

# Palm OS Programming - Gotchyas

[Rumkin.com](#) >> [Reference](#) >> [Palm Programming](#)

## INDEX

[Gotchyas](#)  
[Advanced UI](#)  
[PiIRc](#)  
[C Code](#)  
[Events](#)

This is the text, extracted from a presentation for Palm OS Programming Gotchyas and Pitfalls. I don't use Adobe and I wanted a textual reference instead of printing out slides, so I did this. View the original information at [PalmSource](#), number 106.

### Confusing Form Object ID with Object Index

```
pBtn = FrmGetObjectPtr(MainOKButton);
```

- Most Form Manager routines need object index
- Compiler can't catch: both are type UInt16
- Typical error messages:
  - Object not found in form
  - Index out of range
- Solution: check API parameters carefully and use FrmGetObjectIndex()

### Correct ID, Wrong Resource

```
pLogo = DmGetResource(bitmapRsc, 10004);
```

- If DmGetResource() can't find match in the last opened resource database, continues search through resource chain
- Solution
  - Use Console opened command to display theresource chain
  - Don't use resource IDs <= 128 or >= 10000
  - Use DmGet1Resource()

### Apps Must Use app1 as Database Type

- Runs when you debug, but does not show up in Launcher
- Launcher only displays icons for resource databases of type appl
- App built with a different type can run, but must be launched via code
- Set in PalmRez post linker panel

### Popup Lists

- Assign list object ID to trigger
- Check for popSelectEvent
- Override default behavior of popSelectEvent if list items are not defined
  - In resources
  - By using LstSetListItems()
- Return true from event handler

### FrmCopyTitle & FrmCopyLabel

- These calls copy a string to a buffer
- Buffer size is determined by size of label or title defined in resources
- Solution 1: Make placeholder label/title "big enough" to hold anticipated strings
- Solution 2: Use FrmSetTitle() for forms
- Solution 3: Use text fields instead of labels if you need to change at runtime

### Common C/C++ Problems

- Learn C before learning Palm OS APIs
- Pointers and data type promotion

- Example
  - Char \*p;  
StrCopy(p, "Ooops");  
P = MemPtrNew(60 \* 1024);

#### Standard C Libraries vs. Palm OS APIs

- Use String and Memory Managers
  - Smaller app footprint
  - Support for multibyte character sets
- However, check API parameters carefully
  - StrPrintF is almost the same as sprintf
  - MemSet parameters switched from memset
- Exception handling
  - Use Error Manager instead of C++ exceptions
  - ErrTry, ErrCatch, etc.

#### Form Drawing

- Call FrmDrawForm before drawing on form
  - Problems with modal dialogs and "save behind" bit
  - Debug ROMs will catch
  - Can use WinScreenLock to improve performance
- Beware functions that draw immediately
  - FldDrawField, LstDrawList, etc.

#### Form Updating

- Default frmUpdateEvent calls FrmDrawForm()
  - Erases form
  - Draws form border
  - Draws form objects
  - But not old-style gadgets
- Custom drawing not updated unless you handle the event yourself
- Use extended gadgets to automate
- If you do handle frmUpdateEvent return true to stop system from handling it after you

#### Text Fields

- Don't use FldGrabFocus()
  - Use FrmSetFocus()
  - Call FrmSetFocus() after FrmDrawForm()
- Use PrvSetFieldText() from NetSample sample project
  - Does all the right things
  - Avoids all the standard difficulties
  - Supports immediate/delayed update and appending

#### Database Names and Creator Codes

- Creator codes should be unique
  - Benefits your customers if unique
  - Automates clean-up of device when deleting
  - Register code before coding starts
  - [www.palmos.com/dev](http://www.palmos.com/dev) web form
- Database names must be unique
  - Remember, apps are databases also
  - Use form "AppName-Code"

## Palm OS Data Structure Access

- Don't access UI data structure fields directly
- Use accessor functions
  - FrmGetFormId(), LstGetSelection(), etc.
  - New accessors in SDK 4.0 Update 1
- Good way to prepare for Palm OS Version 5
- Palm OS Emulator reports (non-fatal error)
- Catch in compiler
  - `#define DO_NOT_ALLOW_ACCESS_TO_INTERNALS_OF_STRUCTS`
  - `#include <PalmOS.h>`

## Database Searching

- DmFindSortPosition()
  - Design: maintain sort order when adding records
  - Returns index after any matching records
- When used to find records:
  - If index > 0, compare previous record
  - If record matches, you found search target
  - If record does not match, search target is not in database

## Global Variables

- Not accessible if app is not active
  - Find, receive beam, alarms, etc.
- Global data not always obvious
  - String constants
  - Smart code model intra-segment jumps
  - Access at end of long function call chain
- Use `#pragma warn_a5_access on`
- Use "PC-Relative strings" options

## Error Avoidance

- Require function prototypes
- Turn on all warnings and errors
- Use `DO_NOT_ALLOW_ACCESS_TO_INTERNALS_OF_STRUCTS`
- Use assertions
- Use POSE and debug ROMs

## POSE and Debug ROMs

- Palm OS Emulator will catch numerous errors
  - Memory leaks, data structure access, stack overflow, low memory access, ...
  - Gremlins and new minimize feature automate the hunt
- Debug ROMs check for invalid parameters
  - Range checking, forced update events, etc.
- Always use the latest POSE build
- Test using ROMs for minimum and latest versions of Palm OS

## Suggestions

- Palm OS® Essentials course
  - Lots of lab time
  - Developer instructors
  - Terrific sample applications with source

- Palm OS Recipes
  - Step-by-step solutions to common problems
  - Under Documentation link on [www.palmos.com/dev](http://www.palmos.com/dev)
- Sample code
  - Use CodeWarrior Find feature, grep, etc.
- Third-party books

---

Make it idiot-proof, and someone will make a better idiot.

Tyler Akins <[fidian@rumkin.com](mailto:fidian@rumkin.com)>  
[Legal Info](#)