PopShell

COLLABORATORS

	TITLE : PopShell		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 9, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

PopShell

1.1 Welcome to the PopShell documentation

PopShell Version 6.9

* FreeWare Evaluation Version *
* ShareWare Registered Version *

PopShell is ShareWare and is available to the public in a crippled FreeWare version. It has been crippled (a bit) to give the users a reason to become registered...

What is not available in the FreeWare version? The opportunity to save configuration files from within PopShell - you will have to edit the file using your text editor...

Oh yes, and the "About" requester pops up each time you open the configuration window.

That’s all, folks. Not too terrible, but is it enough to get you to become registered? I really hope so!

1.2 Introduction

Introduction:

PopShell is a commodity that let’s you open shell windows on any public screen (either created by PopShell or any other public screen utility). Furthermore, PopShell can open PubScreens on request by patching LockPubScreen() for use with any program that allows the use of (named) public screens. This feature can be disabled (for those among you who think that patching system functions is evil... :-))

Note:
Yes, this really is the sixth major version of PopShell. Previous versions have been either unreleased or were only available to a very restricted number of users. This is the first major release.

PopShell’s history is quite long:

The first three versions were entirely written in Assembler.

V 1: (1991) simple pubscreen opening tool, controllable only from CLI.
V 2: (1991) new option to start the shell directly from PopShell.
V 3: (1992) added small configuration files.

After V3, PopShell has been rewritten almost completely in C.

V 4: (1992) small GUI with very few options.
V 5: (1993) again, a complete rewrite, implementing the current concept.
V 6: (1994) complete cleanup, new features and optimizations.

1.3 Legal stuff

Copyright and distribution:

PopShell is copyrighted by Tobias Abt and Alexander Kneer. The registered version of PopShell may be distributed by the authors only.

On the other hand, the FreeWare version may be distributed on any possible way given that the archive structure is left untouched and no files are modified. The authors explicitly allow adding of BBS advertisements in the root directory of the archive.

The distibution archive must contain:

```
   PopShell/PopShell
   PopShell/PopShell.config
   PopShell/catalogs/deutsch/PopShell.catalog
   PopShell/Docs.info
   PopShell/Docs/deutsch.info
   PopShell/Docs/deutsch/PopShell.guide
   PopShell/Docs/deutsch/PopShell.guide.info
   PopShell/Docs/english.info
   PopShell/Docs/english/PopShell.guide
   PopShell/Docs/english/PopShell.guide.info
   PopShell/icons/PopShell16.info
   PopShell/icons/PopShell4.info
   PopShell/Install-PopShell
   PopShell/Install_Deutsch.info
   PopShell/Install_English.info
   PopShell.info
```

Disclaimer:

No Warranty of any kind is given that the program described in this document is 100% reliable. You are using this material at your own risk. The authors cannot be made responsible for any damage which is caused by using PopShell.

1.4 How to use PopShell

The graphical user interface of PopShell consists of three major components:

- the PopShell objects,
- the PopScreen objects and
- the menus .

The concept using PopShells and PopScreens offers total and fast control of the entire configuration. This way the user can easily configure shells and screens to suit their needs and likes.

1.5 PopShell objects

PopShell objects are used to link a public screen name, a hotkey and a shell command. You can configure:

- Name: Just for identification purposes for the user.

- PopScreen: Name of the corresponding public screen. Select an appropriate PopScreen from the PopScreen list below and click on the PopScreen button to get the public screen name.

- HotKey: HotKey to launch this PopShell object. Either type into string gadget manually or click on the button and PopShell will interpret keys that you press as a HotKey.

- Windowtitle: Name of the console window when launching, defaults to "AmigaShell".

- Shell Command: Name of the command to start the new shell, defaults to "NewShell".

- Console Handler: Name of the console handler, defaults to "CON:"

- Console Options: Additional options for the console handler.

- Force new Shell: Normally PopShell will simply bring a previously opened shell to the front screen and activate it’s window. This option tells PopShell to open a new shell every time the HotKey is pressed.

The shells will be launched using this command line:

```
<shell cmd> "<con hnd>X/Y/W/H/<Wintitle>/CLOSE/<con opts>/SCREEN<PScName>"
```

1.6 PopScreen objects

PopScreens are used to configure the screens where shells are to be opened. They offer these features:

- Name: This is the name of the public screen. Clicking on this button will open a window listing the currently available public screens. "#FrontMost#" is a special name which stands for the frontmost public screen at the moment of invocation. Very handy for shell or ARexx scripts!

- Displaymode: (Needs asl.library V38+) Opens a screenmode requester which lets the user choose mode, width, height, depth and autoscroll, defaults to HIRES.

- Palette: Opens a palette requester on the PopScreen to adjust colors. Offers menu to load and save IFF palettes. Defaults to Workbench palette. This only works on PopShell created screens.

- Pens: Opens a pens requester for selection of the numerous pens for this screen, defaults to system pens. This only works on PopShell created screens.

- Shells: Lets user configure up to four different window sizes and positions on each PopScreen. Use menus to add or delete windows and to save the current position info. Default: one full sized window.

- Font: Select screen titlebar font. Name and size are shown in the string gadget to the right (e.g. "Courier.13.b" for a bold Courier font with 13 pixels height). This only works on PopShell screens. Defaults to Topaz 8.

- Alien: Decide whether this PopScreen is an external (‘alien’) one or one that PopShell has to open (‘own’). Examples for external PopScreens: "Workbench" and "#FrontMost#". Defaults to own screen.

- Backfill: (V39+ only) When checked, a custom screen backfill hook is activated which fills the background of this PopScreen with a nice pattern. This only works on PopShell screens.

- Clock: Flag which tells PopShell to show a clock and memory display in the titlebar of the PopScreen. This only works on PopShell’s own screens.

The screens will be opened by PopShell. Closing and clock handling is done by a handler task attached to the screen. Once opened, the screen is totally independent from PopShell - although still configurable in some aspects (clock on/off...). A PopScreen will be automatically closed as soon as the last visitor window leaves.

1.7 Menus

The menus are used for global stuff...

```
- Project:
  - Open ...:       Open filerequester to load new configuration file.
  - Save:           Save configuration under the current configfile name,
                    defaults to PopShell.config. Adjustable via tooltype.
                    Available only in the registered version!
  - Save as ...:    Similar to ‘Save’ but with file requester to select new
                    file name.
  - About:          Open about requester (What else did you expect?).
  - Quit:           Quits PopShell. (Usually very rarely used...)

- Special:
  - Active:         PopShell active when checked, otherwise no HotKeys
                    will cause any action...
  - Auto PubScreen: When checked, LockPubScreen() patch is active, so
                    PopScreens can be launched when requested by any
                    program. This also adds own PopScreens to the list
                    of opened public screens allowing other programs to
                    use them e.g. to display a public screen requester.
  - Default:        Reset configuration to internal defaults (two shell
                    objects on an own screen).
```

1.8 Quick start guide

Quick start guide:

Install and start PopShell (preferably from Workbench). To open the configuration window, press the hotkey (default: lcommand p), double click on the PopShell icon a second time or use ‘Exchange’ to show the user interface. The default settings consists of a public screen named ‘PopScreen’ (default screenmode is "HIRES") and two PopShell hotkeys. One hotkey to activate an existing shell on the ‘PopScreen’ or create a new if there is no old one ("lcommand esc") and a second one to always create a new shell whenever this hotkey ("lalt esc") is pressed to get more shells on ‘PopScreen’.

You may play around with the objects, create new ones, delete those you don’t want or do whatever you like. Using PopShell should be no problem for any (at least a bit) experienced Amiga user.

If you want to open a shell e.g. on the Workbench screen, click the button ‘New’ below the ‘Screen Objects’-listview, click on ‘Object Name’ and a requester appears to let you choose the name of the public screen directly. In our example, you would choose "Workbench" now. But you could also accomplish the same by entering "Workbench" in the string gadget right of the button. Now you have a new Screen object associated with the Workbench screen. You will have to tell PopShell now that this is an ‘Alien’ screen, which means that it will not be opened by PopShell. Then you may choose position and size of up to four shell windows clicking on the ‘Shells’ gadget.

Now, create a PopShell object by clicking on the ‘New’-button under the ‘PopShell Objects’ listview. You may give it any name (just for identification purposes), click on the ‘PopScreen’- button to get the name of the current PopScreen object into the text gadget right of it. The contents of this gadget is the name of the PubScreen on which this shell will appear. Now select a HotKey for that shell by clicking the ‘Hotkey’ gadget, which will open a small window and wait for the Hotkey to be pressed (to cancel: press CloseGadget), or enter the hotkey directly into the string gadget. If you want to have a new shell opened each time you press the hotkey, select ‘Force new shell’.

The PopShell object is now ready to go, but you may choose some other shell command than NewShell (e.g. NewWSH), or another console handler than CON: (e.g. CNC:), etc.

When you are done configuring, it is time to save the configuration file... - if you are a registered user...

1.9 Additional features

Addtional features of PopShell:

- There is an ‘Auto PubScreen’ submenu in the ‘Special’ menu (also in the configuration file) which activates the LockPubScreen()-patch. It works like that: if any process calls LockPubScreen() and the requested public screen is not already open, PopShell scans its list of PopScreen objects and if it finds an entry that matches the name of the requested PubScreen and which is not ‘alien’, it opens the screen and returns a pointer of this screen to the calling process. Let’s suppose that there is an entry named ‘foo’ (not of ’alien’ type!) and an application wants to LockPubScreen("foo"), it will open the screen (if it isn’t already open). Additionally those PopShell public screens which are not ’alien’ and not open yet are also added to the global list of public screens and are available to all programs who search this list to e.g. display a requester to let the user choose a public screen.

- If you double-click a PopShell object it gets activated.

- If you press the HotKey again while the Shell associated with the HotKey is in front, the screen will be sent to the back. This was implemented for fast and convenient usage of multiple screens.

- If started from CLI you may give the name of a PopShell-object as an argument. This object will be activated as soon as PopShell is running.

1.10 Description of the configuration file

The configuration file:

A human readable text file (default: PopShell.config) is used for storing the configuration data.

Entries in that file:

```
-AUTOOPENSCREEN=TRUE/FALSE: for  LockPubScreen() patch ,
-"POPSCREEN:...":           for    PopScreen objects   ,
-"POPSHELL:...":            for    PopShell  objects   .
-"# ...":                   for comments...
```

Each entry must have its own, single line, because the file is parsed using ReadArgs()!

```
The ReadArgs()-Templates
------------------------
POPSCREEN:      NAME=N/K/A,MODE=M/K,WIDTH=W/N/K,HEIGHT=H/K/N,
                DEPTH=D/K/N,AUTOSCROLL=AS/S,PALETTE=P/K,PENS/K,
                FONT=F/K,CLOCK/S,ALIEN/S,BACKFILL/S,SHELLS/K
POPSHELL:       NAME=N/K/A,SCREEN=SC/K/A,HOTKEY=H/K/A,
                WINDOWTITLE=WT/K,SHELLNAME=SN/K,CONNAME=CN/K,
                CONPARA=CP/K,NEW/S
```

```
Example configfile (in fact the default one):
-----------------------------------------------------------------
# $VER: PopShell.config V1.0 written by PopShell V6.9
# PopShell configuration file
#
AUTOOPENSCREEN=TRUE
POPSCREEN:NAME="PopScreen" MODE=$8000 WIDTH=-1 HEIGHT=-1 DEPTH=2 AUTOSCROLL  ↩
   PALETTE="" PENS="1" FONT="topaz.8" CLOCK BACKFILL SHELLS="-1,-1,-1,-1"
POPSHELL:NAME="GoShell" SCREEN="PopScreen" HOTKEY="lcommand esc" WINDOWTITLE=""  ↩
   SHELLNAME="" CONNAME="" CONPARA=""
POPSHELL:NAME="NewShell" SCREEN="PopScreen" HOTKEY="lalt esc" WINDOWTITLE=""  ↩
   SHELLNAME="" CONNAME="" CONPARA="" NEW
-----------------------------------------------------------------
```

1.11 Installation

Installation:

PopShell is a commodity which should be placed in the SYS:WBStartup drawer of your Workbench. The appropriate ToolTypes are created by the Installer script, so you should better use it!

ToolTypes:

In the tooltypes (or as CLI parameters) you can use the standard commodities tooltypes (CX_#?), the WBStartup tooltypes (STARTPRI, DONOTWAIT) and ‘CONFIGNAME’ which tells PopShell what file to use as its standard config file (defaults to: PopShell.config).

1.12 Changes to previous versions

Changes between different versions of PopShell:

```
V6.9:  first Aminet release
V6.10-
V6.12: internal versions
V6.13: (NEW) PopShell now launches new shells asynchronously, in order
             to handle new requests while shell has not returned control.
       (NEW) previously, PopShell created shells on the defined positions
             in cyclic order (i.e. it opened the first window, then the
             second and so on), now, it looks if a window on each of the
             defined positions exists and if not, it will be created.
             This is especially convenient if you accidentally close a
             window and you want the window to be opened at the same
             position again. This way, the positions have decreasing
             priorities.
```

1.13 How to become a registered user

There are several options for users to become registered:

a) Send us some money.
Requested amount to become registered: DM 15 or US $10. This includes a disk with the newest version and shipping. If you can be reached via EMail, you will get new versions for free! Otherwise updates/upgrades are DM 5 or US $4. Please no other foreign currencies!
If you want your registered version of PopShell mailed via email, you may only want the executable because of less network traffic.

In Germany, transfer the money to: Tobias Abt, Sparkasse Ulm, BLZ 630 500 00, Kto. 256 861 and send me (Tobias Abt) a letter with your address and a hint that the money is on the way so I can identify the transfer. Within Europe, please send Eurochecks, from outside Europe the simpliest way would be to send the money in a letter. Wrap the money into a sheet of paper so that it will not be visible from outside the letter. And don’t forget to add your name and address!

b) Localize PopShell (and documentation) for a currently not supported language. Please contact us first if you want to do so, as others may already do this!

c) You may offer registered versions of some of your programs to us. The intention behind this offer is to honor the software authors who have done something for the Amiga, too. Please understand that we want to choose if we would like to do so, so please ask us first if we accept, because there are tons of other programs around which we don’t need or even like...! FreeWare authors are certainly welcome, too!

d) Are you someone who contributed to the Amiga in some other way? If you think so and we agree to this, then you might also become a registered user for free.

By the way, standard mail sucks! So if you can, please use EMail! It’s easier, faster and cheaper. Surely for us but probably also for you...

1.14 Authors

PopShell was written by:

Tobias Abt
Bessererstraße 9
89073 Ulm
Germany
phone: ++49/731/63366 (weekends only)
email: zcjc1121@rpool1.rus.uni-stuttgart.de
irc: tabt@#AmigaGer

and

Alexander Kneer
Masurenweg 6a
89233 Neu-Ulm Pfuhl
Germany
phone: ++49/731/712490
email: no email address available due to incompetence of the system
administrators at the CS faculty of the University of Ulm.

PopShell is based on an idea of Ralph Mayer who was also involved in the development of the first versions of PopShell.

Also available from the authors:
- AssignZ, a perfect way to handle unresolved assingments "on demand"
- PatchPens, a comfortable tool to control some or even all of the Workbench colors under Workbench 3.x

1.15 Credits

Our thanks go to:

Our beta-testers (in alphabetical order):
Angela, Felix, Günther, Ingmar, Jens, Lothar, Mats, Michael, Oly,
Ralph, Thomas and to all those I forgot,

Commodore Amiga for the Amiga (sad they are gone...),
SAS for their wonderful C-Compiler (I’m sorry they left the Amiga...),
Michael D. Bayne for his GarshneBlanker and for looking through the
english documentation of PopShell,
Oliver Bausch and Frank Sautter for the oMniBus VGA system

and to all the others who help(ed) to make the Amiga so unique.