ScreenBacker

Nick Christie

COLLABORATORS			
	TITLE : ScreenBacker		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Nick Christie	March 1, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ScreenBacker

1.1 ScreenBacker.guide

ScreenBacker V1.0 By Nick Christie 10th December 1995

In this ScreenBacker Guide:

Introduction. What it's all about. Requirements. What you need. Tooltypes & Arguments. Doing it your way. ARexx Commands. Getting down to business. The Standard Menu. What you get for free. Menu Definition Files. Ordering it a la carte.

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 ScreenBacker: Introduction

Introduction ============

ScreenBacker adds backdrop patterns and customized menus to public screens. It doesn't have a window with gadgets; 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 ScreenBacker.

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

1.3 ScreenBacker: Requirements

Requirements ============

ScreenBacker 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) · asl.library V37 or higher. (requesters)

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

1.4 ScreenBacker: Tooltypes and Shell Arguments

Tooltypes and Shell Arguments =============================

The tooltypes ScreenBacker 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, PR=CX_PRIORITY/K/N, LA=LANGUAGE/K, FL=FLUSH/S, RI=READICON/S

ScreenBacker'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_PRIORITY INT Priority of hotkey among commodities. LANGUAGE STR Specify language catalog. FLUSH BOOL Quit when last backdrop closed. READICON BOOL Read config. from tooltypes (Shell).

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

1.5 ScreenBacker: MASTER Keyword

MASTER Keyword ==============

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

When you pick the "Show Interface" item from the standard menu attached by ScreenBacker to a public screen, ScreenBacker tries to open the ScreenMaster program, launching it if it is not already running. If ScreenMaster is in ScreenBacker's search path this will work, otherwise use this tooltype to tell ScreenBacker 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 ScreenBacker: CX_PRIORITY Keyword

CX_PRIORITY Keyword ===================

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

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

1.7 ScreenBacker: LANGUAGE Keyword

LANGUAGE Keyword ================

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

ScreenBacker 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.

1.8 ScreenBacker: FLUSH Keyword

FLUSH Keyword =============

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

If you set the FLUSH option to TRUE, ScreenBacker will quit as soon as the last backdrop or menu it has open closes.

1.9 ScreenBacker: 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 ScreenBacker 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 ScreenBacker configuration arguments in the command that you supply to the launcher, but by specifying READICON, you can keep your settings in the icon.

1.10 ScreenBacker: ARexx Commands

ARexx Commands ==============

ScreenBacker's ARexx port is called, appropriately enough, 'SCREENBACKER'.

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 CLOSE=OtherScr", a default menu will be attached to PubScr, and any backdrop or menu on OtherScr will be removed.

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 backdrop) return RC_ERROR (10). Finally, errors such as not being able to find a named public screen (for example, CLOSE "NoScreenWithThisName") return 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.

ScreenBacker understands the following commands:

OPEN Opens a new backdrop or menu. CLOSE Closes a backdrop or menu. SILENT A switch to suppress error requesters. QUIT Quits ScreenBacker, if possible. KILL Closes all backdrops/menus and quits.

1.11 ScreenBacker: OPEN Command

OPEN Command ============

Template: OP=OPEN/K,PN=PATTERN/K,PC=PATCENTRE/S,PT=PATTILE/S, ME=MENU/K,NM=NOMENU/S

Returns: 10 if backdrop couldn't be opened. 5 if backdrop already present on that screen. 0 if backdrop opened ok.

The OPEN keyword must be followed by the name of the public screen to attach a backdrop or menu to. You can then specify none or more of the other options. If you don't add any of the options below, the screen will simply get a standard menu attached to it, containing a few useful public screen management functions.

PATTERN <patfile> -----------------

Specifies the name of an IFF ILBM or Workbench Preferences pattern file to apply as a backdrop pattern to this public screen.

ScreenBacker doesn't currently do any pen remapping, so you should make sure the screen's palette is suitable for the backdrop. Also, if given a Workbench 3 pattern preferences file, ScreenBacker will currently only use the screen pattern recorded in it (ie. not the Workbench window or drawer window patterns).

PATCENTRE ---------

If you want the backdrop pattern specified by PATTERN to be centred in the middle of the screen, supply this switch. Unless you specify the NOMENU option as well, the screen will get at least the standard menu attached to it, as well.

PATTILE -------

If you want the backdrop pattern specified by PATTERN to be repeated across and down the screen, supply this switch.

MENU <menufile> ---------------

If you would like to add a custom menu definition to the public screen, supply its filename with the MENU keyword. This option is mutually exclusive with the NOMENU keyword.

NOMENU ------

With this switch you can suppress the attachment of a menu to the screen; use it if you just want to add a backdrop pattern. This option is mutually exclusive with the MENU keyword, and if you use it, you must specify a backdrop pattern with the PATTERN keyword.

1.12 ScreenBacker: CLOSE Command

CLOSE Command =============

Template: CL=CLOSE/K

Returns: 5 if screen could not be found, or no backdrop open on it. 0 if backdrop was closed.

Closes the backdrop and/or menu attached to the named public screen. If the FLUSH switch was specified when ScreenBacker was launched, and ScreenBacker no longer has any backdrops open, the program will quit.

1.13 ScreenBacker: 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.

1.14 ScreenBacker: QUIT Command

QUIT Command ============

Template: QU=QUIT/S

Returns: 10 if ScreenBacker is still has any backdrops open. 0 if ScreenBacker quit.

This switch makes ScreenBacker quit, as long as it doesn't have any backdrops or menus open.

1.15 ScreenBacker: KILL Command

KILL Command ============

Template: KILL/S

Returns: 0 always.

This switch makes ScreenBacker close all backdrops and menus and quit.

1.16 ScreenBacker: The Standard Menu

The Standard Menu =================

The standard menu, titled "Project", that ScreenBacker attaches to a public screen contains the following items:

Open Interface --------------

Opens the ScreenMaster program, launching it if it is not already running.

Make Default ------------

Makes this public screen the system default.

Set Palette... --------------

Opens a palette requester, for adjusting this screen's colours.

Load Palette... ---------------

Loads a new palette from a file, either an IFF ILBM or Workbench Preferences file. A requester is opened to let you choose the file.

Save Palette... ---------------

Saves the screen's palette as an IFF ILBM file. A requester is opened to let you choose the filename.

About... --------

Displays some information about ScreenBacker.

Close Backdrop --------------

Closes the backdrop and menu attached to this screen. If the FLUSH option was specified when ScreenBacker was launched, and no more backdrops or menus are open, ScreenBacker will quit.

Close Screen ------------

Closes the backdrop and menu attached to this screen, and then instructs ScreenDaemon to close the screen itself. This will fail if there are any visitor windows open on the screen. If the FLUSH option was specified when ScreenBacker was launched, and no more backdrops or menus are open, ScreenBacker will quit.

1.17 ScreenBacker: Menu Definition Files

Menu Definition Files ====================

Unless suppressed with the NOMENU switch, ScreenBacker adds a small, standard menu to public screens. You can extend this menu with items that launch your favourite programs, like the Workbench Tools menu. To do this, supply the MENU keyword and the name of a menu definition file. I recommend you use the ScreenMenuEd program to create these, but you don't have to: they are plain text (ASCII) files that can be created and edited with a text editor.

A menu definition file consists of a series of records, one per line, that define consecutive menu titles, item separator bars and menu items. The definition starts with the second menu title (the first being that of the standard menu) and proceeds down with new items and to the right with new menu titles. Each line is parsed with the following template:

MENU/K, BAR/S, ITEM/K, CMD=COMMAND/K, KEY/K, ST=STACK/K/N, PRI=PRIORITY/K/N, OP=OUTPUT/K, CD=CURDIR

MENU/K ------

Start a new menu with the MENU keyword, followed by the desired title. Enclose the title in quotes if it contains embedded spaces. Naturally, the first record of the file must be a MENU definition, and the last record of the file cannot be a MENU definition. No other keywords are allowed in the record if MENU is used.

BAR/S -----

The BAR keyword identifies a separator item. A BAR record must be preceeded and followed by an ITEM record. No other keywords are allowed in the record if BAR is used.

ITEM/K ------

Define a new item for the current menu with the ITEM keyword, with the text you want displayed in the menu for that item. Enclose the text in quotes if it contains embedded spaces. Use the other option keywords in an ITEM record to specify the command you want executed when that item is selected:

CMD=COMMAND/K This argument is required and should be the name of the program to launch - this will be run asynchronously, ie. as a new process. You can embed up to one instance of the formatting sequence "%s" in the command: this will be substituted with the name of the public screen that the menu is attached to. This can be very useful if the program accepts a public screen argument, eg. "MultiView PUBSCREEN %s".

KEY/K Optionally, specify a menu shortcut key for this item.

ST=STACK/K/N Optionally, specify a stack size in bytes for the program. If omitted, the program inherits ScreenBacker's stack size.

PRI=PRIORITY/K/N Optionally, specify a task priority for the program. If omitted, defaults to 0.

OP=OUTPUT/K Optionally, specify a standard output for the program. You can embed one instance of the formatting sequence "%s" in this, which will be substituted with the name of the public screen that the menu is attached to. This can be very useful in conjunction with the SCREEN specifier of console windows. If omitted, the output defaults to: "CON:20/20/600/100/ScreenBacker Output/AUTO/CLOSE/WAIT/SCREEN %s".

CD=CURDIR/K Optionally, specify the current directory for the program. If omitted, the program inherits ScreenBacker's current directory.