt ToolManager die gleiche [] Syntax, wie der AmigaShell-Befehl alias. Dies sieht dann z.B. so aus:

```
Dir [] all
```

Alle Argumente werden vor dem Schlüsselwort all eingefügt.

- Wie kann ich Unter-Docks erzeugen?

Sie müssen dafür Programmobjekte des Typs Dock verwenden. Setzen Sie diese Objekte in Ihr Haupt-Dock und schon können Sie andere Dock-Fenster öffnen bzw. schliessen.

- Dock-Fenster verschwinden, wenn der Workbench-Bildschirm geschlossen und wieder geöffnet wird.

Sie haben vergessen, den Namen des öffentlichen Bildschirms für das Dock-Fenster auf Workbench zu setzen. ToolManager schließt seine Dock-Fenster automatisch, wenn ein öffentlicher Bildschirm geschlossen wird. Aber er muß wissen auf welchem öffentlichen Bildschirm ein Dock-Fenster erscheinen soll, damit er es automatisch öffnen kann, wenn der öffentliche Bildschirm sich wieder öffnet.

## 1.34 ToolManager.guide/History

Entwicklungsgeschichte des ToolManagers

\*\*\*\*\*

Dieses Kapitel wurde original aus der englischen Dokumentation übernommen.

3.0, Release date 23.02.1997

- Again rewritten (almost) from scratch :-)
  - Old object system removed, TM objects are now BOOPSI objects
  - Now uses memory pools
  - Delay parameter removed from Exec Objects
  - Animation support removed from Image Objects
  - Picture.datatype V43 support added to Image Objects
  - Only icon images supported for Icon Objects
  - Only images loadable via picture.datatype are supported in
-

### Dock Objects

- Pattern & Vertical flags and Title parameter removed from Dock Objects
- Dock Objects can now display text and images
- Dock Objects can now be completely borderless
- New preferences file format, hopefully more flexible
- Converter for the ToolManager 2.x format
- Events are now checked while the configuration is read
- Preferences is now a MUI application: resizable window, multiple open edit windows and Drag&Drop support
- Changing an object name automatically updates all references to the object.
- Support for grouping objects.
- All dock objects get the screen notifications
- Added support for DOSPath 1.0
- CLI command lines are not limited to 4KB anymore
- Installer script

### 2.1b, Release date 13.03.1996

- Minor update to 2.1
- Added support for WBStart 2.0

### 2.1a, Release date 26.03.1995

- Minor update to 2.1
- Added support for ScreenNotify 1.0
- Included newer version of WBStart-Handler
- Included missing AutoDocs for toolmanager.library

### 2.1, Release date 16.05.1993, Fish Disks #872 & #873

- New Exec object types: Dock, Hot Key, Network
  - New Dock object flags: Backdrop, Sticky
  - New object type: Access
  - Network support
  - Editor main window is now an AppWindow
  - Gadget keyboard shortcuts in the preferences editor
-

- New tooltypes for the preferences editor
- Several bug fixes
- Enhanced documentation

#### 2.0, Release date 26.09.1992, Fish Disk #752

- Complete new concept (object oriented)
- (Almost) Complete rewrite
- ToolManager is now split up into two parts
- Main handler is now embedded into a shared library
- Configuration is now handled by a Preferences program
- Configuration file format has changed again :-) It is an IFF File now and resides in ENV:
- Multiple Docks and multi-column Docks
- Docks with new window design
- Dock automatically detects largest image size
- Sound support
- Direct ARexx support for Exec objects
- ToolManager can be used without the Workbench. If the Workbench isn't running, it won't use any App\* features.
- Locale support
- Path from Workbench will be used for CLI tools
- Seperate Handler Task for starting WB processes

#### 1.5, Release date 10.10.1991, Fish Disk #551

- Status Window: New/Open/Append/Save As menu items for config file
  - Edit Window: File requesters for file string gadgets
  - Added a Dock Window (a la NeXT)
  - Added a DeleteTool
  - A list of all active HotKeys can be shown
  - Tools can be moved around in the list
  - Icon positioning in the edit window added
  - Name of the program icon can be set
-

- CLI tools can have an output file and a path list
- Uses UserShell for CLI tools
- Maximum command line length for CLI tools is now 4096 Bytes
- AppIcons without a name are supported now
- Workbench screen will be moved to front if you pop up the Status window
- Workbench screen can be moved to front before starting a tool via HotKey
- TM will wait up to 20 seconds for the workbench.library
- Added a DELAY switch which causes TM to wait <num> seconds before adding any App\* stuff
- renamed some tooltypes/parameters
- some visual cues added
- some internal changes

#### 1.4, Release date 09.07.1991, Fish Disk #527

- Keyboard short cuts for tools
- AppIcons for tools
- Menu item can be switched off
- Configuration file format completely changed (hopefully the last time)
- CLI commandline parsing is now done by ReadArgs()
- Status & edit window updated to new features
- Safety check before program shutdown added
- Menu item "Open TM Window" only appears if the program icon is disabled
- WB startup method changed. Now supports project icons
- several internal changes

#### 1.3, Release date 13.03.1991, Fish Disk #476

- Now supports different configuration files
  - Format of the configuration file slightly changed
  - Tool definitions can be changed at runtime
  - Now supports CLI & Workbench startup method
-

- Selected icons are passed as parameters to the tools
- Now uses the startup icon as program icon if started from Workbench
- The position of the icon can now be supplied in the configuration file
- The program icon can now be disabled
- New menu entry "Show TM Window"
- Every new started ToolManager passes its startup parameters to the already running ToolManager process

#### 1.2, Release date 12.01.1991, Fish Disk #442

- Status window changed to a no-GZZ & simple refresh type (this should save some bytes)
- Status window remembers its last position
- New status window gadget "Save Configuration": saves the actual tool list in the configuration file
- Small bugs removed in the ListView gadget handling
- Name of the icon hard-wired to "ToolManager"

#### 1.1, Release date 01.01.1991

- Icons can be dropped on the status window
- Status window contains a list of all tool names
- Tools can be removed from the list

#### 1.0, Release date 04.11.1990

- Initial release

## 1.35 ToolManager.guide/Credits

Der Autor möchte sich bedanken bei...

\*\*\*\*\*

ToolManager hat mehrere große Entwicklungsphasen seit seiner ersten Version Mitte 1990 durchlebt. Diese Entwicklung wäre unmöglich gewesen, wenn ich nicht den enormen Feedback von einigen ToolManager Benutzern gehabt hätte. Viele Ideen und Merkmale stammen aus dieser Quelle...

Daher möchte ich den folgenden Personen danken:

Für die Alpha-/Beta-Tests, Ideen & Bug Reports:

Osma Ahvenlampi, Stephane Barbaray, Olaf Barthel, Fionn Behrens,

---

Mario Cattaneo, Michael van Elst, Michael Hohmann, Markus Illenseer, Frank Mariak, Klaus Melchior, Bernhard Moellemann, Matthias Scheler, Ralph Schmidt, Tobias Walter.

Matthew Dillon

Ohne dein exzellentes C Entwicklungssystem DICE und verschiedener anderer Hilfsprogramme würde es keinen ToolManager geben!

Alle Benutzer, die mir eine Spende geschickt haben:

Eure Unterstützung hat diese Version erst möglich gemacht!

Alle Benutzer, die mir eine Notiz zuschickt haben:

Ich habe es genossen eure Briefe und E-Mails zu lesen!

ToolManager benutzt die folgenden Pakete:

picture.datatype V43

Copyright (C) 1995-1996 Ralph Schmidt, Frank Mariak & Matthias Scheler

WBStart 2.2

Copyright (C) 1991-1996 Stefan Becker

ScreenNotify 1.0

Copyright (C) 1995 Stefan Becker

DOSPath 1.0

Copyright (C) 1996 Stefan Becker

MUI

Copyright (C) 1993-1997 Stefan Stuntz

World Wide Web Homepage: <http://www.sasg.com/>.

Popphotkey.mcc, Popport.mcc, Popposition.mcc

Copyright (C) 1996-1997 Klaus Melchior

Piktogramme

Copyright (C) 1995 Michael W. Hohmann

## 1.36 ToolManager.guide/MUI

Informationen über MUI

\*\*\*\*\*

This application uses

MUI - MagicUserInterface

(c) Copyright 1993-97 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the

---



ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz  
Eduard-Spranger-Straße 7  
80935 München  
GERMANY

Support and online registration is available at

<http://www.sasg.com/>

## 1.37 ToolManager.guide/Index

Index

\*\*\*\*\*

Adressen	Author
Autor	Author
Beispiele für Tastenbefehle	Hotkey examples
Bildobjekte	Image objects
Copyright	Copyright
Credits	Credits
DataTypes	Requirements
Diskinserted	InputEvent classes
Diskremoved	InputEvent classes
Dock-Objekte	Dock objects
DOSPath	Requirements
Einführung über Tastenbefehle	Hotkeys
EMail	Author
Erlaubnisse	Copyright
GiftWare	GiftWare
Homepage	Author
InputEvent-Klassen	InputEvent classes
Installation	Installation
Installer V43.3	Installation
Kontaktadressen	Author
Konzepte	Concepts
Legale Dinge	Copyright
Menüobjekte	Menu objects
MUI	MUI

OS 3.0	Requirements
PictDT V43	Requirements
Piktogrammobjekte	Icon objects
Pophotkey	Requirements
Popport	Requirements
Popposition	Requirements
Postadresse	Author
Programmobjekte	Exec objects
Qualifiers	Qualifiers
Rahmenbedingungen	Requirements
Rawkey	InputEvent classes
Rawmouse	InputEvent classes
ScreenNotify	Requirements
Spenden	GiftWare
Tastenbefehle	Hotkeys
Tastenkodes für rawkey	rawkey key codes
Tastenkodes für rawmouse	rawmouse key codes
Thanks	Credits
Tonobjekte	Sound objects
V39	Requirements
Verbote	Copyright
Vertrieb	Copyright
Voreinsteller	Preferences
WBStart	Requirements
Word Wide Web	Author
Zukunft	Future