

# **director**

Philip Gruebele

**COLLABORATORS**

	<i>TITLE :</i> director		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Philip Gruebele	March 1, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>director</b>	<b>1</b>
1.1	Director Help . . . . .	1
1.2	Legal Notes . . . . .	2
1.3	Introduction to Director . . . . .	2
1.4	Installing . . . . .	3
1.5	Getting Started . . . . .	3
1.6	Tool Types . . . . .	4
1.7	Rexx Commands . . . . .	4
1.8	Main Menu . . . . .	7
1.9	Main Window . . . . .	7
1.10	Macro List . . . . .	8
1.11	Macro Name . . . . .	8
1.12	Play Button . . . . .	8
1.13	Record Button . . . . .	9
1.14	Stop Record Button . . . . .	9
1.15	Play Options Button . . . . .	9
1.16	Record Options Button . . . . .	9
1.17	Registering . . . . .	12
1.18	Reporting Problems . . . . .	13
1.19	Known Problems and Limitation . . . . .	13
1.20	Revision History . . . . .	14

---

# Chapter 1

## director

### 1.1 Director Help

Director 1.6  
Shareware Version  
Copyright © 1996 by Philip A. Gruebele  
All rights reserved.

#### Table of Contents

Legal Notes  
Introduction  
Installing  
Getting Started  
Main Window  
Menus  
Tool Types  
Rexx Commands  
Registering  
Known Limitations and Problems  
Problem Reporting  
Revision History

---

## 1.2 Legal Notes

This is a shareware version of Director. It is fully functional, but if it's not registered, it opens up some irritating reminders to register. This software may be distributed freely. At the end of the trial period of 1 month, you are required to stop using this software if you do not register. If you do register you will receive a key file which makes the irritating requestors go away. This key file will be licensed to you only and may not be redistributed in any form. All of Director is copyrighted and is my property.

The key file simply gives you a single user license to use Director.

To register this software, please see

Registering

. To report problems, please

see

reporting problems

. In using this software, you accept all responsibility

for any damage or loss of productivity or money that may occur through or during its use.

Philip A. Gruebele can in no way be held responsible for any losses.

Other software included in this distribution has its own copyright:

AmigaGuide, AmigaGuide.info, amigaguide.library

(c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.

Reproduced and distributed under license from Commodore.

AMIGAGUIDE SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE;

NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY

OR RESPONSIBILITY IS ASSUMED.

## 1.3 Introduction to Director

Director is the input recorder for the amiga. It allows the user to record all mouse and keyboard actions performed by the user into a macro. This macro can then be played back at any time and any number of times, as fast as the computer can handle it.

Director has some intelligence built into it. Macros are played back to the window they were recorded in. Upon playback, the Director finds the window in question, and plays the macro relative to that window, regardless of its position. Similarly, if an icon or button is selected during the macro recording, then upon playback the Director will find the current position of the icon or button and select it as would be expected.

Director has many uses. The simplest use is to create macros for

actions frequently repeated by the user. For example a macro could be placed in the WBStartup drawer in order to open certain windows or start applications at boot time.

A macro can also be used to record demos of software or hardware at exhibitions. I have also found Director to be useful in doing software testing.

Some of the more advanced features are not available in this unregistered shareware version. For example the interactive requesters are not available for making interactive demos. Please see Registering for further limitations.

Since macros are really AREXX scripts which send commands to Director, the user can modify the scripts to add any logic or add and delete parts of the macro. Macros can also be executed from regular AREXX programs to give extended control over an AREXX capable application.

## 1.4 Installing

To install Director, simply double click on the install icon distributed with this package. Note that Director has to be started at system boot time, so if you want to try Director without installing it, please place it in the WBStartup drawer and reboot.

## 1.5 Getting Started

The Director window can be opened by pressing the hot-key 'ctrl-alt-m'.

This opens the main control window. It contains a list of macros and buttons for starting and stopping a recording, playing back a recording, and setting playback and record settings. Context sensitive help is available for any button or menu by pressing the HELP key while the mouse is over any button or menu item. This feature is available under V39 or above of the AmigaDOS operating system.

The first time you run this software, there will not be any macros in the

### Macro List

. You have to enter a new macro name in the Macro Name button. Then, you may press the Record Button

to start recording

into that macro. The recording will be suspended until you select a window other than the Director window. As soon as you select another window, all mouse and keyboard input is recorded. You may then go back to the Director window and press the

Stop Record Button

to stop the recording. To play back the macro, simply press the Play Button . You may also change the playback speed by pressing the Play Options Button . For the usage of any other buttons, please refer to the documentation for it.

## 1.6 Tool Types

The Director icon contains the following tool types:

- o LEFT - the left edge of the window as saved by the settings menu
- o TOP - the top edge of the window as saved by the settings menu
- o CX\_POPUP - YES to open the window on startup, NO to hide it
- o DONOTWAIT - should always be there
- o FONT - the font to use for the GUI (ex: topaz.font)
- o FONTSIZE - the size for the font to use
- o EDITREQ - the command line to open the editor for requesters (should not return ← until done)
- o DIR - the directory to use to store macros
- o EDIT - the command line to open the editor asynchronously
- o CX\_POPKEY - the key to pop up the main window (default is 'ctrl alt m')
- o REC\_POPKEY - the key to pop up the record control window (default is 'ctrl alt r ←')
- o FRONTSCREEN - YES to open the window on the front most public screen, NO to open ← on Workbench

## 1.7 Rexx Commands

Director is a rexx host. It generates its macros in the Rexx language. When these rexx programs are played back, they send commands to Director. These commands can move the mouse, click mouse buttons, etc. I designed these commands so that they are humanly readable (for the most part). This allows a Rexx programmer to modify macros manually. Some commands take rather cryptic parameters. Letting Director generate these commands can be much easier than generating them by hand. Most commands take a time stamp as the first parameter. This is the time since the beginning of the macro when the given command should complete by. The values are in 1/10's seconds. To obtain a better understanding of commands, you may look at macros generated by Director. The following is a list of the commands supported:

'|' is used to delimit a list of choices.  
 '[' ]' is used to show that the argument is optional  
 'timestamp' is the time stamp as discussed above.  
 'string' is an arbitrary sequence of keyboard input.

BUTT

---

Purpose: Change the state of a mouse button  
Usage: BUTT timestamp L|M|R U|D  
L,M, and R stand for left, middle and right buttons  
U and D stand for up or down.

## CHAR

Purpose: To enter character keyboard input  
Usage: CHAR string

## RAW

Purpose: To enter arbitrary keyboard input  
Usage: Because it's complex, just let Director. generate it.

## MOVEI

Purpose: Move mouse to a workbench icon  
Usage: MOVEI timestamp iconname

## MOVEG

Purpose: Move mouse to a gadget  
Usage: Because it's complex, just let Director. generate it.

## MOVEM

Purpose: Move mouse to a menu item  
Usage: Because it's complex, just let Director. generate it.

## START

Purpose: Tell Director that a macro is starting (block user input etc)  
Usage: START

## STOP

Purpose: Tell Director that a macro ending  
Usage: STOP

## MOVE

Purpose: Move the mouse to a window/screen relative position.  
Usage: MOVE timestamp x y  
x is the offset from the left of the screen/window  
y is the offset from the top of the screen/window  
  
If no window is selected, x/y are relative to the screen ( ↔  
see NOWINDOW).

## MOVEREL

Purpose: Move the mouse to a mouse relative position.  
Usage: MOVEREL timestamp x y  
x and y can be positive or negative.

## SPEED



---

Purpose: Sets the playback speed of the macro.  
Usage: SPEED n  
n is a value from 1 to 30, where 1 is "recorded" speed and ↵  
30 is maximum speed.

NOWINDOW

Purpose: Make all further actions screen relative. To perform further ↵  
window commands (MOVEI, MOVEG etc.), a new window has to be selected (WIN)  
Usage: NOWINDOW

NOSCREEN

Purpose: Unselect the current screen. To perform further mouse commands, a ↵  
screen needs to be selected (SCRN)  
Usage: NOSCREEN

REQ

Purpose: Display a requester to the user  
Usage: REQ T|Q|W|I string  
T will create a text requestor with a "Continue" button.  
Q will create a text requestor with "Yes" and "NO" buttons ↵  
. RESULT will contain YES or NO.  
W will create a text requestor with no buttons. It will ↵  
close automatically when the current window is closed.  
I will create a text requestor with no buttons. It will ↵  
close automatically when the use presses the return key ↵  
. string is the string to be displayed in the requestor.

STORE

Purpose: Store the current mouse position.  
Usage: STORE

RECALL

Purpose: Restore the previously stored mouse position  
Usage: RESTORE

SCRN

Purpose: Select a new screen.  
Usage: SCRN timestamp title special  
title is the title of the screen  
special are some Director generated flags.

WIN

Purpose: Select a new window.  
Usage: Because it's complex, just let Director. generate it.

QUIT

Purpose: Quit Director.

---

Usage:           QUIT

CLOSEWINDOW

Purpose:           Wait until currently selected window closes

Usage:           CLOSEWINDOW

## 1.8 Main Menu

The main window has the following menus:

### Project

- o Edit - open the text editor with the selected macro
- o Help - open the Amigaguide help window
- o About - display copyright and version information
- o Hide - hide the main window
- o Quit - quit Director

### Settings

- o Save As Defaults - save the window position as tool types

## 1.9 Main Window

The main window contains the controls to select macros and record ↔  
or play them.

It also allows the setting of playback and record options.

The following is a list of all the controls available:

Macro List

Macro Name

Play Button

Record Button

Stop Record Button

Play Options Button

Record Options Button

Those controls which the user should not use at a given time will ↔  
be fogged out.

## 1.10 Macro List

The macro list contains the list of available macros. ↔  
 This is simply the  
 files in the s: directory that match the '#?.mac' file pattern. Names may  
 be added through the  
 Macro Name  
 string gadgets. The  
 Play Button  
 ,  
 Record Button  
 , and  
 Play Options Button  
 affect the selected macro from this list.

## 1.11 Macro Name

The macro name gadget displays the name of the currently ↔  
 selected macro.  
 The name may also be changed in order to add a new macro to the  
 Macro List  
 .  
 Please note that the name will always have the '.mac' extension in order for  
 the macro to be found and placed in the list the next time Director is started.

## 1.12 Play Button

The play button will start to play the currently selected ↔  
 macro file. The  
 Main Window  
 window will be hidden during playback. Also, all user input is  
 blocked, since it would interfere with playback. However, the mouse buttons may  
 be used to control the playback. The right mouse button will abort playback.  
 The left mouse button will speed up playback (if possible).  
 When the macro is finished, the  
 Main Window  
 will once again  
 appear and become active. If the macro ever appears to stop playing, but mouse  
 control is not given back to the computer, it usually means that the macro is ↔  
 waiting  
 for a window to be identified. You may regain control of your computer by either  
 waiting until the macro resumes, or by clicking the right mouse button to cancel  
 the macro.

## 1.13 Record Button

The record button will start to record into the currently selected macro file. ↔

Any input that does not go to a Director window is recorded. This means as long as a Director window is active, nothing is recorded. This is so since the user would normally not want to record what is being done to the Director windows, but rather what is being done to some other application. Also, while recording, the

Record Options Button  
can be

used to control some record options.

## 1.14 Stop Record Button

The stop record button will stop the recording. This button is disabled when

Director is not recording a macro.

## 1.15 Play Options Button

The play options button opens the play options window. This window controls some attributes ↔

of the currently from the  
Macro List

. Options that may be set include:

- o 'Speed' - this slider goes from 1 to 30, where 1 is recorded speed, and 30 is maximum speed. ↔
- o 'Hot Key' - the key combination which will automatically execute this macro. This key sequence is identical to that used with 'FKey'. ↔
- o 'Loop' - the number of times to repeat the macro if the 'Loop...' button is pressed. ↔

## 1.16 Record Options Button

The record options button opens the record options window. This window controls how the macro is recorded and provides some extra functionality. Only the registered version has all functions in this window enabled. Since some functions apply only while you are recording a macro, they will be disabled otherwise. ↔

All the "Record Actions" buttons will close this window immediately. To enter another "Record Action", the user will have to reopen this window.

The various functions are as follows:

#### Mouse Position:

These buttons insert mouse position store and recall commands into the macro.

They can for example be used to store the initial position of the mouse when playback starts, and then restore it at some later time. Upon hitting one of these buttons, the window will close automatically.

- o 'Save' - this button will insert the STORE command into the macro. This will store the current mouse position upon playback.
- o 'Restore' - this button will insert the RECALL command into the macro. This will restore the saved mouse position upon playback.

#### Insert Requestors:

These buttons insert requestors into the macro. The preferred editor is brought up on a temporary file. The user should enter the text to be displayed in the given requestor, and select save and exit. This will insert the appropriate requestor with that text into the macro. To get a better understanding for the use of these requestors, you should experiment with them.

- o 'OK...' - this button will insert the "OK" requestor. This requestor displays the given text until the user presses the 'Continue' button, at which point macro playback will resume.
- o 'Yes/No' - this button will insert a "Yes/No" requestor. This requestor displays the given text until the user presses the 'Yes' or 'No' button, at which point macro playback will resume. This requestor returns the yes/no result which can then be used with the "Flow Control" buttons below.
- o 'String Input...' - this button will insert a "String Input" requestor. This requestor displays the given text until the user presses the 'return' key. This for example allows the macro to prompt the user to enter an arbitrary string into a string gadget.
- o 'Window Close...' - this button will insert a "Window Close..." requestor. This requestor displays the given text until the user closes the currently select window. This can be used to prompt the user for arbitrary input into a window, and continue macro playback when the information is entered and the window is closed.

#### Flow Control:

These buttons insert IF-THEN-ELSE logic into the macro. They should be used with the "Yes/No" requestor (see above). For example the user may

---

create a "Yes/No" requestor with the text "Do you want to do the next step  
".  
Immediately after this, the "If Yes" flow control button could be selected  
. This will insert a test like "if last command returned YES, then DO" into  
the macro. The following macro statements will then only be executed if the  
user pressed "YES". The "ELSE" flow control could be used to execute something  
else. Note that for every "If Yes/No" inserted into the macro, the user  
must insert a "End If". Otherwise the macro will encounter an error upon  
playback. Also note that the "If Yes/No" buttons should be pressed immediately after  
the "Yes/No" requestor. This is due to the fact that always the last result  
is used in the test.

- o 'If No' - this button will insert an IF LAST RESULT WAS NO THEN DO THE  
FOLLOWING statement.
- o 'If Yes' - this button will insert an IF LAST RESULT WAS YES THEN DO THE  
FOLLOWING statement.
- o 'Else' - this button will insert an ELSE statement into the macro.
- o 'End' - terminate the conditional expressions above.

#### Record Settings:

These controls give control over what and how a macro is recorded. You  
may select to record just the keyboard input, or both mouse and keyboard. You  
may also select whether mouse actions are recorded relative to the active  
window, screen, or mouse. The default mode is relative to the window.  
This makes all mouse actions relative to the given window. Upon playback, the  
mouse will be positioned relative to that window. If the window cannot be  
found, however, an error will be displayed. The screen relative mode  
makes all mouse actions relative to the screen. Upon playback, no attention will  
be given to where windows are. All input will be played back in absolute  
screen coordinates. The mouse relative mode can be very useful for  
playing a macro relative to where the mouse currently is. Upon playback, the  
macro will not move the mouse to the recorded position, but will start playback  
wherever the mouse currently is.

#### Menus:

There are also some menu options which should normally never be used:

- o 'Look at Gadgets' - this records references to gadgets rather than just window offsets.
- o 'Look at Menus' - this records references to menus rather than just window offsets.
- o 'Smart Window ID' - this uses a more complicated scheme for identifying windows. If there are problems with finding windows, turning this off may help.
- o 'Insert REXX...' - this menu will bring up an editor. The text you enter here will be inserted into the macro.

## 1.17 Registering

This software is distributed as shareware. That means that the unregistered version may be distributed freely. Any one may try out the unregistered version for some period of time (say 1 month). After this period, the user should either register the software, or cease using it. Please be aware of the fact that shareware authors work hard and long hours to create products - just like the rest of you. We depend on registration payments to support development on this and other products. If no one registers, then we will not be able to develop any more shareware software. That would not be good, considering how much good shareware software there is out there - and cheap too! So PLEASE register this software if you feel that it is useful to you. And if it is not quite useful enough to register, maybe you can send me some e-mail and let me know of ways to possibly improve it - thanks.

To register this software please send a check for \$35. This check has to be from a US bank. If you are not able to send a US check, you may send \$35 cash or money order. All orders received will be replied to within 1 week. Please include your full name, mailing address, what you use Director for (optional), and what you would like to see changed. From now on, the registered version will contain a key file which can be used with future revisions of Director. This will allow owners of Director to get the latest shareware release and use it without restrictions.

My mailing address is:

Philip A. Gruebele  
100 Buckingham Dr. #222  
Santa Clara, CA 95051

You will receive a 3.5" disk with the latest software and documentation and no expiration date. The annoying "Please Register" requestor will be gone!

---

## 1.18 Reporting Problems

Since this is the first release of this software, I suspect a number of bugs to be reported to me. If this is the case, I will post another unregistered shareware on Aminet.

Problems may be reported to me in several ways. You may send me mail at the following mailing address:

Philip A. Gruebele  
100 Buckingham Dr. #222  
Santa Clara, CA 95051

You may also send me email at:

philipg@bayarea.net

If you find enforcer hits please send the complete hit. It helps greatly if you run segtracker and enforcer with the following settings:

```
segtracker
enforcer STACKLINES 35 STACKCHECK AREGCHECK
```

This gives me more detailed information needed to find the problem. Also, you should include a brief description of how to recreate the problem, the version of AmigaDOS you are using, the machine type, and any other pertinent information.

## 1.19 Known Problems and Limitation

Surprisingly I have gotten a very small number of bug reports/requests for this software. I guess people have been pretty happy with it! However, here is are some items you should be aware of:

- o requires AmigaDOS 2.04 or greater
  - o AREXX has to be installed and running
  - o only one macro should be placed in WBStartup drawer
  - o some times macros won't play back correctly with a speed of 30, so slow them ↔ down
  - o application with overlapping screens may cause problems (some paint programs)
  - o when using MagicMenu, menus are not played back appropriately.  
To bypass this problem, always click the right mouse button over where the actual menu will appear, rather than clicking and then moving to the appropriate menu. A future release of MagicMenu may fix this.
  - o when using IF-THEN-ELSE constructs, the time stamps for the IF and ELSE cases should start from the same value
  - o also, when doing IF-THEN-ELSE constructs, it is the responsibility of the user to terminate the conditional.
  - o if at playback, a double click is not recognised, change you preferences to make ↔ it longer.
  - o if a "macro returned error" is reported, and you'd like to find out more try ↔ running the macro like this: "rx name.mac"
  - o you should quit Director since it Setfuntions various system calls
-



## 1.20 Revision History

This history lists the changes between releases:

version 1.5 to 1.6

- o updated documentation with new contact information. No bugs were found, so none were fixed! ←

version 1.4 to 1.5

- o added key file for easier upgrading
- o shareware version now no longer expires, but brings up reminders to register

version 1.3 to 1.4

- o generally a little more reliable
- o double clicks no longer get lost
- o keyboard input to shell windows no longer gets lost
- o if active window closes at record time, then at play time macro will wait until the window closes before continuing ←
- o added FRONTScreens tool type so we can pop open windows on Workbench or frontmost public screen ←

version 1.2 to 1.3

- o updated documentation
- o made minor changes

version 1.1 to 1.2

- o compiled for 68000 instead of 68030 (will no longer crash on 68000)

version 1.0 to 1.1

- o Fixed problem that would cause instant crash under AmigaDOS 2.04/2.1. Director was calling AmigaDOS 3.1 functions even under 2.0
- o made it so macros always get '.mac' extension.
- o cleaned up some error handling

version 1.0

- o First public release

Future plans for this software are:

- o Holding down left mouse button will keep a macro from playing. This will allow the user to not execute a macro at boot time.
  - o Add better keyboard input commands that a programmer can use.
  - o Make IF-THEN-ELSE constructs more robust.
  - o Any good suggestions I get!
-