

Utility Launcher™

By George R. Wilde

UtilityLaunch™

A Powerful Apple IIGS Program Selector

by George R. Wilde

CONTENTS

UTILITYLAUNCH™	1
INSTALLING UTILITYLAUNCH.....	4
IMPORTING PREVIOUS MENUS.....	7
UTILITYLAUNCH CAPABILITIES	9
USING UTILITYLAUNCH.....	13
PROGRAM AUTOSTART.....	15
SCREEN BLACKOUT TIMER.....	16
ACCESS CONTROL.....	17
CHANGING SLOTS.....	19
LAUNCHING IBM PROGRAMS WITH THE PC TRANSPORTER.....	22
PRINTING AND SAVING THE SHR SCREEN	23
PARKING HEADS.....	24
MINI UTILITYLAUNCH	25
PROGRAMMING NOTES	26
ACKNOWLEDGMENTS.....	27
SHAREWARE	28
COPYRIGHTS	29
SOFTWARE LICENSE NOTICE.....	29
WARRANTY DISCLAIMER	30
TRADEMARKS	31

UtilityLaunch™

A Powerful Apple IIGS Program Selector

Copyright © 1988 - 1998 by George R. Wilde
24402 Broadwell Ave.
Harbor City, CA 90710-1812
(310) 325-4456

UtilityLaunch (a companion program to UtilityWorks (R)) is a powerful Apple IIGS Program Selector/Launcher which gives the user the capability of launching any program from either a standard file selection dialog box or from an easily configured menu presented in a window on the Desktop. Up to 50 launch selection menus, each containing up to 40 simple button entries or 16 icon button entries, can be quickly configured using a standard file selection dialog box (just point and click on the programs to be configured). UtilityLaunch permits very rapid launching of all programs under GS/OS (much faster than with the Finder).

UtilityLaunch gives the user total control over slots (internal slots or slot cards), system speed (TransWarp or ZIP, fast, or normal speeds), and text, background, border, launch window, and desktop colors. Each of the fifty possible launch selection menus can have its own defined set of slots, speed, and colors to be set upon activation of that menu. Each menu can also have its own defined set of slots, speed, and colors to be set upon shutdown (i.e.; upon switching to another menu, returning to the calling program, or performing a GS/OS shutdown). Moreover each of the 40 possible launch entries on every menu can have its own slots, speed, or colors to be set upon launch of that program. The menu's set of slots, speed, and colors (i.e.; activation settings) are restored upon exiting the called program and returning to UtilityLaunch. Unlike other slot switch programs, slot restoration occurs for both P8 and P16 programs. Setup of slots, speed, and colors is also easy, involving only pointing and clicking the mouse. Changing colors can be useful in executing some programs that appear better in colors other than your control panel defaults. Changing speeds can be useful for some games. Changing slots is sometimes necessary for executing programs that are designed for use with slot cards.

UtilityLaunch includes the following capabilities:

- a) Ability to launch any S16, SYS, EXE, BAS, BIN, or TXT (EXEC) file.
(The latter three types must be capable of being run from BASIC. Only some EXE files can be properly executed).
- b) Ability to switch slots (GS internal slots or slot cards), speed, and colors, as well as certain modem and printer parameters, monochrome/color display settings, and startup slot upon launching of any program and return to default settings at completion of that program.
- c) Ability to set default slot configurations, speed, colors, monochrome/color display, and certain modem and printer parameters independent of the control panel and return to a predefined set of these parameters upon exiting from UtilityLaunch.
- d) Ability to boot any bootable drive or scan any bootable drive.
- e) Ability to autostart any of several programs whenever the disk is booted, and return to UtilityLaunch upon program completion.
- f) Ability to easily configure any UtilityLaunch Menu Screen, including the ability to easily import any icons and the ability to choose any desired colors for the menu buttons.
- g) Ability to set a screen blackout timer to blackout the monitor's screen after a period of nonuse, to avoid permanently damaging the screen when non-changing bright images are displayed for prolonged periods of time.
- h) Ability to limit general access to only selected menus and program features. This will prevent inexperienced users such as children or students from accessing programs that could potentially destroy a hard disk.
- i) Ability to easily insert, edit, delete, and rearrange menu entries. Ability to create Boot Buttons to boot any desired slot, Scan Buttons to scan any desired slot, Menu Buttons to select any desired menu, Purge Buttons to purge memory, and Shutdown Buttons to shutdown the computer. Ability to view the current settings of any menu entry, or printout the entire menu settings or menu screen. Ability to delete or copy entire menu screens.
- j) Ability to purge all purgeable Apple IIGS memory, including ability to purge memory immediately prior to launch of an applications program.
- k) Ability to pass startup paths to programs being launched. This provides a capability for now having separate UtilityLaunch menu buttons to directly launch different IBM programs with the PC Transporter.

- l) Ability to control the TransWarp GS card's speed and IRQ settings or the ZIP GS card's speed and delay settings with any menu entry.
- m) Ability to import all previously constructed menus generated with earlier versions of UtilityLaunch.

Note: UtilityLaunch version 2.3 should be totally compatible with System Disk 6 and will take advantage of the seamless SHR screen transition if System 6 is present and you are launching a Desktop application with the proper Auxiliary Type. If you have a Desktop application which does not contain the proper Auxiliary type (\$DB02 : Desktop Application) you can change it using UtilityWorks. Just make sure the application is truly a Desktop application.

The following paragraphs describe the capabilities implemented in UtilityLaunch and discuss how to best use these capabilities.

Installing UtilityLaunch

UtilityLaunch is an S16 program which must run under GS/OS System Disk 5.0.4 or later. UtilityLaunch can be run out of any directory on any disk. However, if you desire UtilityLaunch to be automatically executed whenever the computer is booted, either the Startup Path Control Panel must be set to launch UtilityLaunch on startup (only applicable to System 6), or UtilityLaunch must be renamed as START and appear in the SYSTEM directory of your boot disk. This is the preferred configuration to achieve maximum performance from UtilityLaunch when running with GS/OS versions earlier than System 6. Any existing START program should be renamed to some other name (e.g.; START.OLD) prior to renaming UtilityLaunch to START. The companion UtilityWorks program can be used to copy and rename UtilityLaunch. You must make sure your boot disk is not write protected and has adequate space for the menu information that is written to the resource fork of either the UtilityLaunch program file or UL.Menu file. Depending on the number and complexity of your menus, a substantial amount of disk space may be required for menu storage. The easiest way to install UtilityLaunch with System 6 is to use one of the four installation scripts included on the disk. Simply run the System 6 Installer Program with the UtilityLaunch disk on-line in one of your 3.5 inch drives. The UtilityLaunch installation scripts will appear in the list of selectable Installer scripts.

The first script will create a System 6 Boot disk containing UtilityLaunch (renamed START) and an empty UL.Menu file. This Boot disk does contain the SCSI hard disk drivers to permit access to your hard disk after booting this 800K boot disk.

A second script installs UtilityLaunch on a hard disk already containing System 6 without disturbing the separate UL.Menu file contained on the hard disk. Use this script to simply replace the UtilityLaunch program without having to regenerate the menus. (Note: Menus must be contained in the separate UL.Menu file and not the resource fork of UtilityLaunch to avoid destroying menus with this script). To boot into UtilityLaunch after installing with this script, use the Startup Path Control Panel to set UtilityLaunch as the Startup path.

A third script functions identically to the above script but also copies an empty UL.Menu file to the directory containing UtilityLaunch. This is the preferred script if you are initially installing UtilityLaunch to a hard disk. Just remember that this script will destroy any past menus you have created.

A fourth script installs UtilityLaunch on a hard disk already containing System 6, and replaces the existing START program with UtilityLaunch (renamed START). Menus will not be destroyed unless they are contained in the resource fork of the previous START program. The difference between this script and the second script is that the START program will be replaced by UtilityLaunch (renamed START) rather than creating a separate file called UtilityLaunch. If you have installed the Startup Path Control Panel with System 6, you can have any program be your startup program

simply by specifying it in the Control Panel. In this way you don't have to replace the START program. Simplify set the Startup Path to UtilityLaunch.

The above scripts will simplify your UtilityLaunch installation process considerably when running with System 6.

Upon loading, UtilityLaunch will load the required RAM based tools and begin execution. A Shareware message will appear the first time UtilityLaunch is run following a Boot, unless a Password has been entered. The Password can be obtained by paying the \$10 Shareware fee. Just press Continue to begin UtilityLaunch execution.

Upon loading, UtilityLaunch will check to see if there is an extended file named UL.Menu in the same directory that contains the UtilityLaunch program. If there is, UtilityLaunch will extract and save all menu information from the resource fork of this file. If UL.Menu does not exist, UtilityLaunch will create a resource fork in its own program file to store all menu information. This gives the user two choices in configuring UtilityLaunch. Using the separate UL.Menu file gives maximum flexibility, and allows easy installation of updates to UtilityLaunch without having to import the previous menu entries. Using the single UtilityLaunch file is conceptually simpler. However, installing an updated version of UtilityLaunch is more complicated. The user must update the resource fork of the new version of UtilityLaunch with the previous menu information BEFORE replacing the old version of UtilityLaunch with the new version. The update of the resource fork information can be accomplished in one of two ways; using the import menu item in UtilityLaunch or copying just the resource fork from the old version to the new version (UtilityWorks is capable of copying just the resource fork of a file).

A capability to use UtilityLaunch from a file server over an AppleTalk network was added in version 2.2.1. UtilityLaunch keeps its menu file resource fork open with write access privileges to allow modification of the menus during menu setup. This will prevent multiple users from utilizing UtilityLaunch at the same time, as may occur when using a file server. (UtilityWorks does not have this problem and should be operable from a file server without the following special actions required for using UtilityLaunch from a file server). To use UtilityLaunch from a file server first change its access privileges to disallow write access (you can do this with UtilityWorks). Now place it in the desired file server directory (you can make it the initial Start program if you so desire). Make sure that you don't copy the UL.Menu file to this directory. Place your private UL.Menu (you can call it any permissible name) file in your own directory on the file server. Now when you first run UtilityLaunch you will be asked to select your UtilityLaunch menu using a standard file selection dialog. Select the directory and file that is your menu and this file will be used for your menu selections. This selection will remain in effect until you reboot your Apple IIGS. This same technique can be used to establish multiple menus (for instance a separate menu for each member of your family). Just make sure you write disable the UtilityLaunch program and don't have any menu in the same directory as UtilityLaunch with the name UL.Menu. As many menus as desired can be placed in the UtilityLaunch directory as long as none are named UL.Menu. UtilityLaunch version 2.2.4 and later implements the new Apple assigned file type and auxiliary type assignments for UtilityLaunch menus, namely file

type \$5A (CFG), auxiliary type \$8023 (known as UtilityLaunch preferences). The current UtilityWorks distribution disks implement these new assignments, but also recognize the older designations except when using the above multiple menu feature. If you intend to use this feature first change the file type and auxiliary type of all existing menus to the new assignment. This can be very easily accomplished using UtilityWorks.

UtilityLaunch version 2.3 and later adds several resources to facilitate better use in conjunction with the new System 6 Finder. rIcons have been prepared for the UtilityLaunch program, the UL.Menu files, and the Boot file. rVersion, rComment, and rBundle resources have also been prepared for use with the System 6 Finder or UtilityWorks version 2.0 or later. Launching UtilityLaunch the first time from the new Finder will install the UtilityLaunch rBundle. After the rBundle is installed, double clicking on any UL.Menu file (regardless of what it is named) from the Finder will launch UtilityLaunch with the selected menu as the current UtilityLaunch menu.

The UL.Menu file provided with UtilityLaunch contains default settings and three sample menus (an icon button menu and two simple button menus) to give the user some examples of how menus can be set up. You can use this file for your own menus (deleting the sample menus using the Remove item described below, if you so desire). You can also use the UL.Menu.Empty file renamed to UL.Menu. This file contains no sample menus, and is the file used by the installation scripts.

Always make sure you have a backup of your menu items regardless of whether the menus are contained in the UtilityLaunch program file or the separate UL.Menu file.

Importing Previous Menus

The following paragraphs describe how to import previously established menus to UtilityLaunch. If you are installing new menus for the first time you can skip this section.

If you have been using a UtilityLaunch program version prior to version 2.0 (i.e.; version 1.9 or earlier) you should import menus to the new version by using the Menus button in the Import Menu dialog. This dialog will either automatically appear when you first load the program, or you can select it at any time by clicking on the Import Menus... item of the Utilities menu. The Menus button will automatically translate all previous menus (i.e.; UWGS.Menu.0 - UWGS.Menu.9, as well as the UWGS.CONFIG file in your boot disk's SYSTEM directory) into resources that are placed in the current UtilityLaunch resource fork. Note that this process will overwrite any existing menus 1 - 10 in the resource fork with whatever menus exist in your boot disk's SYSTEM directory. If you wish to retain a new menu 1 - 10, copy it to a spare menu using the Save As... item described below. The new version of UtilityLaunch does not store any data in the SYSTEM directory. If you do not want a previous menu transported, either rename it, remove it from the SYSTEM directory, or delete it. The newer resource based menus have much more potential power than the older text based menus; however, all relevant menu data is translated from your older text based menus when they are imported to the new format. For instance any old button 1 settings for any menu are also translated to the new menu's autostart settings, since with the old menu format button 1 is used for autostart if autostart is enabled (in the new menu format, any menu can have a separate and distinct autostart entry). Note that menus have been renumbered in the new version of UtilityLaunch. Previous menus 0 through 9 now become menus 1 through 10.

If you have previously been using a UtilityLaunch program version 2.0 or later, you can copy all of the previously used menus by selecting the Resource button in the Import Menu dialog. Upon clicking this button, you are presented with a file selection dialog. Select your previous menu file (contained either in the UtilityLaunch (or START) file or in the separate UL.Menu file) that you desire to import. All resources contained in this previous file will be copied to your new program or menu file. Note that if you previously used a UL.Menu file for your menu information, you will not normally want to perform a menu transport to the new program as long as this old UL.Menu file is contained in the same directory as your new UtilityLaunch program file.

It is possible to use both the Menu and Resource import options to import both your previous text file based menus as well as new resource based menus. In fact it is possible to import multiple Resource based menu files by making multiple Resource selections. However, with each new import, the previous menu having the same menu number will be destroyed. In general it is better to import Text File menus before importing Resource based menus. Note that the default program settings (i.e.;

password, screen blackout timer, etc.) will only be copied during the initial menu translation if you are importing resources.

Always make a backup copy of your UtilityLaunch menus. This will protect you against accidental changes you might make to your menus during Menu Setup. It will also provide protection in the event that your menu resource fork becomes corrupted.

UtilityLaunch has been preset to print menu contents using an ImageWriter printer. The print option will also probably work satisfactory with most other printers, although the special ImageWriter character sequences used to reset the ImageWriter and advance the paper to a new page may cause problems with some printers. UtilityLaunch also assumes you are using the standard serial printer output device unless you are using AppleTalk in which case UtilityLaunch assumes you are using the RPM printer driver. In the event you want other printer defaults or other output devices you can change the UtilityLaunch defaults by selecting your desired printer defaults in the companion UtilityWorks program (version 0.9H or later) and saving these UtilityWorks defaults. Now run a brand new copy of UtilityLaunch (one with no existing resource fork) and select the Resource option in the initial Import Menus dialog. Select the UtilityWorks program as the file you desire to import. This will copy all of the UtilityWorks printer defaults to UtilityLaunch. This step must be the first step. Following this initial copying of default parameters you can now import any menus that you so desire.

UtilityLaunch Capabilities

This section describes the capabilities of UtilityLaunch. When UtilityLaunch initialization is completed, the program will automatically display a Launcher window (containing the currently active set of up to 40 menu entries (only active menu entries will be displayed)). If this is the first time UtilityLaunch has been executed, the menu setup routine will be entered, and all 40 menu entries will be blank. The Launch item in the File menu on the menu bar can now be used to launch any program, however it is recommended that the UtilityLaunch menus be configured for maximum ease of use. To configure the UtilityLaunch menus simply follow the instructions outlined below. When all desired menu entries have been configured in the Setup window, save the menu (click the Save menu button), and then click the Open Launcher entry in the File menu on the menu bar to activate the Launcher window. While in the Launcher window, clicking any menu button will launch the indicated program if it is in an active drive. To readily distinguish between the Launcher window and the UtilityLaunch Setup window, different background colors have been used. The default Launcher window background color is the light blue of the standard Desktop (although you can select any desired color for any menu). The UtilityLaunch Setup window background color is yellow, orange, red, purple, or gray depending on Setup mode. Any active NDAs can be selected from the desktop or from the UtilityLaunch Launcher/Setup windows without leaving UtilityLaunch. If for some reason you want to close the UtilityLaunch Launcher or Setup window, you can click on the standard close window box in the upper left hand corner of the window's title bar, or you can use the Close item in the File menu on the menu bar. This will put you on an empty Desktop. From the Desktop (or from either window) you can go the UtilityLaunch Launcher or Setup windows by clicking the corresponding menu item in the File menu on the menu bar. You can leave UtilityLaunch and return to the calling program (if any) by clicking the Quit menu item in the File menu. Clicking the Shutdown menu item in the File menu will exit to a shutdown/reboot condition, and will not return to the calling program. All removable disks are also ejected on Shutdown.

The menu bar contains five pull-down menus, each containing several items as delineated below. Each individual menu bar item will be discussed in the following paragraphs. Also indicated in brackets are the keys that may be pressed to activate the menu item in lieu of clicking the mouse on the menu item. (Command is the Open-Apple key). When using the keys make certain you hold the indicated keys down until the desired action begins. Quickly depressing and releasing the keys may result in no action being taken.

1. Apple Icon
 - About UtilityLaunch...
 - Speed Up...
 - Any NDAs you have loaded

2. File
 - Open Launcher... [Command-L]
 - Open Setup... [Command-E]
 - Close [Command-W]
 - Enable Access...
 - Page Setup...
 - Print... [Command-P]
 - Launch...
 - Boot...
 - Sleep
 - Shut Down [Command-B]
 - Quit [Command-Q]

3. Edit
 - Undo [Command-Z]
 - Cut [Command-X]
 - Copy [Command-C]
 - Paste [Command-V]
 - Clear
 - Insert Mode
 - Edit Mode
 - Delete Mode
 - Arrange Mode
 - Customize Mode
 - Show Mode
 - Color Buttons

4. Menus

- Icon Menu
- Save... (or Show Startup) [Command-S]
- Set Shutdown... (or Show Shutdown)
- Set Autostart... (or Show Autostart)
- Save As...
- Remove...
- Menu 1 (or menu name) ([1] - if menu 1 exists)
- Menu 2 (or menu name) ([2] - if menu 2 exists)
- Menu 3 (or menu name) ([3] - if menu 3 exists)
- Menu 4 (or menu name) ([4] - if menu 4 exists)
- Menu 5 (or menu name) ([5] - if menu 5 exists)
- Menu 6 (or menu name) ([6] - if menu 6 exists)
- Menu 7 (or menu name) ([7] - if menu 7 exists)
- Menu 8 (or menu name) ([8] - if menu 8 exists)
- Menu 9 (or menu name) ([9] - if menu 9 exists)
- Menu 10 (or menu name) ([0] - if menu 10 exists)
- Menu 11 (or menu name)
- .
- .
- .
- Menu 50 (or menu name)

5. Utilities

- Options...
- Purge
- Desk Color...
- Add Printer...
- Delete Printer...
- Save Panel
- Restore Panel
- Import Menus...
- Password (if not previously correctly entered)

The current date and time are displayed at the right hand corner of the menu bar, unless the date/time display is inhibited in the Options menu.

Beginning with UtilityLaunch version 2.3, the appearance of all dialogs was changed to conform with Apple's latest Human Interface Guidelines. When System 6 is present the AlertWindows will also conform with these guidelines. The buttons at the bottom of the dialog will generally have the action button on the right and the Cancel button (if any) immediately to the left. Pressing enter or return will select the default (bold-outlined) button (usually, but not always, the action button). Pressing esc or Command-period will select the Cancel button if one exists. I Beam cursors will appear if the cursor is positioned over an Edit Line item. With AlertWindow dialogs you can type the first letter of a button name instead of clicking on it (unless some other button's name has the same first letter). This works with or without the Command key.

The Standard File dialog also supports keyboard equivalents with System 6. You can use the up and down arrow keys to select various items in the list. You can also type one or more characters (including the beginning character of an item) to jump the selection down to that item. (To use the first several characters of an item name, you have to type one letter every two-thirds of a second or faster). Command-Tab moves to the next disk. Command-period is the Cancel button, but esc isn't in the Standard File dialogs. Esc and Command-up-arrow are equivalent to the Close button (moving the directory one level closer to the root directory). Command-esc or Command-D jumps to the Disks list. Clicking on the path item at the top of the list displays a pop-up menu with one item for every level in the current path, all the way to the Disks list. Selecting any of these items takes you to the corresponding directory. Command-O and Command-down-arrow are the Open button. In Save dialogs where you enter a file name, Tab chooses between the edit line and the file list. Keystrokes go to whichever one is active.

Using UtilityLaunch

UtilityLaunch uses previously constructed menus generated from the UtilityLaunch Setup routine discussed in the UtilityLaunch Reference. Just click on any indicated program to launch that program. UtilityLaunch checks to see that the requested program is present in an active device prior to launching it. If nothing happens when you click on a previously established menu entry, it means that no previously selected program was chosen in the UtilityLaunch Setup File Selection dialog. If you see an error dialog asking you to insert a specified disk when you click on a menu entry, it means that that desired program is not present on an active disk drive, or the pathname or filename of the desired program has been changed from that which existed when the menu entry was first created. After completion of execution of the selected program (assuming a normal completion), the GS/OS will automatically return to the UtilityLaunch menu for selection of the next program to be run. If UtilityLaunch Setup has not already been performed, it is desirable to do so before proceeding, although the Launch... item can be used without further setup.

UtilityLaunch version 2.3 added keyboard support to allow navigating through the buttons in any menu and selecting programs to be launched without using the mouse. Pressing any letter key on the keyboard will select the next button in the menu whose title begins with that letter (this search is case insensitive). Pressing the up or down arrow key will move the selected button up or down by one position. Pressing the left or right arrow key will move the selected button left or right by one position. Pressing enter or return will launch the selected (highlighted) button.

UtilityLaunch, working in conjunction with the UtilityWorks Set Prefix capability, supports I/O Redirection. A complete discussion of this capability is contained in the UtilityWorks documentation. The Standard In, Standard Out, and Error output prefixes set by UtilityWorks will be passed by UtilityLaunch to any S16 program launched by UtilityLaunch.

UtilityLaunch should work satisfactorily with all P8 and P16 programs which follow the Apple Quit protocol, and all P16 programs which utilize the Apple II GS Memory Manager in accordance with Apple's guidelines. Some early P16 programs and some very old P8 programs do not follow these standards. These programs can still be launched, but they probably will not return to UtilityLaunch upon exit. With certain BASIC programs you will be presented with a "]" prompt at conclusion. Just type "BYE" to return to UtilityLaunch. Incidentally to launch a BAS, BIN, or TXT file you must have the files BASIC.SYSTEM and BASIC.LAUNCHER in the root directory of your boot disk.

UtilityLaunch has been coded to be restartable from memory (see programming notes below), and can request the Operating System to restart it (i.e.; not reload it) upon termination of the program being launched. This technique can improve performance. Unfortunately, restartability will not successfully work when launching some programs. Despite problems with a few programs, for the vast majority of P16

programs, restartability offers a significant performance enhancement. UtilityLaunch lets you have it either way. With the Launch... button, UtilityLaunch will always be reloaded from disk upon termination of the program being launched. This gives slightly poorer performance, but should work with all programs following the Apple Quit protocol. With the menu buttons and the Autostart capability, the UtilityLaunch Setup routine gives you the choice of whether or not to implement the restart feature. With your troublesome programs don't use the Restart feature. With very troublesome programs use Purge Upon Launch. With most other P16 programs, choose the Restart feature for best performance.

UtilityLaunch version 2.3 added a new capability to automatically poll most devices once a second. If an unformatted disk is inserted, a dialog will appear asking to either initialize (format) the disk or eject it.

Program Autostart

Checking the Autostart box (and Saving the defaults) in the defaults dialog box will force UtilityLaunch to automatically launch the Autostart program contained in the Initial menu the next time the disk is booted. This feature can be very useful if you have disks where you normally want to boot into a given program, but still want to return to a powerful program selector upon exit from that program. Upon normal exit from the automatically launched program you will return to UtilityLaunch, where you can now select any other program for execution. If the Initial menu does not contain an Autostart program, UtilityLaunch will examine in succession each increasingly numbered menu until an Autostart program is found. If no Autostart program is found upon reaching Menu 50, UtilityLaunch will perform a normal load of Menu 50. This implementation of Autostart gives you much flexibility in implementation of Autostart, depending on your selection of the default menu number and your selection of Autostart programs for the various menus.

Just remember that unless you deselect the autostart button (and SAVE the defaults), UtilityLaunch will automatically sequence through its Autostart logic every time the disk is booted. UtilityLaunch communicates with itself every time it is run by using the Apple IIGS Toolbox's Message Center. If it finds a previous message to itself it knows that this is not a fresh boot. In this manner UtilityLaunch can release all of its memory and still retain information about the boot state.

Autostart can be bypassed by holding down the Command (Open-Apple) key as the GS/OS Startup splash screen disappears immediately following GS/OS Boot.

Screen Blackout Timer

The Defaults Dialog Box also allows you to select a time interval for shutting down the monitor's screen, to prevent burning an image into the screen after long periods of time where the screen image remains constant. This timer only works while you are actually in UtilityLaunch (it does not work with applications you launch). With this option, if you do leave your computer on while in UtilityLaunch for an extended period of time with no activity, such as going to lunch, the computer's screen will be blanked when this timer expires. The timer can have any value from 1 minute to 60 minutes. Any larger value is interpreted as 0. The default is also 0, which means that the timer is inactive. I have found a value of about 10 minutes is ideal for me. To restore the screen merely move the mouse, press the mouse button, or press any key.

Access Control

UtilityLaunch includes a capability to restrict access to certain of the program's features in order to prevent unauthorized users, such as children or students, from launching programs, such as utilities, which could potentially damage or destroy the contents of a hard disk if not properly used.

Access control is not normally activated. To activate this feature, click the mouse on the Access Password rectangle of the Defaults dialog box (or press the tab key to get to this entry) and enter whatever password you desire (the password is case sensitive). Append an asterisk (*) to the end of the password. The asterisk is NOT later entered with the password. It is used strictly to prevent inadvertent implementation of access control. Press SAVE to save the access password and implement access control the next time UtilityLaunch is executed. To disable access control at a later time, an authorized user may enter a single asterisk (with no other characters) in the Access Password rectangle, and save the entry. This will erase the previous password and return the program to normal operation.

With access control enabled, unauthorized users (i.e.; those users who do not know the password) can only access those menus that have previously been established with unlimited access privileges. Many items within the menus in the menu bar will also be dimmed as discussed in the UtilityLaunch Reference. Likewise, hot keys cannot be used for menu selection, unless the associated menu has unlimited access privileges. The menus that are accessible to all users should have only "safe" programs installed on them. Unauthorized users will only be able to launch these programs. The dimmed and disabled entries will prevent unauthorized users from setting up menus with any other entries, or directly launching any program other than those you have previously set up on one of the unlimited access menus.

Menu access is established when saving (i.e.; Save... item) the menu settings. To limit access to any menu, click on the Limit Access box in the Set Name dialog used to establish the menu name. Unless you check the Limit Access box, the menu will be given unlimited access privileges.

An authorized user can gain access to the "hidden" features at any time by entering his previously selected access password at the Enable Access item in the Files menu. The access password is not shown on the screen as it is being entered. After the Enable Access password has been correctly entered, the Open Setup... menu item will immediately become undimmed and may be selected. Unlimited access will remain in force until the next time the computer is booted or the next time a menu is selected during menu shutdown (which is set in the Set Shutdown menu item), unless a single asterisk is entered and saved in the Defaults dialog box, in which case unlimited access becomes permanent. As long as access control is in effect you must reenter the access password to regain access to the restricted features.

The access control is designed to provide only very limited protection with inexperienced users such as children or students who are not computer experts. An experienced computer user can easily defeat this protection. If you forget what your password is, you can reinstall your menus. You can use the UL.Menu file that comes with UtilityLaunch. This file is set with Access control defeated. Merely import all of your old menus to this file and you should have your current menus with Access control disabled.

Changing Slots

Changing Slots while the Apple II GS is executing is potentially dangerous and must be done with great care. Adhere to the following instructions to do it properly. To minimize problems, it is best to group all of your menu entries involving slot changes on a single menu screen, and have all other menu screens use the defaults for slot changes. If something gets messed up, the program will probably crash or you will get an operating system error. Reboot the computer and check the control panel settings, making any necessary changes. If you need to make slot changes reboot again. In severe cases you may need to turn the computer power off and then back on to correct the problem. In no case will any permanent damage be done to your computer hardware, but it sure can be annoying if the computer crashes.

UtilityLaunch sets slots by changing both the Battery Backup RAM and the corresponding Soft Switches to the appropriate values. If the Resize RAM after Reset option is selected in the Set Slots dialog box, the Power-up byte in the Keyboard Microprocessor RAM will also be reset. Resizing RAM after Reset is unnecessary unless you want to remove the :RAM5 disk the next time you boot from UtilityLaunch (normally this can only be done by turning the power off). The actual resizing of RAM is deferred until the Shutdown menu item is selected. If any program is launched in the interim, the RAM will not be resized.

If you follow the process outlined below, UtilityLaunch will force all control panel settings to be initiated immediately. You never have to wait until the next boot or the next time the computer power is turned off. As mentioned above, switching slots must be done with care. Consider what each slot is used for (internally and externally (i.e.; slot cards)) before you initiate the change. Remember that internal slot 3 supports the 80-column firmware needed by many programs (e.g.; you won't see UtilityLaunch's Launching... message if slot card 3 is selected). Internal slot 4 supports the mouse required by many programs including UtilityLaunch. That doesn't mean you can't use UtilityLaunch to change slot 3 and slot 4 (I do), you just have to do it carefully. Slot RAM cards pose special problems. The Apple II GS does not like to switch out RAM cards, including :RAM5 (slot5, drive2), for fear of losing data, and will not normally implement such a requested control panel change until the computer is actually turned off. With battery backed up RAM cards loss of data, of course, is not a problem. UtilityLaunch can effect instantaneous change from slot RAM or :RAM5 (slot 5/drive 2) to disk or another slot card if properly set up as described below. This procedure probably will not work in slot 6 unless you have new or updated Apple 3.5 in. drives that correct the daisychain interface card problem, which causes occasional anomalies with Apple 5 1/4 in. drives. Until I had my interface cards replaced (a free Apple update) I could not get any slot change program to work with my RAMFACTOR card.

It is probably easiest to explain the use of slot change by example. The following is a slot card setup that I have used in the past:

<u>Slot</u>	<u>Card</u>
1.	free
2.	Apple Super Serial Card (for those Comm programs still requiring it)
3.	Applied Visions FutureSound card
4.	Applied Engineering PC Transporter
5.	AST VisionPlus card
6.	Applied Engineering RamFactor card
7.	Apple SCSI card

In addition to a hard disk in slot 7, I also had two 3.5 in. drives and an Applied Engineering ROM card in slot 5, and two 5 1/4 in. drives in slot 6, as well as drives supporting the PC Transporter in slot 4. As you can see I had a fairly full computer, and many programs on my hard disk required slot changes to work. From my experience it is best to set up your control panel to Apple slot defaults and let UtilityLaunch handle the slot switches. It is necessary that slot 6 be set internal (Disk Port) at power-up and booted in that configuration, or the slot change will not properly be made when going back and forth between the RamFactor and the 5 1/4 in. drives. Slot 5 should also be set internal (Smart Port) if you are using the :RAM5 (slot5, drive2).

For any menu screen having slot change selections, make sure you set all of the slots that you intend to change with individual menu entries to the appropriate menu startup settings (as selected in the Save Menu dialog sequence). By this process, the Startup slot settings will automatically be set (previous values restored) as each launched program returns to UtilityLaunch. These settings for the most part should be set to internal (except slot 7), although I have my slot 6 set to slot card (external) because several programs will not work properly without the RAM disk so set. Make absolutely certain that slots 3 and 4 are set to internal (not slot card) if you intend to change them. You should now set all relevant entries in the Menu Shutdown settings (chosen by the Set Shutdown menu item). I have all of my Shutdown slots set to internal except slot 7. This will be the setting that prevails on the next boot or power up, or when UtilityLaunch is switched to another menu set. Next, for each menu entry requiring a slot card, select only that slot as external (slot card), keeping all other slots at default. Now as you launch the individual programs, the appropriate slot will be instantly changed and properly restored upon return to UtilityLaunch. This above procedure has worked very reliably for me with my configuration. Significant slot changes can occasionally cause problems for GS/OS when UtilityLaunch shuts down and returns to a P8 program selector, especially if the exiting slot setup is different from the entry slot setup. However, if you have UtilityLaunch established as your START program you should not experience these problems.

If you are using :RAM5 (slot5, drive2), you may want to select the Resize RAM after Reset option whenever you switch to another slot 5 device (Your Card). This Resize RAM after Reset option will totally remove the :RAM5 at the next boot. If you do not use the Resize RAM after Reset option, you can retain the contents of :RAM5, even after switching to another slot 5 device, until the computer is powered down. Simply have the :RAM5 specified at the time you boot your computer. Use the select slots

dialog box to select external slot 5 (i.e.; slot card) for the program you want to execute that uses this slot card (mine is the AST VisionPlus card). When this program is launched, UtilityLaunch will switch slots and zero the minimum and maximum RAM sizes in the control panel (the RAM downsizing only occurs when you switch slot 5 from internal to external). Upon return to UtilityLaunch, the original internal slot 5 setting will be restored, but the RAM settings will remain at zero. However, you can still access the data in :RAM5 as though the RAM settings were never set to zero. This is because these particular settings normally do not take effect until the computer is powered down. If you follow the above steps but Resize RAM after Reset, any existing data in :RAM5 will be lost when you reboot your computer. This is an easy way of getting extra memory for some large programs that don't need :RAM5, without having to turn your computer off then back on again to remove :RAM5 (you can use a Custom Shut Down button with the Resize RAM after Reset set to accomplish this). You can verify that the Resize RAM after Reset option has been properly set up by viewing the menu entry in the Show Mode. An example of a Custom button to Resize RAM after Reset (labeled Clear Power Up) is contained in the UL.Menus sample file.

Switching slots works best when the application being run is a P8 application (such as the PC Transporter (AEPC.SYSTEM)). During initial load, GS/OS looks at all currently selected slots, and loads or generates drivers for all connected devices. If you switch a slot, for instance from internal to your card, GS/OS will not have loaded or generated a driver for this new device. If the P16 application that uses this card is now executed, it will not run properly if it depends on a GS/OS driver, because the proper driver has not been installed. This is not a problem with P8 applications, because they use their own drivers, and are not dependent on GS/OS device drivers. If you are switching slots used by P16 applications that are dependent on GS/OS device drivers, make sure that your initial (default) control panel settings are set to "Your Card" so that the appropriate driver will be loaded. UtilityWorks version 2.2.5 and later provides for loading of GS/OS device drivers while launching programs or loading menus. The pathname of the driver can be either manually entered or selected with a Standard File dialog. Drivers for such devices as Apple 5 1/4 in. disk drives can be inhibited from loading using a program such as UtilityWorks. With slot 6 set to Disk Port and the Apple 5 1/4 in. device driver inhibited, you can avoid the lengthy slot 6 polling that occurs with many S16 programs. The 5 1/4 in. drive is still available for P8 programs because they don't use the GS/OS device drivers. If you have a program that requires this 5 1/4 in. device driver, you can easily set up the launch settings for that program to load the driver just prior to program execution.

Note: Switching slots without rebooting is not an Apple approved procedure. It does work well with current versions of GS/OS if you follow the precautions outlined above, but may not work with future versions of GS/OS. GS/OS now includes a skeleton Slot Arbiter feature that is not fully functional. If fully implemented, the Slot Arbiter will provide a GS/OS capability to switch slots without rebooting. UtilityLaunch will be modified to use the Slot Arbiter, if and when it becomes fully functional. In the meantime UtilityLaunch provides a reasonably good alternative to allow switching slots on the fly.

Launching IBM Programs with the PC Transporter

UtilityLaunch includes a provision to launch any IBM program if you have a PC Transporter and the associated Applied Engineering software. Your AUTOEXEC.BAT file must be set up to include AE's launcher.exe file. My AUTOEXEC.BAT file is as follows. Adapt yours to suit your needs.

```
prompt $e[40;36m$e[1m$
echo off
prompt Applied Engineering PC Transporter, on $d at $t$_$p$g
cls
path=e:\;e:\pcdos;e:\aepe;e:\pctools;e:\norton
keyb us 437 \pcdos\keyboard.sys
ver
apclock
launcher.exe
```

You will also need a BAT file in the root directory for each program being launched. For instance my BAT file for Lotus 123 is:

```
LOTUS.BAT

cd e:\lotus
lotus
cd\
```

My MSDOS hard drive is drive e: - yours may be c:. My Lotus 123 program is in a directory called lotus.

Setting up the IBM side is the hard part. To set up UtilityLaunch to launch Lotus 123 I merely select the AEPC.SYSTEM file as the program to launch, set slot 6 to external (assuming your PC Transporter is in slot 6), and enter a startup path of LOTUS. That's it! There is an example of this setup in the UL.Menus samples included with UtilityLaunch.

Printing and Saving the SHR Screen

At almost any point when running UtilityLaunch (except when Alert Windows and Error Windows are being displayed) you can print the current contents of the SHR screen merely by pressing Command-H (Open-Apple-H). Command-H will print the entire contents of the SHR screen, not just the contents of the windows. You can also save the contents of the SHR screen as a Screen Image (Type \$C1, AuxType \$0000) file by pressing Command-J (Open-Apple-J). A Standard File Save dialog will appear. Simply select the directory and file name you desire to save the Screen Image under.

Parking Heads

The Options dialog includes a “Park Heads on Shut Down” selection in UtilityLaunch version 2.3 and later. Most modern drives automatically park their heads so you probably don’t need to use this option. If you select this option the action taken is dependent on the type of drive you have. If you have a non-SCSI hard drive the drive’s heads are positioned over the last track of the last partition on the drive to minimize potential damage to tracks containing programs or data. If you have a SCSI hard drive a SCSI Stop Unit command is issued, which with most SCSI drives will park the heads and shut the unit down. Since this action will probably put your boot drive off line you may get a message to insert your boot disk. You can ignore this message and turn your computer off or you can press Cancel to get the standard Shut Down dialog. Rebooting at this time will turn your hard drive back on, but it may not come up to speed quickly enough to avoid the “Startup Unit not Found” message. If you get this message simply perform an Open-Apple -Control-Reset (Command-Control-Reset in the new Apple terminology) boot to continue.

Mini UtilityLaunch

A reduced size version of UtilityLaunch that will easily fit on a single 800K System 6 Boot Disk has been created and included on the UtilityLaunch distribution disk. This reduced size version is called MiniUL. MiniUL includes essentially all of the UtilityLaunch Launcher mode capabilities, but none of the UtilityLaunch Setup mode capabilities. Menus for MiniUL must be created by the full-up version of UtilityLaunch. Once these menus have been created they can be easily copied to the directory that contains MiniUL.

Programming Notes

UtilityLaunch consists of several segments within its S16 file. To maximize performance, less frequently used features of the program (mainly menu setup and Password entry dialog boxes) are contained in a dynamic segment that is only loaded as needed. This is why you will experience a slight delay when you first use one of these features. The System Loader must retrieve the dynamic segment from disk. Subsequent use of these features will be much faster because the dynamic segment is already in memory. Normal program launching options do not require this dynamic segment. UtilityLaunch S16 file is in ExpressLoad format for best performance in loading under System Disk 5.0.

UtilityLaunch has also been designed to be restartable from memory. Any time a program is launched, UtilityLaunch can request the Operating System (GS/OS) to retain its code segments in memory if the Operating System does not need the memory space for running the program being launched. This results in significant performance improvement when running smaller P16 programs (depending of course on the amount of memory available in your computer), since the System Loader does not have to reload any of UtilityLaunch's code segments. The System Loader does always reload the UtilityLaunch's ~global segment, and the UtilityLaunch itself reinitializes its ARRAYS segment from the configuration and menu files.

UtilityLaunch also incorporates two QuickDraw speed up enhancements (hardware shadowing and fast port) to improve performance.

UtilityLaunch version 2.0 and later stores all menu information as resources in its current resource fork. Resources are only loaded as needed to achieve best overall performance.

UtilityLaunch version 2.1.1 and later prohibits loading of initial program segments into special memory to avoid potential problems in setting up the SHR screen.

UtilityLaunch was developed using TML Pascal and ORCA/M 65816 Assembly Language operating under the Apple IIGS Programmer's Workshop (APW). Resources were developed using Genesys and the APW Resource Compiler. Selected libraries were developed in APW C. The program was linked using LinkIIGS.

Acknowledgments

I wish to thank Tony Vece of ZIP Technology and David Ely, author of the ZIP CDev and CDA, for their help in developing the UtilityLaunch ZIP capability. Portions of the ZIP GS code contained in UtilityLaunch is based upon some of David Ely's ZIP routines.

I wish to express my special thanks to Fred Shader (AG6O on GENie) for the outstanding UtilityWorks Icon he has generated.

SHAREWARE

UtilityLaunch was previously a Shareware program. UtilityLaunch is currently available for free to those who register on the UtilityLaunch Home Page (<http://www.wilde.org/UtilityLaunch>). Users who register on this page will receive an E-Mail which contains the Password to remove the initial Shareware Alert Dialog Box.

The current version of UtilityLaunch is available for download at <http://www.wilde.org/UtilityLaunch>.

COPYRIGHTS

UtilityLaunch is Copyright © 1988-1998 by George R. Wilde. All rights reserved.

Copyright © 1987 TML Systems, Inc. Certain portions of this software are copyrighted by TML Systems, Inc.

ProDOS Copyright © 1983-1988 by Apple Computer, Inc.

GS/OS Copyright © 1988-1993 by Apple Computer, Inc.

SOFTWARE LICENSE NOTICE

The software/documentation described herein is furnished under a nonexclusive license and may be used only in accordance with the terms of the license. Any unauthorized duplication or use of UtilityLaunch software and or documentation, in whole or in part, in print, or in any other storage and retrieval system is forbidden, without the prior written permission of the author. This product is intended for your personal use and may not be distributed commercially without the express written permission of the author.

WARRANTY DISCLAIMER

The author, George R. Wilde, makes no representations or warranties with respect to the contents of UtilityLaunch and its associated documentation, and disclaims any liability or responsibility to the purchaser or any other person or entity with respect to liability, loss, or damage caused or alleged to be caused directly or indirectly by this software. Furthermore, the author reserves the right to revise this product, and to make changes from time to time in the content hereof, without obligation to notify any person of such revisions or changes.

APPLE COMPUTER, INC. ("APPLE") MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE. APPLE DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF APPLE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE APPLE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTY IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE SOFTWARE EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. Apple's liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort (including negligence), product liability or otherwise), will be limited to \$50.

TRADEMARKS

UtilityLaunch is a trademark of George R. Wilde.
UtilityWorks is a registered trademark of George R. Wilde.

Apple, AppleWorks, AppleLink, Apple IIGS, IIGS, Finder, GS/OS, ImageWriter, and ProDOS are registered trademarks of Apple Computer, Inc.

PC Transporter, TransWarp GS, and RamFactor are trademarks of Applied Engineering, Inc.

FutureSound is a trademark of Applied Visions, Inc.

VisionPlus is a trademark of AST, Inc.

ZIP is a trademark of ZIP Technology.