

McNits: Mike's Palm Programming Nitpicks, Part 2

by Mike McCollister

- Originally published circa May 2002 on the “Handheld Computing Developer” website.
- This article can be found at <https://MikeMcCollister.com/palm/articles/mcnits2/>.
- As this article are quite dated, most of the links will be invalid.

Introduction

Twenty-six minutes and twenty seconds into episode #13 of the season one *Babylon 5* episode titled “Signs and Portents” (originally to be titled “Raiding Party” for you die-hard fans) an alien janitor walks past the screen. While this lasted for only four seconds of screen time, it took two and a half hours to apply the makeup onto the extra. I know this because that alien janitor is my dad (he looks a little better in person without that mask on if you really must know). This attention to detail that the producers of *Babylon 5* insisted upon is what made the series a high-quality show. Attention to detail like this is what is needed to design a quality Palm OS program.

This article is part two in a series of Palm OS program nitpicks that I have created. Some of you may think that many of these nitpicks are trivial or trite, but attention to details like these can make your Palm OS programs rise above the others. Increasing the quality of your programs to the finest detail will help catch the attention of potential customers, possibly lead to more sales, and reduce technical support efforts, but they won't help you get a part as an extra on a science fiction television series.



Figure 1: Two and a Half Hours of Makeup for Four Seconds of Screen Time

User Interface Nitpicks

- Do not have a close menu item. This does not make sense on the Palm platform since by switching to another application closes your program.
- Don't have separate versions for color and black and white. If the user beams the software to someone else, it may not work. Include both the color and black and white bitmaps with the program. We are no longer in the days of the 128 Kbyte PalmPilot.
- Make popup list width based on the widest data element, as shown in *Figure 2*. This is easily done by cycling through all of the list elements and getting the pixel width of each item, using the `FntCharsWidth()` function, and storing the maximum value. This value, plus a padding of three pixels, should be the list width. Just make sure that it is not wider than the screen.
- Don't have screen flicker. If needed, do any screen processing in memory, and then copy it to the screen. This makes the application look more professional and gives the illusion that it is faster.

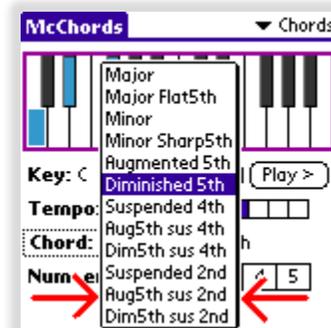


Figure 2: Proper Width of Drop Down List

- If your application is doing some computationally intensive processing, let the user know that something is happening by putting up a form such as a simple “please wait” form or a more sophisticated form with a progress bar. Don’t let the user think that his device has locked up.
- Don’t cram too much information into one form. If need be, use multiple forms. Some applications, such as *HandyShopper* and *WordSmith*, have excellently designed tabbed preference forms.
- Make sure your alert forms use the correct icon. If a dialog is posing a question don’t have the warning icon show up.
- Don’t force your application into a particular color depth. Use the color depth that the device is set to when entered. There may be exceptions to this rule, such as games or image viewers. If you really want to have a color depth that is not standard, give the user the option to enable or disable this.
- Make sure that your application works with a 16-bit color depth. I have seen a number of applications that work fine in 256 colors but fail in 16-bit color mode. 16-bit color mode is the future for the Palm OS platform so don’t be left behind.
- Don’t over-complicate the user interface. Figure out the simplest way of doing something, even if it means more coding. Remember that the user does not care how hard it was to code if the program is easy to use. A computer is supposed to be a tool to make life easier for the human, not the other way around.
- Make sure that your icons invert properly in black and white mode when selected in the built-in launcher. If they do not, then you need to either create a black and white icon or select a different transparent color for your color icons.
- Make sure your application name is not too long so that it is displayed properly in the built-in application launcher. No one wants to see a program name in the launcher that looks like this: “TheBestSmellCh...”.

Support Nitpicks

- Send a “thank you” e-mail to each of your customers when a program is registered, even if no registration code is required. This lets each user know that his hard-earned money has been received by the proper party and it shows him that a means of communication with you is possible.
- When you have an update to your program, send a notice to all your registered users so that they know of improvements to your product.
- When you e-mail someone, make sure that their e-mail address is in the “to” field of the e-mail program. Some people filter out e-mail as spam that does not have their address in the “to” or “cc” field.

VFS Nitpicks

- Make sure that your program works when launched from a VFS card. This is where the future of Palm computing is going so don’t be left behind.
- If your application stores or uses data on to a VFS card, make sure that you look for more than one card as there are devices, such as the HandEra 330, that support more than one removable card.
- Always verify that the removable card that you are accessing is inserted before you do any file operations.

- If you have program specific data stored on a VFS device, put it into the \Palm\Programs\MyProgram directory. Storing that data in the \Palm\Launcher directory may slow down launchers that search for files in that directory.

Compatibility Nitpicks

- Register your application creator ID at PalmSource (<http://dev.palmos.com/creatorid/>). Failing to do so may cause your application to conflict with another and make the user very unhappy with you.
- Make sure that your application works on Palm OS 5. The Palm OS 5 simulator is available from PalmSource at the Resource Pavilion (<http://palmos.com/dev/programs/pdp/login.html>) and free to use. While most applications will work without any modifications, there may be some minor issues that require changes to your code.
- If your program only runs on a subset of Palm OS devices (i.e., requires a particular version of the OS or a particular vendor's device), make sure that it exits gracefully if run on a non-supported device telling the user something useful via a form (don't display something like "Error #0x135A").
- Don't access user interface structures directly. There is no guarantee that these structures will remain constant in future versions of Palm OS. If you need to, try to find a glue function that will do what you need.

Documentation Nitpicks

- Include tips for every non-trivial form. Adding a few lines of documentation in the form, also known as "tips", can reduce technical support requirements.
- The proper way to mark the copyright of your application is to use the copyright symbol followed by the dates as such: "© 1999-2002". Do not use "(c)" or spell out "copyright". The copyright symbol is easy to obtain by putting "\251" into your string. Besides "©" takes less space, which is particularly important on a device with such a small screen.
- In your documentation, you may have to include a picture or a screen dump to help the user understand what you are doing. This is not a problem if you use HTML for your documentation as discussed in "[McNits: Mike's Palm Programming Nitpicks, Part 1](#)". Remember that a picture is worth a thousand nerds, especially if it is bigger than 2,000 bytes (see *Figure 1*).

- 11110 -