

ShellMenus

Mark Ritter

Copyright © Copyright 1993-1996 By Mark Ritter

COLLABORATORS

	<i>TITLE :</i> ShellMenus		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Mark Ritter	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ShellMenus	1
1.1	Shell Menu Documentation	1
1.2	Introduction	1
1.3	System Requirements	2
1.4	Using Shell Menus	2
1.5	Starting Shell Menus	2
1.6	Shell Window	3
1.7	Menu Strip	4
1.8	Project Menu	4
1.9	User Menus	5
1.10	Menus Editor	6
1.11	Requesters Editor	8
1.12	Req Structures	8
1.13	SomePatterns	9
1.14	Arg Structures	10
1.15	Prf Structure	10
1.16	Movement And Control	11
1.17	Preferences Editor	12
1.18	Windows	12
1.19	Format	13
1.20	Keys	13
1.21	Paths	14
1.22	control	15
1.23	Executing Commands	16
1.24	Terminating Shell Menu	16
1.25	GadTools	17
1.26	Locale Support	17
1.27	Operation Notes	17
1.28	Hints And Tips	17
1.29	Reporting Bugs	18
1.30	Incompatibilities	18
1.31	Distribution And Legal Notices	19
1.32	Program History	19

Chapter 1

ShellMenus

1.1 Shell Menu Documentation

Shell Menus

By: Mark Ritter

Version: 3.0 Released: 20.1.95

A Swiss Army Knife Utility

Copyright © 1992-1996 All Rights Reserved

[Introduction](#)

[System Requirements](#)

[Using Shell Menus](#)

[GadTools](#)

[Locale Support](#)

[Operation Notes](#)

[Hints And Tips](#)

[Reporting Bugs](#)

[Incompatibilities](#)

[Distribution And Legal Notices](#)

[Program History](#)

1.2 Introduction

Do you get tired of typing long path names for programs that are not in the directory you are currently in? Do you forget arguments for programs? Do you hate flipping through Workbench drawers, waiting for all the icons to be displayed, just to find one icon? Is your Workbench cluttered with many open drawers because your most often used programs are in different drawers? Do you forget which function key and qualifier has the macro you want? Is your hard-drive root directory packed with files and their icons? If you answered yes to any or all of these, then you need Shell Menus.

Shell Menus is a program to help Shell users be more prolific and save time. Shell Menus creates user definable menus that attach to the Shell window. Shell Menus offers an intuition interface to edit your menus instead of a text editor and many other features not found in other programs that use menus.

There are other programs that attach menus to the Shell window, like Csh, but they are not nearly as advanced as Shell Menus. They only execute programs that do not use command line arguments, or just ones that were written into the script that can't be changed without using a text editor.

Shell Menus can open a window to enter command line arguments before launching the program. It can also open a file requester to select a file(s) to use as an argument. Or they can be mixed for maximum command line editing. A good example of this is using a picture viewer with a variable time delay and selecting one or more pictures to view.

1.3 System Requirements

This program requires at least version V39 of the operating system software. This is to take advantage of the new commands and new tags. It is now easy enough to update your system so it should not pose a problem.

This program requires less than 512K of RAM, therefore it should operate on all systems.

To use the full potential, a harddrive is recommended but not required.

1.4 Using Shell Menus

[Starting Shell Menus](#)

[Shell Window](#)

[Menu Strip](#)

[Menus Editor](#)

[Requesters Editor](#)

[Preferences Editor](#)

[Executing Commands](#)

[Terminating Shell Menus](#)

1.5 Starting Shell Menus

This program may be started from the Workbench or the Shell window.

This program will detach from the Shell so you need not use Run or any other variant. There are a few command line arguments you may use when loading:

OPENSHELL/S

This will open a new Shell window as one of the first things done.

This will allow Shell Menus to be started from the Workbench, or WBStartup, more easily since it will not stop and tell you that there is no Shell window open. If there is already a Shell window open with the same name, "AmigaShell", it may find that window first and use that window instead of the newly opened one.

ex: ShellMenus OPENSHELL

PUBSCREEN/K

Use this argument to inform Shell Menus that the desired Shell window to use is located on a screen other than the default public

screen, usually Workbench.

ex: ShellMenus PUBSCREEN OtherScreen

CUSTOMSHELL/K

Use this option to specify a name other than "AmigaShell" for the Shell window to use. Some programs change the window name, this allows you to use Shell Menus with these programs. The name need not be case sensitive.

ex: ShellMenus CUSTOMSHELL "Csh V5.15L"

PREFSFILE/K

This option is used to load a prefs file other than the default prefs file. You must specify the complete path and name.

ex: ShellMenus PREFSFILE ENVARC:MyPrefs.prefs

EXECUTE/K/N,/N

This will execute the program at the specified menu and item numbers. Both the menu and item numbers must be specified for this to work. If used in the icon, then the menu and item numbers must be separated by `|` in order to work.

ex: ShellMenus EXECUTE 5 20

ex: EXECUTE=5|23 ex: EXECUTE=10|5 (icon tool type)

OPENEDITOR/S

This option will open the Menu Editor when done loading the prefs file and, if specified, the program that was to be executed.

ex: ShellMenus OPENEDITOR

LANGUAGE/K

Preferred locale strings language.

Language must be all lower case.

The arguments are not case sensitive and may be mixed in any order.

ex: ShellMenus OPENEDITOR EXECUTE 10 2

If there is no Shell window open, Shell Menus will offer to open one for you. The window is required to be open at all times while Shell menus is running.

Only one copy may be run at one time. You will be notified if more than one copy is attempted to be executed.

1.6 Shell Window

The Shell window must have the title "AmigaShell" for Shell Menus to find it and attach the menus. If it can not find the window, it will offer to open one for you or you may specify a different name.

Shell Menus will modify the IDCMP flags of the window to receive the Intuition messages. It will do this in a way that is compatible with other programs that might be using the Shell also.

When a program is launched by Shell Menus, the menustrip is disabled but the window is not blocked so input may still be entered while another program is executing.

Clicking the close gadget once will terminate Shell Menus. Clicking it again will close the window. The EndShell command will close the window immediately but will not notify Shell Menus to terminate.

1.7 Menu Strip

Project Menu

User Menus

1.8 Project Menu

Edit Menus...

This will open the Menus Editor window to allow editing of the menus and/or the preferences. If the window is already open it will be brought to the front and made active.

Load Preferences...

This will open a file requester to load the prefs file. A requester will open if you made changes to any of the preferences asking if you are sure so you will not lose any changes.

Save Preferences...

This will open a file requester to save the prefs file.

NewShell...

This will open a Shell window using your defined dimensions or the system defaults.

Launcher...

This is a handy feature that opens a file requester to allow you to launch a program without having to type in the complete path yourself. A useful time saver.

Video Mode

Selecting one of the 4 sub-items will change the video display mode to either NTSC or PAL. See Paths in the Preferences Editor for file paths and names.

NTSC

Change the display mode to NTSC immediately.

PAL

Change the display mode to PAL immediately.

Boot NTSC

Change the display mode to NTSC after the next system reset.

Boot PAL

Change the display mode to PAL after the next system reset.

Format Disk...

Selecting this item will open a requester asking for the new disk name. Quotes will automatically encase the disk name so spaces may be included in the name.

DF0:

Format a disk in floppy drive 0.

DF1:

Format a disk in floppy drive 1.

DF2:

Format a disk in floppy drive 2.

DF3:

Format a disk in floppy drive 3.

Other...

Format a disk or device other than a floppy. A requester will be opened before entering the disk name asking for the device name.

Ex: DH0:, DH1:, RAD:, etc.

Open Options Window

This will open a window before formatting to allow you to change the default settings. The new settings will then be the new defaults each time it is opened.

The system default is not checked.

About...

Will open a large requester with some pertinent information about the program.

Quit

This will terminate the program.

See "Terminating Shell Menus" for more details.

1.9 User Menus

There are 10 user menus with 50 items each for a total of 500 items.

The title of each menu is user-definable and can be up to 12 characters long.

The title of each menu item may be up to 20 characters long.

The width of each menu will adjust to the longest text in that menu. The titles will also adjust accordingly.

If the screen is not tall enough, for a number of reasons, the menus will conform and use multiple columns. If the screen is not wide enough, the last menu will be adjusted back to fit.

If a large font is used, some menus may be completely cut off. This will be corrected if used on a super hi-res screen.

`---' for the menu text may be used to use GadTools' NM_BARLABEL for a divider.

1.10 Menu Editor

This is the main control panel and where you edit the menus.

ListViews

There are 3 ListView gadgets: Menu Items, Keys, and Commands with 51 entries each.

Menu Items

This is the text in the menus. The first line is the title of the menu, the rest are menu items.

Keys

These are the menu short-cut keys.

Commands

These are the command lines with the program path and all structures, if any.

Strings

All normal string editing functions are available.

Menu Items

The title has a maximum length of 12 characters. Menu items have a maximum length of 20 characters.

Keys

Only 1 character is allowed for short-cut keys. Shell Menus will check for duplicate keys and notify you if a key was repeated.

Commands

To allow the use of multiple structures the maximum length is 512 characters.

FR

To ease the entry of paths use the file requester. All structures will be remembered so they do not have to be recreated.

Menu x

This gadget will cycle through the menus. Holding the SHIFT key while pressing the mouse button will cycle backwards.

Close Menu Editor

This will close the Menus Editor but not terminate. To re-open the editor you have to choose the menu item for press the short-cut key in the Shell window.

Clear This Item

This will clear the currently selected item. It will not be

removed.

Clear This Menu

This will clear the title and all items of the current menu. It will not be removed. A requester will ask for confirmation before proceeding.

Clear All Menus

This will clear all menus. They will not be removed. A requester will ask for confirmation before proceeding.

Swap Items

Click this gadget to toggle swapping on and off. You will be asked to select the item to be swapped. You will then be asked to select the item to swap with. Swapping may occur in the same menu or across menus. Entire menus can be swapped by selecting the titles.

Copy Items

Click this gadget to toggle copying on and off. You will be asked to select the item to be copied. You will then be asked to select the item to copy the first item to. Copying may occur in the same menu or across menus. Entire menus can be copied by selecting the titles.

Insert Items

Click this gadget to toggle inserting on and off. You will be asked to select the item you wish to have a blank item inserted in front of. A menu may be inserted by selecting the menu title.

Delete Items

Click this gadget to toggle deleting on and off. You will be asked to select the item you wish to delete. A menu can be deleted by selecting the menu title.

Requesters Editor

This will open the Requesters Editor. See "Requesters Editor" for more details.

Preferences Editor

This will open the Preferences Editor. See "Preferences Editor" for more details.

Load Preferences

This will open a file requester to allow the loading of a prefs file. A requester will open if you made any changes to your preferences.

Save Preferences

This will open a file requester to allow the saving of a prefs file.

Quit

This will terminate the program.

See "Terminating Shell Menus" for more details.

1.11 Requesters Editor

Req Structures

Arg Structures

Prf Structures

Movement And Control

This is where you edit the structures for command lines. There are 3 types of structures: Req (file requester), Arg (arguments), and Prf (preferences).

Structure Position: 0/0

Displays your current position and how many structures are in this command line.

Type: xxx

Clicking this gadget will change the current structure to a different type. Holding the SHIFT key while pressing the mouse button will cycle backwards.

NOTE: Changing types will clear all fields!!!

Filename

The filename of the program you are editing for is displayed at the top of the window.

1.12 Req Structures

This type of structure opens a file requester to select files, paths, devices, etc. To open a plain file requester use an empty structure: {req}. If fields are defined then the structure will look similar to this:

```
{req[TTL=Select a picture(s)][PTH=DH0:Pics/][MLTI][TXT=View]}
```

Title

The title text for the file requester.

The maximum length is 100 characters. [TTL=...]

Path

The starting path at which the file requester will start at when the program is executed.

The maximum length is 255 characters. [PTH=...]

FR

Use this gadget to ease the entry of paths and filenames. It will use the pattern and the status of the Pattern and Display Pattern gadgets.

File

Filename to be displayed in the file string gadget. Can be used to designate the default filename.

The maximum length is 30 characters. [FLE=...]

Pattern

The inclusive or exclusive pattern matching. [Some Patterns](#)

The maximum length is 50 characters. [PAT=...]

Display Pattern

If this gadget is checked the pattern gadget will be displayed in the file requester. If not checked it will not display the pattern gadget. This can be used to force a pattern to be used, such as, #?.prefs.

The system default is checked. [NOPAT] if not checked.

Save File

Check this gadget to allow the saving of a file and/or to allow the creation of directories. This can not be used with Multi-File and will be disabled appropriately.

The system default is not checked. [SAVE]

Multi-File

Check this gadget to allow multi-file selection. This can not be used with Dirs Only and will be disabled appropriately. Only one Multi-File is allowed per command line.

The system default is not checked. [MLTI]

Directories Only

This will open a requester with no files, just directories. This can be used to designate a storage directory. Check Save File to be able to create new directories. This can not be used with Multi-File since no files are displayed.

The system default is not checked. [DIRS]

OK Text

The text displayed in the OK gadget of the requester.

The maximum length is 15 characters. [TXT=...]

1.13 SomePatterns

? Matches any single character.

#<p> Matches 0 or more occurrences of <p>.

<p1>|<p2> Matches if either <p1> or <p2> matches.

~<p> Matches everything but <p>.

(<p1>|<p2>...) Groups items together.

% Matches the null string.

[<p>-<p>] Delimits a character range.

A?B Matches any 3 letter name beginning with A and

ending with B, such as AcB, AzB, and alb.

A#BC Matches any name beginning with A, ending with C, and having any number of Bs in between, such as AC, ABC, ABBC, ABBBC, and so on.

A#(BC) Matches any name beginning with A and followed by any number of BC combinations, such as ABC, ABCBC, and ABCBCBC.

A(B|C)D Matches ABD or ACD.

ABC#? Matches any name beginning with ABC, regardless of what follows, such as ABCD, ABCDEF.info, or ABCXYZ.

#?XYZ Matches any name ending in XYZ, regardless of what precedes it, such as ABCXYZ and ABCDEFXYZ.

[A-D]#? Matches any name beginning with A, B, C, or D.

~(XYZ) Matches anything but XYZ.

~(#?XYZ) Matches anything not ending in XYZ.

A(B|D|%)#C Matches ABC, ADC, AC (% is the null string), ABCC, ADCC, ACCC, and so on.

1.14 Arg Structures

This type of structure opens a window to allow the input of command line arguments. A window with no displayed text can be opened by using an empty argument structure: {arg}. If fields are defined then the structure will look similar to this:

```
{arg[MES=This is the displayed text][ARG=args in string gadget]}.
```

Message

The text that will be displayed in the argument window to let the user know what input is expected.

The maximum length is 75 characters. [MES=...]

Arguments

The default command line arguments for the program.

The maximum length is 75 characters. [ARG=...]

1.15 Prf Structure

This type of structure allows you to have different types of preferences for each program, or use the defaults. This structure must always be the last structure in the command line. The defaults are Multitask and Auto-Close. The structure will look similar to this:

```
{prf[DIM=CON:0/11///A Program's Output/AUTO][MULTITASK][AUTOCLOSE]}.
```

Dimensions

The output window dimensions and name.

The maximum length is 255 characters. [DIM=...]

No Multiple Files

Check is gadget if a program can not handle multiple filenames in the command line, only one at a time. If it is not checked as many filenames will be used in the command line as possible.

The system default is not checked. [NOMULTIPLEFILES]

Multitask

Check this gadget if the program is multitaskable. Programs that open their own window or screen and do not need get passed arguments can use this. Most Shell based programs can not use this since most need arguments, unless Multi-File is not used.

The system default is checked. [MULTITASK] or [SYNCHTASK]

Auto-Close

Check this gadget if you want the output window to close immediately after the program terminates or wait until the user presses the return key or closes the window.

The system default is checked. [AUTOCLOSE] or [WAITCLOSE]

Same Window

Check this gadget if you want to use the same output window for all Multi-File operations instead of opening a new one for each file.

The system default is not checked. [SAMEWINDOW]

Clear Window

Check this gadget if you want the output window cleared between files in Multi-File operation.

The system default is not checked. [CLEARWINDOW]

1.16 Movement And Control

Use

Finished editing. Return to the Menus Editor.

Undo

Restore all fields of the current structure if you have not changed the type or moved any direction.

Cancel

Ignore the changes of the current structure and return to the Menus Editor.

Next

Move to the next structure. All defined fields are copied to the structure.

Previous

Move to the previous structure. All defined fields are copied to the structure.

Append

Add an empty structure based on the Type: xxx gadget to the end of the command line but before the Prf structure. You will have to move to it to edit it. If you are at the Prf structure then this gadget is disabled; use Insert instead.

Insert

Insert an empty structure based on the Type: xxx gadget in front of the current structure and become the current one.

Clear

Clear all fields of the current structure.

Delete

Removes the current structure from the command line. The Prf structure can not be deleted.

1.17 Preferences Editor

Windows

Format

Keys

Paths

Control

1.18 Windows

Console

Default size and name of the output window.

The maximum length is 255 characters.

The system default is

"CON:0/11//189/Shell Menus Output Window/AUTO"

NewShell

Default size and name of the NewShell window.

The maximum length is 255 characters.

The system default is "NewShell".

1.19 Format

Fast File System

This will format the disk using the Fast File System.

The system default is checked.

International Mode

This will format the disk using the international versions of the file system. This mode is not available if running on a Kickstart version less than 39.

The system default under Kickstart 39 or higher is checked.

Directory Cache

This will format the disk using caches for faster directory reads.

This mode is not available if running on a Kickstart version less than 39.

The system default under Kickstart 39 or higher is checked.

Put Trashcan

This will create a Trashcan icon on the disk.

Quick Format

This will erase the root block (and track) and the boot block (and track), and create and bitmap blocks. This is useful when formatting a previously formatted disk.

No Verify

This option is not available with the supplied Format program from Commodore. It is other programs that support this. The command line argument is NOVERIFY.

Install Disk

This will install the disk to make it auto-booting.

Open Options Window

This will open the Options Window before each format to allow the user to change the Format options. These new settings will be used as the new defaults.

1.20 Keys

Edit Menus

Short-cut key to open the Menus Editor window.

The system default is E.

NewShell

Short-cut key to open a NewShell window.

The system default is N.

Launcher

Short-cut key to open the Launcher file requester.

The system default is L.

NTSC

Short-cut key to load the video display mode file NTSC.

PAL

Short-cut key to load the video display mode file PAL.

Boot NTSC

Short-cut key to load the video display mode file Boot NTSC.

Boot PAL

Short-cut key to load the video display mode file Boot PAL.

Format DF0:

Short-cut key to format disk in floppy drive 0.

The system default is 0.

Format DF1:

Short-cut key to format disk in floppy drive 1.

The system default is 1.

Format DF2:

Short-cut key to format disk in floppy drive 2.

The system default is 2.

Format DF3:

Short-cut key to format disk in floppy drive 3.

The system default is 3.

Quit

Short-cut key to terminate Shell Menus.

The system default is Q.

1.21 Paths

Preferences

This is where Shell Menus will look to find the prefs file. You can use the FR gadget to ease entry.

The maximum length is 255 characters.

The system default path is "ENVARC:ShellMenus/"

NTSC

This is where Shell Menus will look to find the video display mode file for NTSC. You can use the FR gadget to ease entry.

The maximum length is 255 characters.

The system default path is

```
"ENVARC:ShellMenus/NTSC" {prf[MULTITASK][AUTOCLOSE]}
```

PAL

This is where Shell Menu will look to find the video display mode file for PAL. You can use the FR gadget to ease entry.

The maximum length is 255 characters.

The system default path is

```
"ENVARC:ShellMenus/PAL" {prf[MULTITASK][AUTOCLOSE]}
```

Boot NTSC

This is where Shell Menu will look to find the video display mode file for Boot NTSC. You can use the FR gadget to ease entry.

The maximum length is 255 characters.

The system default path is

```
"ENVARC:ShellMenus/BootNTSC" {arg[MES=Options for Boot NTSC: i=Install r=Remove][ARG=i]} {prf[MULTITASK][AUTOCL
```

Boot PAL

This is where Shell Menu will look to find the video display mode file for Boot PAL. You can use the FR gadget to ease entry.

The maximum length is 255 characters.

The system default path is

```
"ENVARC:ShellMenus/BootPAL" {arg[MES=Options for Boot PAL: i=Install r=Remove][ARG=i]} {prf[MULTITASK][AUTOCL
```

Format

This is where Shell Menu will look to find the Format program. If you use a program other than the one supplied by Commodore, the command line arguments must be the same or it will not work.

The maximum length is 255 characters.

The system default path is

```
"SYS:System/Format"
```

AmigaGuide help file

This is where Shell Menu will look to find the AmigaGuide help file when you press the Help key.

The maximum length is 255 characters.

The system default path is "PROGDIR:ShellMenus.guide"

1.22 control

Use

Finished editing the preferences. Will return you to the Menu

Editor using the new preferences.

Undo Section

Restores the current preferences to what they were when you first entered the Preferences Editor.

Undo All

Restores all the preferences to what they were when you first entered the Preferences Editor.

Section Defaults

Use the system defaults for the current section.

All Defaults

Use the system defaults for all preferences sections.

Cancel

Ignore the changes to the preferences and return to the Menus Editor.

1.23 Executing Commands

There are 2 ways to execute a command: pick the menu item or use the short-cut key. Your user defined preferences will be used. If none were, then the defaults will be used.

During execution all input is blocked until the program is finished loading and does not require an output window or until the program is terminated if not multitasking.

To allow spaces in the path and/or filename, Shell Menus encases the path and program name quotes. If you are using Multi-File then the filename will also be encased in separate quotes. If you are using an argument window to ask for a specific filename then you must include your own quotes since command line arguments are not encased in quotes.

Ex: "DH0:My Utilities/Super View" -s3 "DH0:Pics/Mona Lisa"

BUT, if you specify some command line arguments that you know are required and do not put them in an argument structure, then you will have to tell Shell Menus not to put them inside quotes with the path and program name.

Ex: DH0:Archives/DMS Read {arg[MES=Enter path and name]} FROM {arg[MES=Enter device to read from]} CMODE HEAVY2 LOW 0 HIGH 79

Normally, the quotes would be placed after Read causing DMS to run improperly. To fix this, you have to use a Stop Quotes Here structure: {""}.

Ex: DH0:Archives/DMS{""} Read {arg[MES=Enter path and name]} FROM {arg[MES=Enter device to read from]} CMODE BEST LOW 0 HIGH 79

Now the end quote will be placed after DMS and the Read argument will be usable.

Selecting the Cancel gadget in the file requester or argument window will abort execution of the command line. Pressing the return key in an empty name gadget in the file requester will also abort execution.

Shell Menus is smart enough to know if a program needs an output window or not. If you close the window before execution is complete it will automatically be re-opened.

The output window does allow for user input, but not if the Multitask flag is set. This is to ensure that programs like DMS and Format will wait for the user to manually start the function to not start automatically and destroy data, i.e., format the wrong disk, DMS Write over the wrong disk, etc.

1.24 Terminating Shell Menus

Before terminating Shell Menus will notify you via a requester if you made any changes to your preferences and give you a chance to save them first.

Shell Menus will reset the Shell window IDCMP flags in a way that will not bother other programs that might be using the Shell window.

1.25 GadTools

Thanks to Olaf `Olsen` Barthel, author of gtlayout.library. It is by far the best user interface package.

The user interface is totally font and screen mode sensitive. All gadgets and the window will automatically adjust to match your preferences.

The latest features are utilized to get the most from the operating system and to make the program the best of it's kind.

1.26 Locale Support

This program is totally localized to allow support for foreign languages. A catalog may not be available for your language at this time, but one will hopefully be available soon.

If you are willing to translate the strings to another language, please let me know. If you do offer, a perfect translation of all strings will be expected. It is all or nothing. I will also be seeking your assistance for future versions that have new strings.

· AFLE for French catalog and AmigaGuide.

1.27 Operation Notes

Press Help for AmigaGuide help.

Shell Menus puts a lock on the Workbench screen to prevent it from being closed. If you need to close it, for a screen mode change or more memory, you will have to terminate Shell Menus first.

If, after running a Shell program, you lose control of Shell Menus, it might be because the program changed the IDCMP flags of the Shell to it's settings, not caring if another program is also using them. See the Incompatibilities section for more details.

Just a reminder: If using Multi-File, there can only be one per command line. If you manually add one or more, the first one encountered will be the one that gets used.

You can not use these strings in your command line because they are structure flags: {""}, {req}, {req[, {arg}], {arg[, {prf[,][, or]]}.

The Multi-File gadget in the Requesters Editor determines how Save, Directories Only, and Multitask will be set. If Multi-File is checked, then the above gadgets will be unchecked and disabled.

When using the FR gadget in the Menus Editor in the Paths section of the Preferences Editor, all command line structures will be remembered so they will not have to be recreated.

The menu strip is laid out using visual information from the default public screen, usually Workbench. Shell Menus' screen will clone the screen when the Menu Editor is opened.

1.28 Hints And Tips

To clean up your harddrive you should group your files in sub-directories like: MyUtilities, Programming, Music, Graphics, Archives, etc. Then with Shell Menus you will not have type in the long path names, just pick the menu item.

Put Shell Menus in the WBStartup drawer and use OPENSHELL to open a Shell window. You will have to use the DONOTWAIT tool type argument.

You can switch to any menu by pressing any numeric key either on the main keyboard or the numeric pad. 0 is menu 10.

If a program allows the use of #? in the command line, then you can also specify this in the file requester by typing #? in the filename string gadget and clicking on the OK gadget. You can not press ENTER and have it be accepted.

1.29 Reporting Bugs

I try to make sure all new additions or changes work correctly before releasing a new version, but, something might get by. If you encounter a bug or a problem with hardware or other programs, please let me know. Do not assume it will be fixed in the next version because if no one tells me, I may not find it.

Please be very specific about what the bug does, when it does it, how you found it, and how I can re-create it to test it. Also let me know what your system consists of: Kickstart version, hard-drives, memory expansion, accelerator card, chip set (original, ECS, AGA), etc.

If you find a program that is not compatible with this program, please send it to me on a disk or via electronic mail.

Programmer: A red-eyed mammal capable of conversing with inanimate objects.

I can be reached at:

Mark Ritter

2008 Valley Vista Dr.

Bettendorf, IA 52722

USA

FIDO: 1:232/6.0

InterNet:

1.30 Incompatibilities

Shell Menus is not compatible with these programs or some of their features:

Csh (Matthew Dillion)

Csh changes the name of the Shell window so you will have to use the CUSTOMSHELL argument. Shell Menus must be executed AFTER Csh is executed since Csh clears the menustrip when it is first executed.

If you are using Csh's menus, then you will like Shell Menus's advanced features even more. You can use Csh's commands and Shell Menus's menus.

Console Buffer (CB) (William J. Fritz)

CB also takes over the Shell window, but in a different way. It modifies the IDCMP flags in an extremely bad way. It takes no consideration for other programs by turning off their flags and setting only the one's it needs. When you quit CB it does not turn the other program's flags back on.

LhA ? (Stefan Boberg)

LhA uses a unique way to use the console.device to use a Paused. Press any key to continue... that is not compatible with the way Shell Menus executes programs or the output window. When LhA should pause at a full page of output it doesn't since it uses its own input/output handle. You must use Ctrl+S and Ctrl+Q

to pause and continue the output.

Format 40.1 (CBM)

The Format program, BETA version 40.1, operates poorly in the output window. To still have the program wait until the user presses RETURN to start formatting, no text output is displayed.

Pressing RETURN is still required to start formatting.

1.31 Distribution And Legal Notices

This program may be freely distributed in it's original form as long as no more than a nominal fee is charged to cover time and copying costs.

This program may be placed on any Public Domain distribution disk without prior consent from the author. A copy of the disk would be appreciated.

This program may not be released with a commercial product without the author's written consent. No exceptions.

This program is NOT Public Domain, it is Freeware. Please respect it.

Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event, unless required by applicable law or agreed to in writing, will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

1.32 Program History

Version 1.0 Completed: 00-Sep-92

- ~~~~~
- Preliminary construction.
 - Completely re-written from original attempt.
 - Beta testing and code improvements.
 - Update from SAS/C 5.10b to SAS/C 6.00
 - Menu preferences:

5 Menus with 10 items each:

Title: 10 characters

Text: 15 characters

Command: 255 characters

Version 1.1 Completed: 00-XXX-00

- ~~~~~
- Even more code improvements.
-

- Added new Project menu items:

Project

Edit Menus... E

Load Prefs

Save Prefs

~~~~~

NewShell

Launcher... L

~~~~~

Video Mode » -----> NTSC

Format Disk » -----> DF0: PAL

~~~~~ DF1: ~~~~~

Quit Q DF2: Boot NTSC

DF3: Boot PAL

~~~~~

FFS

NOICONS

QUICK

~~~~~

Install

Load and Save Prefs use the default path and name. You must use the interface to specify a different one.

NewShell will open a new Shell window. You may edit line 2 of the prefs file to either `NewShell` or a custom string like:

`NewShell "CON:0/100/100/100/Blah/CLOSE"`.

Launcher is another little time saver. It will open a file requester allowing you to choose a program and have SM launch it for you. This is handy to load a program that is on a different volume or imbedded in multiple directories.

Good For: "DF3:Downloads/NewDemo.exe" or  
"DH0:Programming/.C\_Files/My\_Prog/My\_Prog1.0"

Format Disk is just what it says. You can select certain options if you so desire. By default, FFS and NOICONS are checked. To format a disk, select the device from the menu. Multiple formats can be going simultaneously.

NOTES: Selecting only install will still format the disk with the OFS and put a Trashcan icon on the disk. You must type "Install DFX:" in the shell. I will fix this in a later version, I don't know how just yet and not double-up the drive sub-menus.

The disk name is line 7 of the preferences file. Quotes ARE necessary if you have a space in the name.

Video Mode allows you to switch between NTSC and PAL via external programs. Boot NTSC and Boot PAL set the ROM Tags and after a reset the appropriate mode will be used. This is very useful for European megademos that have custom boot blocks.

Paths: The programs are found in the default preferences path and may be substituted for any of your choice. To do so, you must edit your preferences file with a text editor. Lines 3 through 6 are the program paths and command line arguments.

- Fixed bug in load and save code.

Version 1.2 Completed: 25-Oct-92

~~~~~  
- Added additional menu items.

SM now has more menu items for users that use many programs under the same category. There are now 5 menus with 20 items each:

Title: 10 chars

Text: 20 chars

Command: 255 chars

This allows five additional menu-item text characters and ten additional menu-items. This should satisfy all users. It will also allow for more sections of similar programs, such as a diskcopy section of all combinations of drives in the bottom portion of a ¼ filled menu that was sitting empty.

- Created a new Graphical User Interface (GUI).

There are now two LISTVIEW_KIND gadgets, one for menu text and one the commands. The first item in the text gadget is the menu name, it is blank in the command gadget. All twenty items are in the gadgets with the menu title and ten items visible; all others may be scrolled into view.

Clicking on an item or command will activate the string gadgets and place the item and command in them. Menu title's are selected the same way. Once an item and command are selected, the text gadget is selected for input.

Pressing RETURN will jump to the command gadget. Pressing RETURN here will enter the menu item into the menu and gadget and de-activate the string gadgets.

Below the string gadgets are two new gadgets: Swap Items and Clear Items.

These are only active if a menu item is active in the string gadgets.

'Swap Items' is a check-box gadget that, when checked, will allow you to swap the positions of two items in the same menu. After checking, click on the first item, then click on the item to switch with. The

gadget will stay checked for multiple swaps. You must un-check the gadget to stop swapping and/or to change menus.

`Clear Item' will erase only the correct item text and command strings, not the entire menu.

- Fixed font and centering bug.

SM will now handle larger title bar fonts with more than 8 pixels. The window will be adjusted and sized correctly to fit on the Workbench screen like the Workbench window does: Title bar is placed below screen title, height is adjusted for larger fonts, if font is too large the window will be moved up to accommodate.

- Fixed a major menu text bug that plagued SM from day one. SM could not have been released until that problem was fixed.

Version 1.3 Completed: 26-Oct-92

~~~~~  
- If no Shell window is open or one not with the title `AmigaShell', a requester will now open asking if you would like one opened. The NewShell uses your system default dimensions.

- Fixed a problem with the Menu cycle gadget. If you clicked it while holding either shift key the gadget would cycle backward correctly, but the menus would still cycle forward. This now works correctly.

- Menu titles may now be twelve characters long. Ten was always a few short for my needs, so I added two more.

- A requester now asks if you are sure about clearing one or all menus.

- SM now checks if you made any changes to your preferences before loading or quitting via a requester.

Version 1.4 Completed: 02-Nov-92

~~~~~  
- Selecting Load or Save from the menu now opens a file requester like selecting one of the gadgets does. The menu always saved to the default prefs file. This caused some problems if you wanted to save to a different file and chose the menu Save instead. No longer.

- Fixed a few small bugs and sped up part of the code.

- Added an intuition interface for path preferences. Selecting the new `Paths' gadget from the main window will open a window with all user definable options for easy editing. Using a text editor is no longer necessary to change any preferences. Quotes are still necessary if there is a space in a command path or `Disk Name'; arguments are an exception.

- Added the use of the Asl file requester via an intuition interface. Selecting the new `Requester' gadget from the main window will open a window with all

possible requester options.

`Title` is for the requester title bar.

`Path` is the default path to use.

`Name` is the default name to use. If a filename is specified, then the

`Directories Only` option is disabled.

`Pattern` is for including/excluding files. If a pattern is specified then the `Directories Only` option is disabled.

Some Patterns:

- #? Everything
- #?.info Only files ending in ".info" (icons)
- (#?.info#?.doc) Only icons and doc files
- ~(#?.info) Everything EXCEPT icons
- ~(#?.info#?.c#?.iff) Only icons, c files, and IFF files
- Sea#? Files starting with "Sea"
- #?Game#? Anything with "Game" in it
- [d-l]#? Only files between D and L
- ~[j-m]#? Everything EXCEPT file between J and M

`Save/Create File` will create a file or directory if it doesn't exist.

`Directories Only` will only display directories. This options disables

`Name` and `Pattern`.

`OK Text` is the user definable text for the OK gadget.

You can leave all options blank to open a requester to quickly select a file to pass it to the program.

- Fixed some small bugs with some of the string gadgets.
- Fixed/Added the ability to have arguments after the requester structure.

There is still the problem with not having the ability to select multiple files, for example, to create a complete LhA archive in one step. What needs to be done is:

- 1) Use the requester to select the destination and archive name,
- 2) Append the source path to the command line.

ex: LhA -2 a {req[PTH=VDK:][SAVE][TXT=Save]} DH0:Uploads/#?

Same for DMS disk archiver. The source disk will always be what you have specified in the command line, unless you change it.

Version 1.5 Completed: 06-Nov-92

~~~~~

- Can now swap menu items from different menus. This should have been done originally, so now here it is.

- Fixed these bugs in the Asl requester:
  - Now correctly adjust the the height for all flags.
-

- Now uses the correct path for different requester.

Added this new feature:

- If SaveFlag is set it now checks if the file already exists and asks if you wish to overwrite it or not.

- After alot of testing with the requester, multi-select can be used in some situations. To view an entire directory of pictures, for example, open the requester to the desired directory and use a space, " ", for the filename.

This will only work if the program supports reading an entire directory.

- Added command line arguments:

`-O` Will open the main window after loading.

`-P name` Will load the specified prefs file. Complete path is required.

`-E # #` Will execute the command for the specified menu and item.

Version 1.6 Completed: 09-Nov-92

~~~~~  
- Added a new gadget to the Paths window: `Prefs Path`. This gadget allows the user to put the prefs file where-ever he may want. The default is:

:Prefs/Env-Archive/ShellMenus

In previous versions the path was ENVARC:ShellMenus which put the file in RAM: instead to disk. I had changed my assign to put it on disk in the default path.

- Due to an error on my part and lack of testing a function, I missed a major bug dealing with the Shell window. SM detached from the Shell window so it can be closed, but SM wouldn't quit. Since I can not find a workable way to check if the Shell window was closed, I added a port checker.

When a duplicate copy of SM is executed, a requester opens with two new gadget options:

`Quit` will terminate both copies of SM. It will still check for changes and allow you to save them before quitting.

`Pop Up` will bring up the active window, or if none are open, will open the main window. This will allow you to save and then quit if you closed the Shell window.

`Cancel` will do nothing.

- The default paths are as follows:

Prefs Path: :Prefs/Env-Archive/ShellMenus

Window Dims: CON:0/11/640/150/Shell Menu Output Window/AUTO/CLOSE

Disk Name: Empty

- When `Swap Items` is first selected all string gadgets are turned off so you know you have to pick the two items to swap with, not to confuse with the item in the string gadgets.

Fixed a small "active" bug when swapping.

- The file requester now remembers the previous path for preferences and Launcher instead of always using the original starting path.

- A file header is now put in all preferences files to eliminate loading of wrong files by accident. The header is 'Shell Menus Preferences File' and needs to be put in all old prefs files via a text editor before it can be loaded.

- Added 'FR' gadgets in Path and Requester windows to open a file requester to get a path instead of typing it in. Time saver.

Version 1.7 Completed: 22-Nov-92

~~~~~  
- Fixed small bug in file requester.

- Added a new requester type: {arg}

This type opens a small argument window to allow you to enter command line arguments or edit your defaults. Also allows multi-select of files with a certain pattern.

'Message' is for a string to describe what input is expected and for what file.

'Arguments' is your default command line arguments that can be edited when the item is selected. You must leave a space after the last argument so the file-part is copied over correctly.

- Multiple requester types are now available in command lines. Updated Requester window to accommodate by splitting it into two sections. To edit a requester place the cursor at the beginning of the structure and hit the 'Requesters' gadget to edit it. I hope to add an all intuition interface to edit the structures.

If you specify multiple {requester} structures you must separate them by a space to ensure good results: {req...} {arg} {req...} {req} {arg...}

Version 1.8 Completed: 27-Dec-92

~~~~~  
- Added 'Undo' gadgets to the Path and Requesters windows. This gadget will undo all changes made to the current gadget contents to their original state.

- Added user definable command key shortcuts to all menu items. To get total user definition I also made the Project menu keys user definable. This was done for times when you want to use the hard-coded keys for other programs, such as 'N' for NewZap, instead of NewShell.

Not all Project menu items need to be defined nor are all items definable.

Duplicate keys are checked for at loading and upon entry in the new gadgets.

When loading, duplicate keys are set to NULL with no notification to the user.

If you notice, I purposely left space for these new gadgets.

- Added an `FR` gadget to the main window next to the Command string gadget to ease the entrance of complete paths.
- Added `About...` menu item to Project menu for legality reasons.

Version 1.9 Completed: 17-Jan-93

-
- Fixed a bug in the preferences loading routine that would not create the menu strip if the preferences file was not found.
 - The menu strip width dimensions will now conform to the longest item of each menu and the title width also adjusts to the length of the title. The same maximum lengths are still enforced to ensure ease of menu strip creation without worrying about dimension problems.
 - The Requesters structure editor is now all intuition controlled via six new gadgets:
 - `Next` will move to the next structure if exists, otherwise it is disabled.
 - `Last` will move to the previous structure if exists, otherwise it is disabled.
 - `Append` will add a blank {req} structure to the end of the command line. You will stay at the current structure if one or more exists. If the command line previously had no structure(s) then it will become the active one.
 - `Insert` will insert a blank {req} in front of the current structure and the new blank one will become the active one.
 - `Clear` will clear all gadgets for the particular type of structure.
 - `Delete` will remove the current structure and activate the next one if it exists.
 - Improved the handling of structures when executed. Spaces are no longer necessary between structures and you don't have to leave a space after the last argument in the {arg} structure. You can still leave spaces for ease of reading the command line.

- Created a new preferences editor. It is divided into three sections: Operation, Keys and Paths. The same user definable preferences are now grouped under it's appropriate category for ease of finding and knowing what affect it will have.

Version 2.0 Completed: 08-Feb-93

-
- Entire menus can now be swapped by selecting the menu titles. Also made some updates to the code for better handling of swaps.
 - Created a new type of structure: {prf}. This new structure allows for each program to have it's own defined preferences: output window dimensions,
-

multitasking or single-tasking, and auto-close or input closed output window.

- Created a better way to block the Shell window by disabling the entire menustrip instead of blocking all input.
- Fixed a bug in the -E command line argument that wouldn't attach the menustrip to the Shell window if -O argument wasn't used also.
- Fixed a huge bug in the execute command section. The addition of the {prf} created a problem with command line arguments and helped find an old bug. Also spruced up the creation of the command line.
- Added better error handling when loading and saving the prefs file.

Also put the Keys before the Paths in the prefs file to match the order of the Preferences Editor.

Version 2.1 Completed: 14-Mar-93

-
- Fixed a bug in the menustrip creation process that would lose 5K of RAM. If use the command line argument -O everything was fine. If didn't use it then opened the window later, the menustrip was duplicated but not free'ed when you quit, thus losing 5K of RAM.
 - The prefs file now uses the IFFParse library. Error checking is quite extensive and checks the file if it is a Shell Menus prefs file and of the correct version before clearing any preferences data.
 - Fixed a bug that would try to close the main window when it was already closed when loading a prefs file.
 - Fixed some bugs in the Requesters window that would activate/de-activate the wrong gadgets at the wrong time.
 - Implemented Multi-File selection. At this time it does one file at a time for all programs. The multifile struct MUST be the last structure in the command line. Multitasking is disabled for this options.
 - Updated the structure extractor to allow the use the of brackets and braces ([,], {, }) in the command line. The old version wouldn't allow these, preventing patterns like: [f-m] which would only display names of files that start with letters between F and M.
 - Updated routine to ModifyIDCMP() of the Shell window so it won't interfere with other programs, or at least ones that don't use IDCMP_MENU_PICK and IDCMP_CLOSEWINDOW. IDCMP_CLOSEWINDOW may be deleted at any time. Until I find a better way to check if the Shell was closed, it will stay so you can't click close the window.
 - Fixed two bugs in the Requesters window dealing with the Append and Insert gadgets. The addition of the prefs structures threw off the extraction and re-construction of the command line.
-

- Doubled the string buffer size of the Menu Command from 256 to 512 bytes. It was too easily over filled with only three or four 4 fully defined structures and caused a visit from the guru. If need be, it can easily be increased in future version.

- Added `Other...` sub-item under Format Disk to open an argument window to enter other device names (DH0:, RAD:, etc.) to format.

Version 2.2 Completed: 19-Apr-93

~~~~~

- Updated the handling of multiple copies being run simultaneously because of the Shell window being closed, or not. It now double checks if the Shell window is the one it found when SM was first run so it doesn't try to close the old window and cause a guru. It now also handles POPUP better for the Main, Argument, Requesters, and Prefs windows better and re-attaches the menu-strip correctly.

- Wrote a new documentation file in AmigaGuide format. It wasn't worth writing until now because I made so many additions it would have required making extensive changes to the guide every few versions.

- The command line is now encased in quotes before execution so spaces can be used in paths and filenames.

- Added `Save Then Quit` to the requester before quitting.

- Added StopQuotesHere structure, {""}, for better handling of command line arguments with the quotes encaser.

- Updated loading and saving routines by compressing lines.

- Fixed bug in -P argument that when loading a non-existent file wouldn't attach the menustrip.

- Updated Append and Insert in Requesters Editor. It used to always add a {req} structure, but now it adds one of the same type as the Type: xxx gadget is set to.

- Added MoveScreen() to bring the Workbench and all the custom screens back to the top and left edges in case they were moved.

- Added SetWait() and ClrWait() to use with BlockWindow() and ReleaseWindow() but can't use them until I get 3.0 include files.

- Updated the Requesters Editor. The Type: xxx gadget now include Prf for the preferences. The Prf gadgets are off until moved to and the Control gadgets are disabled accordingly.

If no Prf structure was found in the first scan, then

{prf[MULTITASK][AUTOCLOSE]} is appended.

- After executing a command SM now returns to the correct place:

- CheckIDCMP if picked from the ShellWindow

---

- MainBringUp if picked from the MainWindow
- Broke the 100,000 byte mark.
- Fixed a problem in the ExecuteCommand section. If the user manually put a structure, other than Prf, after a Multi-File structure it would not ignore those structures, Prf included, and put the files right after the Prf structure. It now totally ignores them like it should but still extracts from the Prf structure.
- More updates to ExecuteCommand section.
- Changed Code=NEAR to Code=AUTO in scopts. This fixed a few bugs that had been plaguing the program.

Version 2.3 Completed: 09-May-93

- ~~~~~
- Updated strings and error handling in LoadPrefsCheck:

"complete path"

is not a Shell Menus prefs file.

"complete path"

prefs file is an old version.

Use Convert Prefs to update to the latest version.

- Fixed some bugs in the port checker when use POPUP.
- Fixed another bug in ExecuteCommand.
- Made so input is allowed, again, in the output window to wait for user input to start a function, like DMS waits for disk insertion.
- Fixed bug in FormatDisk: caused by the quotes encasing.
- Updated the Block/Release Shell window code:

```
Depth=BlockShell(ShellWindow,Depth);
```

```
Depth=ReleaseShell(ShellWindow,Depth);
```

- Fixed bug that caused MANY gurus to occur when executing commands.

Original:

```
ifh=NIL;
```

```
if (SystemTags(txtbuf,
```

```
  SYS_Input,ifh,
```

```
  SYS_Output,ofh,
```

```
  SYS_Asynch,TRUE,
```

```
  SYS_UserShell,TRUE,
```

```
  TAG_DONE)==-1) MyEasyRequest(NULL,"Can't open output window\nor problem with the program.,"Continue");
```

Changed:

```
if (SystemTags(txtbuf,
```

```
  SYS_Output,ofh,
```

```
  SYS_Asynch,TRUE,
```

```
SYS_UserShell,TRUE,  
TAG_DONE)==-1) MyEasyRequest(NULL,"Can't open output window\nor problem with the program.,"Continue");  
Updated:  
ifh=NULL;  
if (SystemTags(txtbuf,  
SYS_Input,ifh,  
SYS_Output,ofh,  
SYS_Asynch,TRUE,  
SYS_UserShell,TRUE,  
TAG_DONE)==-1) MyEasyRequest(NULL,"Can't open output window\nor problem with the program.,"Continue");
```

It needed a handle to close or it would cause these gurus:

FileHandle Closed More Than Once

Memory Freed More Than Once

Memory Header Not Located

Address Error

- Commented out "Save Then Quit" so I could release it.

- Changed the default path to ENVARC:ShellMenus/

- Released to the public, finally!

Version 2.4 Completed: 18-May-93

~~~~~  
- Updated the routines to get the path and filename out of CompletePath by making ExtractPath() and ExtractFile() subroutines. Much easier and less code.

- Now uses a standard 640x200 or 640x400 screen. Had a problem with the file requester being too tall so it had to move itself up from the 11th line.

- Changes in Requesters Editor and {req} structures:

- Changed "Name" to "File" and [NME= to [FLE=

- Vertically centered the FR gadget between Path and File

- The FR gadget now gets the Path and File in one step and extracts them to the appropriate gadgets

- Created FindStruct() to ease the search in command lines for structures.

- All command line structures will be remembered for CommandFR and the 4 video mode files so they won't have to be recreated if the file is changed using the FR gadget. This was very unhandy for the video display mode files.

- Fixed a bug in the Requesters Editor that when appended a new structure after making a new one without moving to a different one to create the commandline struct, it would erase the first one's data.

- Created SetItem() to CHECK_ITEM, UNCHECK_ITEM, ENABLE_ITEM, and DISABLE_ITEM menu items quick and easy.

- Updated Format options:

Deleted: All options from Format subitems

Changed: FFS -> Fast File System Default

INTERNATIONAL -> International Mode

DIRCACHE -> Directory Cache

NOICONS -> Put Trashcan

QUICK -> Quick Format

Install -> Install Disk

Added to Operation in Prefs:

Default Format options...

Fast File System Put Trashcan

International Mode Quick Format

Directory Cache Install Disk

Added: Options Window... to Format subitems

This window allows you to change the settings before every format, if set, and/or to become new settings.

These prefs are now set in the menu and saved in the prefs file.

- Added the ability to not display the pattern gadget in the file requester via Pattern On gadget in the Requesters Editor. [NOPAT]

- Got "Save Then Quit" to work finally. It wouldn't work the old way I extracted to the path.

Version 2.5 Completed: 29-Jul-93

~~~~~  
- Now uses gadget images for File and Folder instead of the generic FR gadget text.

- Fixed a bug in the Type: xxx cycle gadget that wouldn't cycle correctly when pressing Shift to go backwards.

- Added:

"Press [RETURN], [C]ontinuous, [A]bort or close window..."

to abort a multi-file execution, and

"Press [RETURN] or close window..."

for non-multi-file non-auto-close.

- Incorporated my GT\_SetMenuAttrs().

- Now checks for what devices exist for the Format submenus and disables the non-existent devices.

- Fixed a small bug when swapping items it would turn Requesters on when it shouldn't, it wouldn't turn off Clear Item at the correct time, made the code smaller.

- The Main and Prefs screens now open to OSCAN\_TEXT width and height and centers

---

the display.

Version 2.6 Completed: 29-Aug-93

- 
- Made better loading and saving routines that uses less code and is faster.
  - The Preferences Editor Undo is now global, not just for the current section.
  - Added CheckForVisitors() for successful closing of the Public Screens.
  - Fixed GetVisualInfo() bug that caused the 'Swap Items' gadget to lose its imagery because it was updated too often.
  - Now uses Workbench font for the menus.
  - Fixed a small bug in Requesters Editor dealing with 'Auto-Close'.
  - Added 'Same Window' and 'Clear Window' to Requesters Editor to allow the user to use the same output window for all Multi-File operations, and to clear the output window between files.  
[SAMEWINDOW] and [CLEARWINDOW]
  - Now spread across multiple files for easy updating.
  - Took out PopUp option when run more than one copy because the modules can't goto ... like they used to.
  - Created new CheckChanges() for a more specific message on what was changed.
  - Created UpdateMenus() and got rid of the stpcpy()'s in CopyToMenus:. This resulted in less code, the ability to use NM\_BARLABEL and only have the shortcut key image displayed for only those items that have a shortcut key.  
'---' is used to define NM\_BARLABEL.
  - Updated string gadgets to handle '---' appropriately: Key and Command off when Text is '---', on when not.
  - Changed the maximums to:  
10 menus  
40 items each
  - Now uses dynamic menus. The menustrip is as long or short as the user has defined. The preset empty items are no longer there for a much cleaner and better looking menustrip.
  - Memory usage is cut down to a minimum since the dynamic menus only allocates the memory it needs.
  - The dynamic menus also changed the prefs file AGAIN to accommodate the change. If was left the original way, the prefs file would be 220K+.

Version 2.7 Completed: 05-Oct-93

- 
- Finally got it internally localized but no other catalogs. Always had the strings but not the code to use them.
  - Now uses CreateNewProc() to detach from the Shell instead of using cback.o.
-

- Fixed a problem with the new dynamic menus causing the listviews' imagery to mess up with empty menus.
- Created ArgumentParser() to use ReadArgs() function for better and more "correct" command line arguments.
- O -> OPENEDITOR/S
- P -> PREFSFILE/K
- E -> EXECUTE/K/N,/N
- OPENSHELL/S (new)
- OPENSHELL/S,PREFSFILE/K,EXECUTE/K/N,/N,OPENEDITOR/S
- Added some checks before executing commands that could cause a problem when use the EXECUTE argument. Mostly it is for cosmetic reasons.
- Now has AmigaGuide support at the press of the Help key.
- When using Multi-File it will now fill the command line to the maximum length with filenames. This speeds the process since it won't have to load the program for every file. This saves a lot of time since it will only have to load the program when the command is full.
- By using the new command line argument CUSTOMSHELL <Name>, other names besides 'AmigaShell' may be used for Shell windows. This allows the use of Csh and such programs that changes the name of the Shell. Shell Menus must be run AFTER these programs.
- OPENSHELL/S,CUSTOMSHELL/K,PREFSFILE/K,EXECUTE/K/N,/N,OPENEDITOR/S
- You can now switch menus to a certain one by pressing the numeric keys on the main keyboard or the numeric pad. 0 is menu 10.

Version 2.8 Completed: 28-Feb-94

- ~~~~~
- Updated to handle new Asl.c module.
  - Updated to handle new GadTools.c module.
  - Updated all the gadget lists to use GadTools.c module. This made Shell Menus 100% locale, font, and screen mode sensitive.
  - New Menu Editor gadget layout. Swap Items checkbox is now a toggleselect button.
  - The Requesters Editor is now divided into separate sections, one for each type of requester and a control panel. Only one type is displayed at a time for easier editing in the new control panel.
  - The Preferences Editor is now divided into more sections and works the same as the Requesters Editor. Many new gadgets and functions included in the Screen section.
  - If a window can't be opened, a gadget list can't be created, etc., then it now still allows the program to be used by cleaning up what was opened of the
-

current editor and continues with just input from the Shell window instead of terminating. Some events will still terminate the program.

- Fixed MANY little bugs/quircks that showed up with the use GadTools.c or were in hiding until I did some updating of certain sections of code.

- Changed all the preference's variables into structs to match up with the Preferences Editor's sections. This will cause the prefs file to be changed and old one's will be incompatible. This new way is much better and organized, so it is worth it.

- Created Configurations.c to handle all the new configs. This includes routines for the Undo buffers, setup the defaults, etc.

- Updated PreferencesEditorSupport.c to handle the new configs. Took out the old ComparePrefs() since it is now obsolete and added some new functions and features.

- Changed: Undo -> Undo Section

Added: Undo All

Added: Use Defaults.

to Preferences Editor.

- Under Kickstart V39 or higher: GTBB\_FrameType, BBFT\_RIDGE, is used instead of a recessed border, which is still used for V38 or less.

- Fixed a problem with Asl.c that wouldn't display the pattern gadget under Kickstart V37 and V38 because the tags changed.

- Completed update to Preferences Editor.

- Fixed a bug that would activate the wrong string gadget when choose an item from the listviews.

- Took a 2 month break. Got frustrated with the new Preferences Editor.

Version 2.9 Completed: 08-Mar-94

~~~~~

- Split Asl.c into three modules:

Asl_GetFile.c

Asl_GetFont.c

Asl_GetScreenMode.c

to avoid wasted space used in programs that don't need all the commands.

- Created CheckForDuplicateKeys(). This cut this code usage by 2/3 and will allow for future updates to be added easily.

- Fixed bug in SegmentSplitter() that plagued it for many months. It now detaches from the Shell correctly without using cback.o or Run/RunBack.

- Moved port checker into EntryPoint(). This is better and requires less code.

- Found a major bug relating to SegmentSplitter() that has been the major cause as to why it wouldn't work in any of my programs. SAS/C's fopen(), fgets(),

fputs(), and fclose() are not compatible with SegmentSplitter() somehow. I switched the DOS library's Open(), FGets(), Fputs(), and Close() and had no problems. It also shaved 1.5KB off the output size.

- Took Screen section out of Preferences Editor. It now clones the default or specified public screen so the menus match exactly. There could be a problem if Shell Menus' screen was smaller/larger than the other public screen.

- Only one screen, the MainScreen, is now used. All windows are opened on it for all editors. This is faster, less flickering between screen changes, and makes the editors more the same since they all use the same dims, etc.

This also eliminates ScreenType.

- Completed update to Requesters Editor.

- Updated some of the window handling routines and variables. They are more consistent, precise, etc.

- Fixed some bugs in the code to search for the Shell window. I don't know how they slipped by all this time.

- Added PUBSCREEN/K argument to specify what public screen the Shell window is located on if not Workbench.

- Update Shell search code even more to handle PUBSCREEN/K. It now asks for a public screen name, putting the default/specified name in the requester, then it asks for a Shell window name, putting the default/specified name in the requester. Much more complete.

Version 3.0 Completed: 23.10.94

~~~~~

- Fixed a bug in IDCMP\_VANILLAKEY handling of menu switching with the numeric keys.

- Fixed a bug in the AmigaShell window search routines. When it couldn't find a Shell window and it was told to open one, it wouldn't find it. Fixed now.

- Added checkbox gadgets to Format Options and Format section for No Verify command line argument for Format. Also added Format path gadget to specify an alternate format program.

- Fixed a bug in the way the menus was cleared. Had to give it a name of " " so when tried to swap the menu with another, it would guru trying to copy a blank menu title and such. To totally clear it you will have to use Delete Items/Menus.

- There are now 10 menus with 50 items each for a total of 500 menu items.

- Updated the menu routines to handle the menus even more dynamically. Only the ProjectMenu is defined and the user menus are AllocVec'd as needed. This saves MAJOR memory since only the exact number of menus are defined, not all 500 possible menus.

- Updated Menu Editor with these new gadgets and layout:

Menu Close Menu Editor

Clear This Item Clear This Menu Clear All Menus

Swap Items Copy Items Insert Items Delete Items

Requesters Editor Preferences Editor

Load Preferences Save Preferences Quit

Picking the title will do the action to the whole menu.

Picking the item will do the action to just that item.

- Fixed several little bugs dealing with string comparing. Also one with the

MenuKey string gadget not filling the items correctly.

- Added a TEXT\_KIND gadget below the string gadgets to keep the user informed as to what is expected when using Swap, Copy, Insert, and Delete. Under V39+ the strings are centered, left aligned if less than V39.

- Finally updated all the variables to the new Configurations.c structures.

This changed the prefs file AGAIN so any file before 3.0 is invalid. But this is better since it will allow for easier updating since it now uses structs, so adding options will be quite easy. This should also prevent any new versions from not being backwards compatible with older prefs files.

- Made many updates to code sections that used the old variables or just needed a facelift. Faster, more accurate, cleaner.

- Added 'No Multiple Files' gadget to the Req\_Prfl section. Some programs don't accept multiple file paths on the command line and will not work. Most programs allow for this, but LhA doesn't, so there must be others that don't also. [NOMULTIPLEFILES] is the new flag. MultipleFiles allowed is default.

- Fixed a minor bug in the Requesters Editor dealing with Directories Only.

- MultiFile may now be used anywhere in the command line, not just the last struct before the prfl struct. Too many programs needed this.

- Changed and added these to Preferences Editor:

Use Defaults -> Section Defaults

All Defaults (added)

for more precise control and editing.

- Several bug fixes and code updates.

- Updated AmigaGuide to reflect all the new changes.

- Public release.

- Received French catalog and AmigaGuide from AFLE.

Version 3.1 Completed: 00.00.95

~~~~~  
- Now Kickstart 3.0 minimum.

- Updated to use gtlayout.library.

- Updated to use Modules directory.
 - Updated AmigaGuide.
 - Added command line argument [LANGUAGE <language>] to specify a preferred language for locale strings.
 - Split alot of set up and shut down code into a seperate file, MyProgramSupport.c to speep up compilation and easier updating.
 - Removed alot of extraneous code that I no longer needed or I overlooked in gtlayout.library.
 - Completely new interfaces for all windows. A much needed facelift.
 - Completely revamped the source code to bring it up to date and my cuurent standards.
 - Now uses Exec lists for the menus, so there is now an unlimited number of menus and menu items, memory allowing.
 - Uses better Exec list routines now.
 - Added Volume requester type using ReqTools' volume requester.
-