

DepthKey 1.4
Copyright © 1990, 1991 Tom Thompson
All Rights Reserved

System 7.0 Compatible!!!!

Licensed exclusively by Overpriced Software, Macintosh division

What is it?

DepthKey is a FKEY that displays and changes the mode (bit depth) of Mac monitors. It eliminates the trek to the Monitors cdev when you need to change a monitor's bit depth, typically to view an image in millions of colors, or scroll through a source code window in black and white. Being an FKEY, DepthKey lets you do this from within any application or Desk Accessory (DA) by a combination of keystrokes. DepthKey supports 32-Bit QuickDraw's direct color modes, and is compatible with MultiFinder, A/UX 2.0, and System 7.0.

A FKEY (or function key) is a code resource of type 'FKEY'; hence the name. This code is executed when you press the Command, Shift, and a digit key simultaneously. If you've ever pressed the Comand-Shift-1 sequence to eject the floppy from the Mac's internal drive, then you've used an FKEY. DepthKey's current ID is 7, which means that pressing the sequence Command-Shift-7 activates it.

When you trigger DepthKey, it displays the current bit depth on each monitor hooked to a color Mac. If you do nothing, this display disappears in about five seconds. During the interval that the depth display is present, if you move the mouse pointer to a particular monitor and hold down the mouse button, a popup menu appears. The popup menu shows the various screen modes that this monitor's display board supports. If you chose a different bit depth and release the mouse button, then that monitor's mode is changed to the new selection. Only the chosen monitor's mode is affected. These changes are not permanent; when you reboot, the Mac's screens will be set to the bit depths defined by the Monitors cdev.

DepthKey is shareware; the fee is \$5. I retain all rights to it. You may freely distribute DepthKey, providing that you do not modify the code or data in DepthKey. You must also include, and not modify, this documentation. Bulletin boards and on-line services may distribute DepthKey, as long as they do not make a profit from it. Non-profit user groups such as BCS and BMUG may also distribute DepthKey on disks as part of a software library, charging up to \$5 for reasonable distribution costs. However, those companies that sell shareware/freeware disks *for profit* must seek permission in writing.

History

DepthKey was crafted while I was testing Mac NuBus display boards equipped with graphics accelerators. It turns out that some of them are faster on screen redrawing in the 32-bit mode (or 24-bit mode, take your pick) than in the 8-bit mode. After you've flipped through various screen modes over a dozen times, then strolled to the lab's environmental test chamber for a can of cold Jolt and back, you forget what the monitor's current mode is. The screens look quite the same whether they're showing thousands or millions of colors. To find the screen's bit depth, normally you have to go to the Apple menu, open the Control Panel DA, scroll to the Monitors cdev, click on

the Monitors cdev... You get the idea. So, I wrote DepthKey. With just a key press, I had the bit depth of every monitor displayed on-screen.

OK, so I knew the screen's bit depths. But if the bit depth wasn't what I wanted, I had to change it. That meant going back to the Apple menu, opening the Control Panel, scrolling to the Monitors cdev, and... Well, since DepthKey was already telling me the screen's bit depth, why not have it change the mode as well? That part of DepthKey took a long time to implement, but the efforts were worth it. And I learned a lot about the Macintosh Slot Manager and Color QuickDraw in the process. That's information that's going to be handy later on.

Installing DepthKey

There are several ways to install an FKEY into your Mac system.

1) Use ResEdit. This application lets you to cut and paste resources (such as DepthKey) into the System file. DepthKey's file type and creator are set so that just double-clicking on it automatically launches ResEdit. You can also use ResEdit to modify DepthKey's ID from 7 to a number you prefer. The down side to ResEdit is that it's a developer tool that's not readily available to everyone. And if you're not sure what you're doing, you can really munge up your System file, making your Mac unusable. Make sure you have a copy of the System file on a bootable floppy, just in case you obliterate the Mac's System file. ResEdit is available from APDA for \$25 (there's a \$20/year membership fee, and well worth the price). Call (800) 282-2732.

2) Use a FKEY installer. There are several shareware resource installers that reduce FKEY installation to pointing and clicking, and eliminate the risk of using ResEdit. They can be downloaded from fine on-line systems such as America Online, BIX, Compuserve, and GENie.

3) Use a "resource manager". If you're not using one of these jewels, you should be. A resource manager lets you have practically an unlimited number of desk accessories, FKEYs, fonts, and sounds on your Mac. Better still, these resources don't have to be installed in your System file, but can be stored in separate files. When your Mac boots, the resource manager adjusts things so that the goodies saved in these files are available to you. Two most excellent resource managers are Suitcase II from Fifth Generation Systems, and MasterJuggler from ALSoft. I've tested DepthKey with both products and haven't experienced any problems.

User's guide

Pressing Command-Shift-7 activates DepthKey. (If you've modified DepthKey's ID number, the digit key you press will be different, of course.) At the center of each monitor a small window appears, displaying the monitor's current bit depth. Note that this value is the screen's pixel size, not the number of colors it is using. A help screen with condensed operating instructions appears on the main screen (the monitor with the menu bar).

This display hangs around about five seconds, providing you a visible interval during which you can modify one of the monitor's modes. To do this, move the mouse pointer to the desired monitor and hold down the mouse button. The menubar displays a banner that indicates DepthKey is in control, and a popup menu appears (Figure 1). This popup menu shows the number of screen modes --presented as color counts-- that this monitor's display board supports. The current screen mode is selected and highlighted on the popup menu. To change the bit depth, continue holding

down the mouse button and move the pointer to the desired mode. Once you release the mouse button, DepthKey changes the monitor's bit depth to the new mode. Is that ease of use, or what? If you don't wish to change the mode, just move the mouse pointer away from the popup menu and release the mouse button.

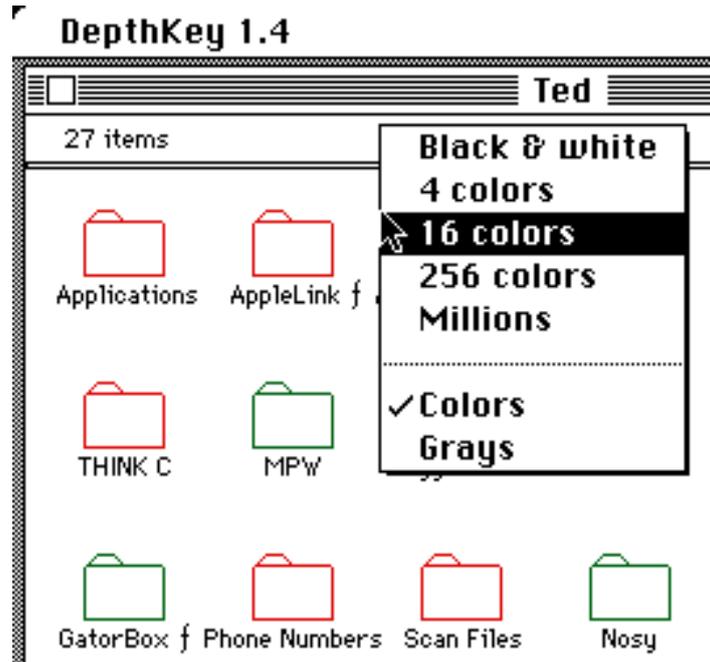


Figure 1

Need to change the monitor from colors to gray scales or back? At the bottom of the menu, two entries, Colors and Grays, deal with color control. Either Colors or Grays is check marked, indicating the current color mode. (It's *really* hard to tell what color mode you're operating at if the display is either 1 bit deep or 32 bits deep.) To change from, say, colors to gray scales, move the pointer to Grays item at the bottom of the menu and release the mouse button.

Tech info

I'm not impressed. You're using `HasDepth()` and `SetDepth()`, right?

Wrongo, moose-breath. Most of DepthKey was written before these two calls, documented in Tech Note #276, were available. Furthermore, these calls are accessible only if you're using System 6.0.5 or later. DepthKey operates on Macs running System 6.0.3 and 6.0.4, and even System 4.2, Finder 6.0 (commonly called "System 5.0").

For those who pay frequent visits to Clarus the DogCow (Apple's developer mascot) and thus are interested in such things, here's a mercifully brief explanation how DepthKey works. When activated, DepthKey walks through the list of GDevices, bagging the depths of those GDevices that are active screens and displaying them. DepthKey then briefly polls for a mouse down event. If it gets one, DepthKey uses the mouse coordinates and the GDevice list to determine which screen received the event. With this information, DepthKey uses Slot Manager calls to query the monitor's display board for the screen modes it supports. Then it builds and presents the popup

menu. Once you change the selection, DepthKey assembles all the pertinent information into a Toolbox call and executes it. Some cleanup is done, and then DepthKey exits.

DepthKey makes no hardware-specific calls and uses only two low-memory globals. This lets DepthKey to work on all color Macs (including those with built-in video, such as the Mac IIx, IIci, and Mac LC), and with all display boards (at least everything I've tested so far). All the information to accomplish this can be found in Addison-Wesley's *Inside Macintosh Volume V* and *Designing Cards and Drivers for Macintosh II and Macintosh SE*. Reading them and sorting it all out was the hardest part. THINK C 4.0's excellent source debugger helped a lot when DepthKey was in its preflight stage as an application.

Operating notes and hints

By using the Slot Manager, DepthKey finds out exactly what modes the display board supports, and only the board in question. It doesn't have to fidget with a 'scrn' resource stashed away somewhere for the information, nor the headache of keeping the information up to date when someone yanks a display board. The disadvantage to doing things this way is that this technique consumes more memory, rather than just using a value out of a file.

DepthKey's way is cool as long you've got the memory (about 10K). Even with the memory, one or more of the Toolbox calls DepthKey uses might force heap compaction, which might cause problems. There's always that poorly-written application that's going to "faw down 'n go boom" because it's expecting something to stay put in memory. Most well-written, mainstream applications don't do this, but you should check DepthKey with your day-to-day application suite for this possibility.

A more insidious problem is how well an application manages its ports. On a color Mac, a list of grafports (kept in a variable-length list called a portlist) helps maintain the color environment. When you alter the screen's bit depth, the Mac OS scans this portlist to make the changes. If an application happens to leave an invalid pointer in this list (typically by disposing a grafport without calling `ClosePort()`), the next time the portlist is scanned, the Mac can crash and burn. This time bomb can lurk around for quite a while, until the next time you use the Monitors cdev or DepthKey to change the bit depth. Check the version history section at the end of this document for problem applications.

So exercise some common sense when using DepthKey. As mentioned earlier, make sure DepthKey and your applications get along. Always save any open files before you change the bit depth. If the applications are already complaining about available memory, don't use DepthKey to set a shallower bit depth in an attempt to scavenge memory. (The actual way to do this is to quit the application, change the bit depth, and then relaunch the application.) Finally, whatever you do, don't use DepthKey while running a disk defrager. Even I'm not crazy enough to try that stunt, and you shouldn't either.

I've made every attempt to ensure that DepthKey is free from bugs. While I have tried it out with a slew of applications on a variety of Macs, there's no way I can cover every permutation of hardware, software, and INITs that might possibly cause a problem. System 7.0 users should exercise care when using DepthKey while running virtual memory, since the code is not reentrant. Therefore, to protect myself and Overpriced Software from lawyers who have nothing better to do,

I have to put in the following legalese, otherwise known as the

Disclaimer

This software is provided “as is” without any warranty of any kind, and the author and distributor expressly disclaim all implied warranties, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as the results and performance of this software and written materials is assumed by you.

Please check the version history section at the end of this document for known problems.

Commonly asked questions about DepthKey:

Why isn't DepthKey an INIT or cdev? It would be easier to install. True. But I'd have to install DepthKey permanently on the system heap, and patch some traps so that you could call it up. This makes for possible INIT conflicts, random crashes, and ties up some memory. The FKEY entry point, on the other hand, is supported by the operating system, and the DepthKey's memory becomes available after it exits. And to be honest, I had enough fun doping out Slot Manager calls without the added complexity of INIT code.

Why digits for the depth display, and a color count for the popup menu? People apparently understand bit depths as well as color counts. At least, I've had no complaint with the bit display, and that makes it a lot easier to code. But requests were made to make the popup menu match the Monitors cdev window. I do listen to my users.

Could DepthKey's popup menu use Command-key equivalents so that I could change the screen from my keyboard? This I could do, but it poses a severe interface problem. First, you might examine the bit depths, judge them to be OK, and start typing while DepthKey is still active. Hit the right key, and DepthKey changes the screen on you. Mondo uncool. This situation gets worse under System 7.0, where you can use the keyboard to maneuver around the Desktop. Also, you might have to move the mouse anyway so that DepthKey would know which monitor you wanted to change. But, if enough people holler for it, maybe I will.

Is there a way to change both the bit-depth and mode, say, from colors to grays, at the same time? Sorry, but it takes two trips through DepthKey 1.4 to do that. (DepthKey 1.3 let you do it in shot, but at the expense of the interface.) Typically, most people want to change either the bit depth or the mode, and in those cases DepthKey 1.4 works well. Trying to change both would require a dialog box of some sort. I'm leeching off the application enough already, and I'd hate to hang it out to dry with a modal dialog box.

Factoids and kudos

DepthKey has been tested on the following systems: Mac II, Mac IIfx, Mac SE/30, Mac IIfx, Mac IIfx, Mac LC and Mac IIsi running System 6.0.3, 6.0.4, 6.0.5, 6.0.7, and System 7.0. A/UX 2.0 testing conducted on Mac IIfx and Mac IIfx, and “Sytem 5.0” testing performed on a Mac II with an Apple Portrait monitor. The following video boards have been checked with DepthKey: Apple Mac II Video Card, Portrait Display, Macintosh Display Card 8•24, and Macintosh Display Card 8•24 GC; SuperMac ColorBoard/24, Spectrum/24 Series III, and Spectrum/24 PDQ; Radius Pivot, DirectColor/24, and TPD boards; RasterOps 24L with Accelerator and 364 board, Data Translation's ColorCapture 2.0 board, and a Micron Xceed SE/306-48 color board for the Mac

SE/30. I use MasterJuggler 1.53 or Suitcase II 1.2.10 to install DepthKey, and CE Software's QuickKeys 2 to assign an idle Extended Keyboard function key to fire DepthKey.

DepthKey is inspired by Radius' Bit-Depth Changer feature supplied with their monitors. Special thanks to Apple Computer, SuperMac Technology, and Radius, Inc., whose monitors and display boards helped me test out DepthKey. Thanks to Data Translation and RasterOps for the loan of the display boards. Kudos to Addison-Wesley, whose Mac reference books enabled me to figure out how to change the video modes.

I'd also like to thank:

Larry Loeb -	provided the moose.
Don Sample -	spotted some gruesome bugs in my code; interface suggestions.
Chris Gibson,	
Howard Eglowstein,	
Jim Reekes -	made valuable suggestions. Pointed out problems with DepthKey's interface and proposed new features.
Brendan Larson -	SE/30 beta tester.
Stephen Fleming -	Mac IIcx beta tester; interface suggestions.
Athena Little -	Who graced Figure 2 of the beta document.

Where your DepthKey dollar goes

DepthKey is shareware; the fee is \$5. I retain all rights to it. You may freely distribute DepthKey, providing that you do not modify the code or data in DepthKey. You must also include, and not modify, this documentation. I'm not asking for much, since I can only provide very limited support.

Please send \$5, payable in U.S funds, to:

Overpriced Software
P.O. Box 202
Peterborough, NH 03458

I'd appreciate any comments (good or bad), or suggestions for improvements or new features. And, of course, bug reports (arrgggghhhh!).

Contact me on the following on-line services:

America Online: Cthulu
BIX: tom_thompson
GENie: TTHOMPSON

Party on, dudes, and be excellent to each other.

Version history

Version 1.0 4-Jun-90 First shipping version. Depth display and mode switch only; no color/gray control.

Known problem: the tool palette in Adobe Illustrator 1.9.3 won't reappear after a mode change. I need to force an update into the event queue, if I can find a safe way to do it.

Version 1.1 12-Jun-90 Major fix to interface.

Version 1.0 required you jump on the mouse button immediately to get the depth popup menu. This took practice, and some people couldn't get to this feature at all because of this design problem. Now DepthKey polls for a mouse down while displaying the bit depths. If there's a mouse down during this "window of opportunity", the mode control routine is called.

The problem with Adobe Illustrator 1.9.3 is fixed. I use a `NewWindow()/DisposeWindow()` pair to trigger a screen update, which makes Illustrator's tool palette reappear. If there's a more elegant way to do this, I'd love to hear it.

Version 1.2β Never released.

Version 1.3β 12-Nov-90. Uses less memory; has color/gray control. Limited distribution.

Holding down the Option key while doing a mouse down presents a set of extended menus. These extended menus let the user set the display for colors or gray scales as well as the depth. Also shows a banner in the menubar for version control. Major internal revisions reduce memory usage. Heap management is much improved, preventing major memory fragmentation problems that deep-sixed version 1.2β. Improved screen clean-up eliminates the `NewWindow()/DisposeWindow()` pair required to redraw the screen.

Known problem: Dies horribly in `InitGDevice()` when using MindWrite 2.1. I suspect this application is one of those that mismanages its ports.

Version 1.3 4-Jan-91. Depth display windows made larger; help screen added.

Known problem: Same as version 1.3β.

Version 1.4β 8-Feb-91. Interface changes; fix for System 7.0b4. Limited distribution.

Option key funkiness gone, you can now access color/gray control from the same popup menu. Help window simplified as a result; now put an icon in the help window. Fixed screen clean-up for System 7.0b4.

Known problem: Same as version 1.3.

Version 1.4 20-May-91. General distribution. Tested out on System 7.0 (the shipping version) before release.

Known problem: Same as version 1.3. Note: DepthKey's code is non-reentrant; therefore, use with caution when using virtual memory.