

ScreenDaemon

Nick Christie

Copyright © ScreenDaemon.guide1.0 Copyright ©1995 Nick Christie

COLLABORATORS

	<i>TITLE :</i> ScreenDaemon		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Nick Christie	March 1, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ScreenDaemon	1
1.1	ScreenDaemon.guide	1
1.2	ScreenDaemon: Introduction	1
1.3	ScreenDaemon: Requirements	1
1.4	ScreenDaemon: Tooltypes and Shell Arguments	2
1.5	ScreenDaemon: MASTER Keyword	2
1.6	ScreenDaemon: CX_POPKEY Keyword	2
1.7	ScreenDaemon: CX_PRIORITY Keyword	3
1.8	ScreenDaemon: LANGUAGE Keyword	3
1.9	ScreenDaemon: FLUSH Keyword	3
1.10	ScreenDaemon: READICON Keyword	3
1.11	ScreenDaemon: ARexx Commands	3
1.12	ScreenDaemon: OPEN Command	4
1.13	ScreenDaemon: DEFAULT Command	5
1.14	ScreenDaemon: TOFRONT Command	5
1.15	ScreenDaemon: TOBACK Command	5
1.16	ScreenDaemon: CLOSE Command	5
1.17	ScreenDaemon: SHANGHAI Command	5
1.18	ScreenDaemon: POPPUBSCREEN Command	6
1.19	ScreenDaemon: SILENT Command	6
1.20	ScreenDaemon: QUIT Command	6

Chapter 1

ScreenDaemon

1.1 ScreenDaemon.guide

ScreenDaemon V1.0 By Nick Christie 10th December 1995

In this ScreenDaemon Guide:

Introduction. What it's all about. **Requirements.** What you need. **Tooltypes & Arguments.** Doing it your way. **ARexx Commands.** Getting down to business.

These are in ScreenMaster's Guide:

Distribution. Some rules. Copyrights. It's mine, all mine! Disclaimer. But I'm not responsible. Package Contents. What's in the box. Installation. How to get started. Bugs and Improvements. What could be better. Version History. In the beginning... Credits. I'd like to thank... Author. The person responsible.

1.2 ScreenDaemon: Introduction

Introduction =====

ScreenDaemon is the most important part of the ScreenMaster package, as it is the program that actually opens and closes public screens. However, it doesn't have a window with gadgets, as you might expect. Instead, it is an ARexx host and you can only talk to it via the ARexx language.

You already have ARexx installed on your Amiga; although originally a separate, commercial product, ARexx was so useful that Commodore (in a rare fit of wisdom) included it in Workbench Release 2 and above.

Unfortunately, in a more typical penny-pinching exercise, C= didn't supply an ARexx manual with the budget model Amigas. If you don't have an ARexx manual, you can get a very comprehensive (if a bit technical) guide to it as ARexxGuide in the util/rexx directory of AmiNet, or from your favourite PD supplier. This is a document in AmigaGuide format which you can read using AmigaGuide (WB2) or MultiView (WB3). In addition, several popular Amiga magazines have run ARexx tutorials.

Using ARexx you can perform all the usual computation-like activities normal in any programming language, but in addition, you can control other applications that have an ARexx port, including ScreenDaemon.

If this all sounds a bit heavy, you might prefer to use a graphical (windows and gadgets) interface to ScreenDaemon - have a look at the ScreenMaster User Guide instead.

1.3 ScreenDaemon: Requirements

Requirements =====

ScreenDaemon requires:

· AmigaOS 2.04 / Kickstart V37 or higher. · ARexxMast to be running. (inter-process communication)

The following disk-based Commodore libraries are required, but are part of AmigaOS 2.04 (and later releases) anyway:

· rexxsyslib.library V36 or higher. (ARexx support) · commodities.library V37 or higher. (commodity support)

Additionally, ScreenDaemon will use the ReqTools library (V38 or higher) for its requesters, if it is available.

1.4 ScreenDaemon: Tooltypes and Shell Arguments

Tooltypes and Shell Arguments =====

The tooltypes ScreenDaemon looks for in its program icon are described below. The arguments used from a Shell are identical, except where stated otherwise. The full Shell template (with abbreviations) is:

MA=MASTER/K, PK=CX_POPKEY/K, PR=CX_PRIORITY/K/N, LA=LANGUAGE/K, FL=FLUSH/S, RI=READICON/S

ScreenDaemon's tooltypes consist of three kinds:

String - You must supply a string of some kind; a filename, public screen name, etc. For example, "LANGUAGE=deutsch".

Integer - You must supply a numeric value (a whole number). For example, "CX_PRIORITY=5".

Boolean - In tooltypes you must supply one of the following: YES, NO, TRUE or FALSE. For example, "FLUSH=YES". You can also omit the value entirely, which indicates the YES or TRUE state. On a Shell command line, just state the keyword without a value to get the TRUE state, omit the keyword for FALSE.

Tooltype Type Short Description -----
 ----- MASTER STR Location of ScreenMaster program.
 ----- CX_POPKEY STR Commodities popup hotkey definition.
 ----- CX_PRIORITY INT Priority of hotkey among commodities.
 ----- LANGUAGE STR Specify language catalog.
 ----- FLUSH BOOL Quit when last screen closed.
 ----- READICON BOOL Read config. from tooltypes (Shell).

If you put ScreenDaemon in your WBStartup drawer, you should add the tooltype "DONOTWAIT" so Workbench won't wait for ScreenDaemon to quit before continuing with its start up.

1.5 ScreenDaemon: MASTER Keyword

MASTER Keyword =====

Type: String Default: ScreenMaster Shell: MA=MASTER/K Tooltype: MASTER

When you press ScreenDaemon's hotkey, it tries to open the ScreenMaster program, launching it if it is not already running. If ScreenMaster is in ScreenDaemon's search path this will work, otherwise use this tooltype to tell ScreenDaemon where to find it.

I recommend you add the READICON keyword as well, so that ScreenMaster, when started, reads its configuration from its tooltypes. For example: "MASTER=Work:Things/ScreenMaster READICON".

1.6 ScreenDaemon: CX_POPKEY Keyword

CX_POPKEY Keyword =====

Type: String Default: lcommand lshift p Shell: PK=CX_POPKEY/K Tooltype: CX_POPKEY

Set a popup hotkey using this keyword; when you press it, ScreenDaemon will try to open the ScreenMaster program, launching it if it is not already running. See your Amiga manuals for the syntax of hotkey definitions.

1.7 ScreenDaemon: CX_PRIORITY Keyword

CX_PRIORITY Keyword =====

Type: Integer Default: 0 Shell: PR=CX_PRIORITY/K/N Tooltype: CX_PRIORITY

Use this tooltype to set ScreenDaemon's priority relative to the other commodities running. Normally there is no need to change this.

1.8 ScreenDaemon: LANGUAGE Keyword

LANGUAGE Keyword =====

Type: String Default: <system default> Shell: LA=LANGUAGE/K Tooltype: LANGUAGE

ScreenDaemon supports the locale library of AmigaOS 2.1 and above, allowing it to display text and messages in your preferred language. You would normally specify the language with the Locale Preferences program; this tooltype allows you to override that setting if necessary.

1.9 ScreenDaemon: FLUSH Keyword

FLUSH Keyword =====

Type: Boolean Default: FALSE Shell: FL=FLUSH/S Tooltype: FLUSH

If you set the FLUSH option to TRUE, ScreenDaemon will quit as soon as the last screen it has open closes, ie. when it is no longer in charge of any public screens.

1.10 ScreenDaemon: READICON Keyword

READICON =====

Type: Boolean Default: FALSE Shell: RI=READICON/S Tooltype: <not applicable>

This keyword is only available from the Shell command line. It makes ScreenDaemon read its configuration from its icon's tooltypes rather than the command line. This is useful in conjunction with program launchers that run programs as if from a Shell. Ordinarily you would have to duplicate all your preferred ScreenDaemon configuration arguments in the command that you supply to the launcher, but by specifying READICON, you can keep your settings in the icon.

1.11 ScreenDaemon: ARexx Commands

ARexx Commands =====

ScreenDaemon's ARexx port is called, appropriately enough, 'SCREENDAEMON'.

Multiple commands on a single line are legal, and are evaluated in the order they appear below. So if you supply, for example, "OPEN=PubScr DEFAULT=PubScr", the screen will be opened and then made the default public screen.

Failure to complete a command results in a requester being displayed with an appropriate error message, unless the **SILENT** keyword is supplied.

All the commands return a primary result code, available in the ARexx variable RC, but no secondary result (RESULT variable). Errors during the processing of a command (such as an invalid value for a keyword) return RC_FATAL (20). Errors that occur while carrying out the command (such as being unable to open a screen) return RC_ERROR (10). Finally, the error of not being able to find a named public screen (for example, CLOSE "NoScreenWithThisName") returns RC_WARN (5).

No further commands on a given line are processed after an RC_FATAL or RC_ERROR condition occurs. RC_WARN does not stop further processing.

Remember that ARexx tends to capitalize everything, so quote public screen names, which are case sensitive. Also quote any names with embedded spaces. This includes filenames. The commands themselves are case-insensitive.

ScreenDaemon understands the following commands:

OPEN Opens a new public screen. **DEFAULT** Makes a public screen the default. **TOFRONT** Moves a public screen to the front. **TOBACK** Moves a public screen to the front. **CLOSE** Closes a public screen. **SHANGHAI** Sets the state of the Shanghai flag. **POPPUBSCREEN** Sets the state of the PopPubScreen flag. **SILENT** A switch to suppress error requesters. **QUIT** Quits ScreenDaemon, if possible.

1.12 ScreenDaemon: OPEN Command

OPEN Command =====

Template: OP=OPEN/K,TI=TITLE/K,MI=MODEID/K,MN=MODE/K,WI=WIDTH/K/N,HE=HEIGHT/K/N,DP=DEPTH/K/N,OS=OVERSCAN/K,PE=PENS/K,FT=FONTTYPE/K,FO=FONT/K,PA=PALETTE/K,NA=NOAUTOSCROLL/S,IL=INTERLEAVED/S,SP=SHAREPENS/S,BE=BEHIND/S,AC=AUTOCLOSE/S

Returns: 10 if screen cannot be opened. 0 if screen opened ok.

The OPEN keyword must be followed by a name for the new public screen. All the additional parameters listed in the template are, in fact, optional. They simply give you control over the characteristics of the screen to be opened.

TITLE <title> ----- Title for the new public screen. Default: Same as name.

MODEID <mode-hex> ----- Desired displaymode as a hexadecimal number, eg, "0x8004" or "\$8004". Default: 0.

MODE <mode-name> ----- Desired displaymode as a name, eg. "PAL:High Res Laced". Mutually exclusive with MODEID.

WIDTH <num> and HEIGHT <num> ----- Pixel width/height of screen. Default: Displaymode's nominal size.

DEPTH <num> ----- Colour depth (1-8). Default: 1.

OVERSCAN <TEXT|GRAPHICS|CUSTOM> ----- Desired overscan mode. If CUSTOM, supply DCLIP, below. Default: TEXT.

DCLIP <minx,miny,maxx,maxy> ----- Display clip, relative to view. eg, "-8,-8,655,527". Must be quoted. CAUTION! Read the details in the ScreenMaster guide before messing about with this one!

PENS <detail,block,text,shine,...> ----- A set of system pens, supplied as pen numbers, eg. "0,1,1,3,4,2,6". This must be quoted. AmigaOS2 has max 8 pens, OS3 has 11. Default: System default pens.

FONTTYPE <SYSTEM|SCREEN|CUSTOM> ----- Choice of font. If CUSTOM, supply FONT, below. Default: SYSTEM.

FONT <customfontname/size> ----- Custom font name and size, eg. "Times/15" or "Times.font/15". Default: None.

PALETTE <filename> ----- Screen palette filename, either ILBM or Prefs/Palette file. Default: System default.

NOAUTOSCROLL ----- A switch, turns off autoscroll. Default: Autoscroll on.

INTERLEAVED ----- A switch, requests an interleaved bitmap (OS3+ only). Default: Off, simple bitmap.

SHAREPENS ----- A switch, stops Intuition allocating all screen pens exclusively. Default: Off, no pen sharing.

BEHIND ----- A switch, opens the new screen behind all others. Default: Off, open in front.

AUTOCLOSE ----- A switch, autocloses the screen when the last visitor window has left it. Default: Off, no autoclose.

1.13 ScreenDaemon: DEFAULT Command

DEFAULT Command =====

Template: DE=DEFAULT/K

Returns: 10 if screen cannot be made the default 5 if screen cannot be found 0 if screen was made the default

Sets the named public screen to be the system default public screen.

1.14 ScreenDaemon: TOFRONT Command

TOFRONT Command =====

Template: TF=TOFRONT/K

Returns: 10 if screen could not be moved to front 5 if screen could not be found 0 if screen was moved to the front

Moves the named public screen to the front, in front of all other screens.

1.15 ScreenDaemon: TOBACK Command

TOBACK Command =====

Template: TB=TOBACK/K

Returns: 10 if screen could not be moved to back 5 if screen could not be found 0 if screen was moved to the back

Moves the named public screen to the back, behind all other screens.

1.16 ScreenDaemon: CLOSE Command

CLOSE Command =====

Template: CL=CLOSE/K

Returns: 10 if screen could not be closed 5 if screen could not be found 0 if screen was closed

Closes the named public screen, if possible. A screen can't be closed when it is private, or has visitor windows open on it. If the FLUSH switch was specified when ScreenDaemon was launched, and ScreenDaemon no longer has any screens open, the program will quit.

1.17 ScreenDaemon: SHANGHAI Command

SHANGHAI Command =====

Template: SG=SHANGHAI/K

Returns: 20 if argument is not TRUE, FALSE, YES or NO. 0 otherwise.

Sets the state of the Shanghai flag: supply as argument one of TRUE, FALSE, YES or NO. Any other argument is treated as an error.

1.18 ScreenDaemon: POPPUBSCREEN Command

POPPUBSCREEN Command =====

Template: PO=POPPUBSCREEN/K

Returns: 20 if argument is not TRUE, FALSE, YES or NO. 0 otherwise.

Sets the state of the PopPubScreen flag: supply as argument one of TRUE, FALSE, YES or NO. Any other argument is treated as an error.

1.19 ScreenDaemon: SILENT Command

SILENT Command =====

Template: SI=SILENT/S

Returns: N/A

This is a switch; when SILENT is specified with any other command, it suppresses any requester that might appear to indicate an error. A primary result code, in ARexx variable RC, is still returned.

1.20 ScreenDaemon: QUIT Command

QUIT Command =====

Template: QU=QUIT/S

Returns: 10 if ScreenDaemon is still has any public screens open. 0 if ScreenDaemon quit.

This switch makes ScreenDaemon quit, as long as it doesn't have any screens open. This switch can be supplied with other commands and is the last to be acted on.
