

Using the Palm OS[®] Emulator

This chapter describes how to use the Palm OS[®] Emulator program, a hardware emulator for the Palm Computing[®] platform. You can use the Palm OS Emulator to test and debug programs that you have developed for this platform.

NOTE: The Palm OS Emulator has previously been referred to as POSE or Poser. The name Palm OS Emulator is used throughout this book and in new versions of other Palm documentation. In this chapter, Emulator is sometimes used as an abbreviated form of Palm OS Emulator.

About the Palm OS Emulator

The Palm OS Emulator is a hardware emulator program for the Palm Computing platform, which means that it emulates the Palm hardware in software, providing you with the ability to test and debug Palm OS software on a Macintosh, Unix, or Windows-based desktop computer.

When you run a Palm OS application with the Palm OS Emulator on your desktop computer, the Palm OS Emulator fetches instructions, updates the handheld screen display, works with special registers, and handles interrupts in exactly the same manner as does the processor inside of Palm Computing platform handhelds. The difference is that the Palm OS Emulator executes these instructions in software on your desktop computer.

The Palm OS Emulator displays an on-screen image that looks exactly like a Palm[™] connected organizer, as shown in [Figure 1.1](#). You can select which type of Palm handheld device you want to

Using the Palm OS® Emulator

About the Palm OS Emulator

emulate, and you can also specify that you want the Palm OS Emulator to display the screen in double size, which continues to provide an accurate representation and makes the Palm screen easier to view

Figure 1.1 The Palm OS Emulator display



You can use the mouse on your desktop computer just as you use the stylus on a Palm connected organizer. You can even use the Graffiti® power writing software with Palm OS Emulator and your mouse. And Palm OS Emulator includes additional keyboard shortcuts that you can use on your desktop computer. These shortcuts are described in [Using the Hardware Buttons](#).

You can use the Palm OS Emulator to perform some debugging of your applications, and you can use the Emulator with Palm Debugger to perform extensive debugging of your applications. When you connect the Emulator with Palm Debugger, you can debug in exactly the same manner as debugging with your application running on an actual hardware handheld device.

Standard Device Features

Palm OS Emulator accurately emulates Palm Computing platform hardware devices, and includes the following features:

- an exact replica of the Palm device display, including the silkscreen and Graffiti areas
- emulation of the Palm stylus with the desktop computer pointing device
- emulation of the Palm device hardware buttons, including:
 - power on/off button
 - application buttons
 - up and down buttons
 - reset button
 - HotSync® button
- ability to zoom the display for enhanced readability and presentation
- screen backlighting
- communications port emulation for modem communications and synchronizing

Extended Emulation Features

Palm OS Emulator also provides the following capabilities on your desktop computer that extend the standard Palm device interface.

- ability to enter text with the desktop computer
- configurable memory card size, up to 8MB

Debugging Features

Palm OS Emulator provides a large number of debugging features that help you to detect coding problems and unsafe application operations. Palm OS Emulator includes the following debugging features and capabilities:

- ability to use an automated test facility called Gremlins, which repeatedly generates random events

- support for external debuggers, including Palm Debugger, the Metrowerks CodeWarrior debugger, and gdb.
- monitoring of application actions, including various memory access and memory block activities
- logging of application activities, including events handled, functions called, and CPU opcodes executed by the application
- profiling of application performance

For more information about testing and debugging applications with Palm OS Emulator, see [Testing and Debugging With Palm OS Emulator](#).

Using ROM Images

To run Palm OS Emulator, you need to transfer a ROM image to it. The ROM image contains all of the code used for a specific version of the Palm OS. You can obtain ROM images for different Palm OS versions from the Palm developer zone web site, or you can tell Palm OS Emulator to download the ROM from a handheld that has been placed in the device cradle and connected to the desktop computer. For more information about transferring a ROM image to Palm OS Emulator, see [Loading ROM Images](#).

When you download a ROM image from the Palm web site, you can obtain a debug ROM image, which contains information that Palm OS Emulator uses to help you debug Palm OS applications. For more information about the debugging capabilities in Palm OS Emulator, see [Testing and Debugging With Palm OS Emulator](#).

Downloading and Running Palm OS Emulator

You run Palm OS Emulator just like you would any other program. When Palm OS Emulator starts up, it displays an image of a handheld device, as shown in [Figure 1.1](#).

NOTE: The first time that you start the Emulator, it does not display an image of a handheld device; instead, it asks you to create a new session. After you have defined a session configuration, the Emulator creates a new session based on those settings when it launches.

You can then use the keyboard and mouse to interact with the emulated device, as described in [The Palm OS Emulator User Interface](#), and use the menus to interact with Palm OS Emulator.

Palm OS Emulator Runtime Requirements

The Palm OS Emulator requires one of the following runtime environments:

- Windows 98
- Windows 95
- Windows NT
- MacOS 7.5 or later
- Unix: some versions, including Linux

The Emulator is a multi-threaded 32-bit program. It does not run on Windows 3.1, even with Win32s installed.

Downloading Palm OS Emulator

The most recent released version of Palm OS Emulator for both the Macintosh and Windows is always posted on the Internet in the Palm developer zone:

<http://www.palm.com/devzone>

Follow the links from the developer zone main page to the emulator page to retrieve the released version of the Emulator. If you want to test-drive the version of Palm OS Emulator that is currently under development, you can follow links from the developer zone page to the Emulator seed page.

The Palm OS Emulator package that you download includes the files shown in [Table 1.1](#).

Using the Palm OS® Emulator

Downloading and Running Palm OS Emulator

Table 1.1 Files included in the Palm OS Emulator packa g

File name	Description
Binder.exe (Windows NT)	A program that binds the Palm OS Emulator with a ROM image for kiosk and demonstration purposes.
Emulator.exe (Windows) Palm OS Emulator (Macintosh) pose (Unix)	Main Palm OS Emulator executable
Emulator_Profile.exe (Windows) Palm OS Emulator - Profile (Macintosh)	Palm OS Emulator with added profiling facilities
Docs (directory)	Palm OS Emulator documents, including: <ul style="list-style-type: none">• _ReadMe.txt, which describes the files in the Docs directory• _News.txt, which describes changes in the most recent version• _OldNews.txt, which describes previous version changes
ROM Transfer.prc	Palm OS program to send the Palm.ROM file to your desktop.
HostControl.h	C/C++ header file declaring functions that can be used to control the Palm OS Emulator.

Versions of Palm OS Emulator

Each released version of Palm OS Emulator has a version number that uses the following scheme:

`<majorVers>.<minorVers>.<bugFix>[dab]<preRel>`

Each field has the following semantics:

majorVers The major version number

minorVers The minor version number.

bugFix	The optional bug repair revision number
dab	The prelease stage of the product, as follows: <ul style="list-style-type: none">d Indicates that the version is currently under development, and features are still being added.a Indicates alpha status, which means that the feature set is complete and some quality assurance testing has been performed.b Indicates beta status, which means that bugs uncovered in the alpha version have been addressed, and more extensive testing has been performed.
preRel	The developmental, pre-release version number.

Some examples of version numbers are shown in [Table 1.2](#)

Table 1.2 Version number examples

Version	Description
2.0	Official release version 2.0
2.1d19	The 19th developmental release of version 2.1.
2.1a2	The 2nd alpha release of version 2.1.

Profile Versions

Some releases of Palm OS Emulator include a profile version, with the word profile appended to the program name. Each profile version adds the ability to perform selective profiling of your program's execution, and to save the results to a file.

The code required to add profiling capability slows down your application, even when you are not using profiling. That means that you are better off using the non-profiling version of Palm OS Emulator if you don't expect to use the profiling capabilities.

For more information about profiling with Palm OS Emulator, see [ProfilingYour Code](#).

Command Line Options

If you are running Palm OS Emulator on a Windows-based desktop computer or on a Unix system, you can supply the session parameters as command-line parameters. For example:

```
Emulator -psf C:\Data\Session1.psf
```

[Table 1.3](#) shows the options that you can specify on the Windows command line. You can also change most of these options by starting a new session with the **New** menu command, as described in [Configuring a New Session](#).

NOTE: The command line options are not available on Macintosh computers.

Note that the command line option specifications are not case sensitive.

Table 1.3 Palm OS Emulator command line option

Option syntax	Parameter values	Description
-horde <num>	A Gremlin number	The number of the Gremlin to run after the session is created or loaded. Note that this is equivalent to supplying the same Gremlin number for the <code>horde_first</code> and <code>horde_last</code> options.
-horde_first <num>	A Gremlin number	The first Gremlin to run in a horde.
-horde_last <num>	A Gremlin number	The last Gremlin to run in a horde.
-horde_apps <app name list>	A comma-separated list of applications	The list of applications to which the Gremlin horde is allowed to switch. The default is no restrictions.

Table 1.3 Palm OS Emulator command line options

Option syntax	Parameter values	Description
<code>-horde_save_dir</code> <code><path></code>	A path name	The name of the directory in which to save session and log files. The default log location is the directory in which the Palm OS Emulator application is stored.
<code>-horde_save_freq</code> <code><num></code>	An event count	The Gremlin snapshot frequency. The default value is to not save snapshots.
<code>-horde_depth_max</code> <code><num></code>	An event count	The maximum number of Gremlin events to generate for each Gremlin. The default value is no upper limit.
<code>-horde_depth_switch</code> <code><num></code>	An event count	The number of Gremlin events to generate before switching to another Gremlin in the horde. The default is to use the same value as specified for the <code>horde_depth_max</code> option.
<code>-psf <fileName></code>	Any valid <code>.psf</code> file name	The emulator session file to load upon start-up. You can also load a session file with the Open menu command.
<code>-rom <fileName></code>	Any valid ROM file name	The name of the ROM file to use.

Table 1.3 Palm OS Emulator command line options

Option syntax	Parameter values	Description
-ram <size>	One of the following kilobyte size values:	The amount of RAM to emulate during the session.
or	128	
	256	
-ramsize <size>	512	
	1024	
	2048	
	4096	
	8192	
-device <type>	One of the following device type values:	The device type to emulate during the session.
	Pilot	
	PalmPilot	Note that Pilot1000 and Pilot5000 are synonyms for Pilot.
	PalmIII	
	PalmIIIX	
	PalmV	
	PalmVx	Also note that PalmPilotPersonal and PalmPilotProfessional are synonyms for PalmPilot.
	PalmVII	
	PalmVIIIEZ	
	ColorDevice	
-load_apps <file name list>	A list of valid file names, separated by commas	A list of .prc or other files to load into the session after starting up.
-log_save_dir <path>	A path name	The name of the directory in which to save the standard log file. The default log location is the directory in which the Palm OS Emulator application is stored.

Table 1.3 Palm OS Emulator command line options

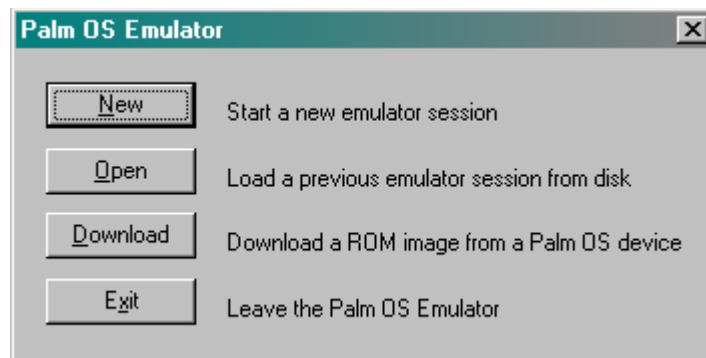
Option syntax	Parameter values	Description
<code>-quit_on_exit</code>	None	If the <code>-run_app</code> option was specified, this option indicates that Palm OS Emulator should quit after that application terminates.
<code>-run_app <app name></code>	Application name	The name of an application to run in the session after starting up. You must specify the name of the application, not the name of the application's file.
<code>-silkscreen <type></code> or <code>-skin <type></code>	One of the following silkscreen types: english japanese	The silkscreen type to emulate during the session.

How Palm OS Emulator Starts Execution

When Palm OS Emulator starts execution, it determines its configuration by sequencing through the following rules:

1. If the *Caps Lock* key is on, the Startup dialog box is always displayed. The Startup dialog box is shown in [Figure 1.2](#).

Figure 1.2 The Palm OS Emulator startup dialog box



Using the Palm OS® Emulator

Downloading and Running Palm OS Emulator

NOTE: The dialog box shown in [Figure 1.2](#) is displayed when you are running Palm OS Emulator on a Windows-based computer.

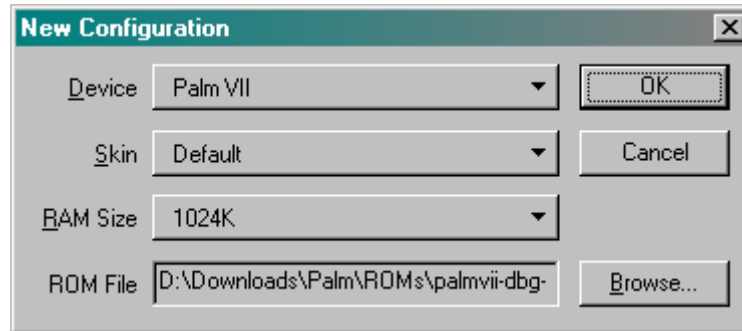
If you are using a Macintosh computer, the New Session dialog box, shown in [Figure 1.6](#) is displayed instead.

If you are using a Unix system, Palm OS Emulator does not provide an automatic startup sequence; instead, it presents you with a window that displays a device graphic, and you must right-click in that window to display the new session menu.

2. If the Caps Lock key is not on, Palm OS Emulator scans the command line for options. If an error is encountered on the command line, Palm OS Emulator displays an error message and then presents the Startup dialog box.
3. If a session (`.psf`) file was specified on the command line, Palm OS Emulator attempts to load the file. If the file cannot be loaded, Palm OS Emulator displays an error message and then presents the Startup dialog box.
4. If any other options are specified on the command line, Palm OS Emulator attempts to start a new session with those values. If any of the four values is missing, Palm OS Emulator displays the session configuration dialog box, as shown in [Figure 1.3](#).

If any of the command line options are not valid, or if the user cancels the dialog box, Palm OS Emulator displays an error message and then presents the Startup dialog box.

Figure 1.3 The session configuration dialog box



5. If no command line options are specified, Palm OS Emulator attempts to reopen the session file from the most recent session, if one was saved. If the file cannot be opened, Palm OS Emulator displays an error message, and then presents the Startup dialog box.
6. Palm OS Emulator attempts to create a new session based on the setting most recently specified by the user. If an error occurs, Palm OS Emulator displays an error message, and then presents the Startup dialog box.

Probably the most common scenario is when you start Palm OS Emulator without any command line parameters, and it restarts with saved information from the previous session.

NOTE: When it starts up, Palm OS Emulator looks for the most recently saved .psf file. On Windows and Unix, the Emulator uses the full path name of that file; on Macintosh systems, the Emulator uses aliases to locate the file. If it cannot find that file, Palm OS Emulator looks for the file name in the directory in which the Palm OS Emulator executable is located.

The Palm OS Emulator User Interface

This section provides a description of the user interface for Palm OS Emulator, including descriptions of the menus and keyboard usage.

The Palm OS Emulator Display

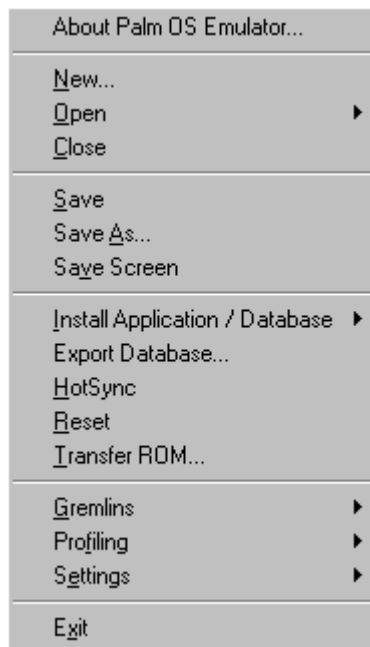
The Palm OS Emulator display looks very much like a real Palm Computing handheld device. You can use your mouse to perform actions that you perform with the stylus on handheld devices, and you can use the menus to access the Palm OS Emulator functionality.

Using the Menus

You can also access features that are specific to Palm OS Emulator by choosing menu commands:

- If you are using Windows, you right-click on the Palm OS Emulator screen display to access the menu items, or press the F10 key. The Palm OS Emulator menu displays, as shown in [Figure 1.4](#).

Figure 1.4 The Windows version of the Palm OS Emulator menu



- If you are using a Macintosh, select menu commands from the menu bar. The Macintosh menu presents the same

commands in four different menus, as described in [Table 1.4](#). The Macintosh version is only slightly different:

- The Macintosh version of Palm OS Emulator uses the **Preferences** command instead of the **Properties** command to access the option-setting dialog box.
- The Macintosh version of the Emulator features the **Undo**, **Cut**, **Copy**, **Paste**, and **Clear** commands, which are not available in the Windows version.
- The Macintosh version of the Emulator uses the **Quit** command instead of the **Exit** command.
- The Macintosh version does not feature the **Breakpoints** command.

Table 1.4 Palm OS Emulator Macintosh menus

Menu	Commands
File	New Open Close <hr/> Save Save As Save Screen <hr/> Install Application/Database HotSync Reset Transfer ROM <hr/> Quit
Edit	Undo <hr/> Cut Copy Paste Clear <hr/> Preferences Logging Options Debug Options Skins
Gremlins	New <hr/> Step Resume <hr/> Stop
Profile	Start Stop Dump

- If you are using Unix, Palm OS Emulator provides the same commands as are included with the Windows version, except that the **Breakpoints** command is not available. The Unix version of the menu pops up like the Windows version, and uses a different hierarchy, but presents the same commands.

[Table 1.5](#) provides a brief description of the Palm OS Emulator menu commands, listed in alphabetical order.

Table 1.5 The Palm OS Emulator menu commands

Command	Description
Close	Closes and optionally saves the current emulator session.
Exit	Exits Palm OS Emulator. If you have unsaved changes in your session file, Palm OS Emulator optionally prompts you to save the file before exiting.
Gremlin:New	Create a new Gremlin and start running it.
Gremlin:Step	Step a Gremlin, after stopping.
Gremlin:Resume	Resume running of the Gremlin. NOTE: this command is only shown in Windows versions, and is not yet implemented.
Gremlin:Stop	Stop running the Gremlin.
Gremlin: Resume from control file	Resumes running of Gremlins from data that was previously saved in a file. For more information, see Using Gremlins .
HotSync	Allows you to synchronize the emulator session environment with the desktop computer. See Using the HotSync Application With the Palm OS Emulator for more information about the cabling requirements and other considerations for this command.

Table 1.5 The Palm OS Emulator menu commands

Command	Description
Install App/DB	Allows you to install an application into the emulator session, in the same way that a user would install it on the handheld with the Palm Install tool. For more information, see Installing Applications .
Export Database...	Exports a database to your desktop computer as a .pdb or .pqa file, or exports an application to your desktop computer as a .prc file.
New	Displays the new configuration dialog box for initiating a new session.
Open	Displays the open file dialog box for opening a saved emulator session file.
Profiling:Start	Start profiling your application.
Profiling:Stop	Stop profiling your application.
Profiling:Dump	Save the profiling information to a file. For more information, see Profiling Your Code .
Reset	Resets the current emulation session, as if the reset button on the back of the handheld was pressed.
Save	Saves the current emulator session to an emulator .psf file.
Save As	Saves the current emulator session to an emulator .psf file.

Table 1.5 The Palm OS Emulator menu commands

Command	Description
Save Screen	Saves the current screen image as a bitmap file. <hr/> TIP: The Save Screen command is a very convenient means of capturing screen images for documentation of Palm OS® applications. <hr/>
Settings: Properties	Presents the properties dialog box, as described in Palm OS Emulator Properties .
Settings: Logging	Presents the logging options dialog box, as described in Logging Options .
Settings: Debug	Presents the debug options dialog box, as described in Debug Options .
Settings: Skins	Presents the skins dialog box, as described in Changing the Emulator's Appearance .
Settings: Breakpoints	Presents the breakpoints dialog box, as described in Setting Breakpoints .
Transfer ROM	Allows you to download a ROM image and save it to disk. You can then initiate a new session based on that ROM image. For more information, see Transferring a ROM Image From a Handheld .

Using the Hardware Buttons

Palm OS Emulator emulates each of the hardware buttons on Palm Computing devices. You can click on a button to activate it, and you can press and hold down a button just as you would on a handheld. Palm OS Emulator also allows you to activate the hardware buttons with keyboard equivalents, as shown in [Table 1.6](#).

Table 1.6 Keyboard equivalents for the hardware buttons

Button	Keyboard equivalent
On/off	Esc
Palm Date Book	F1
Palm Address Book	F2
Palm To Do List	F3
Palm Memo Pad	F4
Up	Page Up
Down	Page Down

Entering Data

Palm OS Emulator allows you to use your desktop computer pointing device to tap and to draw Graffiti characters, just as you do with the stylus on the handheld.

Palm OS Emulator also allows you to enter text from the desktop computer keyboard. For example, you can type the text for a note by tapping in the note text entry area and then using the keyboard.

Control Keys

Palm OS Emulator also supports a set of control keys that you can use for input. These keys, which are shown in [Table 1.7](#), are the same control keys that you can use with the Palm OS Simulator program.

Table 1.7 Palm OS Emulator Control Keys

Control key combination	Description
Control - A	Displays the menu
Control - B	Low battery warning
Control - C	Command character
Control - D	Confirmation character

Table 1.7 Palm OS Emulator Control Keys

Control key combination	Description
Control - E	Displays the application launcher
Control - F	Displays the onscreen keyboard
Control - M	Enters a linefeed character
Control - N	Jumps to the next field
Control - S	Automatic off character
Control - T	Sets or unsets hard contrasts
Control - U	Turns backlighting on or off

Loading ROM Images

Since the Palm OS Emulator emulates the Palm Computing Platform hardware, all components of the hardware must be present. This includes a ROM image file, which is not shipped with the Emulator. There are two ways to obtain a ROM image:

- download a ROM image from the Palm web site
- transfer a ROM image from a handheld

Downloading a ROM Image Obtained From Palm

To download a debug ROM image from Palm, go to the Palm developer zone web site, which is a rich source of resources for Palm OS developers. The developer zone URL is:

<http://www.palm.com/devzone>

The ROM image files are found in the Palm Provider Pavilion.

Transferring a ROM Image From a Handheld

To transfer a ROM image from a handheld, you need to follow these steps:

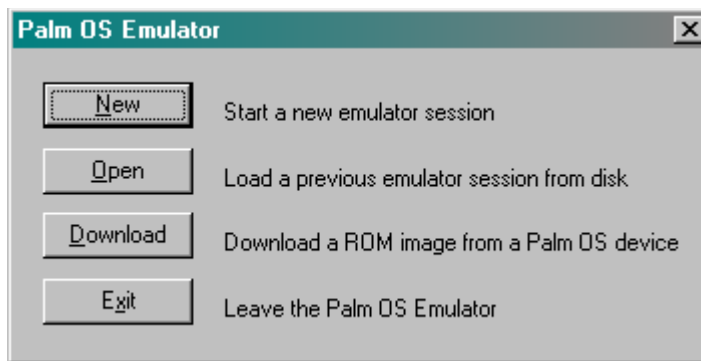
1. Install the Palm OS application named `ROM Transfer.prc` on your handheld device. You can use the Install program in the Palm Desktop organizer software and then synchronize with the handheld to install this program.
2. Place the handheld in the HotSync cradle that is connected to your desktop computer.
3. Follow the steps in the appropriate section below.

Transferring a ROM File in Windows

This section describes how to transfer a ROM image from a handheld on a Windows-based desktop computer. Before proceeding, you must have the `ROM Transfer.prc` program installed on the handheld, as described in the previous section.

If you are running the program for the first time, Palm OS Emulator presents the Startup dialog box shown in [Figure 1.5](#). Click on **Download** to begin the transfer of a ROM image from a handheld.

Figure 1.5 The Palm OS Emulator startup dialog box



If you are not running Palm OS Emulator for the first time, it usually restarts the session that you most recently ran, as described in [How Palm OS Emulator Starts Execution](#). To transfer a new ROM image for Palm OS Emulator to use, you can right-click on the Palm OS

Emulator display (the Palm device image) and select the **Transfer ROM** menu choice.

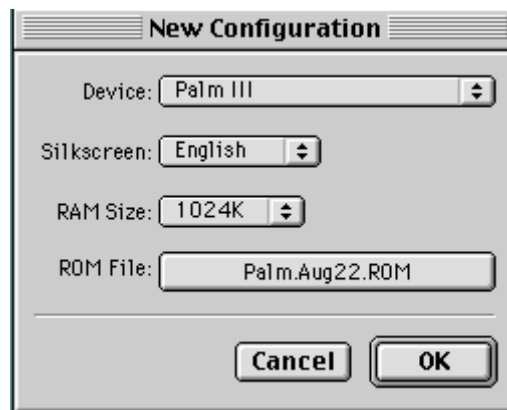
Once you have chosen to transfer a ROM image, Palm OS Emulator presents you with a sequence of dialog boxes that guide you through the process.

Transferring a ROM File On a Macintosh

This section describes how to transfer a ROM image from a handheld on a Macintosh desktop computer. Before proceeding, you must have the `ROM Transfer.prc` program installed on the handheld, as described in the previous section.

If you are running the program for the first time, Palm OS Emulator presents the dialog box shown in [Figure 1.6](#).

Figure 1.6 Running Palm OS Emulator for the first time on a Macintosh system



You can dismiss this dialog box and choose the **Transfer ROM** command from the File menu.

If you are not running Palm OS Emulator for the first time, it usually restarts the session that you most recently ran. To transfer a new ROM image for Palm OS Emulator to use, select the **Transfer ROM** command from the File menu.

Once you have chosen to transfer a ROM image, Palm OS Emulator presents you with a sequence of dialog boxes that guide you through the process.

Using a ROM Image in Palm OS Emulator

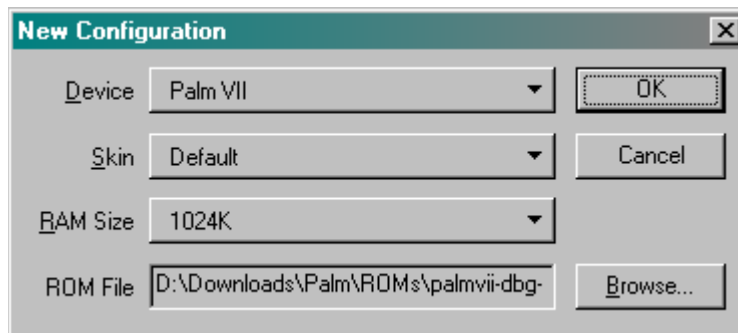
Once you have transferred a ROM image to disk, you need to create a new session that is based on the image. To initiate the new session, you select the **New** command. [Table 1.8](#) shows the first step in creating a new session for each transfer method.

Table 1.8 Initiating a new session after transferring a ROM image

Method used to initiate ROM transfer	New session method
Clicked Download initial dialog in Windows	Click New in the dialog box.
Selected Transfer ROM command in Windows	Select either the New command or the Close command from the File menu.
Selected Transfer ROM menu command on a Macintosh	Select the New command from the File menu.

After you initiate the session, Palm OS Emulator presents the new configuration dialog box, which is described in [Configuring a New Session](#). The Windows version of this dialog box is shown in [Figure 1.7](#).

Figure 1.7 The New Configuration dialog box



After you select your parameters and click **OK**, Palm OS Emulator begins an emulation session.

Drag and Drop a ROM Image

You can use drag and drop to start a new Emulator session in either of two ways:

- Drag and drop a ROM image file onto the Emulator screen to start a new session.
- Drag and drop a ROM image file onto the Emulator executable or shortcut (alias) to start the Palm OS Emulator program.

You can also drag and drop other file types, as described in [Dragging and Dropping Files](#).

Using the Binder to Create an Executable

If you are running the Palm OS Emulator on Windows NT, you can use the Binder program to create an executable that binds the Emulator program with a ROM image and optionally a RAM image. The bound program can then be used for demonstrations, training, and kiosk systems.

Testing and Debugging With Palm OS Emulator

This section provides an overview of testing and/or debugging an application with Palm OS Emulator.

Testing Software

Testing software is probably the most common use of Palm OS Emulator. This section provides a quick summary of the steps to load and test an application.

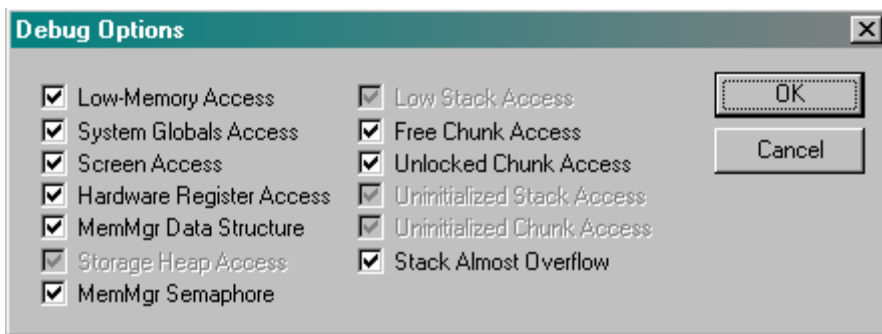
Debug Options

The Palm OS Emulator monitors the actions of your application while it is emulating the operation of the handheld device. When your application performs an action that does not strictly conform

to Palm Computing’s programming guidelines, the Emulator displays a dialog box that explains what is happening.

The debugging options dialog box, which is shown in [Figure 1.8](#), allows you to enable or disable the monitoring activities applied to your application. Use the **Debug Options** command to display this dialog box.

Figure 1.8 The Palm OS Emulator debugging options dialog box



[Table 1.9](#) describes each of the debugging options.

Table 1.9 Emulator debugging option

Option	Description
Low-Memory Access	Monitors low-memory access by applications. Low-memory access means an attempt to read from or write to a memory location in the range 0x0000 to 0x00FF.
System Globals Access	Monitors access to system global variables by applications. System global variable access is defined as reading from or writing to a memory location in the range from 0x0100 to the end of the trap dispatch table.

Table 1.9 Emulator debugging options (continued)

Option	Description
Screen Access	<p>Monitors LCD screen buffer access by applications.</p> <p>LCD screen buffer access is defined as reading from or writing to the memory range indicated by the LCD-related hardware registers.</p>
Hardware Register Access	<p>Monitors accesses to hardware registers by applications.</p> <p>Hardware register access is defined as reading from or writing to memory in the range from 0xFFFFF000 to 0xFFFFFFFF.</p>
MemMgr Data Structure	<p>Monitors access to Memory Manager data structures, which is restricted to only the Memory Manager.</p> <p>Memory Manager data structures are the heap headers, master pointer tables, memory chunk headers, and memory chunk trailers.</p>
Storage Heap Access	<p>Monitors naked access to the storage heap by applications. To access the storage heap, your application should use the <code>DmWrite</code> functions.</p>
MemMgr Semaphore	<p>Monitors how long the Memory Manager semaphore has been acquired for write access using the <code>MemSemaphoreReserve</code> and <code>MemSemaphoreRelease</code> functions.</p> <p>Your applications should not be calling these functions; however, if you must call them, you should not hold the semaphore for longer than 10 milliseconds.</p>
Low Stack Access	<p>Monitors access to the range of memory below the stack pointer.</p>

Table 1.9 Emulator debugging options (continued)

Option	Description
Free Chunk Access	Monitors access to free memory chunks. No process should ever access the contents of a chunk that has been deallocated by the <code>MemChunkFree</code> , <code>MemPtrFree</code> , or <code>MemHandleFree</code> functions.
Unlocked Chunk Access	Monitors access to unlocked, relocatable memory chunks, which is restricted to the Memory Manager.
Uninitialized Stack Access	Monitors read accesses to uninitialized portions of the stack. You can use this option to detect read accesses to uninitialized local variables.
Uninitialized Chunk Access	Monitors read access to uninitialized portions of memory chunks that have been allocated by the <code>MemChunkNew</code> , <code>MemPtrNew</code> , and <code>MemHandleNew</code> functions. You can use this option to detect read accesses to uninitialized portions of dynamically allocated memory chunks. Note that your application's global variables are stored in memory chunks allocated by these functions, so enabling this option also detects read accesses to uninitialized global variables.
Stack Almost Overflow	Ensures that the stack pointer has not dipped below the space allocated for it by the kernel. When this option is enabled, Palm OS Emulators warns you when the application stack is getting close to full. Note that you are always warned of a stack overflow, even if this option is disabled.

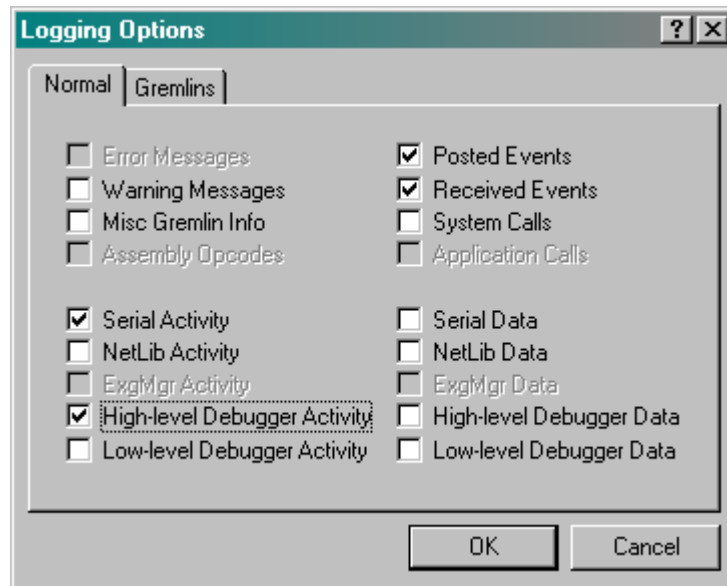
Logging Options

The Palm OS Emulator also logs various actions taken by your application to help you debug and performance tune your code. The

logged information is automatically written to a text file that is saved in the same directory as the Emulator executable.

You can control the logging activity with the logging options dialog box, which is shown in [Figure 1.9](#). Use the **Logging Options** command to display this dialog box.

Figure 1.9 Palm OS Emulator logging options



The logging options dialog box features separate tabs for logging during normal operations, and for logging while a Gremlin is running. Both tabs offer the same options, which are described in [Table 1.10](#)

Table 1.10 Emulator logging option

Option	Description
Error Messages	Not yet implemented.
Warning Messages	Logs any message that is displayed in a dialog box that can be dismissed by tapping the Continue button.
Misc Gremlin Info	Logs information about Gremlins that is mostly useful for debugging the Gremlins themselves.

Table 1.10 Emulator logging options (*continued*)

Option	Description
Assembly Opcodes	Logs assembly-level trace information, including registers, the program counter, opcodes, and related information. This option is not yet implemented.
Posted Events	Logs events that have entered into the system by way of calls to the <code>EvtAddEventToQueue</code> , <code>EvtAddUniqueEventToQueue</code> , <code>EvtEnqueuePenPoint</code> , and <code>EvtEnqueueKey</code> functions.
Received Events	Logs events returned by calls to the <code>EvtGetEvent</code> , <code>EvtGetPen</code> , and <code>EvtGetSysEvent</code> functions.
System Calls	Logs calls to Palm OS® functions.
Application Calls	Logs calls to functions in your application. This option is not yet implemented.
Serial Activity	Logs changes in serial port settings, and the opening and closing of the serial port.

Table 1.10 Emulator logging options (continued)

Option	Description
Serial Data	<p>Logs data sent and received over the serial port. Data is logged as it is being transferred over the host serial port</p> <p>Incoming data follows this path:</p> <ol style="list-style-type: none">1. Serial port2. Emulated hardware registers3. Palm OS4. Palm application <p>Palm OS Emulator logs the serial port data.</p> <p>Outgoing data follows this path:</p> <ol style="list-style-type: none">1. Palm application2. Palm OS3. Emulated hardware registers4. Serial port <p>Again, Palm OS Emulator logs the serial port data.</p>
NetLib Activity	Logs calls to <code>NetLib</code> functions, including parameter and return values.
NetLib Data	Logs data sent and received via <code>NetLib</code> calls.
ExgMgr Activity	Not yet implemented.
ExgMgr Data	Not yet implemented.
High-level Debugger Activity	Logs messages received back from an external debugger, and the messages sent back to the debugger.
High-level Debugger Data	Logs details of the messages sent to and received from an external debugger.

Table 1.10 Emulator logging options (*continued*)

Option	Description
Low-level Debugger Activity	Traces the low-level mechanisms that receive raw data from external debuggers and send data back to external debuggers.
Low-level Debugger Data	Logs the raw data being sent to and received from an external debugger.

Using Gremlins

You can use Gremlins to automate testing of an application. A **Gremlin** generates a series of user input events that test your application's capabilities. You can have a Gremlin to run a specified number of times, or to loop forever, which allows you to set up a Gremlin and allow it to run overnight to thoroughly test your application.

A **Gremlin horde** is a range of Gremlins that you want Palm OS Emulator to run. The Emulator generates a stream of events for each Gremlin and then moves onto the next Gremlin. The Emulator cycles through the Gremlins until the maximum number of events have been generated for the hor

The Palm OS Emulator generates a stream of events for each Gremlin in the horde until one of the following conditions occurs:

- An error such as a hardware exception or illegal memory access is generated.
- The maximum number of events for a single Gremlin have been generated.
- The maximum number of events for the horde have been generated.
- You stop the horde by choosing the Stop or Step command from the Emulator menu or from the Gremlin Status dialog box.

If a Gremlin generates an error, it is halted and the Palm OS Emulator does not include it when cycling through the horde again.

Gremlin Characteristics

Each Gremlin has the following characteristics:

- it generates a unique, random sequence of stylus and key input events to step through the user interface possibilities of an application
- it has a unique “seed” value between 0 and 999
- it generates the same sequence of random events whenever it is run
- it runs with a specific application or applications
- it displays a report immediately when an error occurs

Gremlin Horde Characteristics

Each Gremlin horde has the following characteristics:

- The number of the first Gremlin to run. This must be a value between 0 and 999.
- The number of the last Gremlin to run. This must be a value between 0 and 999.
- The switching depth of the Gremlin horde. This is the number of events to run for each Gremlin. After this many events have been generated for the Gremlin, it is suspended, and the next Gremlin in the horde starts running.
- The maximum number of events for each gremlin in the horde. The Emulator stops running the Gremlin after it posts this many events, or after it terminates with an error.
- With which applications the Gremlins are to run. You can select a single application, a group of applications, or all applications.
- Errors that occur are logged to the log file and the emulation continues with the next Gremlin in the horde.

When Palm OS Emulator runs a Gremlin horde, it actually maintains a separate stream for each Gremlin in the horde. When it starts a horde, the Emulator first saves the complete state of the emulation to a session (.psf) file. Then, the Emulator:

- Starts the first Gremlin. When the Gremlin has posted a number of events equal to the specified switching depth, the

Emulator saves its state to a new file and suspends the Gremlin.

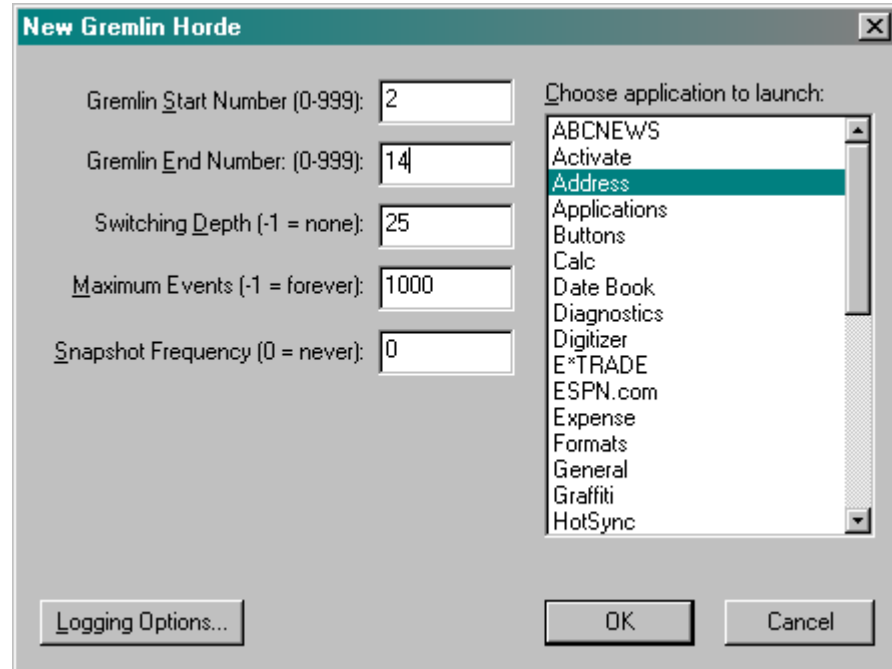
- Reloads the original session state.
- Starts the second Gremlin and runs it until it posts that number of events, at which time its state is saved to another file, and the Gremlin is suspended.
- Runs each Gremlin in the horde, until each has been suspended or terminated:
 - A Gremlin is terminated when an error occurs while the Gremlin is posting events.
 - A Gremlin is suspended when it has posted a number of events equal to the switching depth for the horde.
- Returns to the first suspended Gremlin in the horde, reloads its state from the saved file, and resumes its execution as if nothing else had happened in the meantime.
- Continues cycling through the Gremlins in the horde until each Gremlin has finished. A Gremlin finishes when either of these conditions occurs:
 - the Gremlin has terminated due to an error
 - the Gremlin has posted a total number of events equal to the maximum specified for the horde.

Running a Gremlin Horde

Select the **New Gremlin** command to start a Gremlin. The new Gremlin dialog box displays, as shown in [Figure 1.10](#). You use this dialog box to specify the characteristics of the Gremlin horde that you want to run.

NOTE: If you wish to run a single Gremlin, simply set the Gremlin Start Number and Gremlin End Number fields to the same value.

Figure 1.10 The Gremlin horde dialog box



When Palm OS Emulator runs the example shown in [Figure 1.10](#), the horde will operate as follows:

- The Emulator will only run the Address application when generating key and stylus events for this horde.
- The Emulator will use a seed value of 2 for the first Gremlin in the horde and a seed value of 14 for the last Gremlin in the horde. It also runs all intervening Gremlins: numbers 3 through 13.
- The Emulator will generate 25 events for each Gremlin before switching to the next Gremlin in the horde.
- The Emulator will cycle through the Gremlins in the horde until a total of 1000 events have been generated for each Gremlin. Thus, a total of 13,000 events will be generated.

This means that the Emulator will generate the following sequence of Gremlin events:

1. Gremlin #2 runs and receives twenty-five events, after which Gremlin 2 is suspended.

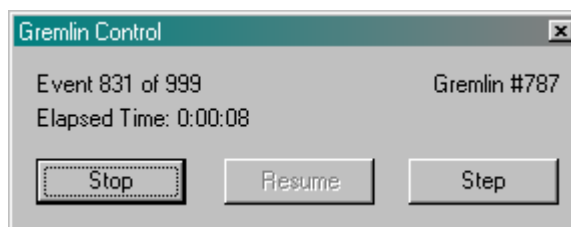
2. Gremlin #3 runs and receives twenty-five events, after which Gremlin #3 is suspended.
3. Similarly, each Gremlin (#4 through #14) runs and receives twenty-five events, after which it is suspended.
4. The Emulator loops back to Gremlin #2 and runs it, sending it twenty-five events before again suspending it.
5. Gremlin #3 runs again, receives twenty-five events, and suspends.
6. This looping through the Gremlins and sending each events until the switch depth (25) is reached continues until the maximum number of horde events (1000) have been generated.
7. All activity for the Gremlin horde completes.

NOTE: If an error occurs while a specific Gremlin is running, Palm OS Emulator halts that Gremlin rather than suspending it. This means that the Gremlin is not run when the Emulator next iterates through the horde.

Stepping and Stopping Gremlins

After the horde starts running, Palm OS Emulator displays the Gremlin control dialog box, which is shown in [Figure 1.11](#). You can use the commands in this dialog box to stop, resume, and single-step a Gremlin. You can also use the **Gremlins** menu command to perform these actions.

Figure 1.11 The Gremlin status dialog box



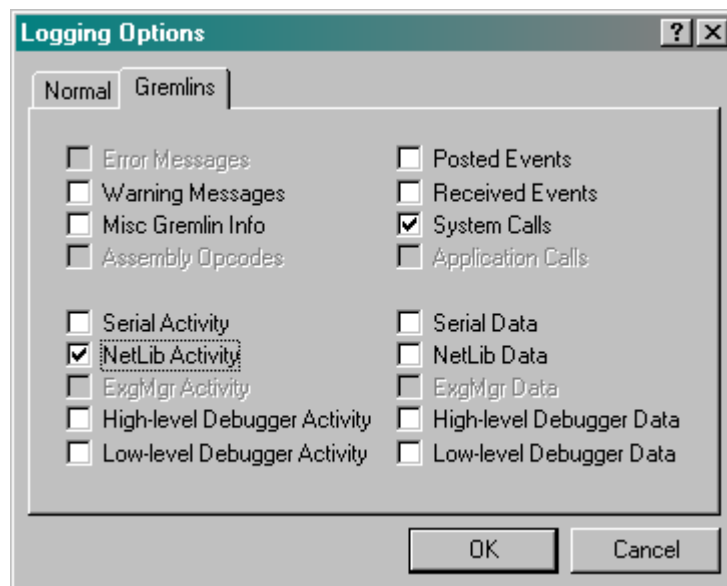
Gremlin Snapshots

When you start a new Gremlin horde, you can specify that you want the Palm OS Emulator to take a snapshot on a regular basis. You specify a frequency value, as shown in [Figure 1.10](#), and the Emulator saves a session file each time that many Gremlins have run. Each snapshot is a .psf file that captures the current state of the emulation. You can open the snapshot in the Emulator as a new session and begin debugging from that state.

Logging While Gremlins Are Running

Palm OS Emulator allows you to specify separate logging options to use while Gremlins are running. [Figure 1.12](#) shows the Gremlin logging options dialog box. Each of the options is described in [Logging Options](#).

Figure 1.12 Gremlin logging options



Setting Breakpoints

You can set breakpoints in your code with the Emulator. When the Palm OS Emulator encounters a breakpoint that you have set, it halts and takes one of the following actions:

Using the Palm OS® Emulator

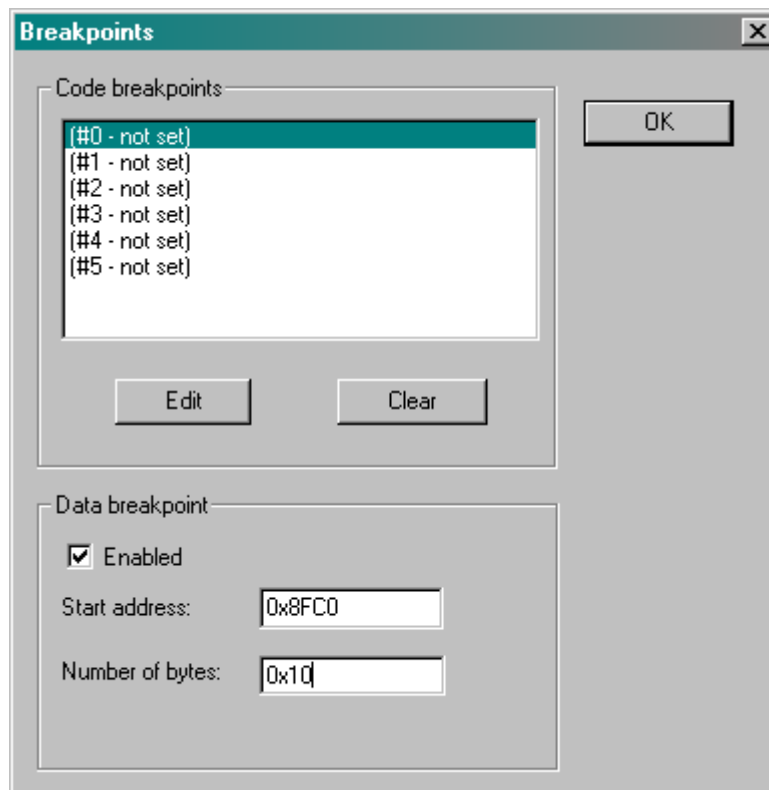
Testing and Debugging With Palm OS Emulator

- If you are running the Emulator connected to a debugger, the Emulator sends a message to the debugger, informing it that the breakpoint was hit. The debugger then handles that command as it sees fit.
- If the Emulator is not connected to a debugger, the Emulator displays an error message. This message will typically say something like “TRAP \$0 encountered.”

To set a breakpoint, select the **Breakpoints** command from the **Settings** menu. The Breakpoints dialog box is displayed, as shown in .

NOTE: You cannot use the Breakpoints feature on the Macintosh or Unix versions of the Palm OS Emulator.

Figure 1.13 Setting a breakpoint



Setting the Data Breakpoint

You can set exactly one data breakpoint. While your program is executing, the Emulator watches the specified address range; if it is written to, the Emulator generates a break. You can specify both the address and number of bytes fields in either hexadecimal (0x) or decimal.

Setting Conditional Breakpoints

You can set up to six independent conditional breakpoints. The Emulator generates a break for a conditional breakpoint when both of the following are true:

- the program counter reaches the specifies address
- the specified condition is true

To set one of these breakpoints, select the breakpoint number in the list at the top of the dialog box, and click on the **Edit** button. This displays the Code Breakpoint dialog box, which is shown in [Figure 1.14](#).

Figure 1.14 Setting a code breakpoint



To set the breakpoint, specify an address and the break condition. You can specify the address in hexadecimal (0x) or decimal.

The condition that you specify must have the following format:

`<register> <condition> <constant>`

register One of the registers: A0, A1, A2, A3, A4, A5, A6, A7, D0, D1, D2, D3, D4, D5, D6, or D7.

condition One of the following operators: ==, !=, <, >, <=, or >=.

constant A decimal or hexadecimal constant value.

WARNING! All comparisons are unsigned.

Source Level Debugging

Palm OS Emulator provides an interface that external debugger applications can use to debug an application. For example, Metrowerks has developed a plug-in module that you can use to debug an application that Palm OS Emulator is running, in exactly the same manner as you would debug an application running on the handheld. This plug-in module is shipped with the latest version of CodeWarrior for Palm OS.

Connecting the Emulator With Palm Debugger

You can use the Palm Debugger with the Palm OS Emulator to perform extensive debugging of your applications. To use Palm Debugger with the Emulator, follow these steps:

1. Start the Palm Debugger and Palm OS Emulator programs.
2. In the Palm Debugger Communications menu, select **Emulator**. This establishes the emulator program as the “device” with which Palm Debugger is communicating.
3. In the debugger window, type the `att` command.

You can now send commands from the Palm Debugger to the Palm OS Emulator .

Connecting the Emulator With External Debuggers

Palm OS Emulator can communicate with external debuggers using the methods shown in [Figure 1.11](#).

Table 1.11 Palm OS Emulator Connections

Connection type	Platforms
TCP	All
PPC Toolbox	Macintosh
Memory-mapped files	Windows

NOTE: Currently, PalmDebugger uses TCP only when running on Windows. The CodeWarrior plug-in uses TCP if you select the `Use sockets` checkbox in the debugger preference panel.

However, although you can configure the TCP port that Palm OS Emulator uses, you cannot configure which TCP port that either PalmDebugger or the CodeWarrior plug-in uses.

If you are communicating with a debugger using TCP, you can configure which socket port the debugger connects to by editing the value of the `DebuggerSocketPort` preference setting in your preferences file. You can disable the TCP connection by setting the value of the `DebuggerSocketPort` preference to 0.

NOTE: In some versions of Palm OS Emulator, you may notice that an unwanted PPP dial-up starts whenever you begin a new emulation session. You can disable this behavior by disabling the use of TCP for communications, which you do by setting the `DebuggerSocketPort` preference to 0.

Profiling Your Code

One of the features of the Palm OS Emulator that is most useful for developers is the ability to profile your application while it is running, and to save the resulting data to a file that you can examine.

When the Emulator profiles your application, it monitors and generates statistics about where your code is spending its time, which enables you to focus your optimization efforts in the most productive manner.

You can start a profiling session by choosing the **Profiling Start** command. While profiling is active, the Palm OS Emulator monitors which application and system functions are executed, and the amount of time executing each. The Emulator collects the timing information until you select the **Profiling Stop** command.

Using the Palm OS® Emulator

Palm OS Emulator Session Features

You can then save the profiling information to a file by selecting the **Profiling Dump** command. The information is saved to file in two different formats. Both of these files are stored in the directory in which the Emulator executable is located:

File name	Description
Profile Results.txt	A text version of the profiling results.
Profile Results.mwp	A Metrowerks Profiler version of the results, which can be used with the MW Profiler application bundled with CodeWarrior Pro.

IMPORTANT: The MW Profiler is only available on Macintosh computers.

You do not have to prepare your code in any special way for Palm OS Emulator to profile it, because the Emulator can determine when functions are entered and exited on its own. And the Emulator performs its profiling calculations between cycles, thus the timing information is quite accurate.

NOTE: It is a good idea to set your compiler's switch to embed debug symbols in your code so that you can easily interpret the profiling results.

Palm OS Emulator Session Features

Palm OS Emulator uses the concept of an emulation session, which is a testing or debugging session for a combination of the following items:

- the handheld device type to emulate
- the amount of RAM to emulate
- the ROM file to use for the emulation

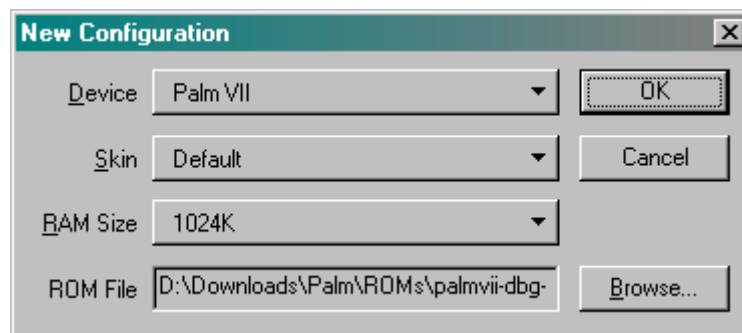
You can start new emulation sessions during a single run of Palm OS Emulator. You can also save the current state of a session and restore it in a later session. This session describes these features of Palm OS Emulator.

Configuring a New Session

You can start a new session in Palm OS Emulator by choosing the **New** command from the Palm OS Emulator menu. If you are already running an emulation session, Palm OS Emulator will optionally ask if you want to save the session in a Palm OS Emulator session (.psf) file before starting the new session. You set this option in your preferences.

[Figure 1.15](#) shows the New Configuration dialog box, which Palm OS Emulator displays when you choose the **New** command from the menu.

Figure 1.15 Configuring a new session



You need to make the following choices in this dialog box:

- Select the Palm handheld device that you want to emulate in the session. You can choose from among the following choices:
 - Pilot (1000/5000)
 - PalmPilot (Personal/Pro)
 - Palm III
 - Palm IIIx
 - Palm V

Using the Palm OS® Emulator

Palm OS Emulator Session Features

- Palm VII
- Palm VII EZ
- Color Device
- Select the silkscreen that you want displayed on the emulation screen. Alternative silkscreens, such as the Japanese silkscreen, are only available for certain device types. The Default choice is always available, even when alternatives are not available.
- Select the amount of memory that you want emulated. You can choose from the following RAM sizes:
 - 128K
 - 256K
 - 512K
 - 1024K
 - 2048K
 - 4096K
 - 8192K

Note that 1 MB is most often the right amount of RAM to emulate.

- Select the ROM file on your desktop computer that you want to use for the session. You can use the **Browse** button to navigate to the file. For more information about ROM files, see [Loading ROM Images](#).

After you click the **OK** button, Palm OS Emulator begins an emulation session.

Dragging and Dropping Files

You can drag and drop the following file type categories onto the Palm OS Emulator LCD screen:

- .prc, .pdb, and .pqa files
- .rom files
- .psf files

When dragging and dropping files, you must observe the following rules:

- You can drag and drop only one `.rom` file at a time.
- You can drag and drop only one `.psf` file at a time.
- You can drag and drop any number of `.prc`, `.prb`, and `.pqa` files.
- You cannot drag and drop files from more than one of the file type categories in the same operation.

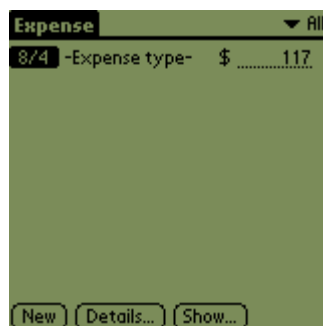
Saving and Restoring Session State

You can save the current state of a Palm OS Emulator session to a session file for subsequent restoration. Palm OS Emulator saves a session to a session file. The Emulator uses Save and Save As in the standard manner, with one addition: you can automate what happens when closing a session by changing the Save options.

Saving the Screen

You can save the current screen to a bitmap file by selecting the **Save Screen** menu command, which saves the contents of the emulated Palm handheld device screen.

Figure 1.16 A Palm OS Emulator screen shot

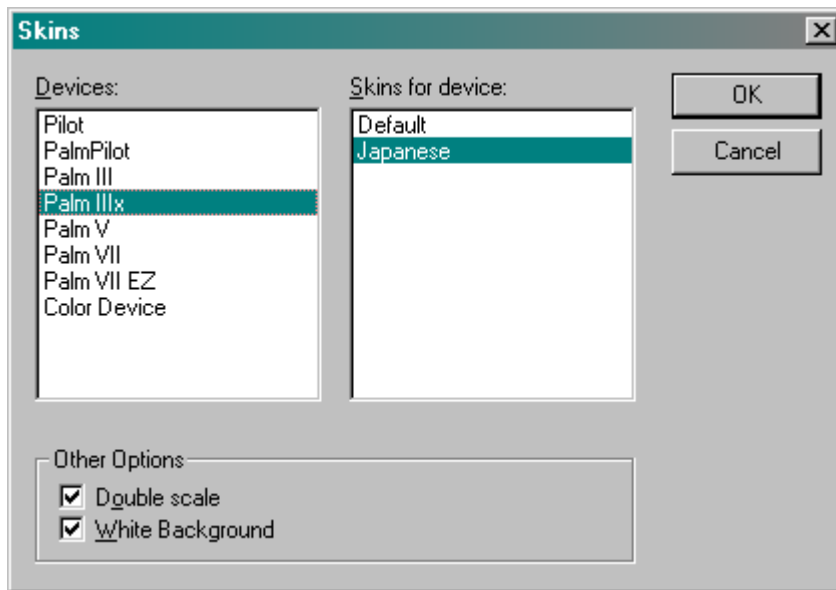


Palm OS Emulator saves screen images on Windows-based systems as `.bmp` bitmap images, and saves screen images on MacOS-based systems as SimpleTextimage files.

Changing the Emulator's Appearance

You can change the appearance of the Palm OS Emulator by choosing the **Skins** command from the Settings submenu. This displays the Skins dialog box, which is shown in [Figure 1.17](#).

Figure 1.17 Changing the Palm OS Emulator appearance



The Skins dialog box provides three appearance options that you can use:

- Select or deselect the **Double scale** option to display the emulated device in double size or actual size on your monitor.
- Select or deselect the **White Background** option to display the emulated device LCD background color in white or green on your monitor.

NOTE: The term “skin” is used to refer to a set of graphics that an application uses to create its appearance. You can change the appearance of an application by changing its skin.

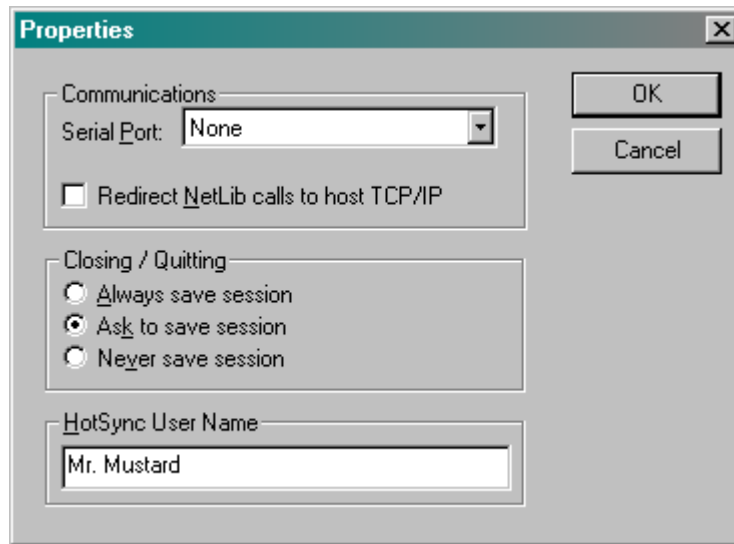
The Palm OS Emulator Runtime Environment

This section describes how you can modify the Palm OS Emulator runtime environment, including changing the properties and installing applications in the emulator session.

Palm OS Emulator Properties

You can use the Properties dialog box to modify characteristics of your Palm OS Emulator sessions. To display this dialog box, choose the **Properties** menu command on a Windows system, or the **Preferences** menu command on a Macintosh system. The Properties dialog box is shown in [Figure 1.18](#).

Figure 1.18 Changing the Palm OS Emulator properties



[Table 1.12](#) describes the options available in the properties dialog box.

Table 1.12 Palm OS Emulator properties

Option	Description
Serial Port	Specifies which serial port the Palm OS Emulator uses to emulate serial communications on the handheld device.
Redirect Netlib calls	Redirects Netlib calls in emulated software to TCP/IP calls on the desktop computer.
Session saving	Selects what action the Palm OS Emulator takes when you close a session or quit the program.
User name	Selects the user account name for synchronizing from Palm OS Emulator with the desktop computer HotSync application.

Preferences Files

Your properties are stored in a preferences file on your computer. Each property is stored as a text string that you can view with a text editor. The location of your preferences file depends on the type of computer that you are using, as shown in [Table 1.13](#).

Table 1.13 Palm OS Emulator preference file locations

Platform	File name	File location
Macintosh	Palm OS Emulator Preferences	In the Preferences folder.
Windows	Palm OS Emulator Preferences.ini	In the Windows System directory.
Unix	.poserrc	In your home directory.

Installing Applications

You can use the **Install** command to load applications or databases directly into the current Palm OS Emulator emulation session.

- in Windows, right-click on the Palm OS Emulator screen display and choose the Install Application/Database command
- on a Macintosh, select the Install Application/Database command from the File menu

The **Install** command displays an open file dialog box in which you can choose the application (.prc), database (.pdb), or Palm Query Application (.pqa) file that you want installed.

Palm OS Emulator immediately loads the file into emulated RAM. If Palm OS Emulator finds another application or database with the same creator ID, that application or database is deleted before the new version is loaded.

WARNING! If you install an application while the Palm OS Launcher is running, the Launcher does not update its data structures, and thus does not reflect the fact that a database has been added or modified. It is best to use the Install command while an application is running in the emulated session.

Serial Communications and Palm OS Emulator

The Palm OS Emulator supports emulation of the Palm device serial port connection. It does so by mapping Palm OS serial port operations to a communications port on the desktop computer. To select which port the Emulator uses, use the **Properties** (on Macintosh computers, this is **Preferences**) menu command, as described in [Palm OS Emulator Properties](#).

When emulated software accesses the Dragonball or Dragonball EZ serial port hardware registers, Palm OS Emulator performs the appropriate actions on the specified serial port on the desktop computer. This means that serial read and write operations work as follows:

- when outgoing data is written to the UART's `tx` register, the Emulator redirects that data to the desktop computer's serial port.
- when the emulated software attempts to read data from the UART's `rx` register, the Emulator reads data from the desktop computer's serial port and places the data into that register.

Using the HotSync Application With the Palm OS Emulator

You can perform a HotSync operation from your emulated session in one of two ways:

- If you are using a Windows-based computer, you can use the Network HotSync option, which greatly simplifies your communications efforts.
- If you are not using a Windows-based computer, or your computer is not connected to a network, you can use a null-modem cable to connect two ports together and perform a HotSync operation.

Synchronizing From Palm OS Emulator With a Network

To synchronize when you are connected to a network, you need to set up your HotSync Manager application to perform a network synchronization. You do not need to use a null-modem cable when performing a network synchronization with the Palm OS Emulator

Synchronizing From Palm OS Emulator Without a Network

To synchronize when you are not connected to a network, you need to connect the serial port that the HotSync application uses to communicate with the handheld device to another serial port that the Palm OS Emulator uses. You connect these ports together with a null modem cable, such as a LapLink cable.

For example, if you are using a Windows-based computer and your HotSync application uses the COM1 port, follow these steps:

1. Select the **Properties (Preferences on a Macintosh)** command and specify the COM2 port for use the Palm OS Emulator.

2. Connect COM1 and COM2 together with a null modem cable.
3. Select the HotSync command from the Palm OS Emulator menu.

The HotSync application synchronizes with the Palm OS Emulator just as it does with an actual hardware handheld device.

TIP: The desktop HotSync application is CPU intensive, which is not generally an issue; however, when you are using the HotSync application with the Palm OS Emulator, the two programs are sharing the same CPU, which can dramatically the synchronization down.

A handy trick to deal with this problem is to click on the Palm OS Emulator window after the HotSync process starts. This brings the Emulator back into the foreground and allows it to use more CPU time, which improves the speed of the overall process.

If your desktop computer has two ports and you use a serial mouse on one of them, you can temporarily disable the mouse, perform a synchronization, and reenale the mouse. Follow these steps:

1. Disable your mouse.
2. Restart Windows.
3. Connect the serial ports together with a null modem cable.
4. Start the Palm OS Emulator.
5. Press F10 to display the menu, then H to begin the HotSync operation.
6. After the HotSync operation completes, reenale your mouse.
7. Restart Windows again.

TIP: When you first perform a HotSync operation with the Palm OS Emulator, the HotSync application asks you to select a user name. It is a good idea to create a new user account, with a different name, for use with the Emulator.

Palm OS Emulator Error Handling

This section describes the error handling and reporting features of the Palm OS Emulator program, including the following information:

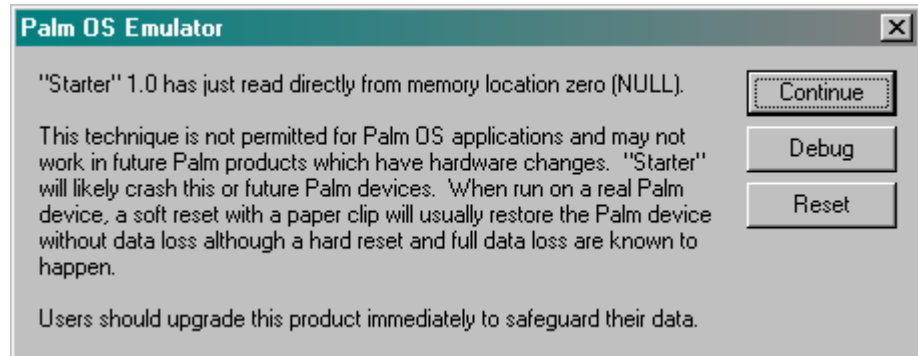
- which conditions are detected
- what the Emulator does upon detecting an error condition
- the message displayed for each error condition
- the options available to the user when an error condition occurs

Detecting an Error Condition

When Palm OS Emulator detects an error condition, it generates error message text and displays the error dialog box. If you select the Debug button in the error dialog box, the Emulator attempts to send the text to an external debugger such as Palm Debugger or MWDebug; if successful, the Emulator then stops emulating opcodes until the external debugger sends a command specifying that it can resume emulation.

If the Emulator cannot send the text to a debugger, it presents the error text to the user in a dialog box like the one shown in [Figure 1.19](#).

Figure 1.19 Palm OS Emulator error dialog box



You can click one of the three buttons in the dialog box:

Button	Description
Continue	Continues emulation, if possible.
Debug	Enters the external debugger, if one is running.
Reset	Performs a soft reset on the emulated device ROM.

Error Condition Types

The Palm OS Emulator detects condition types:

- A *processor exception* involves the CPU pushing the current program counter and processor state onto the stack, and then branching through a low-memory vector.
- A *memory access exception* involves access to a memory location that the application is not supposed to access.
- An *application error message* is a message displayed when software running on the handheld device calls a system function such as `ErrDisplayFileLineMsg` or `SysFatalAlert`.

The Palm OS Emulator uses four levels of accessibility when checking memory accesses:

- Applications have the least access to memory. An application is any software running in RAM on the handheld device.

- The system has more access to memory than do applications. The system is any software running in ROM on the handheld device.
- The memory manager has the most access to memory. The memory manager is any function operating within the context of a memory manager call, which means any function that runs while a memory manager function is still active.
- Some sections of memory cannot be accessed by any software.

Error Messages

[Table 1.14](#) shows the Palm OS Emulator error messages. Note that you can prevent some of these messages by disabling the relevant debugging option, as described in [Debug Options](#).

Table 1.14 Palm OS Emulator error messages

Error type	Description	Message example
Hardware register access	The application or system software has accessed a Dragonball or Dragonball EZ hardware register.	"Mytest" 1.0 has just read directly from the hardware registers.
Low-memory access	The application or system software has accessed low memory (the first 256 bytes), which contains the exception vectors.	"Mytest" 1.0 has just read directly from low memory. or "Mytest" 1.0 has just read directly from NULL (memory location zero)
System variable access	The application or system software has accessed a system variable, which resides in a memory location between low memory and the the end of the system function dispatch table.	"Mytest" 1.0 has just read directly from Palm OS global variables.

Table 1.14 Palm OS Emulator error messages (*continued*)

Error type	Description	Message example
LCD screen buffer access	The application or system software has accessed the screen buffer, which is defined by the LCD-related hardware registers.	"Mytest" 1.0 has just read directly from screen memory.
Memory Manager data structure access	The application or system software has accessed a memory manager data structure, which includes heap headers, master pointer tables, chunk headers, and chunk trailers.	"Mytest" 1.0 has just read directly from memory manager data structures.
Unlocked chunk access	The application or system software has accessed an unlocked memory chunk.	"Mytest" 1.0 has just read directly from an unlocked memory chunk.
Low-stack access	<p>The application or system software has accessed an area of the stack below the stack pointer.</p> <p>The stack is defined by values returned by the <code>SysGetAppInfo</code> function when it is called during system startup.</p> <p>If Palm OS Emulator cannot determine the stack range, it does not monitor low-stack accesses.</p>	"Mytest" 1.0 has just read directly from an invalid section of memory known as the "stack" .

Table 1.14 Palm OS Emulator error messages (*continued*)

Error type	Description	Message example
Uninitialized stack access	The application or system software has accessed uninitialized memory, which is memory that has not previously been written.	"Mytest" 1.0 has just read directly from an uninitialized section of memory known as the "stack" .
Free chunk access	The application or system software has accessed an unallocated memory chunk.	"Mytest" 1.0 has just read directly from an unallocated chunk of memory.
Uninitialized chunk access	The application or system software has attempted read access to uninitialized memory.	"Mytest" 1.0 has just read directly from an uninitialized chunk of memory.
Storage heap access	The application has accessed the storage heap.	"Mytest" 1.0 has just tried to write to the storage heap and that's just plain not allowed! Try using DmWrite.
Stack overflow	The application pushed more information onto the stack than is allocated for the stack.	"Mytest" 1.0 has just overflowed its stack.
Stack almost overflowed	The stack is close to overflowing, which means that the stack pointer is within a small number of bytes (typically 100) of the end of the stack.	"Mytest" 1.0 is getting close to overflowing the stack.

Table 1.14 Palm OS Emulator error messages (*continued*)

Error type	Description	Message example
Memory Manager semaphore acquisition time	The application or system software has acquired the Memory Manager semaphore for write access, and has held it for more than 10 milliseconds.	"Mytest" 1.0 has held the "Memory Manager semaphore" for approximately 20 milliseconds. It is recommended that applications not hold the semaphore longer than 10 milliseconds.
Invalid heap	Heap corruption detected during a regular heap check. The Palm OS Emulator regularly checks the heap.	During a regular checkup, the Emulator determined that the dynamic heap got corrupted. (corruption type) is one of the following message fragments: • The chunk was not within the heap it was supposed to be • The size of the chunk (chunk_size) was larger than the currently accepted maximum (chunk_max) • Some unused flags were set to "1" • The "hOffset" field of the chunk header did not reference a memory location within a master pointer block • The master pointer referenced by the "hOffset" field in the chunk

Table 1.14 Palm OS Emulator error messages (*continued*)

Error type	Description	Message example
Invalid program counter	The program counter has been set to an invalid memory location, which is a location outside of a 'CODE' resource.	"Mytest" 1.0 has just set the Program Counter (PC) to an invalid memory location.
Unimplemented trap.	<p>The application or system software has attempted to invoke an unimplemented system function.</p> <p>An unimplemented system function is one with a trap number outside of the the numbers in the system function dispatch table, or one whose table entry matches that of the SysUnimplemented function.</p>	"Mytest" 1.0 tried to call Palm OS routine trapNum (trapName). This routine does not exist in this version of the Palm OS.

Table 1.14 Palm OS Emulator error messages (*continued*)

Error type	Description	Message example
SysFatalAlert	<p>The application or system software has called the SysFatalAlert function.</p> <p>The Palm OS Emulator patches the SysFatalAlert function and present the message in its own dialog box, to allow the user to choose how to respond to the error.</p>	"Mytest" 1.0 has failed, reporting "attempted divide by 0". If this is the latest version of "Mytest", please report this to the application author.
Unhandled exception	The application or system software has caused an exception that the Palm OS Emulator cannot handle itself.	"Mytest" 1.0 has just performed an illegal operation. It performed a "exception". If this is the latest version of "Mytest" 1.0, please report this to the application author.

Sending Commands to Palm OS Emulator

You can use RPC packets to send commands to the Palm OS Emulator. You can invoke any function in the Palm OS dispatch table, including the Host Control functions, which are described in [Appendix B](#), “[Host Control API](#).”

The RPC packets use the same format as do packets that are sent to the debugger interface, which is described in [Appendix A](#), “[Debugger Protocol Reference](#).”

You use the socket defined by the `RPCSocketPort` preference to make RPC calls to Palm OS Emulator. When you send a packet to the emulator, you must set the `dest` field of the packet header to the value defined here:

```
#define slkSocketRPC (slkSocketFirstDynamic+10)
```

NOTE: You can disable the RPC command facility by setting the value of the `RPCSocketPort` preference to 0.

You can send four kinds of command packets to the emulator:

- ReadMem
- WriteMem
- RPC
- RPC2

The first three packet types are described in [Appendix A](#), “[Debugger Protocol Reference](#).” The fourth packet type, RPC2, is an extension of the RPC packet format that allows support for a wider range of operations.

The RPC2 Packet Format

```
#define sysPktRPC2Cmd    0x20
#define sysPktRPC2Rsp    0xA0

struct SysPktRPCParamInfo
{
    UInt8    byRef;
    UInt8    size;
    UInt16   data[1];
};

struct SysPktRPC2Type
{
    _sysPktBodyCommon;
    UInt16   trapWord;
    UInt32   resultD0;
    UInt32   resultA0;
    UInt16   resultException;
    UInt8    DRegMask;
    UInt8    ARegMask;
    UInt32   Regs[1];
    UInt16   numParams;
    SysPktRPCParamTypeparam[1];
};
```

```
};
```

Almost all of the RPC2 packet format is the same as the RPC format that is described in [Appendix A, “Debugger Protocol Reference.”](#) The RPC2 packet includes the following additional fields:

<code>resultException</code>	Stores the exception ID if a function call failed due to a hardware exception. Otherwise, the value of this field is 0.
<code>DRegMask</code>	A bitmask indicating which D registers need to be set to make this call.
<code>ARegMask</code>	A bitmask indicating which A registers need to be set in order to make this call.
<code>Regs[1]</code>	A variable length array containing the values to be stored in the registers that need to be set.

Only the registers that are being changed need to be supplied. Most of the time, `DRegMask` and `ARegMask` are set to zero and this field is not included in the packet.

If more than one register needs to be set, then the register values should appear in the following order: D0, D1, ..., D6, D7, A0, A1, ..., A6, A7. Again, only values for the registers specified in `DRegMask` and `ARegMask` need to be provided.

Getting Help With Palm OS Emulator

Palm OS Emulator is constantly evolving, and Palm Computing is always interested in hearing your comments and suggestions.

Palm provides a forum (emulator-forum@ls.palm.com) for questions and comments about Palm OS Emulator.

You can find the latest information about Palm OS Emulator in the Palm developer zone on the Internet:

<http://www.palm.com/devzone>.

Using the Palm OS® Emulator

Getting Help With Palm OS Emulator

NOTE: The source code for Palm OS Emulator is available on the Palm OS Emulator seed page: <http://www.palm.com/devzone>. You can create your own emulator by modifying this code.

For more information on the protocol used in Palm OS Emulator to send requests to and receive responses from a debugging target, see [Chapter](#), “[Debugger Protocol Reference](#).”
