



Palm OS® Cobalt Simulator Guide

Palm OS Cobalt Version 6.1

Written by Brian Maas and Denise Stone.

Technical assistance from Patrick Porlan, Puneet Mishra, Benjamin Combee, Madhavi Sunkara, David Navarro, Ezekiel Sanborn de Asis, Regis Nicolas, John Dance, Denis Berger, Dianne Hackborn, Ken Kru-gler, and Keith Rollin.

Copyright © 1996–2004, PalmSource, Inc. and its affiliates. All rights reserved. This technical documentation contains confidential and proprietary information of PalmSource, Inc. (“PalmSource”), and is provided to the licensee (“you”) under the terms of a Nondisclosure Agreement, Product Development Kit license, Software Development Kit license or similar agreement between you and PalmSource. You must use commercially reasonable efforts to maintain the confidentiality of this technical documentation. You may print and copy this technical documentation solely for the permitted uses specified in your agreement with PalmSource. In addition, you may make up to two (2) copies of this technical documentation for archival and backup purposes. All copies of this technical documentation remain the property of PalmSource, and you agree to return or destroy them at PalmSource’s written request. Except for the foregoing or as authorized in your agreement with PalmSource, you may not copy or distribute any part of this technical documentation in any form or by any means without express written consent from PalmSource, Inc., and you may not modify this technical documentation or make any derivative work of it (such as a translation, localization, transformation or adaptation) without express written consent from PalmSource.

PalmSource, Inc. reserves the right to revise this technical documentation from time to time, and is not obligated to notify you of any revisions.

THIS TECHNICAL DOCUMENTATION IS PROVIDED ON AN “AS IS” BASIS. NEITHER PALMSOURCE NOR ITS SUPPLIERS MAKES, AND EACH OF THEM EXPRESSLY EXCLUDES AND DISCLAIMS TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, ANY REPRESENTATIONS OR WARRANTIES REGARDING THIS TECHNICAL DOCUMENTATION, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY WARRANTIES IMPLIED BY ANY COURSE OF DEALING OR COURSE OF PERFORMANCE AND ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, ACCURACY, AND SATISFACTORY QUALITY. PALMSOURCE AND ITS SUPPLIERS MAKE NO REPRESENTATIONS OR WARRANTIES THAT THIS TECHNICAL DOCUMENTATION IS FREE OF ERRORS OR IS SUITABLE FOR YOUR USE. TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, PALMSOURCE, INC. ALSO EXCLUDES FOR ITSELF AND ITS SUPPLIERS ANY LIABILITY, WHETHER BASED IN CONTRACT OR TORT (INCLUDING NEGLIGENCE), FOR DIRECT, INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, EXEMPLARY OR PUNITIVE DAMAGES OF ANY KIND ARISING OUT OF OR IN ANY WAY RELATED TO THIS TECHNICAL DOCUMENTATION, INCLUDING WITHOUT LIMITATION DAMAGES FOR LOST REVENUE OR PROFITS, LOST BUSINESS, LOST GOODWILL, LOST INFORMATION OR DATA, BUSINESS INTERRUPTION, SERVICES STOPPAGE, IMPAIRMENT OF OTHER GOODS, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR OTHER FINANCIAL LOSS, EVEN IF PALMSOURCE, INC. OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR IF SUCH DAMAGES COULD HAVE BEEN REASONABLY FORESEEN.

PalmSource, Palm OS, Palm Powered, Graffiti, HotSync, and certain other trademarks and logos are trademarks or registered trademarks of PalmSource, Inc. or its affiliates in the United States, France, Germany, Japan, the United Kingdom, and other countries. These marks may not be used in connection with any product or service that does not belong to PalmSource, Inc. (except as expressly permitted by a license with PalmSource, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits PalmSource, Inc., its licensor, its subsidiaries, or affiliates. All other product and brand names may be trademarks or registered trademarks of their respective owners.

IF THIS TECHNICAL DOCUMENTATION IS PROVIDED ON A COMPACT DISC, THE SOFTWARE AND OTHER DOCUMENTATION ON THE COMPACT DISC ARE SUBJECT TO THE LICENSE AGREEMENTS ACCOMPANYING THE SOFTWARE AND OTHER DOCUMENTATION.

Palm OS Cobalt Simulator Guide

Document Number 3129-002

August 26, 2004

For the latest version of this document, visit

<http://www.palmos.com/dev/support/docs/>.

PalmSource, Inc.

1240 Crossman Avenue

Sunnyvale, CA 94089

USA

www.palmsource.com

Table of Contents

About This Book	vii
What This Book Contains	vii
Additional Resources	vii
 1 Introducing Palm OS Cobalt Simulator	 1
What Is Palm OS Cobalt Simulator?	1
Palm OS Applications	2
Palm OS System Code	2
Palm Application Compatibility Environment	3
Device Driver Set	3
How Does Palm OS Cobalt Simulator Compare to Palm Simulator for Macintosh?	4
Prerequisites	4
Security	4
 2 Using Palm OS Cobalt Simulator	 7
Installing Palm OS Cobalt Simulator	7
Starting Palm OS Cobalt Simulator	8
Specifying Command Line Arguments	9
Using the Initialization File	17
Loading ROM Images	19
Running Palm OS Cobalt Simulator	19
Palm OS Cobalt Simulator Display	19
Displaying Menu Items	20
Entering Data	20
Installing Applications	20
Using AutoRun, AutoLoad, and AutoRunAndQuit	21
Using AppCreator to Start an Application	22
Starting Simulator with a Storage Snapshot File	22
Using Communication Functions	23
Testing Infrared Communication (Beaming)	23
Performing a HotSync Operation	24
Using External Debug Tools with Palm OS Cobalt Simulator	24
Using Palm OS Cobalt Simulator with 68K Debuggers	25

3 Using Skin Files	27
Using Simulator Skin Files	27
Using Skin Files	27
Modifying or Creating Skin Files	27
4 Using Palm OS Reporter	33
About Palm OS Reporter	33
Palm OS Reporter Features	34
Installing Palm OS Reporter	34
Palm OS Reporter Package Files	34
Adding Trace Calls to Your Application	35
Tracing ARM-Based Applications	35
Trace Manager Functions in a Code Sample	36
Tracing 68K-Based Applications	36
Specifying Trace Strings	38
Displaying Trace Information in Palm OS Reporter	39
Starting a Palm OS Reporter Session	40
Filtering Information in a Palm OS Reporter Session	41
Using the Palm OS Reporter Toolbar	42
Troubleshooting Palm OS Reporter	43
5 Palm OS Cobalt Simulator User Interface Reference	45
Menu Reference Summary	45
Displaying the Palm OS Cobalt Simulator Menu Items	46
Reset Menu	46
Soft	46
Hard	46
No Notify	46
View Menu	47
Databases	47
Heaps	48
Events	49
Install Menu	50
Settings Menu	51
Display > Resolution	51
Display > Orientations	52

Display > Color Depth	52
Display > Renderer	53
Display > Skin	53
Display > Magnification	53
Display > Always on Top	54
Communication Ports	54
Networking > Standard Palm OS TCP/IP	55
Networking > Winsock-Based STREAMS Module	55
Networking > Winsock-Based SocketLib Replacement	56
Dock Status	56
Memory	56
Battery	57
HotSync Name.	58
Enable Sound	58
Traces.	58
Storage Menu	59
Load	59
Save	60
Export Database	60
Volumes	60
Screenshot Menu	62
About Menu	62
Exit Menu	62
Keyboard Equivalents Reference	63
Hardware Buttons	63
Additional Keyboard Functions	64

About This Book

Palm OS Cobalt Simulator Guide provides conceptual, guidance, and reference information for developers who want to use Palm OS Cobalt Simulator to test their applications.

This edition describes Palm OS Cobalt Simulator for Palm OS version 6.1.

What This Book Contains

This volume has the following organization:

- [Chapter 1, “Introducing Palm OS Cobalt Simulator,”](#) on page 1 describes general concepts that will help you understand how to use Palm OS Cobalt Simulator.
- [Chapter 2, “Using Palm OS Cobalt Simulator,”](#) on page 7 provides task information describing how to use Palm OS Cobalt Simulator functions.
- [Chapter 4, “Using Palm OS Reporter,”](#) on page 33 describes how you can use Palm OS Reporter with Palm OS Cobalt Simulator to do trace analysis of your application.
- [Chapter 5, “Palm OS Cobalt Simulator User Interface Reference,”](#) on page 45 documents the Palm OS Cobalt Simulator menu commands and cross references relevant task information.

Additional Resources

- Documentation
PalmSource publishes its latest versions of this and other documents for Palm OS developers at
<http://www.palmos.com/dev/support/docs/>
- Training
PalmSource and its partners host training classes for Palm OS developers. For topics and schedules, check
<http://www.palmos.com/dev/training>

- Knowledge Base

The Knowledge Base is a fast, web-based database of technical information. Search for frequently asked questions (FAQs), sample code, white papers, and development documentation at

<http://www.palmos.com/dev/support/kb/>

Introducing Palm OS Cobalt Simulator

This chapter provides conceptual information that will help you learn about Palm OS® Cobalt Simulator.

This chapter contains the following sections:

- [“What Is Palm OS Cobalt Simulator?”](#) on page 1
- [“How Does Palm OS Cobalt Simulator Compare to Palm Simulator for Macintosh?”](#) on page 4
- [“Prerequisites”](#) on page 4
- [“Security”](#) on page 4

What Is Palm OS Cobalt Simulator?

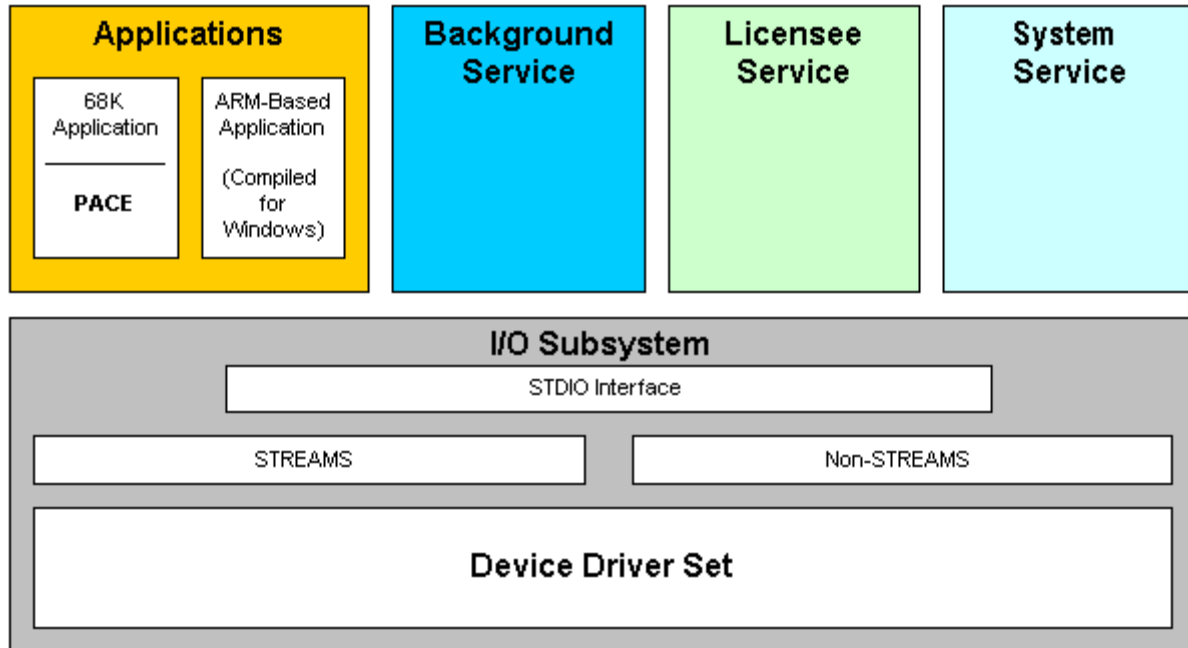
Palm OS Cobalt Simulator is Palm OS recompiled for a desktop machine processor. Palm OS Cobalt Simulator combines all of the Palm OS Cobalt architecture components into a single execution environment.

- Palm OS applications
- Palm Application Compatibility Environment (PACE)
- Palm OS system code
- Device driver set implementation

Introducing Palm OS Cobalt Simulator

What Is Palm OS Cobalt Simulator?

Figure 1.1 Palm OS Cobalt Simulator components



Palm OS Applications

Palm OS Cobalt Simulator includes all of the built-in Palm OS applications, such as Address Book, Date Book, Memo Pad, and To Do List. The built-in Palm OS applications are included in the Simulator ROM file.

You can add other Palm OS applications to a Simulator session as well. For more information, see [“Installing Applications”](#) on page 20.

Palm OS System Code

Palm OS Cobalt Simulator includes all of the Palm OS Cobalt system code, compiled to run on Windows.

Palm OS Runtime Services

Palm OS Cobalt Simulator implements the Palm OS Cobalt runtime services by using desktop system library files (DLLs). These DLLs are located in the Palm OS Cobalt Simulator directory, and are loaded when the code that they contain needs to be executed.

Communication Stacks

Palm OS Cobalt Simulator uses the communication stacks for Sockets API and Telephony API components. Palm OS Cobalt Simulator can optionally redirect Socket API calls to the host machine TCP/IP stack.

Palm Application Compatibility Environment

When you use Simulator to test your existing 68K application, the application is run in the Palm Application Compatibility Environment (PACE). PACE provides a 68K application environment that is equivalent to Palm OS 4.1.

PACE handles the data translation required for a 68K application to run on Palm OS Cobalt. For example, 68K applications read and write data in big-endian mode, but Palm OS Cobalt views data in little-endian mode. When a 68K application calls a Palm OS function, PACE handles the translation of the parameters, objects, and structure layouts so that existing applications do not have to be updated to handle the change of endianness. PACE creates “shadow structures” for the 68K application’s data that allow the 68K application to run under Palm OS Cobalt.

Device Driver Set

The device driver set is responsible for insulating Palm OS from the underlying system and hardware. By running Palm OS on top of the device driver set for the desktop machine, Palm OS Cobalt Simulator recreates the Palm OS program execution environment in the desktop machine.

Applications that run in this environment see the same functionality provided by the Palm OS managers, libraries, and applications as on a Palm Powered handheld.

Because Palm OS Cobalt Simulator runs on a desktop machine, it can be integrated with the desktop tools for Palm OS development. This integration allows full source level debugging of the code that is targeted for Palm Powered devices.

How Does Palm OS Cobalt Simulator Compare to Palm Simulator for Macintosh?

Palm OS Cobalt Simulator has significant advantages over the Macintosh simulator in use with Palm OS 4:

- All of the applications and shared libraries that can be loaded onto a handheld can be loaded in Palm OS Cobalt Simulator at the same time. This allows analyzing the interactions between the applications, shared libraries, and Palm OS with a much better accuracy than before.
- Palm OS Cobalt Simulator supports multi-threading.
- Palm OS Cobalt Simulator runs the same Palm OS code that runs on a Palm Powered handheld. The only difference between Palm OS running on a handheld and Palm OS Cobalt Simulator is the driver set.
- The Macintosh simulator required that components be statically linked together. Because Palm OS Cobalt Simulator does not have this requirement, the simulation reproduces accurately the Palm OS runtime architecture on top of the desktop machine system.

Prerequisites

Palm OS Cobalt Simulator runs on Windows 2000 and Windows XP.

Palm OS Cobalt Simulator is flexible enough to be used with most C/C++ development chains, with or without an associated integrated development environment (IDEs such as Palm OS Developer Suite, CodeWarrior, and Visual C++).

Security

Palm OS Cobalt Simulator should not be viewed as a secure execution environment because it is intended as a development platform. When you are debugging applications, you need extensive access to the state of the application as it is running in Palm OS.

Palm OS Cobalt Simulator supports debugging of 68K applications using a TCP/IP connection, usually on port 2000. Simulator opens this port when starting and when accepting connections from debugger applications, like Palm OS Debugger or CodeWarrior for Palm OS. If you block access to these ports, you will not be able to debug with these programs. You may want to configure your firewall to only allow connections from the local machine to limit any security risks.

- Use a firewall if the machine is connected to the internet.
- Don't run untrusted applications within Palm OS Cobalt Simulator.

Using Palm OS Cobalt Simulator

This chapter provides guidance and reference information that will help you use Palm OS® Cobalt Simulator.

- [“Installing Palm OS Cobalt Simulator”](#) on page 7
- [“Starting Palm OS Cobalt Simulator”](#) on page 8
- [“Specifying Command Line Arguments”](#) on page 9
- [“Using the Initialization File”](#) on page 17
- [“Loading ROM Images”](#) on page 19
- [“Running Palm OS Cobalt Simulator”](#) on page 19
- [“Using Communication Functions”](#) on page 23
- [“Using External Debug Tools with Palm OS Cobalt Simulator”](#) on page 24

Installing Palm OS Cobalt Simulator

Palm OS Cobalt Simulator consists of the following:

- The executable file: `PalmSim.EXE`
- A ROM file
- The DLLs required by the ROM file

The ROM file is specific to Palm OS Cobalt Simulator; the ROM file is not the same as ROM files used with Palm OS Emulator. ARM-based PRCs and 68K-based PRCs are embedded in this ROM file. However, ARM-based PRCs don’t really contain code; rather, they reference external DLL files. As a result, there is at least one DLL per ARM-based application or shared library.

Using Palm OS Cobalt Simulator

Starting Palm OS Cobalt Simulator

The DLLs required by the ROM file can be in any directory specified by the `PALMSOURCE_SIM_PATH` environment variable. The DLLs can also be in the same directory as the executable file `PalmSim.EXE`, or in the subfolder for the locale-specific ROM file (such as `enUS` or `jaJP`).

To use Palm OS Reporter's tracing functions with Palm OS Cobalt Simulator, you need to have the file `PalmTrace.DLL` in a directory specified by the `PATH` environment variable. For more information on Palm OS Reporter, see [Chapter 4, "Using Palm OS Reporter,"](#) on page 33.

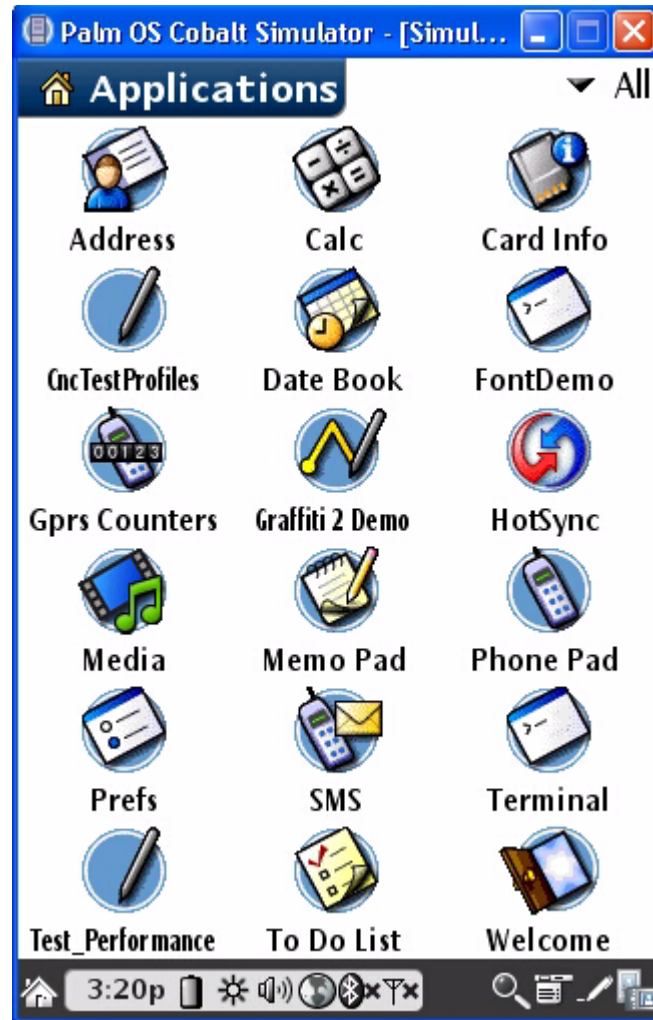
Starting Palm OS Cobalt Simulator

To start Palm OS Cobalt Simulator, run `PalmSim.EXE`. The first time you start Palm OS Cobalt Simulator, you are asked to select a ROM file.

You can also start Palm OS Cobalt Simulator by dragging and dropping a Simulator ROM file onto the `PalmSim.EXE` icon. (Note again that the Simulator ROM file is not the same as the ROM files used for Palm OS Emulator. You should not drop an Emulator ROM file on the `PalmSim.EXE` icon.)

When Palm OS Cobalt Simulator starts, the main window is displayed, as shown in [Figure 2.1](#) on page 9.

Figure 2.1 Palm OS Cobalt Simulator's main window



Specifying Command Line Arguments

You can supply the session parameters for Palm OS Cobalt Simulator as command line options. For example:

```
PalmSim.EXE  
-rom:Simulator_Full_EFIGS_Release.rom
```

[Table 2.1](#) on page 10 shows the options that you can specify with the command line version of Palm OS Cobalt Simulator.

Using Palm OS Cobalt Simulator

Specifying Command Line Arguments

NOTE: The command line options are not case sensitive, but the values specified for the options might be (for example, the four-character application creator ID for the `-appcreator` option).

Table 2.1 Command line options

Option Syntax	Parameter Values
<code>-68kdebuggerport:</code> <code>[host:port None]</code>	<p><i>host</i> - The name of the host used for the 68K debugger.</p> <p><i>port</i> - The port used for the 68K debugger.</p> <p>Example:</p> <pre>-68KDebuggerPort: localhost:2000</pre> <p>For more information, see “Communication Ports” on page 54.</p>
<code>-68kdebuggerporttype: type</code>	<p><i>type</i> - The type of port used for the 68K debugger. The default is TCP/IP.</p> <p>For more information, see “Communication Ports” on page 54.</p>

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-additionalPorts: <i>portList</i></code>	<p><i>portList</i> - A list of additional communications ports, enclosed in parentheses and separated by commas.</p> <p>Each port definition consists of a name, a port type, and the port information:</p> <p>(<i>portName</i>, <i>portType</i>, <i>portInfo</i>)</p> <p>where:</p> <p><i>portName</i> - A unique name for a port.</p> <p><i>portType</i> - [Standard RS-232 Modem IR through RS-232 IR through JetEye TCP/IP]</p> <p><i>portInfo</i> - The port information, either <i>host:port</i> or <i>COMx</i>.</p> <p>Example:</p> <pre>-AdditionalPorts: "(FirstPort, Standard RS- 232, COM3) , (SecondPort, TCP/ IP, LOCALHOST:1000) "</pre> <p>Note: The quotation marks are required if the parameter string contains spaces.</p> <p>For more information, see "Communication Ports" on page 54.</p>
<code>-alwaysontop: [on <u>off</u>]</code>	<p>Indicates whether the Simulator window should stay in front of other windows on the desktop. The default is <code>off</code>.</p> <p>For more information, see "Display > Always on Top" on page 54.</p>

Using Palm OS Cobalt Simulator

Specifying Command Line Arguments

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-appcreator:xxxx</code>	<p>xxxx - A four-character Creator ID indicating the application to start.</p> <p>For more information, see Chapter 2, “Using AppCreator to Start an Application,” on page 22.</p>
<code>-autoload:[filename filelist directory]</code>	<p><i>filename</i> - The PRC or PDB file that you want Palm OS Cobalt Simulator to automatically load.</p> <p><i>filelist</i> - A list of PRC and PDB files that you want Palm OS Cobalt Simulator to automatically load, each item in the list separated by semicolons. If a filename includes spaces, the filename must be enclosed in quotation marks.</p> <p>For example:</p> <pre>-autoload: "C:\Sample\My Sample.prc"; "C:\Sample\My Data.pdb"; C:\Program.prc</pre> <p><i>directory</i> - The directory containing the PRC and PDB files that you want Palm OS Cobalt Simulator to automatically load.</p> <p>For more information on using AutoLoad, see “Using AutoRun, AutoLoad, and AutoRunAndQuit” on page 21.</p>

Table 2.1 Command line options (continued)

Option Syntax	Parameter Values
<code>-autorunandquit: [filename directory]</code>	<p><i>filename</i> - The PRC or PDB file that you want Palm OS Cobalt Simulator to automatically run.</p> <p><i>directory</i> - The directory containing the PRC and PDB files that you want Palm OS Cobalt Simulator to automatically run.</p> <p>For more information on using <code>AutoRunAndQuit</code>, see “Using AutoRun, AutoLoad, and AutoRunAndQuit” on page 21.</p>
<code>-autorun: [filename directory]</code>	<p><i>filename</i> - The PRC or PDB file that you want Palm OS Cobalt Simulator to automatically run.</p> <p><i>directory</i> - The directory containing the PRC and PDB files that you want Palm OS Cobalt Simulator to automatically run.</p> <p>For more information on using <code>AutoRun</code>, see “Using AutoRun, AutoLoad, and AutoRunAndQuit” on page 21.</p>
<code>-bitdepth: [4 8 16 24 32]</code>	<p>Sets the color depth for the display setting. The default is 16.</p> <p>For more information, see “Display > Resolution” on page 51.</p>
<code>-cradleport: COMx</code>	<p><i>COMx</i> - The communications port used to talk to the cradle port.</p> <p>For more information, see “Communication Ports” on page 54.</p>

Using Palm OS Cobalt Simulator

Specifying Command Line Arguments

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-dllpath:directories</code>	<p><i>directories</i> - A list of directories, separated by semicolons (;), that specify where you want Palm OS Cobalt Simulator to look to find DLLs.</p> <p>This command line option is similar to the <code>PALMSOURCE_SIM_PATH</code> environment variable (described in “Installing Applications” on page 20).</p> <p>If you use both the environment variable and the command line option, Palm OS Cobalt Simulator looks in the paths specified by the environment variable first.</p>
<code>-infraredport:[COMx None]</code>	<p><i>COMx</i> - The communications port used for infrared.</p> <p>For more information, see “Communication Ports” on page 54.</p>
<code>-infraredporttype:type</code>	<p><i>type</i> - The type of port used for infrared communication.</p> <p>For more information, see “Communication Ports” on page 54.</p>
<code>-preferredLocale:llCC</code>	<p><i>llCC</i> - A language/country descriptor identifying the language you want Palm OS Cobalt Simulator to use.</p> <p>When you use this command line option, Palm OS Cobalt Simulator skips the language selection application for a ROM that supports multiple languages.</p>

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-ram: <i>size</i></code>	<p><i>size</i> - An integer value indicating the amount of RAM to emulate during the session, specified in kilobytes. The size has to be the exact number of bytes. For example, to specify 16 MB, use the value 165536.</p> <p>For more information, see “Memory” on page 56.</p>
<code>-redirectsocketlibcalls: [on off]</code>	<p>Indicates whether you want to redirect Socket API calls to the host machine’s TCP/IP stack. The default is <code>off</code>.</p> <p>For more information, see “Networking > Standard Palm OS TCP/IP” on page 55.</p>
<code>-rom: <i>romname</i></code>	<p><i>romname</i> - The name of the ROM file.</p> <p>If you do not specify a value for this option when you first start Simulator, Simulator opens a dialog box asking for you to specify a ROM file. If you do not specify a value for this option on subsequent uses of Simulator, Simulator loads the ROM file name from the initialization file.</p>
<code>-sound: [on off]</code>	<p>Indicates whether Simulator should output sound. The default is <code>on</code>.</p> <p>For more information, see “Enable Sound” on page 58.</p>
<code>-storagesnapshotfile: <i>ssfname</i></code>	<p><i>ssfname</i> - The name of the storage snapshot file (SSF), indicating the saved storage snapshot that you want Simulator to load at startup or upon hard reset.</p> <p>For more information, see “Storage Menu” on page 59.</p>

Using Palm OS Cobalt Simulator

Specifying Command Line Arguments

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-tracetarget:[machine file stderr]</code>	<p><i>machine</i> - The identifier of the machine to which you want traces redirected. For example:</p> <pre>-tracetarget: tcp:localhost:25998</pre> <p><i>file</i> - The file to which you want traces redirected. For example:</p> <pre>-tracetarget:file:myfile.txt</pre> <p><i>stderr</i> - Redirects trace output to standard error. For more information, see “Traces” on page 58.</p>
<code>-usehostethernetadapter: [on off]</code>	<p>Indicates whether Palm OS Cobalt Simulator can use the host ethernet adapter. The default is on. However, the ethernet driver does not load if you have not installed the WinPcap library.</p> <p>For more information, see “Networking > Standard Palm OS TCP/IP” on page 55.</p>
<code>-windoworiginx:integer</code>	<p><i>integer</i> - Specifies Simulator’s horizontal distance from the left of the screen when the window is opened. The default is 0.</p>
<code>-windoworiginy:integer</code>	<p><i>integer</i> - Specifies Simulator’s vertical distance from the top of the screen when the window is opened. The default is 0.</p>
<code>-zoom:[1 2 3 4]</code>	<p>Magnification level. The default is 1.</p> <p>For more information, see “Display > Magnification” on page 53.</p>

Using the Initialization File

The command line arguments can be set in the Simulator initialization file, `palmsim.ini`. Any options specified on the command line will override the initialization file settings. (Command line options are described in [Table 2.1](#) on page 10.)

When you exit Simulator, the session's values are written to `palmsim.ini`; these values are loaded from the initialization file the next time you start Simulator.

To reset the values to the original defaults, simply delete the `palmsim.ini` file.

Listing 2.1 Sample `palmsim.ini` File

```
[Settings]
ROM=C:\SDK\ROM\Simulator_Full_EFIGS_Release.rom
Sound=1
AlwaysOnTop=0
Fullscreen=0
Zoom=1
Renderer=0
BitDepth=16
RedirectSocketLibCalls=0
UseHostTCP=0
CradlePort=
CradlePortType=Standard RS-232
UseHostBatteryInfo=1
BatteryPluggedIn=1
BatteryPercent=75
BatteryKind=2
DockStatus=0
UseSkin=0
RAM=32768
LastSilkScreen=
InfraredPort=localhost:6417
InfraredPortType=TCP/IP
68KDebuggerPort=localhost:2000
68KDebuggerPortType=TCP/IP
VirtualPhonePort=localhost:6416
VirtualPhonePortType=TCP/IP
AdditionalPorts=
LastStorageSnapshot=C:\SDK\Simulator\Release\storage
snapshot.ssf
ScreenDensity=144
```

Using Palm OS Cobalt Simulator

Using the Initialization File

```
ScreenWidth=320
ScreenHeight=480
ScreenOrientation=0
ScreenRotations=14
HasSilkScreen=0
TraceTarget=tcp:localhost:25998
ConsoleVisible=0
EventsVisible=0
ThreadsVisible=0
HeapsVisible=0
EventsFilter=
AutoStartReporter=0
UseHostEthernetAdapter=1
WindowOriginX=0
Window0Right=326
WindowOriginY=0
Window0Bottom=505
```

The options that you can also set on the command line are described in [Table 2.1](#) on page 10.

NOTE: The initialization file options use an equal sign (=) to separate the option from the value, rather than the colon (:) used in the command line version. Also, boolean values in the initialization file are indicated using 0 and 1 rather than `off` and `on`.

Loading ROM Images

When you first run Palm OS Cobalt Simulator, you can specify the ROM image filename using the `-rom` command line option. If you do not specify a value for this option, Simulator opens a dialog box asking for you to specify a ROM file.

When you restart Simulator, it assumes you want to use the ROM file that you specified when you first started Simulator.

To run Simulator with a different ROM file, you can do one of the following:

- Change the `ROM` option value in the `palmsim.ini` file.
- Specify a new value using the `-rom` command line option.
- Hold down the **SHIFT** key when you start Simulator. Then, Simulator opens a dialog box asking for you to specify a ROM file.
- Drag and drop a ROM file onto the `PalmSim.EXE` icon.

NOTE: If you are running the Release version of Palm OS Cobalt Simulator, you must use a Release version ROM file. If you are running the Debug version of Palm OS Cobalt Simulator, you must use a Debug version ROM file.

Running Palm OS Cobalt Simulator

This section provides an overview how to use Palm OS Cobalt Simulator.

Palm OS Cobalt Simulator Display

The Palm OS Cobalt Simulator display looks very much like a real Palm Powered 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 Palm OS Cobalt Simulator functions.

Displaying Menu Items

Right-click (use mouse button 2) on the Palm OS Cobalt Simulator screen display to access the menu items. The Palm OS Cobalt Simulator menu displays, as shown in [Figure 2.2](#).

Figure 2.2 The Palm OS Cobalt Simulator menu



For more information about the Palm OS Cobalt Simulator menu items, see “[Menu Reference Summary](#)” on page 45.

Entering Data

Palm OS Cobalt Simulator supports handwriting recognition. You can draw characters using a mouse.

Palm OS Cobalt Simulator also supports keyboard input. When a field is active, you can use the keyboard to enter text. You can also use keyboard equivalents for hardware buttons and other functions. For more information, see “[Keyboard Equivalents Reference](#)” on page 63.

Installing Applications

To install Palm OS applications, you can either use the **Install** menu item or drag and drop files onto Simulator.

NOTE: Palm OS applications written for Palm OS 4 or earlier (68K applications) run in Palm OS Cobalt Simulator without any changes.

However, if you are installing a Palm OS application that uses ARM-native code, you must recompile the ARM-native code into a Windows DLL.

Palm OS Cobalt Simulator recognizes a call to an ARM-native subroutine as a call into a DLL. A fully ARM-native application must also be built as a PRC that calls a Windows DLL in order for Palm OS Cobalt Simulator to use it.

Place the DLL either in the Palm OS Cobalt Simulator directory or in a directory specified by using the `-dllpath` command line option.

Using the Install Menu Item

Use **Install > Database** to open the Install Database dialog box. You can install a single PRC or PDB file, or you can use SHIFT-click to select multiple databases for installation. You will receive a warning message if Palm OS Cobalt Simulator cannot use a PRC or PDB file.

Using Drag and Drop

Drag and drop PRC and PDB files onto the Palm OS Cobalt Simulator main window. You will receive a warning message if Palm OS Cobalt Simulator cannot use a PRC or PDB file.

Using AutoRun, AutoLoad, and AutoRunAndQuit

Palm OS Cobalt Simulator also supports the AutoRun, AutoLoad and AutoRunAndQuit features that make it easier to test you applications.

You can use these features either by using the command line options (`-autorun`, `-autoload`, `-autorunandquit`) or by creating special directories.

Using Palm OS Cobalt Simulator

Running Palm OS Cobalt Simulator

For information about the command line options, see [Table 2.1](#) on page 10.

To use an AutoRun directory:

- Create a subdirectory of the Simulator directory called `AutoRun`.
- Place the PRC and PDB files that you want to automatically run in the `AutoRun` directory.
- When you start Simulator, Simulator automatically loads the PRC and PDB files.

NOTE: If you include more than one PRC file in the `AutoRun` directory, Simulator loads all of the PRC files, but runs only the last PRC file loaded (the last PRC filename when sorted alphabetically).

To use `AutoLoad` or `AutoRunAndQuit`, follow the steps listed above using `AutoLoad` or `AutoRunAndQuit` as the directory name rather than `AutoRun`.

Using AppCreator to Start an Application

To have Simulator switch to a specific application at startup, you can either set the `AppCreator` value to the application's creator ID in the `palmsim.ini` file, or you can specify the `-AppCreator` command line argument. See "[Specifying Command Line Arguments](#)" on page 9 for more information about command line arguments.

Starting Simulator with a Storage Snapshot File

To have Simulator load a specific storage snapshot file at startup or upon hard reset, you can either:

- Set the `storagesnapshotfile` value in the `palmsim.ini` file
- Specify the `-StorageSnapshotFile` command line argument.

Note that the storage image size specified by the storage snapshot file must match the current storage size setting.

See “[Specifying Command Line Arguments](#)” on page 9 for more information about command line arguments. For more information about storage snapshot files, see “[Storage Menu](#)” on page 59.

Using Communication Functions

Palm OS Cobalt Simulator supports the following communication functions:

- “[Testing Infrared Communication \(Beaming\)](#)” on page 23
- “[Performing a HotSync Operation](#)” on page 24

Testing Infrared Communication (Beaming)

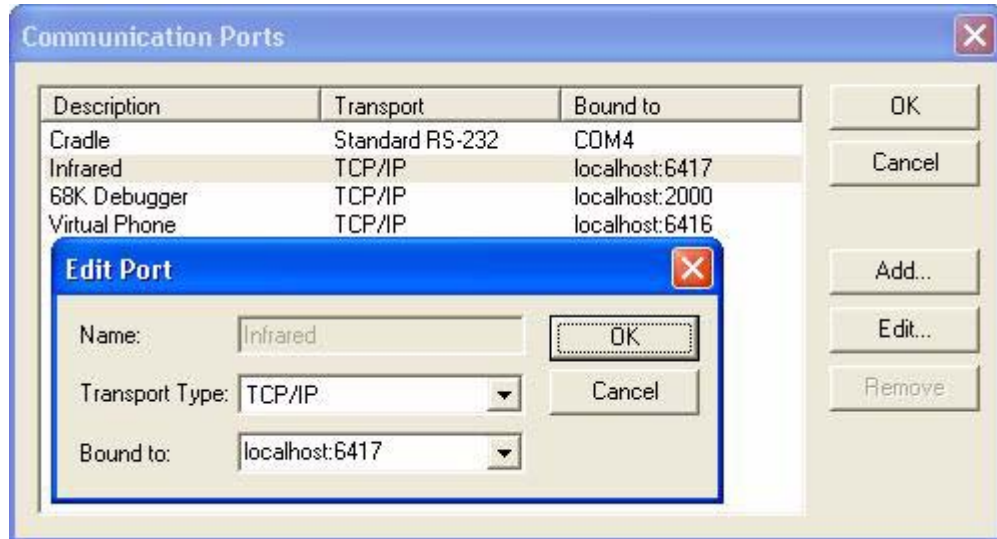
You can beam data between two Palm OS Cobalt Simulator sessions to test infrared communication in your applications.

1. Select [Communication Ports](#) to open the Communication Ports dialog box.
2. Select **Infrared** from the listed ports, and click **Edit**.
3. Set the **Transport Type** to TCP/IP, and set the **Bound to** value to `localhost:6417`, as shown in [Figure 2.3](#) on page 24.

Using Palm OS Cobalt Simulator

Using External Debug Tools with Palm OS Cobalt Simulator

Figure 2.3 Setting communication ports to test beaming



Performing a HotSync Operation

You can perform a HotSync® operation through serial connection, through IrDA, and through TCP/IP.

NOTE: To perform a HotSync operation, you may want to first select **Settings > HotSync Name** to enter a user name.

HotSync Operation with Two Serial Ports

1. Connect the serial ports with a NULL serial cable.
2. Set the properties for the HotSync application to perform a local HotSync operation with one of the serial ports.
3. Using the Simulator menu **Settings > Communication Ports**, set the Cradle port to Standard RS232 bound to the other serial communication port.

Using External Debug Tools with Palm OS Cobalt Simulator

Palm OS Cobalt Simulator can be used with a 68K debugger, to examine the state of the 68K emulated applications. Palm OS Cobalt

Simulator can also be used with a Windows debugger, to examine the state of ARM-native code that has been recompiled to run as a PRC calling a DLL.

Using Palm OS Cobalt Simulator with 68K Debuggers

Palm OS Cobalt Simulator is a debug target, just as an actual device or Palm OS Emulator. You can use Palm Debugger, Metrowerks CodeWarrior Debugger, or any other debugger you use to debug 68K-native applications.

Using Palm OS Cobalt Simulator

Using External Debug Tools with Palm OS Cobalt Simulator

Using Skin Files

Using Simulator Skin Files

Palm OS Cobalt Simulator uses skin files to present the image of a handheld. Note that the skin is simply a graphic; it does not change the ROM or the handheld being emulated. The skin simply changes the appearance of the Simulator window.

The skin choices available are dependent on the handheld selection. When you select a handheld, Simulator reads through the available SKIN files for the skin names that support the selected handheld.

Palm OS Emulator comes with a built-in **Generic** skin. This skin is suitable for doing your own application testing and debugging.

However, there are times when you want Simulator to look more like a specific handheld, such as when you are using Simulator to demonstrate your application to others. This section describes how to use additional skins that are available, and how to modify or create your own skins.

Using Skin Files

You can select the skin file for your simulation session by using **Settings > [Display > Skin](#)**.

Modifying or Creating Skin Files

Skins are defined by a pair of files: an image file and a SKIN file that describes the image file. The image file is a graphic; currently, only BMP format is supported.

The associated SKIN file is a text file that describes the image. The text file is made up of a series of lines, each line defining an attribute of the image. Each definition is of the form:

`<attribute>=<value>`

Using Skin Files

Using Simulator Skin Files

This is similar to the way INI files are stored on Windows, and how Simulator saves its own preferences.

Conditions for Skin File Entries

The following conditions apply for the definitions in SKIN files:

- The attribute is case-sensitive. For example, "Name" and "name" are not equivalent.
- There can be only one definition of each attribute. For example, if the skin can be used with multiple handhelds, specify both handhelds on the same "Devices" definition. This definition is correct:
`Devices = Pilot1000, Pilot5000`
However, this definition is not correct:
`Devices = Pilot1000`
`Devices = Pilot5000`
- White space is optional, both around the equal sign and in the specification of the value. For example, "color=1,2,3" is the same as "color = 1, 2, 3".
- The file can include comments, which are ignored when the file is parsed. Comments appear on their own lines, and start with "#" or ';'.
- Invalid files are detected and silently ignored. There is currently no error reporting when invalid values are encountered. Your only indication that something is wrong is that your skin won't show up in the Skins menu or dialog box.

Specifying Attributes in Skin Files

This list defines the attributes that you can use in skin files, and a describes how to specify the attribute's values.

Name

This is the name of the skin. The value is what appears in the Skin menu in the New Session dialog box and in the Skins dialog box.

Example:

`Name = Keith's Cool Skin`

File1x

This is the name of the single-scale BMP image file.

Example:

```
File1x = MySkin1.bmp
```

BackgroundColor

This field defines the normal color used when displaying the LCD area of Simulator's display. The value is specified as an RGB set of values. The three components are provided as hexadecimal or decimal values in the range from 0 to 255, separated by commas.

Example:

```
BackgroundColor = 0x7B, 0x8C, 0x5A
```

HighlightColor

Note: This setting is not currently supported in this release of Palm OS Cobalt Simulator.

This field defines the backlighting color used when displaying the LCD area of Simulator's display. (That is, the color used for when the user turns on backlighting by holding down the power button.)

The value is specified as an RGB set of values. The three components are provided as hexadecimal or decimal values in the range from 0 to 255, separated by commas.

Example:

```
HighlightColor = 132, 240, 220
```

Devices

Provides the list of handhelds with which this skin can be used. One or more handhelds can be provided, separated by commas. The current list of valid handhelds is:

```
Pilot, Pilot1000, Pilot5000, PalmPilot,  
PalmPilotPersonal, PalmPilotProfessional,  
PalmIII, PalmIIIC, PalmIIIE, PalmIIIX, PalmV,  
PalmVx, PalmVII, PalmVIIIEZ, PalmVIIx, PalmM100,  
m100, PalmM105, m105, PalmM125, m125, PalmM130,  
m130, PalmM500, m500, PalmM505, m505, PalmM515,  
m515, PalmI705, i705, Symbol1500, Symbol1700,
```

Using Skin Files

Using Simulator Skin Files

Symbol11740, TRGpro, HandEra330, Visor,
VisorPlatinum, VisorPrism, VisorEdge

Examples:

Devices = Pilot1000, Pilot5000

Devices = PalmIIIc

Element#

A class of attributes that describes the layout of the image. There is one attribute for each item in the image that can be clicked on. There are also attributes for the LCD and touchscreen areas.

The value for each attribute is a list of 5 items: the name of the element and its coordinates on the screen. The current set of valid element names is:

PowerButton

The hardware on/off button.

UpButton

The hardware scroll up button.

DownButton

The hardware scroll down button.

App1Button

The first application button (usually the Date Book application button).

App2Button

The second application button (usually the Address Book application button).

App3Button

The third application button (usually the To Do List application button).

App4Button

The fourth application button (usually the Memo Pad application button).

CradleButton

The HotSync[®] operation button.

Antenna

The trigger for raising the antenna.

ContrastButton

The hardware dial for setting screen contrast.

Touchscreen

The full screen area, including the area.

LCD

The application screen area, excluding the area.

Symbol-specific Values:

TriggerLeft

TriggerCenter

TriggerRight

UpButtonLeft

UpButtonRight

DownButtonLeft

DownButtonRight

All elements except for Touchscreen and LCD are optional.

The coordinates of each element are provided by specifying the left coordinate, the top coordinate, the element width, and the element height. Only single-scale coordinates can be provided; double-scale coordinates are derived from these. Coordinate values can be specified in hexadecimal or decimal.

Each attribute name must start with the text “Element”, and must be suffixed with characters that make it unique from all the other element-related attributes.

[Listing 3.1](#) shows an example of a skin file.

Listing 3.1 Example of a Skin File

```
# This is a skin file for Palm OS Cobalt Simulator. See the ReadMe.txt
# file in this directory for a description of its contents.
```

```
Name                = Standard-English
File1x               = Palm_III_16.bmp
BackgroundColor      = 0x7B, 0x8C, 0x5A
HighlightColor       = 0x64, 0xF0, 0xDC
Devices              = PalmIII
```

```
#                                x    y    w    h
```

Using Skin Files

Using Simulator Skin Files

#		-----	-----	-----	-----
Element1	= PowerButton,	10,	295,	16,	24
Element2	= UpButton,	110,	292,	20,	21
Element3	= DownButton,	110,	313,	20,	21
Element4	= App1Button,	37,	295,	23,	25
Element5	= App2Button,	76,	297,	23,	25
Element6	= App3Button,	141,	297,	23,	25
Element7	= App4Button,	180,	294,	23,	25
Element11	= Touchscreen,	39,	44,	160,	220
Element12	= LCD,	39,	44,	160,	160

Using Palm OS Reporter

This chapter describes Palm OS Reporter, which you can use to do trace analysis of your Palm OS® applications. The following topics are covered in this chapter:

- “[About Palm OS Reporter](#)” - An introduction to Palm OS Reporter concepts
- “[Installing Palm OS Reporter](#)” on page 34 - How to download and install the Palm OS Reporter package
- “[Adding Trace Calls to Your Application](#)” on page 35 - How to add Trace Manager calls to your ARM application or Host Control trace calls to your 68K application.
- “[Displaying Trace Information in Palm OS Reporter](#)” on page 39 - How to open a Palm OS Reporter session to view the trace information
- “[Troubleshooting Palm OS Reporter](#)” on page 43 - How to make sure Palm OS Reporter is running correctly

About Palm OS Reporter

Palm OS Reporter is a trace utility that can be used with Palm OS Cobalt Simulator. As an application runs on Palm OS Cobalt Simulator, it can send information in real time to Reporter. This information can help pinpoint problems that might be hard to identify when executing code step-by-step or when specifying breakpoints. To view the realtime traces, simply start Reporter before you run Palm OS Cobalt Simulator.

Palm OS Reporter Features

Palm OS Reporter has a number of features that make it useful:

- High throughput of trace output, allowing for realtime traces
- Trace output filtering, searching, saving, printing, and copying
- Display of Trace output through a TCP/IP connection

Installing Palm OS Reporter

Palm OS Reporter is a component of the Palm OS Cobalt developer tools package. When you install the developer tools package, you install Palm OS Reporter.

To check for product fixes or other between-release updates, check

<http://www.palmos.com/dev/tools>

Palm OS Reporter Package Files

The Palm OS Reporter package includes the following files:

Table 4.1 Files included in the Palm OS Reporter package

File	Description
Reporter.exe	Main Palm OS Reporter program file
PalmTrace.dll	Palm OS Cobalt Simulator add-on that relays traces to Palm OS Reporter

Palm OS Reporter requires either Palm OS Cobalt Simulator or Palm OS Emulator. When you installed Palm OS Reporter, the installer placed the PalmTrace library (PalmTrace.dll) in the same folder as the Palm OS Cobalt Simulator executable. Palm OS Cobalt Simulator is not able to send trace information to Reporter if it cannot find and load the PalmTrace library.

The Palm OS Reporter executable can be located in any folder on your system; it does not need to be in the same folder as Palm OS Cobalt Simulator.

Adding Trace Calls to Your Application

Traces are generated by system calls that are recognized by Palm OS Cobalt Simulator but ignored by actual handheld devices.

Tracing ARM-Based Applications

For ARM-based applications recompiled to run on Windows for Palm OS Cobalt Simulator, you use the system calls listed in `TraceMgr.h`, which is part of the Palm OS SDK. For more information about the Trace Manager API, see the book *Exploring Palm OS: System Management*.

The Trace Manager system calls pertinent to tracing are listed in the following table:

System Call Format	Function Description
<code>void TmOutputT(status_t mod, const char* fmt, ...)</code>	Output a string to Reporter (<code>printf</code> format).
<code>void TmOutputTL(status_t mod, const char* fmt, ...)</code>	Output a string to Reporter (<code>printf</code> format) with an additional line break.
<code>void TmOutputB(status_t mod, const void* buff, long len)</code>	Send binary data to Reporter.
<code>void TmOutputVT(status_t mod, const char* fmt, va_list vars)</code>	Output a string to Reporter (<code>vprintf</code> format).
<code>void TmOutputVTL(status_t mod, const char* fmt, va_list vars)</code>	Output a string to Reporter (<code>vprintf</code> format) with an additional line break.

Using Palm OS Reporter

Adding Trace Calls to Your Application

Trace Manager Functions in a Code Sample

```
void function(void)
{
    unsigned char theBuffer[256];
    unsigned long theUInt32 = 0xFEDC1234;
    unsigned short theUInt16 = 0xFE12;
    int i;

    TmOutputTL(appErrorClass, "This is an Int32:");
    TmOutputTL(appErrorClass, " unsigned (lu) [4275835444]=[%lu]", theUInt32);
    TmOutputTL(appErrorClass, " signed (ld) [-19131852]=[%ld]", theUInt32);
    TmOutputTL(appErrorClass, " hexa (lx) [fedc1234]=[%lx]", theUInt32);

    TmOutputTL(appErrorClass, "This is an Int16:");
    TmOutputTL(appErrorClass, " unsigned (hu) [65042]=[%hu]", theUInt16);
    TmOutputTL(appErrorClass, " signed (hd) [-494]=[%hd]", theUInt16);
    TmOutputTL(appErrorClass, " hexa (hX) [FE12]=[%hX]", theUInt16);

    TmOutputTL(appErrorClass, "This is a string (s) [Hello world]=[%s]", "Hello world");
    TmOutputTL(appErrorClass, "This is a char (c) [A]=[%c]", 'A');

    TmOutputTL(appErrorClass, "This is a buffer:");

    for (i = 0 ; i < 256 ; i++) theBuffer[i] = (unsigned char) i;

    TmOutputB(appErrorClass, theBuffer, 256);

}
```

Tracing 68K-Based Applications

For 68K-based applications, you use the system calls listed in `hostcontrol.h`, which is part of the Palm OS SDK. For more information about the Host Control API, see the book *Exploring Palm OS: System Management*.

The Host Control system calls pertinent to tracing are listed in the following table:

System Call Format	Function Description
<code>void HostTraceInit(void)</code>	Initiate a connection to Reporter
<code>void HostTraceOutputT(UInt16 mod, const char* fmt, ...)</code>	Output a string to Reporter (<code>printf</code> format)
<code>void HostTraceOutputTL(UInt16 mod, const char* fmt, ...)</code>	Output a string to Reporter (<code>printf</code> format) with an additional line break
<code>void HostTraceOutputB(UInt16 mod, const char* buff, UInt32 len)</code>	Send binary data to Reporter
<code>void HostTraceOutputVT(UInt16 mod, const char* fmt, va_list vars)</code>	Output a string to Reporter (<code>vprintf</code> format)
<code>void HostTraceOutputVTL(UInt16 mod, const char* fmt, va_list vars)</code>	Output a string to Reporter (<code>vprintf</code> format) with an additional line break
<code>void HostTraceClose(void)</code>	Close the connection to Reporter

All `HostTraceOutput` functions take an error class identifier as their first parameter. This parameter allows filtering of traces according to their origin. Recognized error classes are listed in `SystemMgr.h`. For example, applications should specify the error class `appErrorClass`.

Host Control Functions in a Code Sample

```
void function(void)
{
    unsigned char theBuffer[256];
    unsigned long theUInt32 = 0xFEDC1234;
    unsigned short theUInt16 = 0xFE12;
    int i;

    HostTraceInit();

    HostTraceOutputTL(appErrorClass, "This is an Int32:");
```

Using Palm OS Reporter

Adding Trace Calls to Your Application

```
HostTraceOutputTL(appErrorClass, " unsigned (lu) [4275835444]=[%lu]",
theUInt32);
HostTraceOutputTL(appErrorClass, " signed (ld) [-19131852]=[%ld]", theUInt32);
HostTraceOutputTL(appErrorClass, " hexa (lx) [fedc1234]=[%lx]", theUInt32);

HostTraceOutputTL(appErrorClass, "This is an Int16:");
HostTraceOutputTL(appErrorClass, " unsigned (hu) [65042]=[%hu]", theUInt16);
HostTraceOutputTL(appErrorClass, " signed (hd) [-494]=[%hd]", theUInt16);
HostTraceOutputTL(appErrorClass, " hexa (hX) [FE12]=[%hX]", theUInt16);

HostTraceOutputTL(appErrorClass, "This is a string (s) [Hello world]=[%s]",
"Hello world");
HostTraceOutputTL(appErrorClass, "This is a char (c) [A]=[%c]", 'A');

HostTraceOutputTL(appErrorClass, "This is a buffer:");

for (i = 0 ; i < 256 ; i++) theBuffer[i] = (unsigned char) i;

HostTraceOutputB(appErrorClass, theBuffer, 256);

HostTraceClose();
}
```

Specifying Trace Strings

Trace strings use the following format:

% <flags> <width> <type>

<flags>

-

Left-justify display (Default is right justify).

+

Always display the sign character (Default is to display the sign character for negative values only).

space

Display a space (when a value is positive) rather than displaying a "+" sign.

#

Alternate form specifier.

<width>

Must be a positive number.

<type>	
%	Display a “%” character
s	Display a null-terminated string
c	Display a character
ld	Display an Int32 value
lu	Display a UInt32 value
lX or lx	Display an Int32 or UInt32 value in hexadecimal
hd	Display an Int16 value
hu	Display a UInt16 value
hX or hx	Display an Int16 or UInt16 value in hexadecimal

NOTE: The following types are not supported for <type>: o, e, E, f, F, g, G, p, l, n, d, i, u, X, or x.

Displaying Trace Information in Palm OS Reporter

To view trace information in Palm OS Reporter, you need to do the following:

- Add trace calls to your application and build your application.
- Start a Palm OS Reporter session.
- Start a Palm OS Cobalt Simulator session.

Using Palm OS Reporter

Displaying Trace Information in Palm OS Reporter

- Redirect the Simulator's Socket API calls to the host machine's TCP/IP stack. Select the Simulator menu **Settings > [Networking > Winsock-Based SocketLib Replacement](#)**.
- Install your trace-enabled application in the Simulator session.
- Run your trace-enabled application in the Simulator session.

Starting a Palm OS Reporter Session

To start a Palm OS Reporter session, run the `Reporter.exe` file. After starting Palm OS Reporter, you should see an empty window. This window serves as a container for other windows which display the trace information.

Testing ARM-Based Applications with Palm OS Reporter

Each `TmOutput` call sends information into the current trace window. The `TmOutput` call is ignored if there is no active trace window.

Testing 68K-Based Applications with Palm OS Reporter

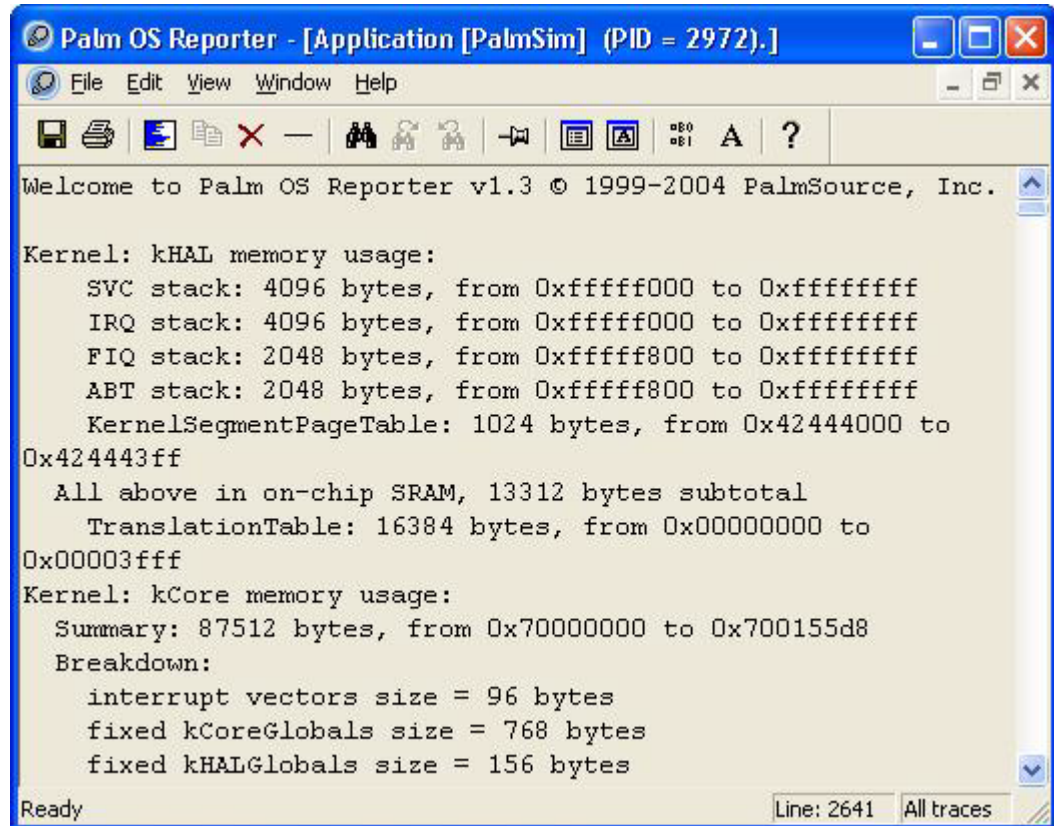
A new trace window is created for each `HostTraceInit` to `HostTraceClose` sequence in your trace-enabled application.

Each `HostTraceOutput` call sends information into the current trace window. The `HostTraceOutput` call fails if there is no active trace window, which can happen if Reporter is not running when the `HostTraceInit` function is called.

Also, a reset in Simulator closes any pending connection. That is, Simulator calls the `HostTraceClose` function for your application if you used `HostTraceInit` to open a trace connection.

[Figure 4.1](#) shows a Palm OS Reporter session window.

Figure 4.1 Palm OS Reporter session window



Filtering Information in a Palm OS Reporter Session















You can control the type of trace information Palm OS Reporter displays. You control this information by setting *filters*. Filters can be set either globally, by using the **Global filters...** menu, or for the current window, by using the **Active view filters...** menu. By enabling or disabling the filters, you can choose to view traces sent by corresponding modules in your application. Global filter settings are saved when you exit the Palm OS Reporter session.

Using Palm OS Reporter

Displaying Trace Information in Palm OS Reporter

Using the Palm OS Reporter Toolbar

Palm OS Reporter provides a toolbar with the following functions:

Toolbar Icon	Function
	Save the contents of the Reporter window to a file
	Print the contents of the Reporter window
	Select all of the text in the Reporter window
	Copy the selected text into the system clipboard
	Clear the contents of the Reporter window
	Draw a horizontal line across the Reporter window
	Search the Reporter window for specified text
	Search the Reporter window for the next occurrence of specified text
	Search the Reporter window for the previous occurrence of specified text
	Set “on top” mode to keep the Reporter window always visible on the screen
	Set filters for the current window only
	Set font for the current window only
	Set filters for all new windows
	Set font for all new windows

Troubleshooting Palm OS Reporter

Table 4.2 How to solve possible Palm OS Reporter problems

Symptom	Solution
The PalmTrace library (PalmTrace.dll) doesn't appear in the folder where you decompressed the Reporter's archive.	Check to see if your system is configured to "Hide system files."
Nothing appears in the Palm OS Reporter session window.	<p>Make sure that:</p> <ul style="list-style-type: none"> • The PalmTrace library (PalmTrace.dll) is in the same folder as the Palm OS Cobalt Simulator executable. • Your application code is calling HostTraceInit, if the application is a 68K-application. • Your filters are set correctly, and traces are emitted with the right modules.

Table 4.3 Palm OS Reporter error messages

Error Message	Problem	Possible Solution
An error occurred while trying to listen for traces.	Default reception port is already in use.	Check that no other instance of the Reporter is running.
An error occurred while ObjectSet was initializing TCP/IP.	TCP/IP related failure.	Check that TCP/IP networking is correctly set up.
Unable to start a reader thread.	Reporter could not create receiver thread.	Free up system resources.
Unable to start a format thread.	Reporter could not create displayer thread.	Free up system resources.

Palm OS Cobalt Simulator User Interface Reference

This chapter provides a reference for the Palm OS® Cobalt Simulator user interface elements.

Menu Reference Summary

The Palm OS Cobalt Simulator menus include:

- “[Reset Menu](#)” on page 46
- “[View Menu](#)” on page 47
- “[Install Menu](#)” on page 50
- “[Settings Menu](#)” on page 51
- “[Storage Menu](#)” on page 59
- “[Screenshot Menu](#)” on page 62
- “[About Menu](#)” on page 62
- “[Exit Menu](#)” on page 62

Displaying the Palm OS Cobalt Simulator Menu Items

Right-click (use mouse button 2) on the Palm OS Cobalt Simulator screen to display the menu items.

Figure 5.1 Palm OS Cobalt Simulator menu items



Reset Menu

Use the **Reset** menu to perform a reset of the current simulation session.

Soft

Performs a soft reset of the current simulation session. This is equivalent to pressing the reset button on the back of a handheld.

Hard

Performs a hard reset of the current simulation session. A hard reset erases all data in the simulation session, restoring it to the equivalent of a new handheld.

No Notify

Performs a “no notify” reset of the current simulation session. A “no notify” reset is a soft reset that does not send the `sysAppLaunchCmdSystemReset` notification to applications. This is equivalent to holding down the “Page Up” key while pressing the reset button on the back of a handheld.

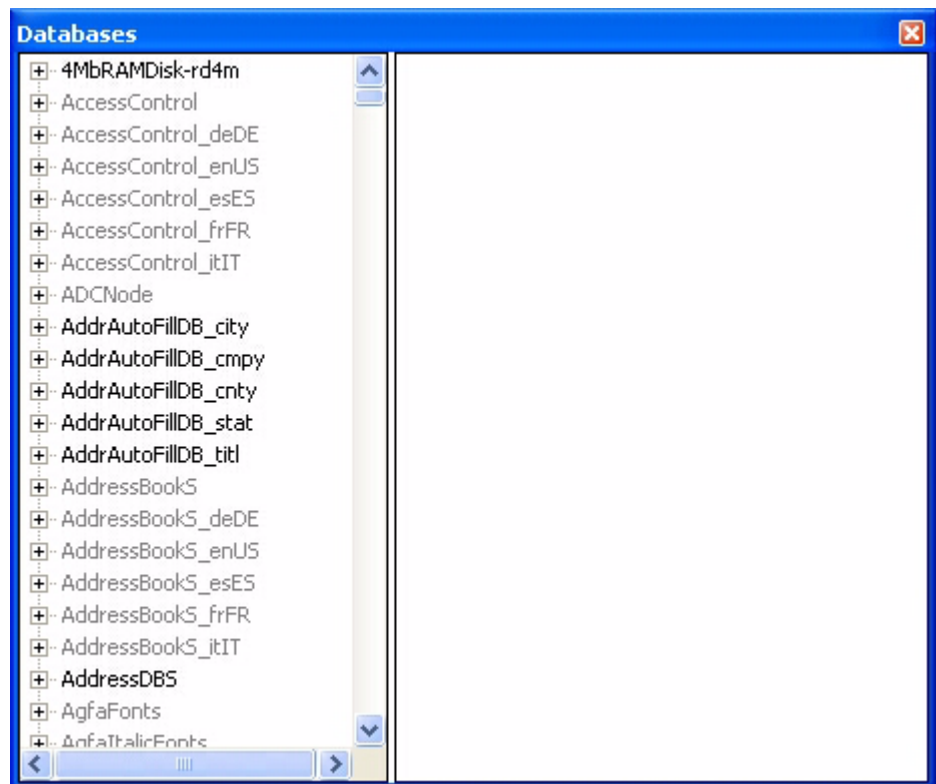
View Menu

Use the **View** menu to view information about the installed databases, the contents of the heap, or the user interface events.

Databases

Opens the Databases dialog box, shown in [Figure 5.2](#), which lists the databases included in the current simulator session.

Figure 5.2 View > Databases dialog box



Using the Databases Dialog Box

- Click on the plus sign icons to view the details of a database.
- Click on the minus sign icons to close the details view of a database.
- For databases that have contents, click on a record to view the record's contents in the right view pane.

- Right-click a database (use mouse button 2) to open the Export Database dialog box and save the contents of the database.

NOTE: The dimmed databases in the Databases dialog box are the read-only databases.

Heaps

Opens the Heaps dialog box, shown in [Figure 5.3](#), which displays the contents of the heaps for the current simulator session.

Figure 5.3 View > Heaps dialog box

Heaps						
Process ID#	Description					
0	System					
1	DataMgr					
2	IOS					
3	Persistent					
4	App					
5	App					
Heap #	Type	VA span	Mapped	Free chunks		
0	Dynamic (RW)	48720000-4931FFFF (12 MB)	12 MB	0 chunk (0 byte)		
1	Legacy (RW)	4670111C-486FFFFF (31 MB)	56 KB	1 chunk (4 MB)		
2	ROM (RO)	01010314-0166FFFF (6 MB)	6 MB	0 chunk (0 byte)		
3	Storage (RO)	42701108-446FFFFF (31 MB)	31 MB	0 chunk (0 byte)		
Chunk #	Ptr	MemHandle	Lock count	Size	Data Size	Owner
0	0x48b20008	No handle	Non-movable	6 KB	6 KB	15
1	0x48b218b8	No handle	Non-movable	344 bytes	336 bytes	15
2	0x48b21a18	No handle	Non-movable	1 KB	1 KB	1
3	0x48b21e78	No handle	Non-movable	16 bytes	8 bytes	1
4	0x48b21e90	No handle	Non-movable	32 bytes	24 bytes	1
<div> <div>00000000 5010B248 A014</div> <div>P. 2H .</div> </div>						

Using the Heaps Dialog Box

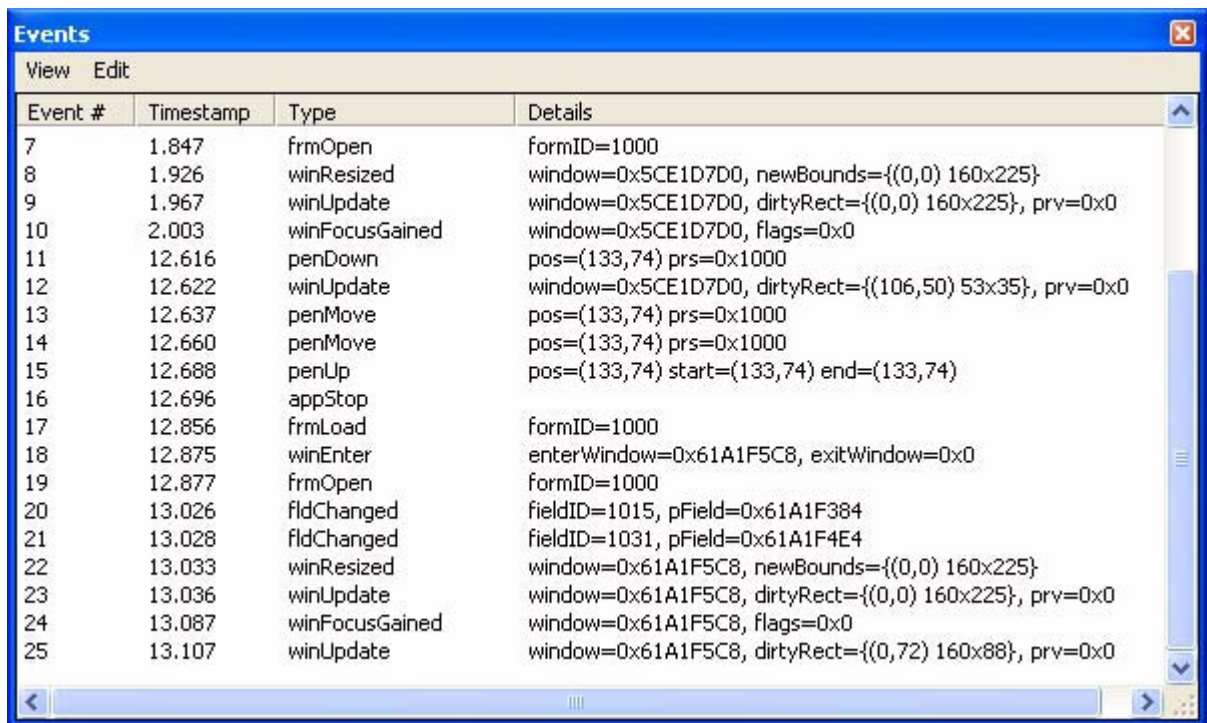
- Click on a process ID number to show the heaps in that process.
- Click on a heap number to show the chunks in that heap.

- Click on a chunk number to view the contents of a chunk in the bottom view pane.

Events

Opens the Events dialog box, shown in [Figure 5.4](#), which displays the user interface events for the current simulator session.

Figure 5.4 View > Events dialog box



Using the Events Dialog Box

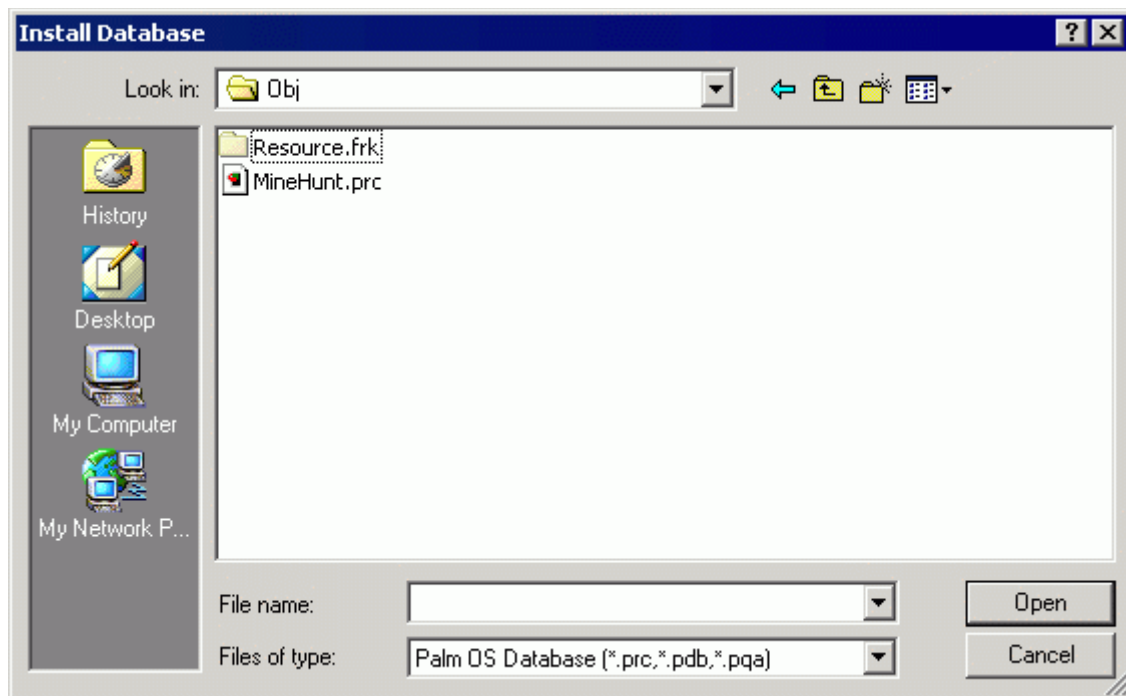
- Use the **View > Lock** menu to keep the events information from being updated.
- Use the **View > Unlock** menu to allow the events information to be updated.
- Use the **View > Filters** menu to select the type of events you want displayed in the event viewer.
- Use CTRL-A to select all the events information.

- Use the **Edit > Clear** menu to clear the events information that is displayed.
- Use the **Edit > Copy** menu or CTRL-C to copy the events information to your system clipboard.

Install Menu

Use the **Install > Database** menu to install PRC and PDB files. The **Install > Database** menu displays the Install Database dialog box, shown in [Figure 5.5](#).

Figure 5.5 Install Database dialog box



Using the Install Database Dialog Box

- To install a single database, select a PRC or PDB file, and click **Open**.
- To install multiple databases, use SHIFT-click to select multiple PRC and PDB files, and click **Open**.

Settings Menu

Use the **Settings** menu to change the settings for the current simulation session.

NOTE: If you select high values for display resolution and color depth, the Simulator session may run out of system memory and display an error dialog box. To prevent this problem, first increase the memory setting for the Simulator session by selecting **Settings > [Memory](#)**.

Display > Resolution

Set the screen resolution for this simulation session.

1.5x Density

- 240x240 108dpi - Select this option to simulate 1.5x density with a static input area.
- 240x320 108dpi (QVGA) - Select this option to simulate 1.5x density with a dynamic input area.

2x Density

- 320x320 144dpi - Select this option to simulate 2x density with a static input area.
- 320x350 144dpi - Select this option to simulate 2x density with a small screen dynamic input area.
- 320x480 144dpi (HVGA) - Select this option to simulate 2x density with a dynamic input area.

3x Density

- 408x640 216dpi (VGA) - Select this option to simulate 3x density.

NOTE: If you change the setting for **Resolution**, Palm OS Cobalt Simulator performs a soft reset.

Display > Orientations

Set the screen orientation and rotation for this simulation session.

Orientation

- Portrait Dominant - Select this option to simulate a portrait orientation.
- Landscape Dominant - Select this option to simulate a landscape orientation

NOTE: If you change the orientation selection, Palm OS Cobalt Simulator performs a soft reset.

Rotation

- No Hardware Rotation
- 90 Degree Rotation
- 180 Degree Rotation
- 270 Degree Rotation

NOTE: If you change the rotation selections, tap the screen rotation icon in the status bar to apply the rotation settings.

Display > Color Depth

- Set the color depth for this simulation session. 8 Bit (Grayscale)
- 32 Bit (Color)

NOTE: If you change the setting for **Color Depth**, Palm OS Cobalt Simulator performs a soft reset.

NOTE: For best results, set your PC monitor to display 16-bit color or higher.

Display > Renderer

Sets the graphics renderer simulation.

- **x86 (Fastest)** - Select this option to use the Windows-based renderer. This option is the default because it is the fastest option.
- **Emulated ARM** - Select this option to use an emulated version of the ARM renderer. This renderer uses the same ARM instructions that are used on ARM-based devices. This option has the slowest performance.
- **Reference** - Select this option to use the reference renderer. This option uses a simple generic rendering implementation that is designed to be correct rather than fast. However, in some cases, this reference renderer may not be identical to the ARM renderer.

NOTE: Independent of the Renderer setting, the Palm OS graphics subsystem may issue a warning message about the application not being reactive to updates while you are using Simulator to do step-by-step debugging. To disable this warning, set the variable `INHIBIT_UPDATE_TIMEOUT_WARNING=1`.

Display > Skin

Select **Settings > Display > Skin > Load** to specify a skin file for this simulation session.

Select **Settings > Display > Skin > Basic** to use the basic skin file.

Display > Magnification

Sets the scaling size for the simulation session display:

- 1:1 (This is the default selection.)
- 2:1
- 3:1
- 4:1

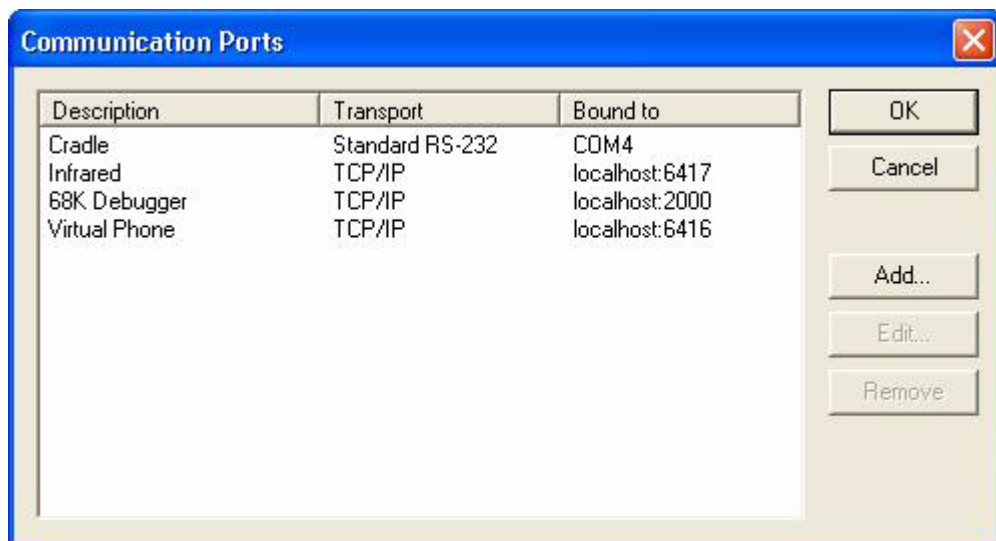
Display > Always on Top

Indicates whether the Simulator window should stay on top when you switch to other application windows.

Communication Ports

Displays the Communication Ports dialog box, shown in [Figure 5.6](#), so that you can change the communication port settings for the current simulation session. For more information on setting communication ports, see “[Using Communication Functions](#)” on page 23.

Figure 5.6 Communication ports dialog box



Using the Communication Ports Dialog Box

Cradle:

Use the **Cradle** selection to select the transport and port Palm OS Cobalt Simulator uses for Cradle communication.

The Cradle selection defines the port that Palm OS Cobalt Simulator uses to simulate a connection to a cradle or similar cable connection.

Infrared:

Use the **Infrared** selection to indicate which transport and port Palm OS Cobalt Simulator uses for infrared communication. For additional information, see “[Using Communication Functions](#)” on page 23.

68K Debugger :

Use the **68K Debugger** selection to indicate which transport and port Palm OS Cobalt Simulator uses for communication with a 68K-based debugging tool such as Palm Debugger, Palm OS Debugger, or the CodeWarrior Debugger. For additional information, see “[Using External Debug Tools with Palm OS Cobalt Simulator](#)” on page 24.

Virtual Phone:

Use the **Virtual Phone** selection to indicate which transport and port Palm OS Cobalt Simulator uses for communication with Virtual Phone.

Networking > Standard Palm OS TCP/IP

Indicates that Socket API calls use the standard Palm OS TCP/IP implementation.

NOTE: To use the host ethernet adapter, you must install the WinPcap library. To get the WinPcap library, visit this web site: <http://winpcap.polito.it>

Networking > Winsock-Based STREAMS Module

Indicates that Socket API calls use your PC's ethernet card to simulate an ethernet connection, using the Palm OS TCP/IP implementation.

If you have difficulty establishing communication between Palm OS Cobalt Simulator and your host PC, you may need to change your Windows ethernet driver properties. Using the System Property's Device Manager, open your Ethernet Controller's properties pages,

and disable the **RX Checksum Offload** and **TX Checksum Offload** features.

Networking > Winsock-Based SocketLib Replacement

Indicates that Socket API calls are redirected to the Windows TCP/IP API (Winsock) rather than the Palm OS implementation.

Dock Status

Indicates whether to simulate a connection with a hardware device such as a modem, cradle, or other peripheral.

- **Attached To Modem** - Indicates that the handheld is connected to a modem.
- **Attached To Dock** - Indicates that the handheld is connected, typically to a serial cradle.
- **Attached To USB Cradle** - Indicates that the handheld is connected to a USB cradle.
- **Attached To USB Peripheral** - Indicates that the handheld is connected to a USB peripheral (not a cradle).
- **Attached To Serial Peripheral** - Indicates that the handheld is connected to a serial peripheral (not a cradle).

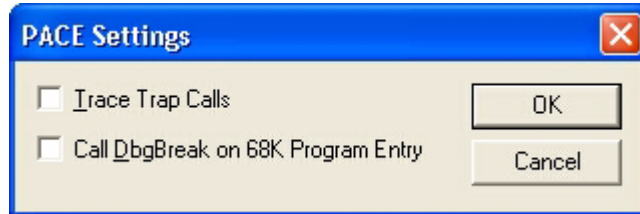
Memory

Sets the RAM size for this simulation session:

- 16 MB
- 32 MB
- 64 MB
- 128 MB
- 256 MB

NOTE: If you change the value of the RAM size setting, Palm OS Cobalt Simulator performs a hard reset.

Figure 5.7 PACE Settings dialog box

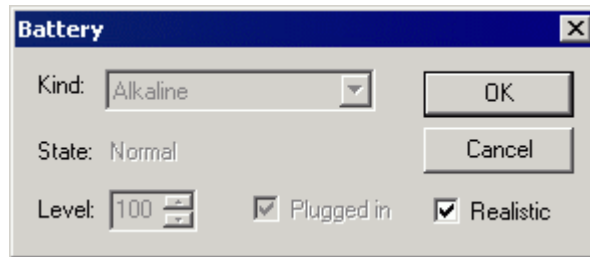


- Check **Call DbgBreak on 68K Program Entry** to enable an automatic `DbgBreak()` call when a 68K application is launched.

Battery

Displays the Battery dialog box, shown in [Figure 5.8](#) on page 57, so that you can change the simulated battery settings.

Figure 5.8 Battery dialog box



Using the Battery Dialog Box

- Select the **Kind** of battery you want to simulate from the supported list:
 - Alkaline: a hydrous alkaline (non-rechargeable) battery
 - NiCad: a rechargeable nickel cadmium battery
 - LiIon: a rechargeable lithium ion battery
 - RechAlk: a rechargeable alkaline battery
 - NiMH: a rechargeable nickel metal hydride battery
 - LiIon1400: a rechargeable lithium ion battery with a capacity of 1400 mAh (milliamperes hours)
- The battery **State** is dependent on the **Level** selected:

State	Level
Normal	100 to 21
Low	20 to 11
Critical	10 to 6
Shutdown	5 to 0

HotSync Name

Select **HotSync Name** to enter a user name for simulating a HotSync operation.

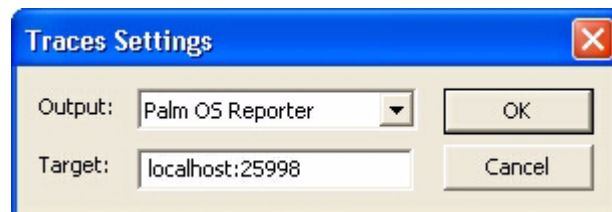
Enable Sound

Select **Enable Sound** to enable sound simulation for this simulation session.

Traces

Displays the Traces Settings dialog box, shown in [Figure 5.9](#).

Figure 5.9 Traces Settings dialog box



This dialog box allows you to redirect trace output to any of the following:

- None - Discards trace output.
- Palm OS Reporter - Send trace output to the Palm OS Reporter tool. For more information about Palm OS Reporter, see [Chapter 4, "Using Palm OS Reporter,"](#) on page 33.
- Text file - Writes trace output to the target file specified.
- stderr - Writes trace output to standard error.

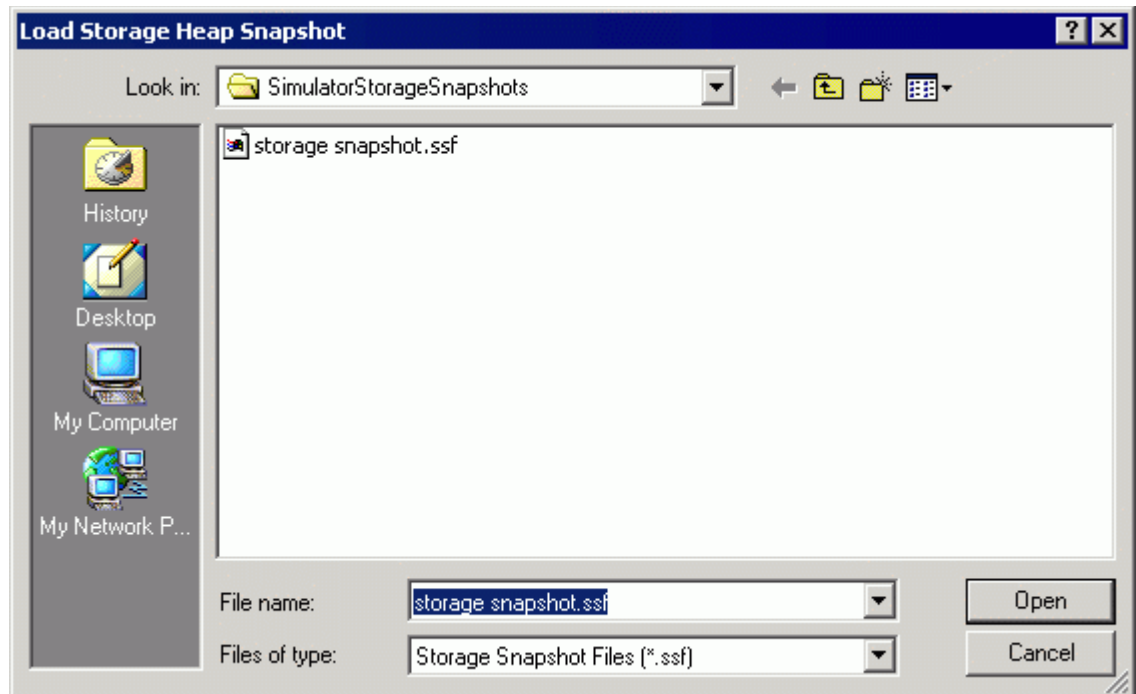
Storage Menu

Use the **Storage** menu to load and save a storage snapshot file (SSF file), to export database files from Simulator, or to mount a virtual file system (VFS) volume.

Load

Opens the Load Storage Heap Snapshot dialog box, shown in [Figure 5.10](#), so that you can load the storage heap with contents that were previously saved to a storage snapshot file (SSF file).

Figure 5.10 Load Storage Heap Snapshot dialog box

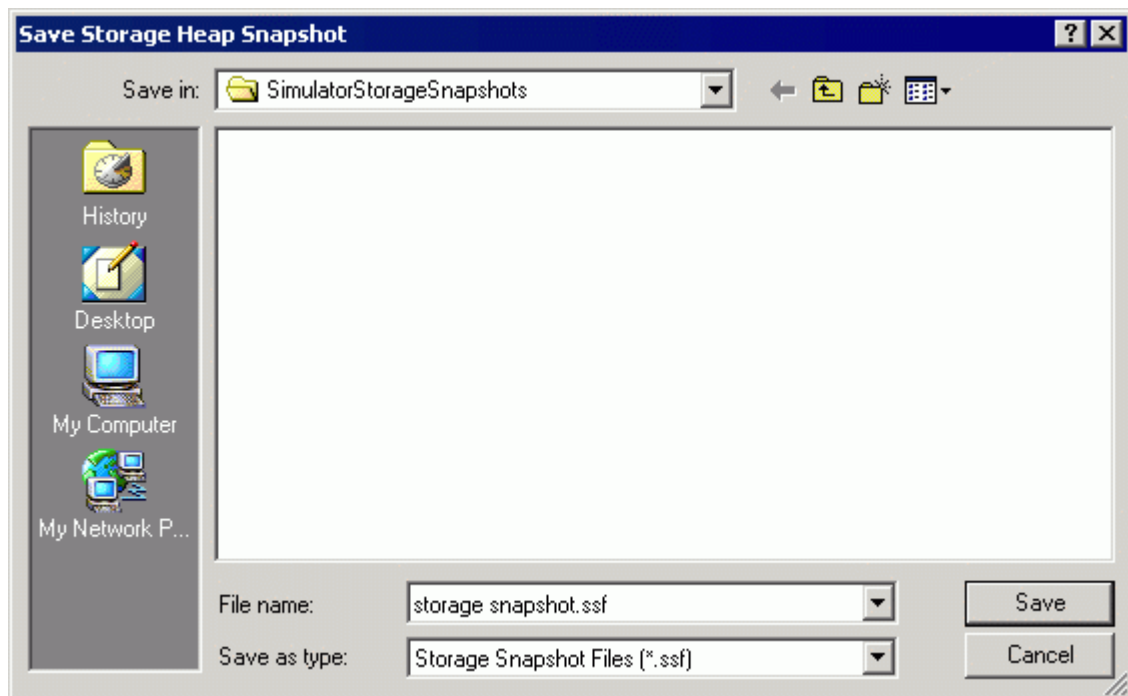


NOTE: You can load a storage snapshot file that has been created on the same PC. However, the storage snapshot file is not portable to another PC. Palm OS Simulator's method for calculating locations within memory is dependent on the PC's version of the Windows OS and on the PC machine configuration.

Save

Opens the Save Storage Heap Snapshot dialog box, shown in [Figure 5.11](#), so that you can save the current contents of the storage heap to a storage snapshot file (SSF file).

Figure 5.11 Save Storage Heap Snapshot dialog box



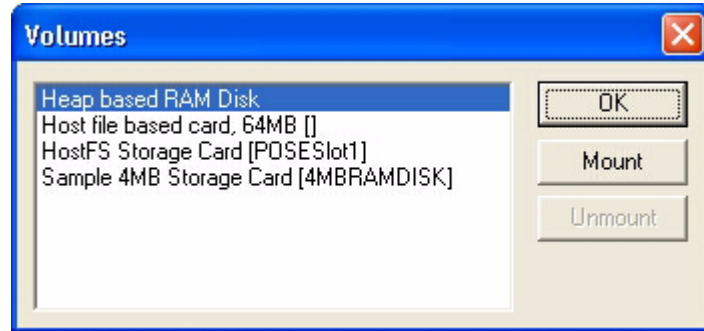
Export Database

Opens the Export Database dialog box, so that you can select the database that you want to export.

Volumes

Opens the Volumes dialog box, shown in [Figure 5.12](#), so that you can mount or unmount a virtual file system (VFS) volume.

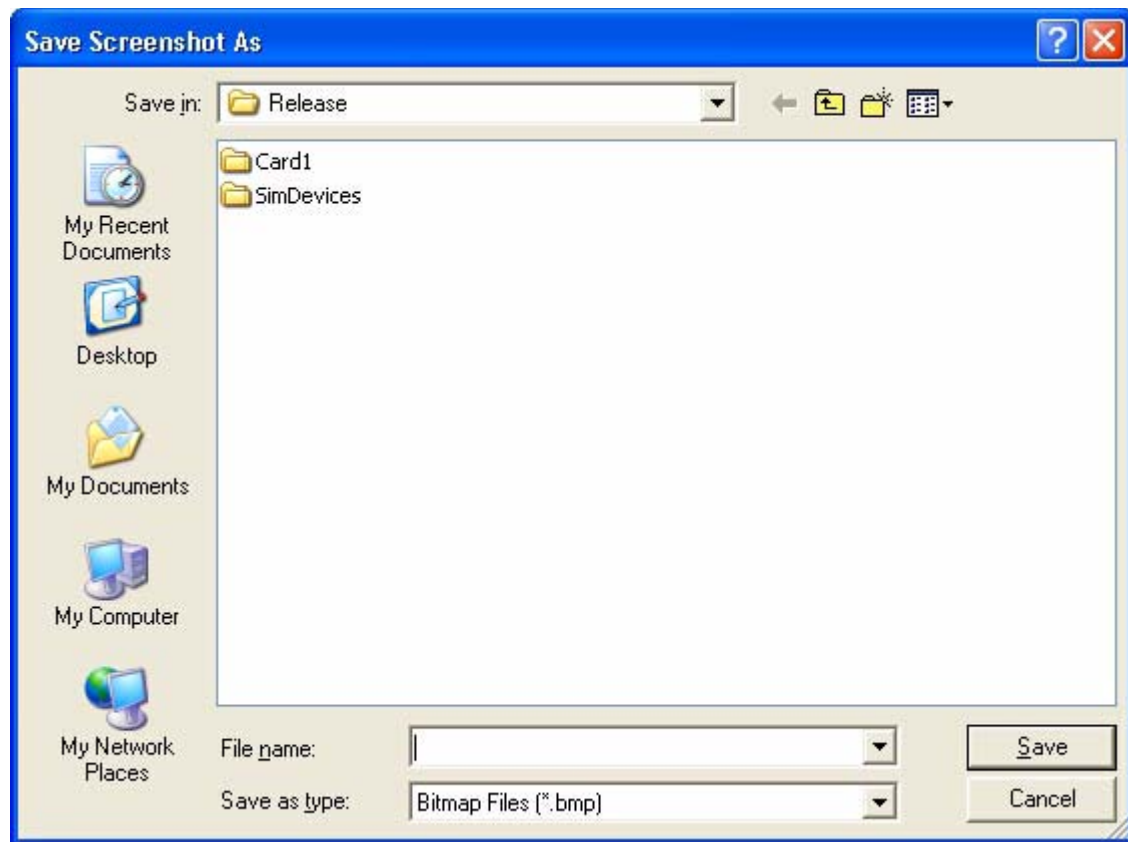
Figure 5.12 Volumes dialog box



Screenshot Menu

Opens the Save Screenshot As dialog box, shown in [Figure 5.13](#), so that you can save the current Simulator display as a bitmap image.

Figure 5.13 Save Screenshot As dialog box



About Menu

Use the **About** menu to get release information about Palm OS Cobalt Simulator.

Exit Menu

Use the **Exit** menu to exit Palm OS Cobalt Simulator.

When you exit Palm OS Cobalt Simulator, the current option values are written to the file `palmsim.ini` for the next time you start Simulator. For more information on using `palmsim.ini`, see [Chapter 2, “Using the Initialization File,”](#) on page 17.

Keyboard Equivalents Reference

Because it is more difficult to use the mouse in place of a handheld stylus, Palm OS Cobalt Simulator provides keyboard equivalents for many functions.

Hardware Buttons

The keyboard equivalents for handheld hardware buttons is similar to the key mapping recognized by Palm OS Emulator.

Table 5.1 Keyboard equivalents for hardware buttons

Hardware Button	Keyboard Equivalent
Power (on/off)	F5
Hardware button 1 (Date Book application)	F1
Hardware button 2 (Address Book application)	F2
Hardware button 3 (To Do List application)	F3
Hardware button 4 (Memo Pad application)	F4
Scroll up	PGUP
Scroll down	PGDN
HotSync Operation button	F6

Additional Keyboard Functions

In addition to hardware button equivalents, Palm OS Cobalt Simulator provides the following functions that can be invoked from the keyboard.

Table 5.2 Palm OS Cobalt Simulator keyboard functions

Function	Keyboard Equivalent
Display the pop-up menu.	CTRL + A
Enter the menu command stroke.	CTRL + C
Enter a confirmation character.	CTRL + D
Tap the Applications icon.	CTRL + E
Display the onscreen keyboard (Tap the “abc” in the input area).	CTRL + F
Tap the Find icon.	CTRL + I
Tap the Calculator icon.	CTRL + K
Enter a linefeed character.	CTRL + M
Tab to the next field.	CTRL + N
Tab to the previous field.	CTRL + P
Perform a soft reset.	CTRL + R
Perform a hard reset.	CTRL + SHIFT + R
Enter the 68K debugger.	CTRL + PAUSE
This function is applicable only when you are running a 68K application.	

Index

A

- about menu 62
- adding trace calls 35
- alkaline battery 57
- always on top menu 54
- AppCreator 22
- AutoLoad 21
- AutoRun 21
- AutoRunAndQuit 21

B

- battery
 - dialog box 57
 - menu 57
 - settings 57

C

- CodeWarrior Debugger 24
- command line arguments 9
- communication
 - functions 23
 - ports dialog box 54
 - ports menu 54
 - settings 56
 - stacks 3
- communication>communication ports menu 54
- communication>redirect socketlib calls to host
 - TCP/IP menu 56
- critical battery state 58

D

- databases 47
 - dialog box 47
- debugging 24, 64
- display>always on top menu 54
- display>magnification 53
- displaying trace information 40
- drag and drop 21
- driver set 3

E

- enable sound menu 58
- entering

- data 20
 - the debug tool 64
- events 49
 - dialog box 49
- EXE file 8
- exit menu 62
- external debug tools 24

F

- file
 - DLL 2
 - EXE 8
 - INI 17
 - PalmSim.exe 8
 - palmsim.ini 17
 - PDB 21
 - PRC 21
 - ROM 7, 15
 - SSF 15, 59
 - storage snapshot 15
- filtering trace information 41
- fixing reporter problems 43

G

- generic skin 27

H

- hard reset 46, 64
- hardware buttons 63
- heaps 48
 - dialog box 48
- host control
 - adding trace calls 35
 - using reporter 35
- host TCP/IP 56
- HotSync operation 24

I

- INI file 17
- initialization file 17
- install
 - database dialog box 50
 - menu 50
- install>database menu 50

- installing
 - applications 20
 - databases 50
 - Palm OS Cobalt Simulator 7
 - reporter 34

K

- keyboard
 - equivalents 63
 - functions 64
- kind of battery 57

L

- liion battery 57
- liion1400 battery 57
- load
 - menu 59
 - storage heap 59
 - storage heap snapshot dialog box 59
- loading StorageSnapshotFile 59
- low battery state 58

M

- magnification 53
- main window 9
- menu items 20
- menus 45
 - about 62
 - displaying 46
 - exit 62
 - reset 46
 - settings 51
 - storage 59
 - view 47
- Metrowerks CodeWarrior Debugger 24

N

- next field 64
- nicad battery 57
- nimh battery 57
- normal battery state 58

O

- overview 1

P

- PACE 3
- Palm Application Compatibility Environment 3
- Palm Debugger 24
- Palm OS
 - applications 2
 - communication stacks 3
 - runtime services 2
- Palm OS Cobalt Simulator 4
 - command line arguments 9
 - communication functions 23
 - components 2
 - debugging 24
 - display 19
 - entering data 20
 - initialization file 17
 - installing 7
 - installing applications 20
 - main window 9
 - menu items 20
 - overview 1
 - performing a HotSync operation 24
 - prerequisites 4
 - ROM images 19
 - running 19
 - starting 8
- Palm OS Reporter 33
- palmsim.ini 17
- PDB file 21
- performing
 - a hard reset 46
 - a soft reset 46
- PRC file 21
- previous field 64

R

- rechalk battery 57
- redirect SocketLib calls 56
- redirect SocketLib calls to host TCP/IP menu 56
- reporter 33
 - about 33

- adding trace calls 35
- features 34
- filtering information 41
- functions 42
- installation errors 43
- installing 34
- package file contents 34
- sample code 36, 37
- session window 41
- toolbar 42
- trace session 40
- trace strings 38
- troubleshooting 43
- using simulator 40

reset menu 46

reset>hard menu 46

reset>soft menu 46

ROM

- file 7, 15
- images 19

running Palm OS Cobalt Simulator 19

runtime services 2

S

save

- menu 60
- storage heap 60
- storage heap snapshot dialog box 60

saving StorageSnapshotFile 60

scaling size 53

setting

- the communication ports 54
- the scaling size 53
- the screen magnification 53

settings menu 51

settings>battery menu 57

settings>communication menu 54

settings>enable sound menu 58

showing

- heap contents 48
- installed databases 47
- user interface events 49

shutdown battery state 58

simulator

- using reporter 40

skins

- generic 27

Socket API calls 56

soft reset 46, 64

sound settings 58

SSF file 15, 59

starting Palm OS Cobalt simulator 8

stay on top 54

storage

- heap contents 59
- menu 59
- snapshot file 59

storage>load menu 59

storage>save menu 60

StorageSnapshotFile 22, 59

T

tabbing 64

TCP/IP communication 56

testing 68K applications 3

trace

- analysis 33
- strings 38

tracing

- applications 33
- sample code 36, 37

U

user interface events 49

using

- a mouse 20
- AppCreator 22
- AutoRun, AutoLoad, and AutoQuit 21
- drag and drop 21
- external debug tools 24
- reporter 33
- StorageSnapshotFile 22
- the initialization file 17

V

view menu 47

view>databases menu 47

view>events menu 49

view>heaps menu 48

viewing
 databases 47
 events 49
 heaps 48

W

window size 53