

McNits: Mike's Palm Programming Nitpicks, Part 1

by Mike McCollister

- Originally published circa April 2002 on the “Handheld Computing Developer” website.
- This article can be found at <https://MikeMcCollister.com/palm/articles/mcnits1/>.
- As this article are quite dated, most of the links will be invalid.

Introduction

There are literally thousands of programs available for Palm OS platforms, many of which are great, and many of which are not so great. Over the years, I have compiled a list of nitpicks that I have concerning those not-so-great Palm OS programs. Some of these are suggestions brought forth by PalmSource, some by me, others are just common sense. Keep in mind that most of these are not hard-pressed rules, nor are they written in stone. At best, they will be written on paper. These nitpicks, or rules, are like the fashion rule that states that if you have a pair of pants with belt loops, wear a belt. Don't even ask me about what I think about those kids who wear their pants all the way down to their knees. See what happens if you don't wear a belt?

User Interface Nitpicks

- Read the *Palm OS Companion* for good programming style and the *User Interface Guideline* for knowing what a standard Palm user interface should look like. You can find these documents, as well as many others, at <http://www.palmos.com/dev/support/docs/>. PalmSource has put much thought into how things should be done. If your program deviates too much from the standard, then the user is hindered by having to learn something new. Keeping to the standard may very well reduce technical support.
- Make sure your button widths are at least 38 pixels wide if there is room. This makes the buttons easier to press. However, this may not be possible, especially if you have more than three buttons, as Figure 1 illustrates.
- Don't have “Palm” as part of the name of your program. It is redundant and locks your program into the Palm platform. Besides, PalmSource might not like it and there are other Palm compatible devices that are not made by Palm.
- Use modal forms for preferences. This lets the user know that something is different or something special is happening. For example, shows the main screen of a timer application called *TikTok*. Figure 2 shows the preferences screen, a modal form, for *TikTok*. Modal forms should be used for preference forms, about forms and input forms.
- Make sure that the modal forms are the width of the screen and are aligned to the bottom of the screen, regardless of the height of the form.

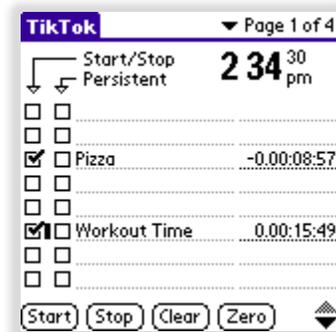


Figure 1: Sample Application



Figure 2: Preference Screen Using a Modal Form

- If you have an “OK” and a “Cancel” button, put the “OK” button to the left of the “Cancel” button. This is how most other programs order these buttons and may prevent a user from pressing the wrong button if he is not paying full attention to his actions.
- Use “OK”, not “ok”, “Ok”, “okay” or “Okay”.
- Have a small icon for your program. For goodness sakes, small icons have been available ever since Palm OS 3.0.
- Have color icons for your program, both small and large. Also, make sure that you specify which color is transparent as this will be especially important for Palm OS 5 or if a user is using third party applications such as *Chrome* or *Khroma* which can make the background of a form or the launcher to be non-white.
- Use shortcuts in menus. For example, one can use “/C” for copy and “/P” for paste. Use these types of shortcuts to allow users to bypass the menu and take a shortcut to the desired feature.
- When your program has a preference form, use “/R” for the shortcut. This is the standard way, and it will prevent the user from possibly doing something unwanted.
- Make sure you have an about box.
- Your about box should include the version number, web address and e-mail addresses. This makes it easier for the user to get in contact with you or find updated versions. If you don't have an about box, see previous rule.
- No small dots at the bottom of the screen. If you are using a modal form and don't want the thick border, then make sure that you don't have those one-pixel dots at the bottom left and right of the screen. *Figure 3* shows those dots at the bottom of the screen with a form positioned at (0, 0) with a width of 160 and a height of 160. Also, note that the “i” button at the upper right of the screen (program tips icon) is too high as it does not have the dark border above it as it does as the bottom. Both issues can be fixed by positioning the form at (0, 2) with a width of 160 and a height of 159.
- Include “Keyboard” and “Graffiti Help” in “Edit” menu if appropriate.
- Use hard up and down buttons if appropriate.
- Don't put version number in program title. This gives the program a “techie” feel to it that may scare off novice users.
- Make sure pen operations are activated when pen is released. For example, a button is considered pressed when pen is lifted. Don't stray from this, it is confusing.



Figure 3: Those Ugly Dots

Documentation Nitpicks

- Include documentation both internally to your application (in the forms of tips) and externally. The internal tips, or documentation, will cover the basic features and will be readily available to the user. The external documentation can contain a more robust information to aid the user in a more verbose fashion.
- It is important that you proof read your documentation. Watch out for words that spell checkers will show as correct. If need be, have someone else read the documentation.

- Have an HTML version of your documentation. It looks better and you don't have the carriage return problem between platforms (i.e., PC, Macintosh and Unix). You can also put links to your home page or to sales pages.

Bug Nitpicks

- Visit the Palm OS Developer page at <http://www.palmos.com/dev/>. It is full of great information for beginning to advanced developers. This page is updated frequently so visit often.
- Use the debugging tools available to you. Run your code on a debug version of the Palm OS 4.x ROM or greater. Starting with OS 3.5 the debug ROM captures many more memory leaks than previous versions. Also, use the latest version of the Palm OS Emulator (POSE). Version 3.4 of POSE introduced a memory leak detection feature that will help find those hard-to-find leaks. POSE also has a gremlin feature that will press various buttons and input text in a way that no user would even think of doing. It is like one of the infinite monkeys trying to write one of Shakespeare's plays using your program. Gremlins will find errors that you had no idea existed. There is no excuse for shipping a product with memory leaks if these tools are available.
- Make sure you call `FrmCloseAllForms()` at the end of your program. Missing this function call will result in a memory leak.
- If your program has an alarm, make sure it survives a reset.

-30-