

//Help/english/ToolManager

COLLABORATORS

	<i>TITLE :</i> //Help/english/ToolManager		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	//Help/english/ToolManager	1
1.1	//Help/english/ToolManager.guide	1
1.2	ToolManager.guide/Copyright	2
1.3	ToolManager.guide/GiftWare	3
1.4	ToolManager.guide/Future	4
1.5	ToolManager.guide/Author	4
1.6	ToolManager.guide/Requirements	5
1.7	ToolManager.guide/Installation	6
1.8	ToolManager.guide/Concepts	6
1.9	ToolManager.guide/Exec objects	7
1.10	ToolManager.guide/Image objects	8
1.11	ToolManager.guide/Sound objects	8
1.12	ToolManager.guide/Menu objects	8
1.13	ToolManager.guide/Icon objects	9
1.14	ToolManager.guide/Dock objects	9
1.15	ToolManager.guide/Preferences	10
1.16	ToolManager.guide/MainWindow	11
1.17	ToolManager.guide/ExecWindow	12
1.18	ToolManager.guide/ImageWindow	13
1.19	ToolManager.guide/SoundWindow	14
1.20	ToolManager.guide/MenuWindow	14
1.21	ToolManager.guide/IconWindow	15
1.22	ToolManager.guide/DockWindow	16
1.23	ToolManager.guide/GroupWindow	18
1.24	ToolManager.guide/ClipWindow	18
1.25	ToolManager.guide/GlobalWindow	18
1.26	ToolManager.guide/Hotkeys	19
1.27	ToolManager.guide/InputEvent classes	20
1.28	ToolManager.guide/Qualifiers	21
1.29	ToolManager.guide/Key codes	22

1.30	ToolManager.guide/rawkey key codes	22
1.31	ToolManager.guide/rawmouse key codes	23
1.32	ToolManager.guide/Hotkey examples	23
1.33	ToolManager.guide/Questions	23
1.34	ToolManager.guide/History	25
1.35	ToolManager.guide/Credits	30
1.36	ToolManager.guide/MUI	31
1.37	ToolManager.guide/Index	32

Chapter 1

//Help/english/ToolManager

1.1 //Help/english/ToolManager.guide

ToolManager 3.1 Documentation

Important information:

Copyright

Copyright and other legal stuff

GiftWare

If you like ToolManager...

Future

About the future of ToolManager

Author

Where to send bug reports, comments & donations

Usage:

Requirements

What is required to run ToolManager?

Installation

How to install ToolManager

Concepts

Concepts behind ToolManager

Preferences

How to configure ToolManager

Appendices:

Hotkeys

	How to define a Hotkey
Questions	Frequently asked questions
History	History of ToolManager
Credits	The author would like to thank...
MUI	Information about MUI
Index	Index for this document

1.2 ToolManager.guide/Copyright

Copyright and other legal stuff

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

COPYRIGHT

Copyright (C) 1990-1998 Stefan Becker

No program, document, data file or source code from this software package, neither in whole nor in part, may be included or used in other software packages unless it is authorized by a written permission from the author.

NO WARRANTY

There is no warranty for this software package. Although the author has tried to prevent errors he can't guarantee that the software package described in this document is 100% reliable. You are therefore using this material at your own risk. The author cannot be made responsible for any damage which is caused by using this software package.

DISTRIBUTION

This software package is freely distributable. It may be transferred to any media which is used for the distribution of free software like Public Domain disk collections, CDROMs, FTP servers or bulletin board systems.

In order to ensure the integrity of this software package distributors should use the original archive files:

ToolManager3_1Binaries.lha (file name on Aminet: ToolManagerBin.lha)
ToolManager3_1Developer.lha (file name on Aminet: ToolManagerDev.lha)
ToolManager3_1Extras.lha (file name on Aminet: ToolManagerExt.lha)
ToolManager3_1Locale.lha (file name on Aminet: ToolManagerLoc.lha)
ToolManager3_1Sources.lha (file name on Aminet: ToolManagerSrc.lha)

The author cannot be made responsible if this software package has become unusable due to modifications of the archive contents or of the archive files itself.

There is no limit on the fee taken by distributors, e.g. for the media costs of floppy disks, streamer tapes or compact discs, or the process of duplicating. Such limits have proven to be harmful to the idea of freely distributable software, e.g. the software package was removed instead of reducing the price of a floppy disk below the limit.

Although the author does not impose any limit on these fees he would like to express his personal opinions on this matter:

- * This software package should be made available to everyone free of charge whenever this is possible.
- * If you have purchased this software package under normal conditions from a Public Domain dealer on a floppy disk and have paid more than 5 DM or US \$5 then you have definitely paid too much. Please don't support this improper profit making any longer and switch to a cheaper source as soon as possible.

USAGE RESTRICTIONS

No program, document, data file or source code from this software package, neither in whole nor in part, may be used on any machine which is used

- * for the research, development, construction, testing or production of weapons or other military applications. This also includes any machine which is used in the education for any of the above mentioned purposes.
- * by people who accept, support or use violence against other people, e.g. citizens from foreign countries.

1.3 ToolManager.guide/GiftWare

If you like ToolManager...

ToolManager is GiftWare, not ShareWare!

So if you like the program and use it very often, you should consider to send a little donation or gift to honor the work that I have put into this program. I suggest a donation of US \$10-\$20 or 10-20 DM. Please don't send cheques or money orders from outside Europe, because most often cashing those items costs more than what they amount

to.

If can't afford to send a donation you don't have to feel bad about it. But you have to send me at least a postcard or letter saying that you are using ToolManager (I like to get fan mail :-). See

Author

.

1.4 ToolManager.guide/Future

About the future of ToolManager

Since the last major release of ToolManager 2.1 in May 1993 it has been a troubled time for the Amiga and its user community. At the time of this writing the future still doesn't look too optimistic. Despite of this I have decided to develop a new version of ToolManager 3.0, because of the enormous feedback I got from hundreds of satisfied users.

My trustworthy A3000 is now getting old and with the current situation it is uncertain which direction the Amiga will take. I can't afford to buy every upgrade or any of the (eventually) new machines, because this project is only my hobby. Thus it depends on your feedback and donations if I'm able to work on future versions of ToolManager.

This is also a call to the companies which are working on the future Amigas (Amiga Technologies, Phase5, ProDAD, VisCorp or whoever is currently involved). I'm only a FD author and can't afford to buy every possible system or to pay the developer material for every system. So I need your support if you wan't to see ToolManager running on your system. Remember that ToolManager is one of (if not the) most-used tools on the Amiga and therefore it will be a bonus for your system.

The future of ToolManager depends on YOUR support!

1.5 ToolManager.guide/Author

Where to send bug reports, comments & donations

The author can be reached at the following addresses:

Post address:

Stefan Becker
Bonner Ring 68
D-50374 Erfstadt
GERMANY

Electronic Mail:

stefanb@yello.ping.de

There is also a ToolManager homepage available in the World Wide Web:

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

1.6 ToolManager.guide/Requirements

What is required to run ToolManager?

ToolManager needs at least:

AmigaOS 3.0 (V39)

or better for memory pools and the picture.datatype.

WBStart 2.2

This enables ToolManager to start Workbench programs.

DOSPath 1.0

This enables ToolManager to handle AmigaDOS paths.

Additionally it supports:

ScreenNotify 1.0

This package enables ToolManager to open and close its dock windows when public screens open or close.

picture.datatype V43

The extensions of this enhanced picture.datatype are automatically supported if it is installed on your system. See Credits

.

The Preferences editor requires:

AmigaOS 3.0 (V39)

or better for memory pools.

MUI 3.7

The object-oriented GUI system. See MUI

.

Pophotkey, Popport, Popposition

MUI custom classes for popups. See Credits

.

1.7 ToolManager.guide/Installation

How to install ToolManager

Please use the supplied Installer script to install ToolManager. It requires the AmigaOS Installer V43.3. This version can be found on the Aminet.

Make sure that you quit the old ToolManager first before installing the new version!

1.8 ToolManager.guide/Concepts

Concepts behind ToolManager

ToolManager is a program which lets you start your tools in a very easy way. You can start programs by using keyboard shortcuts (see

Hotkeys

), by selecting an entry from the Workbench's Tools menu or by clicking an icon either on the Workbench or in special dock windows. You can even drag icons from Workbench drawers on those icons to supply files to the programs. Additionally you can attach a sound to each of these actions.

All these things are controlled by ToolManager objects. Each object has a specific task and contains information how to accomplish this task. F.ex. programs are represented as Exec objects and contain information about the program name and the stack size.

There are two types of objects. The object types Exec, Image and Sound are called basic objects, because they only contain information about one program, one image and one sound.

The object types Menu, Icon and Dock are called compound objects, because they bind several basic objects together to perform their tasks. F.ex. an icon on the Workbench is represented by an Icon object which has a link to an Image object for the icon imagery and a link to an Exec object which will start a program when the icon is activated.

Exec objects
Programs

Image objects
Images

Sound objects
Sounds

Menu objects
 Entries in the Workbench Tools menu

Icon objects
 Icons in the Workbench window

Dock objects
 Button windows

1.9 ToolManager.guide/Exec objects

Programs

=====

An Exec object contains information about a program which is started when the object is activated. The object can be activated directly by the user using a

 hotkey

 or through a compound object. On activation a set of files can be supplied which are forwarded to the program as startup parameters. Exec objects are configured with the

 Exec object edit window

.

ToolManager can execute different kinds of programs:

Shell

The program is executed as shell program just as the user had typed in the command line into the shell. Shell scripts have to be started in this mode. You can use the usual [] place holders to specify where the file arguments should be placed on the command line.

Workbench

A start from the Workbench is simulated. All files are supplied as Workbench arguments. Shell-only programs and shell scripts will not work when started as Workbench program.

ARexx

An ARexx script or command is executed.

Dock

The command specifies a ToolManager Dock object which should be activated. You can use this program type to create docks which are embedded in other docks.

Hotkey

A hotkey is generated. This might be used to control another program from ToolManager.

1.10 ToolManager.guide/Image objects

Images

=====

An Image object contains information about an image which is used by a compound object. ToolManager can load Workbench icons which are used by

Icon objects

. When a

Dock object

uses an Image object the data is

loaded using the DataTypes system. Thus ToolManager can use every image for which you have a valid datatype installed on your system. Image objects are configured with the

Image object edit window

.

1.11 ToolManager.guide/Sound objects

Sounds

=====

A Sound object contains information about a sound command. This sound command is sent as an ARexx command to an external sound player. Sound objects are configured with the

Sound object edit window

.

1.12 ToolManager.guide/Menu objects

Entries in the Workbench Tools menu

=====

A Menu object is a compound object which binds an

Exec object

and a

Sound object

together to create an entry in the Workbench Tools menu.

Whenever this menu entry is selected the Exec object and the Sound object are activated. Every selected icon on the Workbench is used as startup parameter for the program. Menu objects are configured with the

Menu object edit window

.

1.13 ToolManager.guide/Icon objects

Icons in the Workbench window

=====

An Icon object is a compound object which binds an
Exec object
, an

Image object
and a

Sound object

together to create an icon in the

Workbench window. The Image object is used to create the icon image.

Icon objects are configured with the

Icon object edit window

.

Icons can be activated in two ways. The user can double-click the icon or he can select icons on the Workbench and drop them on the icon. Whenever the icon is activated the Exec object and the Sound object are activated. The icons which have been dropped onto the icon are used as startup parameter for the program.

1.14 ToolManager.guide/Dock objects

Button windows

=====

A Dock object is a compound object which presents a window with
button rows to the user. Each button binds an

Exec object
, an

Image object
and a

Sound object

together. Each button can display

either a text, an image or both. The name of the Exec object is used
for the text. The Image object is used to for the image. Dock objects
are configured with the

Dock object edit window

.

Each button can be activated in two ways. The user can click on the button or he can select icons on the Workbench and drop them on the button. Whenever the icon is activated the Exec object and the Sound object are activated. The selected icons are used as startup parameter for the program.

A Dock object can be "activated" with a
hotkey

. When the dock window is closed and the user enters the hotkey then the dock window is opened and vice versa.

If the `screennotify`.library is installed then ToolManager can open and close dock windows automatically. Every time a screen is going to be closed all dock windows on this screen are closed first. When the public screen opens again all dock windows for this screen are opened again.

1.15 ToolManager.guide/Preferences

How to configure ToolManager

The ToolManager preferences editor is used to configure ToolManager.

MainWindow

The main window

Object edit windows:

ExecWindow

How to configure Exec objects

ImageWindow

How to configure Image objects

SoundWindow

How to configure Sound objects

MenuWindow

How to configure Menu objects

IconWindow

How to configure Icon objects

DockWindow

How to configure Dock objects

Miscellaneous windows:

GroupWindow

How to rename an object group

ClipWindow

Clipboard for objects

GlobalWindow

Global ToolManager options

The preferences editor understands the standard Workbench tool types and Shell command parameters:

FROM (Shell only)

Specifies the file name from which the editor should load the configuration.

EDIT (default)

Edit the configuration.

USE

Use the specified configuration temporarily.

SAVE

Use the specified configuration permanently.

CREATEICONS

Create icons for the configuration files when they are saved. When started from Workbench the preferences editor creates icons by default. When started from the shell no icons are created by default.

1.16 ToolManager.guide/MainWindow

The main window

=====

The main window contains the object lists. By clicking on the object type you can select which list is currently visible. Each list can contain several groups. Each group can hold several objects.

A double-click on the name of a group opens the
group edit window

.

Clicking on the symbol left to the group name opens and closes the group. If a group is open you can see all the objects in this group. A double-click on the name of an object opens the object edit window.

To move a group you first select one, drag it to the new position while holding the left mouse button and then release the mouse button. You can also move objects between groups with this method.

Attached to the list are four buttons:

New Group

Creates a new empty group. The
group edit window
will open so that
you can set the name of the new group.

New Object

Create a new object in the selected group. The object edit window will open so that you can edit the properties of the new object.

Delete

Deletes the selected group or object. If a group is selected also all objects in this group will be deleted.

Sort

If an object or an open group is selected then the contents of this group are sorted alphabetically. Otherwise the groups are sorted alphabetically.

With the buttons at the bottom of the main window you can tell the preferences editor where to store the configuration. The name of the configuration file is ToolManager.prefs.

Save

Store the configuration to ENVARC: and ENV:. The new configuration will be taken into use automatically and survive a reboot. After storing the preferences editor exits.

Use

Store the configuration to ENV:. The new configuration will be taken into use automatically but it will not survive a reboot. After storing the preferences editor exits.

Test

Store the configuration to ENV:. The new configuration will be taken into use automatically but it will not survive a reboot. The preferences editor does not exit.

Cancel

The preferences editor exits. All changes which have not been saved will be discarded.

1.17 ToolManager.guide/ExecWindow

How to configure Exec objects

=====

Exec objects

contain information about programs. The edit window has the following gadgets:

Name

Name of the object.

Exec Type

Type of the program. You can choose between Shell, Workbench, ARexx, Dock, Hotkey and Network. The type Network is currently not supported.

Command

The name of the program. This is either the file name, the name of a dock objects or a hotkey description depending on the Exec Type.

Hotkey

A

hotkey

description string which activates this Exec object.

Stack

The stack size for the program. ToolManager will enforce a minimum size of 4096 bytes.

Priority

The priority for the program. Usually you should only use the default value 0.

Arguments

If this gadget is selected then files are forwarded to the program as startup arguments. Otherwise the files are ignored.

To Front

If this gadget is selected then the specified public screen is moved to front before starting the program.

Current Directory

The program is started from this directory.

Path

You can here supply a list of directories (separated with semicolons) which is used by shell programs to search for other programs.

Output File

The output of shell programs is redirected to this file. If you specify a console window here then the output and the input of the shell program is redirected to this window.

Public Screen

Specifies the public screen which will be moved to the front before starting the program.

1.18 ToolManager.guide/ImageWindow

How to configure Image objects

=====

Image objects

contain information about images. The edit window has the following gadgets:

Name

Name of the object.

File

The name of the file from which the image data should be loaded. Usually you have to remove the ending .info if you want to load an icon file.

1.19 ToolManager.guide/SoundWindow

How to configure Sound objects

=====

Sound objects

contain information about sounds. The edit window has the following gadgets:

Name

Name of the object.

Command

The ARexx command which should be send to the external sound player.

ARexx Port

The ARexx port name of the external sound player. The default is PLAY which is used by the program upd.

1.20 ToolManager.guide/MenuWindow

How to configure Menu objects

=====

Menu objects

contain information about entries in the Workbench Tools menu. The edit window has the following gadgets:

Name

Name of the object. This is also used to create the menu entry.

Exec Object

Link to the attached
Exec object
. Use Drag&Drop from the
main window
or a

clipboard
to attach an object. You can edit the
attached object by clicking on it.

Sound Object

Link to the attached
Sound object
. Use Drag&Drop from the

main window
or a
clipboard
to attach an object. You can edit the
attached object by clicking on it.

1.21 ToolManager.guide/IconWindow

How to configure Icon objects

=====

Icon objects
contain information about icons in the Workbench window.

The edit window has the following gadgets:

Name

Name of the object.

Exec Object

Link to the attached
Exec object
. Use Drag&Drop from the

main window
or a
clipboard
to attach an object. You can edit the
attached object by clicking on it.

Image Object

Link to the attached
Image object
. Use Drag&Drop from the

main window
or a
clipboard
to attach an object. You can edit the
attached object by clicking on it.

Sound Object

Link to the attached
Sound object

. Use Drag&Drop from the
main window
or a
clipboard
to attach an object. You can edit the
attached object by clicking on it.

Position

Specifies the X and Y coordinates of the icon, e.g. for X = 100 and Y = 55 you would enter 100/55. The coordinates are relative to the top left corner of the Workbench window.

Show Name

If this gadget is selected then the name of the object is shown below the icon.

1.22 ToolManager.guide/DockWindow

How to configure Dock objects

=====

Dock objects

contain information about dock windows. The edit window has the following gadgets:

Name

Name of the object. This also used as window title.

Hotkey

A

hotkey

description string which open and closes the dock window.

Public Screen

Specifies the public screen on which the dock window appears. You must specify a valid public screen name if you want to use the automatic open and close feature for dock windows.

Font

Use this font for the button texts.

Columns

Number of button columns in the dock window. Columns will be filled with buttons from left to right. If the last column in a row is filled then a new button row is added. All buttons will have the same width and height.

Position

Specifies the X and Y coordinates of the dock window, e.g. for X = 150 and Y = 200 you would enter 150/200. The coordinates are relative to the top left corner of the screen.

Entries

Each entry in this list creates one button. The left column contains the link to the attached Exec object, the middle column the link to the attached image object and the right column the link to the attached sound object. Use Drag&Drop from the main window or a clipboard to attach an object. You can edit the attached objects by double-clicking on them. You can use Drag&Drop to sort the entries in the list. When you press the Delete button the currently selected entry is removed from the list. The attached objects itself are not deleted.

Activated

The dock window will be opened when the configuration is loaded.

Backdrop

The dock window is moved to the back after it has been opened.

Border

When this gadget is selected the dock window looks like a normal window with a border and window gadgets. Otherwise it will have no border at all. Note that you can move the dock window only if it has a border.

Menu

A menu is attached to the dock window. The menu allows you to close the dock window, start the ToolManager preferences editor or to quit ToolManager.

Frontmost

When this gadget is selected then the dock window will always open on the frontmost public screen.

Pop Up

The dock window closes automatically after a button has been selected.

Centered

The dock window opens centered around the current mouse position.

Sticky

Usually a dock window remembers its position when you close it. It will open on this position if you open it again. If this gadget is selected then the dock window will always open at the same position.

Images

The buttons in the dock window will display images. Note that you have to attach

Image objects
to the dock entries in this case.

Text

The buttons in the dock window will display the name of the attached

Exec object
. Note that you have to attach Exec objects to
the dock entries in this case.

1.23 ToolManager.guide/GroupWindow

How to rename an object group

=====

You can change the name of the group with the name string gadget.

1.24 ToolManager.guide/ClipWindow

Clipboard for objects

=====

This window contains a list with attached

Exec objects

,

Image objects

and

Sound objects

. You can drag objects from this list

and drop them on edit windows. You can edit the attached objects by double-clicking them. When you press the Delete button the currently selected object is removed from the list. The object itself is not deleted.

The clipboard can be opened from the menu in the main window. You can have several clipboards open at once.

1.25 ToolManager.guide/GlobalWindow

Global ToolManager options

=====

This window lets you change the global options of ToolManager. It can be opened from the menu in the main window. It has the following gadgets:

Current Directory

Set the current directory for the ToolManager process. All files without an absolute path name will be opened relative to this directory. The default directory is the boot volume.

Preferences editor

Path of the ToolManager preferences editor binary. The default is SYS:Prefs/ToolManager.

Enable Network

Currently not supported.

Enable Remap

Enable colour remapping for picture.datatype. Disable this only if you the images in the dock windows come up with the wrong colours.

Remap Precision

Sets the precision of the colour remapping. You might try change this value if the colour choices of the remap algorithm are unsatisfactory on your system.

1.26 ToolManager.guide/Hotkeys

How to define a Hotkey

This chapter describes how to define a Hotkey as an Input Description String, which is then parsed by Commodities. Each time a Hotkey is activated Commodities generates an event which is used by ToolManager to activate Exec objects or to toggle Dock objects. A description string has the following syntax:

[<class>] {[-][<qualifier>]} [-][upstroke] [<key code>]

All keywords are case insensitive.

class describes the InputEvent class. This parameter is optional and if it is missing the default rawkey is used. See InputEvent classes

.

Qualifiers are "signals" that must be set or cleared by the time the Hotkey is activated; otherwise no event will be generated. For each

qualifier that must be set you supply its keyword. All other qualifiers are expected to be cleared by default. If you want to ignore a qualifier, just set a - before its keyword. See
Qualifiers
.

A Hotkey event is usually generated when a key is pressed. If the event should be generated when the key is released, supply the keyword upstroke. When both press and release of the key should generate an event, use -upstroke.

The key code is depending on the InputEvent class. See
Key codes
.

InputEvent classes

Qualifiers

Key codes

Hotkey examples

Note: Choose your Hotkeys carefully, because Commodities has a high priority in the InputEvent handler chain, i.e. it will override existing definitions. ←

1.27 ToolManager.guide/InputEvent classes

InputEvent classes

=====
Commodities supports most of the InputEvent classes that are generated by the input.device. This section describes those classes that are most useful for ToolManager Hotkeys.

rawkey

This is the default class and covers all keyboard events. For example rawkey a or a creates an event every time when the key "a" is pressed. You must specify a key code for this class. See

rawkey
.

rawmouse

This class describes all mouse button events. You must specify a key code for this class. See

rawmouse
.

diskinserted

Events of this class are generated when a disk is inserted in a drive. This class has no key codes.

diskremoved

Events of this class are generated when a disk is removed from a drive. This class has no key codes.

1.28 ToolManager.guide/Qualifiers

Qualifiers

=====

Commodities supports the following qualifiers:

lshift, left_shift
Left shift key

rshift, right_shift
Right shift key

shift
Either shift key

capslock, caps_lock
Caps lock key

caps
Either shift key or caps lock key

control, ctrl
Control key

lalt, left_alt
Left alt key

ralt, right_alt
Right alt key

alt
Either alt key

lcommand, lamiga, left_amiga, left_command
Left Amiga/Command key

rcommand, ramiga, right_amiga, right_command
Right Amiga/Command key

numericpad, numpad, num_pad, numeric_pad
This keyword must be used for any key on the numeric pad.

leftbutton, lbutton, left_button
Left mouse button

midbutton, mbutton, middlebutton, middle_button
Middle mouse button

rbutton, rightbutton, right_button
Right mouse button

repeat
This qualifier is set when the keyboard repeat is active. This is only useful for the InputEvent class rawkey.

1.29 ToolManager.guide/Key codes

Key codes

=====

Each InputEvent class has its own key codes:

rawkey

rawmouse

1.30 ToolManager.guide/rawkey key codes

Key codes for InputEvent class rawkey

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

f1, f2, ..., f10, f11, f12
Function keys

up, cursor_up, down, cursor_down
left, cursor_left, right, cursor_right
Cursor keys

esc, escape, backspace, del, help
tab, comma, return, space, spacebar
Special keys

enter, insert, delete
page_up, page_down, home, end
Numeric Pad keys. Each of these key codes must be used with the numericpad qualifier keyword!

1.31 ToolManager.guide/rawmouse key codes

Key codes for InputEvent class rawmouse

```
-----  
mouse_leftpress  
    Press left mouse button  
  
mouse_middlepress  
    Press middle mouse button  
  
mouse_rightpress  
    Press right mouse button
```

Note: To use one of these key codes, you must also set the corresponding qualifier keyword, e.g.

```
rawmouse leftbutton mouse_leftpress
```

1.32 ToolManager.guide/Hotkey examples

Examples for Hotkeys

```
=====
```

```
ralt t  
    Hold right Alt key and press "t"  
  
ralt lalt t  
    Hold left and right Alt key and press "t"  
  
alt t  
    Hold either Alt key and press "t"  
  
rcommand f2  
    Hold right Amiga key and press the second function key  
  
numericpad enter  
    Press the Enter key on the numeric pad  
  
rawmouse midbutton leftbutton mouse_leftpress  
    Hold middle mouse button and press the the left mouse button  
  
diskinserted  
    Insert a disk in any drive
```

1.33 ToolManager.guide/Questions

Frequently asked questions

```
*****
```

Here are the answers to the most asked questions about ToolManager:

- When I start the ToolManager preferences editor only a requester with the text "Program initialization failed" appears. What's wrong?

The preferences editor checks the basic requirements before opening the first window. Please check that these requirements are

fulfilled! Your system might also be running out of memory. You may have to quit some other applications first before enough memory is available to run the ToolManager preferences editor. Also there can be only one preferences editor running at one time.

- How can I run the ToolManager preferences editor on another public screen than the Workbench screen?

Select the entry MUI... from the settings menu in the main window

menu. Now select the System settings page and enter the name of the public screen into the string gadget. For further details please consult the MUI documentation.

- After converting my old ToolManager 2.x configuration some pictures are missing in the dock windows and some dock windows don't appear at all!

ToolManager 3.1 only supports picture files in dock windows which can be accessed via the picture.datatype. In your old configuration you have attached some Image objects to the Dock objects which refer to icon files. In order to use icon files you have to install an icon datatype on your system. You may find such a datatype e.g. on the Aminet. You can also convert them to e.g. IFF Brushes.

- After converting my old ToolManager 2.x configuration some of the icons in the Workbench window are missing!

ToolManager 3.1 only supports icons files for Icon objects. In your old configuration you have attached some Image objects to the Icon objects which refer to IFF Brushes. You have to convert them to icon files in order to use them with ToolManager 3.1 Icon objects.

- When I use icon files for images in the dock windows then there are some additional texts attached to the image or it has a thick border. What's wrong?

You have installed an icon.datatype on your system which inserts additional information from the icon into the image. Please check the documentation of the icon.datatype how this information can be suppressed. If you don't like the thick border you also have to tell the icon.datatype not to generate the usual icon borders.

- Why can't ToolManager create multiple "Tools" menus or sub-menus?

Multiple menus or sub-menus are currently not supported by the Workbench. To create them, you have to hack them into the AmigaOS, which can result in an unstable system. Therefore I won't implement it in ToolManager.

- How can I create a horizontal dock window?

Just set the number of columns to the number of entries in the dock object.

- How can I create an output window for shell programs?

Output windows can be created by using the CON: device. Use the following file name to create an auto-open window with a close gadget which doesn't close after the program has quit:

```
CON:10/10/640/100/Output-Window/AUTO/CLOSE/WAIT
```

The CON: device has many options, please consult your AmigaDOS manual for further information.

- How can I put the arguments in the middle of a Shell/Arexx command line?

Usually all arguments are appended to the command line. To insert the arguments anywhere in the command line, ToolManager uses the same [] syntax, which is used by the AmigaShell command alias. So for example

```
Dir [] all
```

will insert all arguments before the keyword all.

- How can I create sub-docks?

You must use Exec objects of the type Dock. Put such objects in the entries of your main dock and they will open/close the other docks.

- The dock windows disappear when the Workbench screen is closed and opened again.

You have forgotten to set the public screen name for the dock window to Workbench. ToolManager will close dock windows when the public screen closes. But it has to know on which public screen the dock windows should appear in order to open them automatically when the public screen opens again.

1.34 ToolManager.guide/History

History of ToolManager

3.1, Release date 01.06.1998

- Clipboard list also shows the object type now
- Only requires locale.library V38 now
- Icon drop corrected for dock windows with borders
- The __geta4 qualifier was missing for some BOOPSI dispatchers. This might correct some strange bugs which were reported?
- Installer script now checks for required libraries

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
- Separate 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

The author would like to thank...

ToolManager has gone through many major evolutionary phases since its first implementation in mid-1990. This development would have been impossible if I hadn't received the enormous feedback from various ToolManager users. Many ideas & features resulted from this source...

Therefore I would like to thank:

For Alpha/Beta testing, ideas & 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.

For the translations:

The Amiga Translators Organization (ATO). Please check the file Readme.Locale for the list of translators.

Matthew Dillon

Without your excellent C development system DICE and various other tools, ToolManager wouldn't exist!

All users who sent me money:

Your support made this release possible.

All users who sent me a note:

I really enjoyed reading your letters and E-Mails!

ToolManager uses the following packages:

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

Pophotkey.mcc, Popport.mcc, Popposition.mcc

Copyright (C) 1996-1997 Klaus Melchior

Icons

Copyright (C) 1995 Michael W. Hohmann

1.36 ToolManager.guide/MUI

Information about 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	
Addresses	Author
Author	Author
Concepts	Concepts
Contact addresses	Author
Copyright	Copyright
Credits	Credits
DataTypes	Requirements
Diskinserted	InputEvent classes
Diskremoved	InputEvent classes
Distribution	Copyright
Dock objects	Dock objects
Donations	GiftWare
DOSPath	Requirements
EMail	Author
Examples for Hotkeys	Hotkey examples

Exec objects	Exec objects
Future	Future
GiftWare	GiftWare
Homepage	Author
Hotkeys	Hotkeys
Icon objects	Icon objects
Image objects	Image objects
InputEvent classes	InputEvent classes
Installation	Installation
Installer V43.3	Installation
Introduction to Hotkeys	Hotkeys
Key codes for rawkey	rawkey key codes
Key codes for rawmouse	rawmouse key codes
Legal stuff	Copyright
Menu objects	Menu objects
MUI	MUI
OS 3.0	Requirements
Permissions	Copyright
PictDT V43	Requirements

Pophotkey	Requirements
Popport	Requirements
Popposition	Requirements
Post address	Author
Preferences	Preferences
Prohibitions	Copyright
Qualifiers	Qualifiers
Rawkey	InputEvent classes
Rawmouse	InputEvent classes
Requirements	Requirements
ScreenNotify	Requirements
Sound objects	Sound objects
Thanks	Credits
V39	Requirements
WBStart	Requirements
Word Wide Web	Author
