

Pocket Smalltalk FAQ 1.1

By [Thierry Reignier](#) — October 10, 2001 (Messages 1250 - 1612).
 Revised and updated by N. Chamfort (January 20, 2002).
 This FAQ covers PST version 1.6a4

Table of Contents

1. [Questions](#)
 - 1.1. [Versions and Documentation](#)
 - 1.2. [IDE](#)
 - 1.3. [Development](#)
 - 1.4. [Testing](#)
 - 1.5. [Applications](#)
 - 1.6. [Virtual Machine](#)
 - 1.7. [Known Limitations](#)
2. [Known Errors](#)
 - 2.1. [IDE](#)
 - 2.2. [Development](#)
 - 2.3. [Testing](#)
3. [Corrections](#)
 - 3.1. [Packages](#)
 - 3.2. [Virtual Machine](#)
 - 3.3. [Upgrading from PST v1.5 to PST v1.6](#)
4. [Improvements](#)
 - 4.1. [IDE](#)
 - 4.2. [New Packages](#)
 - 4.3. [New Methods](#)
5. [Contributors](#)
6. [Document History](#)

1. Questions

1.1. Versions and Documentation

Q1. What is the latest PST version?

There are 2 different latest versions: one built with Dolphin Smalltalk (Dolphin v4.0), and one built with Squeak Smalltalk (Squeak v2.7).

Development of the Dolphin version of the IDE is more up-to-date than that of the Squeak version. Both versions can be found at: www.pocketsmalltalk.com/pst-download.html. New users are encouraged to begin with the Dolphin version of the IDE.

The latest Dolphin-PST version is 1.6a4. It runs only on Windows. The latest Squeak-PST version is 2.0a. It runs on all platforms supported by Squeak (Windows, Mac, Linux).

Note that GEOS port for Nokia is broken from v1.6. It still works for v1.5 and v2.0a.

The next release, probably called something like 2.0w (2.0 on Windows) will include small improvements and a LOT of changes to the VM. The VM is expected to be about 35% faster. [EA]

Q2. Where can the FAQ be found?

This FAQ is available at www.pocketsmalltalk.com.

Q3. Where do I get documentation for PST?

Documentation is available both at www.pocketsmalltalk.com and in the PST distribution.

Q4. Which Palm OS versions are supported?

PST 1.6a4 supports all sysstraps through Palm OS version 3.5. Palm OS v4.1 sysstraps are not yet available. However, PST will run on Palm OS 4 if your application does not call any specific 4.1 sysstraps.

A check-program could be written to verify the minimum and maximum versions a certain project requires. Going through all methods of "save in project" packages, looking for the sysstraps they call and listing what version the traps are implemented in would tell what Palm OS versions this application can run on. These stamps could even be included upon saving a package with a special menu option.

The file `pocketstug>>files>>PRJ, osVersion-v1-and-v2.zip` contains code to check at launch time what version the device runs.

Q5. Is there a port of PST for WinCE?

Stefan Matthias Aust is developing one.

Q6. Where do I find more information on Smalltalk?

Start your quest at the [PST links page](#).

Q7. Where do I get technical information about Palm Sysstraps?

See the Palm OS documentation at:

www.palmos.com/dev/tech/docs/
 Palm OS programmer's Companion, Vol 1
 Palm OS programmer's API reference

See books on Palm OS development at:

www.palmos.com/dev/tech/docs/other.html
 Palm Programming, The developer's guide

Slides from Palm Source 2000 in San Diego (were available) at:

www.palmsource.com/slides/Track_600/601.pdf

Download the Palm OS SDK 4.1 at:

www.palmos.com/dev/tech/tools/

The SDK contains the latest Palm OS header files, and is useful for retrieving constants or C enum definitions.

Q8. Where do I find an update of Joey's version 1.5 tutorial (ManHourCalculator)?

An update of the PST tutorial for 1.6a4 can be found at `pocketstug>>FAQ + tutorial`.

A new chapter was added to show the debugger.

Q9. What was Eric's nickname at school?

System Child ;)

1.2 IDE

Q10. Why can't I find senders of systraps?

For example, the method SYSTRAP>>DmOpenDatabase: arg1 with: arg2 with: arg3 does not return any senders though Database>>openID:mode: calls it this way:

```
SYSTRAP DmOpenDatabase: 0 localID: aLocalID mode: aMode
```

Ben Hultink:

The name of the SYSTRAP parameters does not matter (with: or index: or fool: or...) it's only the number position and type that does count. The parameters are pushed on the stack and, in that way, passed to the VM machine where the systrap is executed.

1.3 Development

Q11. How can I add more systraps?

What is the procedure to add more systraps in systrap.st?

Palm OS 4.1 systraps are not available yet. In the palmos/ folder of the PST installation, a SysTraps.txt file lists all supported systraps. It might be the starting point for generating systraps.st. See the file CoreTraps.h from the Palm OS SDK for a list of all official traps per OS version.

Look at the class named SYSTRAP. All the methods that implement traps for Palm OS are available as class-side protocol. If you would like to add new entries, follow the same pattern used in each of the class methods there. There are MANY examples of how to define an OS call there. [EA]

Q12. How are C enums supported by PST for systrap calls?

To illustrate, consider the SYSTRAP specified for getting preferences (see the Palm OS reference guide for details):

```
UInt32 PrefGetPreference (SystemPreferenceChoice choice)
```

where choice is a constant like prefVersion, prefCountry, prefShowPrivateRecords from SystemPreferenceChoice enum (see preferences.h in SDK). The question is, how to pass these enums as SYSTRAP parameters in PST?

C expects the relevant bits to be in the first byte. Enums indeed start at 0. So if you use (x bitShift: 8) where x is the element you need counted from zero, things work fine. This is presumably a big / little endian thing. What happens if there are more than 255 elements? For enums bigger than 255 the first and last byte of the word needs to be swapped but I have not tested this. [BH]

So, the SYSTRAP call would look like:

```
SYSTRAP PrefGetPreference: (##prefCountry bitShift: 8).
```

where ##prefCountry is a constant defined to be 2 since it is the second enum item in SystemPreferenceChoice enum. A more readable version of this call would be:

```
SYSTRAP PrefGetPreference: ##prefCountry asBigEndian
```

where asBigEndian is implemented as follows:

```
Integer>>asBigEndian
^self bitShit: 8
```

Likewise, if you want to interpret a value returned from a systrap, you need to bit-shit it; e.g.:

```
code := SYSTRAP MemPtrResize: (CPointer new) size: 10.
"code is 257, bit-shifted is 1"
(code bitShift: -8) = 1 ifTrue: [self error: 'Could not resize'].
```

A simpler version of this call would be:

```
code asLittleEndian = 1 ifTrue: [self error: 'Could not resize'].
```

where asLittleEndian is implemented as follows:

```
Integer>> asLittleEndian
```

```
^self bitShit: -8
```

Q13. What is class PopupApplication for?

Make a subclass of PopupApplication if you need a modal windows to popup. Such a window often asks a question or asks for confirmation. Call #popupFor: to start it and call #leave to exit it.

Note that in the default 1.6a4 installation there is not support for the calling application to receive any return value from child popup-applications. Install `returningFromApplication.zip` from `pocketstug>>files>>PRJ` to add this feature. This modification enables your application to receive the exit state by looking at the value returned by #popupFor:.

Q14. Graphics: how to avoid flickering while refreshing?

"I wrote a small program displaying a clock with an update of the hands every second. The display is built as a bitmap showing the dial and figures (3-6-9-12), and lines for the hands. Nothing special, the program works. But: every second both bitmap and hands are redisplayed, but there is a noticeable flickering of the hours- and minutes- hands.

Two questions:

1. Is there any way to "freeze" the form, and to unfreeze when the update is done?
2. The reason why I redisplay the bitmap is that I move the hands by "erasing" the last position before redrawing in a new position. Doing so also removes parts of the bitmap. Is there a way to put the bitmap on the screen once, and to move the hands only, without damaging the bitmap?"

There are some pointers on 3com's site for how to do games, moving graphics. It has to do with off-screen buffers and such. I don't have an exact url, but do a search on the palm developer's area and you should find it. It'll be specific to programming in c, but you should be able to extrapolate to Smalltalk. [JD]

Q15. How can I store objects in Palm databases with PST?

A persistency framework is under development by TR.

The C equivalent can be found at `pocketstug>>files>>prj`

Q16. What are "no name" and "" packages for?

Joey:

The problem has to do with the 'new way' of handling packages, uncommitted code, and not losing changes when quitting. Before 1.6a4, any code that you did not assign to a package was lost when you shut down PST. The current version creates a 'no name' package to hold this code, which is then inspected on shutdown to force you to move it to your own package. Unfortunately, the save logic seems to be a bit pear shaped here. To 'fix' this problem: before trying to save the project, create a new package, set it as the default and then remove the 'no name' package. (If you've already created code and it is in the 'no name' package, move it to your package first.) Then save the project. I just recreated your problem, and this is what worked for me to get around it. I will fix that logic for the next release.

Thierry:

My wild guess at the beginning was that uncommitted would be the package were my not-assigned-to-package code would be saved and that no-name was just there at start to force one to create an own package.

Joey:

Maybe that's what it was. Eric put that code in earlier this year because people were complaining about losing their changes (it's happened to me, I can tell you). I wasn't involved with this change so I can't address it fully (I could if I dug into the code...). Anyway, perhaps I stated the problem/fix incorrectly, but if you do what I said in that message, then you will be able to save your code with no worries.

Eric:

Oops, I'll fix that [error upon saving when 'no name' package is still present]. The problem is the "no name" project as you so correctly guessed. I do not need to have that when a developer loads a project. The "no name" project is where things go now when you create a brand new project. The intent is for you to save it when you are ready.

Thierry:

This was never corrected. See workaround later.

Q17. How do I set a custom font?

I tried to do this and it seems all to be right. It works, no error messages and seems to set the font. But when I try to switch to my new font, I still get a font with ID 0. What I do first is to define font in .rcp:

```
FONT ID 1000 FONTID 128 "font.txt"
```

and then I reference it in my code.

```
| newFont handle temp |
newFont := SYSTRAP MemHandleLock: (SYSTRAP
DmGetResource: (Integer fromBytes: 'NFNT')
fntID: 1000).
SYSTRAP FntDefineFont: 128 resID: newFont.
"Seems like we defined the font"
"Now set it current"
Window font: 128.
"Try to draw string"
Window
  drawString: ('ABCDEFGH', (SYSTRAP FntGetFont) printString)
  x: 5
  y: 45.
```

The string is output with font 0, but FntGetFont returns 128.

Ben Hultink:

(from an old issue, message 985)

```
font: fontID
"Answers the old font ID."
^SYSTRAP FntSetFont: (fontID bitShift: 8).
```

Works for me! Don't forget bitShift.

Thierry:

Palm OS reference says that: your font ID "has to be greater than fntAppFontCustomBase (currently 128). Values less than or equals to 128 are reserved for system use. Note that the font IDs are 8-bit unsigned

value, and so must be less than 256."

Q18. How can I handle real Palm buttons?

In a regular Palm OS application, this is done by adding a handler for the `keyDownEvent`, which handles both graffiti characters and hard-keys (you need to do a test to see which applies).

In PST, override the method `#Application>>onKeyDownEvent` in your application class, and examine the "key" and "modifiers" arguments to see what you've been passed. [VPutz]

Q19. Does PST support video pages?

This functionality is handled by the Palm OS window manager functions.

In particular, look at the Palm OS documentation for `WinCreateOffscreenWindow`, `WinSetDrawWindow`, and the like (available in PST through the class `SYSTRAP>>WinCreateOffscreenWindow:with:with:with` messages and the like).

Q20. Is it possible to beam with PST?

Q: I'm writing an application and would like to be able to "beam" information to another computer (using either the Exchange Manager or raw IR). Do I need to use SYSTRAPS or add primitives to the VM?

The key here is that exchange manager will actually send in a launch code notifying you that something has come in. The current VM does not currently support this launch code. This launch code has serious limitations like sub-launch (no globals and such). You would need to add the stuff to the VM and recompile. The new source [1.6a4] is coming out soon. [EA]

Q21. How can I add an item to a popupTrigger?

Q: How should an application add an item to the list of a `popupTrigger`, and then get the new entry to become the selected item, and have this reflected on the Form?

From a C perspective, but the same should apply in Smalltalk, wait until you get the event that's going to pop up the list before setting up the list and the selection. This is a `ctlSelectEvent` for the popup trigger. You can let the OS actually pop up the list, and wait for the `lstSelectEvent` that tells you which item was selected, if any. [PE]

Q22. Are Delays supported by PST?

For implementing a clock displaying time (and some other information) I need a class which sends a callback every second (or whatever timeframe). I haven't found anything in the standard packages. Is such a class available?

If I have to implement it myself, which SYSTRAPS do I have to use?

Hans Scholz:

I suggest you overwrite "eventTimeout" in your application (should be msec). Also overwrite `handleLastEvent`, it will call super `handleLastEvent` after the screen is refreshed. This is not exactly "a class which sends a callback" but might fit your requirements.

Q23. Which method actually starts my application?

`Smalltalk>>start` is the method that calls and starts your application. Your application should override it and it should be saved in one of your packages.

Q24. How does my application exit PST?

To exit your application, your exit method (preferably implemented as a subclass of class `Application`) should send "self class eventuallyExitSmalltalk".

Q25. When do I need mathLib.prc?

Before running on your device or from the emulator you must ensure that `mathLib.prc` is installed as soon as your application uses any Double or Fraction objects.

1.4 Testing

Q26. Uploading ROMs via USB connection

I'm trying to use the Palm Emulator (3.3). I have a Palm m505 and want to upload the ROM to my PC (for the emulator), but I only have the USB connection (No serial connection). I can only select COM1 or COM3 to download. [TS]

The latest version of the Emulator (3.3) supports downloading ROMs via USB. Alternatively, you may use a MAC, if you have one lying around. [JJM]

Q27. Is there a debugger?

PST includes a very simple debugger to show the last smalltalk method that was executed.



The upper list of the debugger shows methods on the stack in decreasing order. Press the arrow buttons to browse up and down the stack. If you select a method you can then view its associated parameters and local variables.

1.5 Applications

Q28. Have any business applications been built with PocketSmalltalk?

So far, PST has been used primarily for games (See `pocketstug>>files`).

Ben Hultink used PST to write a commercial application for registering car usage, needed to satisfy the Dutch tax office. Ben reports that after one year of free usage, users have to register and they actually do! For details, see: www.becorora.com or www.autorit.becorora.com for details (in Dutch).

Post a message to list if you learn of details about other commercial applications.

Q29. Where can I find code examples?

In pocketstug>>files>>PRJ you can find:

- BlackJack
- VolleyScore
- Sokodan (only prc)
- ClueTrain (change vm path and exit method if re-generate with 1.6),
See message #1517 from Nathan Eaton for bug corrections
- NewFlippy (change vm path and exit method if re-generate with 1.6)
- SunitF101 (never completed because of PST limitations, see readMe file)

In the standard PST release installation, you can find:

- PST demo
- ManHourCalculation (see Joey's tutorial and Thierry's update for 1.6)

Other:

- PocketSet (Jason Dufair, ask for it)

Mail me ([Thierry Reignier](#)) if you want your application be listed here.

1.6 Virtual Machine

Q30. Why is there a difference in number of vm invoke and return calls?

Q: Counting the number of `invoke_method()` and the number of `local_return()` + `nonlocal_return()` calls in the VM, I thought they would be the same number but they're not : there are a few more `invoke_method()` than `return()`. Is this normal ?

(I would like to make a Smalltalk profiler, so I need to know exactly in which method I am)

A: The asymmetry between `invoke_method` and `*_return` is normal. The Smalltalk stack is basically independent of the C stack. If you are trying to build a profiler into the VM, your best bet is to use the global variables `'instruction_ptr'` and `'receiver'` along with the functions `'infer_selector'` and `'infer_class'` to find out where in the Smalltalk program execution is taking place. [AB]

1.7 Known Limitations

Q31. Are PalmOS launch flags supported?

No.

Q32. Are callbacks supported?

No.

Consequences: any special widgets that use callbacks (like tables) cannot be used in PST.

Q33. Are Smalltalk exceptions supported?

No.

The original PST whitepaper (www.pocketsmalltalk.com/whitepaper.html#exceptions) mentions support for exception handling (ExceptionHandler class) but this class does not exist in the core package (1.6a4 and 2.0a).

2. Known Errors

2.1 IDE

E1. Error #indexOfSubCollection:startingAt:

As of 1.6a4, if you receive this error: "undefinedObject does not understand #indexOfSubCollection:startingAt:", browse packages, set a package as default via the "set as default" option and remove the "no name" project.

Eric:

Oops, I'll fix that. The problem is the "no name" project as you so correctly guessed. I do not need to have that when a developer loads a project. The "no name" project is where things go now when you create a brand new project. The intent is for you to save it when you are ready.

E2. A "no name" package is still shown upon reloading an old project

As of version 1.6a4, this package is always part of a PST project after a reload. It should not be reintroduced if previously removed. To be able to save your project, remove it first.

E3. "PstRuntimeSessionManager does not understand #inspectItFor:"

"Inspect it" in a Workspace now produces the following error [1.6a4]:

```
"PstWorkspace does not understand #sourceCode"
```

followed by:

```
"PstRuntimeSessionManager does not understand #inspectItFor:"
```

Joey:

If you have the IDE source loaded, you can just change the #inspectIt method of class PstShell to look like this:

```
self sourceCodePart inspectItFor: self receiverForEvaluations.
```

The problem was sourceCode should have been sourceCodePart. If you don't have the IDE source loaded, you'll have to wait for a patched version of 1.6a4

E4. "PstMethodListBrowser does not understand #classLibrary:"

Loading the BlackJack application (pocketstug), selecting "Database" in initialize method and choosing "Browse it" returns this error:

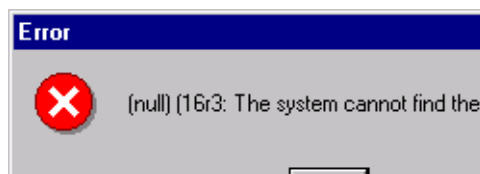
```
BlackJackMatch>>Initialize
database := nil.
(Database findDatabaseIDByName: 'BkJk' )
```

Pressing ok gives more errors. Should not be a ok, cancel warning, just a ok one.

E5. File location error when generating an application

The error occurs because the path to the VM is set incorrectly. Look at ##resourceDB1 in the Constants Browser and make sure it's valid.

E6. Path spelling error when generating an application



An extra back-slash in the string for resourceDB2 (##ResourceDB2*: 'resources*.bin') may give an error.

E7. Package error

Symptom:

Add a method (let's say #organizer) to a superclass and then remove it from 2 subclasses. Then, when saving the project from the PST launcher window, the error "method not in any class" is raised:



Going through the package browser, saving packages one by one I could locate the error: an old reference to #organizer was still filed under "methods", though I deleted it from a "browse implementers" browser. I could find a workaround (re-implementing the method, seeing twice the same row in package->method tab and choosing remove from package for both).

Note that removing from package gave the same error if I did not re-implement the method in the first place.

If the error can't be fixed, fine, but a more specific error message would be better: "method: #organizer not in any category for class: Blabla, package: blabla", and maybe: "Do you want PST to remove it?" (Sent to Joey, 29.08.2001)

E8. Error when saving a project

Symptom:

Load a project, then save it. The IDE gives an error that the "none" package still exists. If you press ok, the project file is not corrupted but if you press cancel it gets corrupted.

Saving a project should first make a back-up (or keep a backup in memory). If E1 is corrected then this problem will be resolved.

E9. Error when saving a method

I saved a method to a package different than the one where the class is defined. Modifying this method afterwards and then saving it, moves it back to the default package. The method should stay in the given package if this package is savable.

E10. Removing packages

Removing a package does not remove or unload classes.

E11. Hover help for the save project icon says: "Save project"

1.6a4 shows an incorrect spelling.

E12. Error when selecting code

Double-clicking on a method ending with a semi-colon does not select that last semi-colon character.

2.2. Development

E13. Trouble with garbage collection

Every time I call the `#collectGarbage` manually I get a mark stack overflow error (using the Palm Emulator). I seem to get the same problem when Smalltalk calls it automatically, because after several button-clicks in my application I get the same message.

I tried the `vm16a4.prc` but I still have the same problems. I can only make a limited number of pen-clicks, before the mark stack overflows.

The mark stack is used by the garbage collector to hold objects that have been "marked" during the mark & sweep collection. What it probably means is that you have too many objects being referenced, but one would need to check the VM source to confirm this. [JH]

2.3 Testing

E14. Error upon exiting any application

Symptom:

On hitting the Done button on the Demo app, you see an "application accessed invalid chunk" or something like that, unless the check was turned off in the Palm Emulator, and running the application on a physical Palm device was fine.

Solution:

It turns out that the way we were exiting from a form was BAD. The implementation of the Done button method handler called:

```
Smalltalk exit
```

If you are INSIDE a form, this is BAD.

Do not use `Smalltalk exit` in a form event handler, UNLESS you are handling the `appStopEvent`.

Instead, use:

```
Application eventuallyExitSmalltalk
```

Here's why: calling `#exit` will exit the VM. Well, the OS is not done sending events to the application. So what happens is that the app disappears, and the OS sends events to someone who is no longer there. The correct way to exit is to tell the OS somehow you wish to exit, and to let it send an `appStopEvent`. When your application receives this event, no more events will come through. So this is the right time to exit. PST 1.6a4 includes the method `Application class>>#eventuallyExitSmalltalk` which does the trick. Keep in mind that the event loop must run after this call is done in order for the exit to actually occur. [EA]

E15. Error upon leaving any popup application

Symptom:

Errors after using the #leave method in PopupApplication.

Ben Hultink:

This is an old problem, this weekend I tried the Handera emulator so I changed it a little because it didn't work any more. Here is the stuff for PopUpApplication:

Create an instanceVariable named finished and modify the following two methods:

```

PopUpApplication>>leave
    "BJH"
    finished ifFalse: [finished := true].

PopUpApplication>>handleLastEvent
    "BJH"
    | event |
    finished ifTrue: [
        finished := false.
        self release.
        Form returnToForm: self owner class formID.
        [event := CEvent waitForEvent: -1.
         event eventType == ##winEnterEvent]
        whileFalse.
        ^false].
    ^super handleLastEvent.

```

Explanation: it used to wait for the winExitEvent but I don't get this for the Handera. Actually you should resend the winEnterEvent again so the owner can pick it up.

Christian Schmidt-Setzwein:

PopupApplication>>leave. The Palm OS generates two more events (CtlSelectevents) for the done button (from the Popup form, which is already closed), which should be handled from the main form (which is now the active one).

To bypass this problem and make all PST programs, even with PopupApplication forms, run properly on the emulator, add the following:

```

Application class>>consumeCtlSelectEvents
    | event |
    [event := CEvent waitForEvent: -1.
     CEvent sysHandleEvent: event.
     event eventType == ##ctlSelectEvent]
    whileTrue.

PopupApplication>>leave
    finished ifFalse: [
        finished := true.
        Form returnToForm: self owner class formID.
        self class consumeCtlSelectEvents].

```

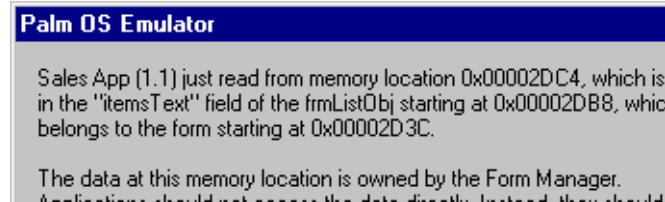
Thierry:

Christian's solution is the one implemented in 1.6a4. Ben: does it work for you on your Handera with Christian's solution?

E16. Listbox error upon opening the debugger

Symptom:

Using the latest Emulator (3.3) and a non-debug Palm OS 3.5 ROM, the following warning occurs four times whenever the application enters the debugger:

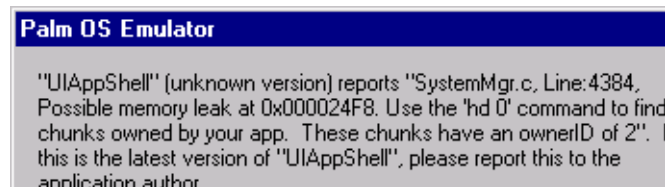


It seems like the VM in primitive 100 is not following Palm OS rules and accesses Palm resources directly.

```
Listbox>>primSetListChoices: choices
      controlID: controlID freeVisibleItems: boolean
<primitive: 100>
^self primitiveFailed.
```

E17. Memory Leak upon exiting applications

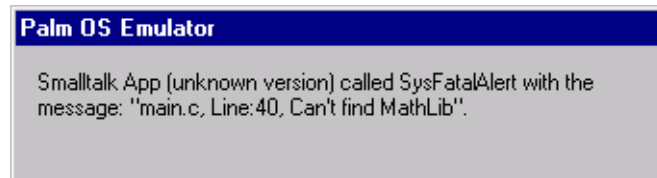
Running the latest emulator and a Palm OS 3.5 debug ROM gives the following error:



The VM apparently does not release all memory it acquired. Solution: do not use a debug ROM or try to fix the VM.

E18. MathLib error

Running the tutorial, for example < or any application using Double or Fraction, without having installed mathLib.prc, gives the following error:



So next time, don't forget MathLib.

E19. Debugger

Symptom:

- Every time my application shows the debugger, I always get an empty upper pane with just 1 row: "Object>>error:"
- an empty lower pane with just 1 row of parameters



Pressing the arrow button at the right-hand side has no effect for both panes.

What could be wrong with my app?

Have I set wrong System Properties settings? (In this case, `##debug` is set to true).

E20. Error with `##debug` option set

Symptom:

I went through the tutorial and wanted to see effects of turning the `Constant Browser>>System Properties>>##debug` to false. When running the newly generated tutorial, I get an error:

```
"ManHour Calc (unknown version) called SysFatalAlter
with the message: "main.c, Line 40, unsued primitive"
```

Could the VM show what primitive number failed? What's wrong?

3. Corrections

C1. General: Reporting Bugs or Corrections

To report bugs, send them to the pocketstug discussion group with subject starting with the text "[Bug]" on the subject line, or else send them to me ([Thierry Reigner](#)). I will see that the FAQ is updated with known bugs.

You might also want to publish them in the files directory: `pocketstup>>files>>bugs`.

Known bugs were found in PST 1.6a4.

If you send a smalltalk package containing only your bug corrections, please follow this naming convention:

```
correctedXxx.st
```

— where Xxx is the name of the original package. Example: `correctedCore.st`.

3.1 Packages

C2. Bug missing method: SortedCollection>>reallyExpandAtEndBy:

```
SortedCollection>>reallyExpandAtEndBy: amount
  "TREG: missing method"
  | oldSort |
  oldSort := sortBlock.
  super reallyExpandAtEndBy: amount.
  sortBlock := oldSort
```

C3. Bug in ListModel>>selectionIndex:

```
ListModel>>selectionIndex: newIndex
  "TREG: added selection to changed method"

  selectionIndex == newIndex ifFalse: [
    selectionIndex := newIndex.
    self changed: #selection with: self selection].
```

It provides the selected object in the `#update:with:` method.

C4. Bug in Application>>showFor:

TREG: This method is unclear. Is it really needed? How is the next form opened?

```
Application>>showFor: instance
  "TREG: replaced self by instance class in first line"

  | temp |
  Form gotoForm: instance class formID.
  [instance isNil] whileFalse: [
    self consumeExcessEvents.
    instance create.
    instance handleLastEvent.
    instance eventLoop.
    temp := instance nextToRun.
    instance nextToRun: nil. "clear forward reference"
    "Note: hard assignment is used here to avoid keeping
    a perpetual reference to the first shown application...this
    is one of the few places where := is actually required"
    instance := temp].
```

C5. Bug in Dictionary>>new:

Jean-Jacques Moreau: The class Dictionary currently allows dictionaries to be created with a size of 0 which later on causes hashing problems (division by zero). A brief look at Set>>new: suggests the following fix:

```
Dictionary>>new: capacity
  ^(self basicNew: (capacity max: 1) * 2) initialize.
```

C6. Bug in Integer>>readFrom:radix:

```
Integer>>readFrom: stream radix: radix
  " ^ nil or Integer
  TREG: if the stream has no digit return nil"

  | value char position |
  position := stream position.
  [(char := stream peek) notNil and: [char isDigit]]
    whileTrue: [
      value := 0.
      stream next].
  value ifNil: [^nil].
  stream position: position.
  value := 0.
  [(char := stream peek) notNil and: [char isDigit]]
    whileTrue: [
      char digitValue >= radix
      ifTrue: [^value].
      value := value * radix + char digitValue.
      stream next].
  ^value.
```

C7. Bug in TextField>>grabFocus

```
TextField>>grabFocus
  "1.6a4 implementation: SYSTRAP FldGrabFocus: self getPointer
  Palm: you rarely need to call this function directly.
  Instead use FrmSetFocus which calls FldGrabFocus for you.
  One instance you need to call it directly is to set the focus in
  a field contained in table cell.

  TREG: if you only call FldGrabFocus, copy, paste, cut won't work"

  Better to call this method from your application via widgetName:
  rather than directly calling Form>setFocus:"

  | formPtr index |
  formPtr := SYSTRAP FrmGetActiveForm.
  index := SYSTRAP FrmGetObjectIndex: formPtr id: id.
  SYSTRAP FrmSetFocus: formPtr index: index
```

C8. Bug in Database>>recordCount

```
Database>>recordCount
  "Return the number of records in the database.
  Deleted records are also included.
  See recordCountInCategory: aCategory to get only active records.
  Record indexes are 0..recordCount-1"
  "TREG: If the database has just been created, systrap
  fails so return 0"

  dmOpenRef fNil: [^0].
  ^SYSTRAP DmNumRecords: dmOpenRef.
```


C9. Bug in Database>> recordCountInCategory: aCategory

```
Database>>recordCountInCategory: aCategory
" ^ <Integer>aCategory <Integer>
Return the number of records in the database for a specific category
Use ##dmAllCategories to excluded deleted records"

"TREG: If the database has just been created, systrap fails so return 0"
dmOpenRef ifNil: [^0].
^SYSTRAP DmNumRecordsInCategory: dmOpenRef category: aCategory
```

C10. Bug in String>>trimBlanks

Jean-Jacques Moreau:

"The current version of String>>trimBlanks does not work with either empty or blank strings. Appended is a suggested fix, which reverses the order of the tests."

```
String>>trimBlanks
| first last |
first := 1.
last := self size.
[first < last and: [(self at: first) isWhitespace]]
  whileTrue: [first := first + 1].
[first <= last and: [(self at: last) isWhitespace]]
  whileTrue: [last := last - 1].
^self copyFrom: first to: last
```

C11. Possible bug in Transcript >>nextPutAll:

Jean-Jacques Moreau:

"Currently, the Transcript does not scroll when displaying long strings. Even worse, it crashes when displaying strings >= 100 characters. Here is a suggested fix."

```
Transcript >>nextPutAll: string
| window stream line |
window := Window.
stream := string readStream.
[stream atEnd] whileFalse:
  [line := stream upTo: Character cr.
  window
    drawString: line
    x: x
    y: 150 - window fontHeight.
  x := x + (window widthOfString: line).
  self cr].
```

C12. Possible bug in Integer>>~=

Jean-Jacques Moreau:

I believe there is a bug in the ~= comparison method for integers. I have two collections, of identical sizes. The following expression generates an error (but it shouldn't):

```
coll1 size ~= coll2 size ifTrue: [self error: 'different sizes']
```

The following expression works as expected, and does not generate an error:

```
coll1 size = coll2 size ifFalse: [self error: 'different sizes']
```

3.2 Virtual Machine

C13. Bug in vm.c>>infer_selector()

Jean-Jacques Moreau:

The method infer_selector(class, ip) in vm.c can fail to return the corresponding selector. This is due to an improper test:

```
if (ip > ip_start)
```

The correct test is:

```
if (ip >= ip_start)
```

3.3 Upgrading from PST 1.5 to PST 1.6

C14. To make old 1.5 projects work:

Don't forget these 2 changes:

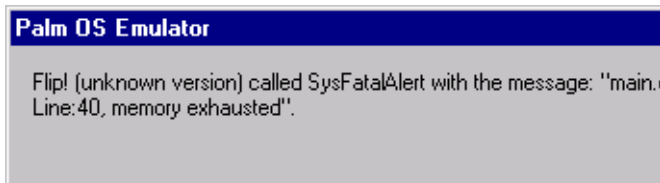
- Change the vm resource path in the Constants Browser (called vm16a4.prc or vm-math.prc)
- Change main application's exit method from:

```
Smalltalk exit
```

to:

```
self class eventuallyExitSmalltalk
```

Otherwise you might get this error message:



4. Improvements

I. How should I submit suggested improvements to Pocket Smalltalk?

If you are proud of a some functionality that you've built, don't hesitate to post the packages. Send them to the pocketstug discussion group or to me. I will make sure the FAQ is updated. You might also want to publish them in: `pocketstup>>files>>prj`

If you send a smalltalk package adding functionality to original packages, please follow this naming convention:

```
ExtendedXxx.st
```

where Xxx is the name of the original package. Example: `extendedCore.st`.

If you send a totally new packages or set of packages, please also include a simple demo to ease understanding.

4.1 Suggested Improvements to the IDE

I1. List of 4 last projects in File menu

A request was made to have a list in the File menu showing the 4 last projects as in MS-Excel or Word.

I2. Is it possible to automatically launch a project?

Is it possible to pass a project file as parameter to PST? Add a pocketSt /? e.g., to see all parameters in Windows?

I3. When exiting, could PST offer more options for saving the project?

PST could offer the choice: "Do you want to save before exiting?: Yes, No, Cancel"

I4. Remembering categories?

Quick categorize should show all categories, even those entered manually.

I5. Categorize SYSTRAPs?

Add categories per Palm OS Manager and version.

I6. Add a "project properties" window?

From the File menu a "Properties" option could display a new window with major properties:

- Project name (human readable)
- Pilrc path
- Start directory (give the whole path from the top)
- System directory (local to start directory)
- Working directory (local to start directory)
- Some items from the "System Properties" constant categories

Display project name without path in title bar or PST.

I7. Add a version number to packages? — a date is not enough

To ease manipulation of packages, a version number could be added as it was done for the vm (alias vm16a4.prc).

I8. Subclass CPointer into Pointer and Handle (HeapAllocator is super class)?

Ask for my handleRedo.zip, though changes are required in the VM that I haven't been able to do yet.

I9. Sort by class (and method names) all new methods of a package?

Sorting could be done upon saving from special save menu

I10. Add "Copy to another class" in class browser?**I11. Reloading packages?**

Reload a large project without having sorted packages causes problems because classes are in wrong package. Pressing "Continue" goes fine. Could get a list of errors?

I12. Add Find and Replace in method code?

Add these menu picks to the Browser.

I13. Why do windows still have the "sun" icon, not the balloon?

While the dialog windows have a balloon icons?

I14. How can a PST application be assigned to a category on the PalmOS?

I15. Renaming resources?

Why does PST have names like ##ResourceDB1 and ##ResourceDB2? Why not GuiResources and WmResource? and also PilrcResource if it's possible to generate .rcp files from PST.

I16. Save chosen font?

Chosen font is not saved and restored.

I17. Cancelling loading project?

Changing paths manually in a PST file and reloading the project can give an understandable error "The system cannot find the path specified" but pressing "No" should really stop the loading process and maybe revert to the previous fully-loaded image. This should also be the case upon errors when loading single packages from the Package Browser.

I18. Path definition of packages?

Changing paths manually in the PST file and reloading the project can give an understandable error: "The system cannot find the path specified". I edited it to specify the right local paths given the working directory. I had 3 local packages (Black jack.pst). I updated all 3 but got errors on loading the second one:

PST adds the local path. Suggestion: define the system and working directories.

I.e., it could look like:

```
PstSystem systemDirectory: 'palmos\!'
PstSystem workingDirectory: 'demos\black jack!'
PstSystem addSystemPackage: 'core.st' dontSave: true!
PstSystem addSystemPackage: 'forms.st' dontSave: true!
PstSystem addSystemPackage: 'database.st' dontSave: true!
PstSystem addWorkingPackage: 'Emulated UI.st' dontSave: true!
PstSystem addWorkingPackage: ' CardFramework.st' dontSave: true!
```

I19. Palm OS Constants?

PST could be able to read Palm OS header files from SDK and automatically generate constants from C constants and enums. If this is too complex there could be constant packages created.

I20. Determining minimum and maximum required OS versions?

A check-program could be written to verify the minimum and maximum OS versions a certain project requires. Going through all methods of "save in project" packages, looking for SYSTRAP they call and listing what version the traps are implemented in would tell what Palm OS versions this application can run on. These stamps could even be included upon saving a package with a special menu option.

4.2 New Packages and VM features

I21. Ben's C calls and VM optimisation

Ben Hultink has released a [distribution-2001-07-25.zip](#) (in the discussion group file area `pocketstug>>files`) that includes changes to the virtual machine to support C calls and vm as GLIB. This could be part of a later PST release.

I22. Ben's table package

Ben Hultink has released a [distribution-2001-07-25.zip](#) (in the discussion group file area `pocketstug>>files`). This files contains a `table.st`. This could be part of a later PST release and there could be a demo.

I23. Ben's augment package

It can be found in `pocketstug>>files`.

I24. BlackJack's EmulatedUi.st

It can be found in `pocketstug>>files`. Should it be part of standard packages?

I25. SerialTest.st

It can be found in `pocketstug>>files`. Should it be part of standard "communication" package?

I26. SerialPort.zip

It can be found in `pocketstug>>files`. Should it be part of standard "communication" demo?

I27. Palm OS Version package

It can be found in `pocketstug>>files>>prj`. The file `osVersion-v1-and-v2.zip` contains 2 ways of retrieving the OS version your device runs on. It also shows how to test the version at launch time. The 3 methods of `OsVersion-v2.st` are worth considering for `core.st`.

I28. Returning for Application package

It can be found in `pocketstug>>files>>prj`. `ReturningFromApplication.st` implements return values that your application can receive upon closing popup as well as other non-popup applications. This value can then be used by the calling application to take the appropriate action.

The code uses predefined constants but it could be more general and could return instead the button or menu option value that was pressed.

I29. Daniel Enting's SUnit

It can be found in `pocketstug>>files`. However, as long as PST has no exceptions, it is not possible to do ordinary unit testing. After a failure it is not possible to continue testing within the running application.

I30. Add AddOnsVW package

It can be found in a message to the `pocketstug` discussion group. Search for that package name.

4.3 New Methods**I31. Useful method: `ReadStream>>upToSeparator`**

Jean-Jacques Moreau:

```
ReadStream >>upToSeparator
  "Answer a subcollection from position to the occurrence
  (if any, exclusive) of a separator. The stream is left positioned
  after the separator. If no separator is found answer everything."
  | index start end |
  start := index := position + 1.
  [index <= limit] whileTrue: [
    (collection at: index) isWhitespace ifTrue: [
      position := index.
      ^collection copyFrom: start to: index - 1].
    index := index + 1].
  position := limit.
  ^collection copyFrom: start to: limit
```

I32. Useful method: ReadStream>>space and tab

Jean-Jacques Moreau:

Here are two utility methods for class Transcript.

```
Transcript>>space
  ^self current nextPut: $\space.
Transcript>>tab
  ^self current nextPut: $\tab.
```

I33. Useful method: Integer>>asBigEndian / asLittleEndian

When a parameter has to be a value within a C enum, a bitshift 8 in PST is needed. To make code more understandable than just adding "bitshift: 0", use:

```
Integer>>asBigEndian
  "Convert myself into a 8 bit-left-shifted integer.
  To be used as input parameter for SYSTRAP when referring to
  C enum values and some C constants"
  ^self bitShift: 8

Integer>>asLittleEndian
  "Convert myself into a 8 bit-right-shifted integer.
  To be used on SYSTRAP return values when retrieving
  C enum values and some C constants"
  ^self bitShift: -8
```

The method name was suggested by Sunir Shah.

[Note from N.C.: Isn't it the case that big-endian / little-endian should swap bytes, not just shift them?]

Note: Behavior to be tested — is the bitshift or byte reversal needed at all?

I34. Storing in and reading from the Clipboard

Eric:

Here is the C function used to do it, it's a systrap:

```
void ClipboardAddItem (const ClipboardFormatType format,
                      const VoidPtr ptr, Word length)
enum clipboardFormats {
  clipboardText,
  clipboardInk,
  clipboardBitmap };

typedef enum clipboardFormats ClipboardFormatType;
```

Daniel Enting:

I worked one day to make the clipboard function and the result is well worth it. I don't think there are simpler ways to implement this. Why not add it to the Forms package?

```
PalmOS class>>getClipboardString
| handle length string |
handle := SYSTRAP ClipboardGetItem: 0 with: PadBuffer.
length := PadBuffer wordAt: 0.
string := String new: length.
SYSTRAP
  MemMove: string
  from: handle lock
  bytes: length.
handle unlock.
^string

PalmOS class>>putClipboardString: aStringOrByteArray
SYSTRAP
  ClipboardAddItem: 0
  with: aStringOrByteArray
  with: aStringOrByteArray size
```

Thierry:

ClipboardFormatType enum values could be defined as constants in category: "PalmOS: Clipboard Format Type":

Constant Name	Enum Position
##clipboardText	0
##clipboardInk	1
##clipboardBitmap	2

In Daniel's code 0s can now be replaced by "##clipboardText asBigEndian".

5. Contributors

Thanks to the following persons for their supportive replies and contributions:

Joey Gibson
 Eric Arseneau
 Ben Hultink
 Andrew Brault
 Jason Dufair
 Nathan Eaton
 Daniel Enting
 Peter Epstein
 Jean-Jacques Moreau
 Christian Schmidt-Setzwein
 Hans Scholz
 Sunir Shah
 VPutz
 N. Chamfort

Send any comments, additions, corrections, questions, or answers to:
thierry_reignier@hotmail.com

6. Document History

Version	Editor	Comments
1.1	N. Chamfort	December 19, 2001 — Converted to HTML; set in new page layout; extensive editing to make the text more readable by newbies.
1.0	Thierry Reignier	September 10, 2001 — Created document; messages 1250 -- 1612 October 6, 2001 — Published document at pocketstug

Last updated: Jan 20, 2002

Palm Powered is a trademark of Palm Inc. Pocket Smalltalk is trademark of Pocket Smalltalk Group. Copyright (c) 1998 - 2001 Pocket Smalltalk Group.