



Palm OS® Cobalt Simulator Guide

Palm OS Cobalt, Release 6.0

Written by Brian Maas and Denise Stone.

Technical assistance from Patrick Porlan, Madhavi Sunkara, David Navarro, Ezekiel Sanborn de Asis, Regis Nicolas, John Dance, Puneet Mishra, Asif Shabbir, Denis Berger, Dianne Hackborn, Ken Krugler, and Keith Rollin.

Copyright © 2003 - 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, the PalmSource logo, BeOS, Graffiti, HandFAX, HandMAIL, HandPHONE, HandSTAMP, HandWEB, HotSync, the HotSync logo, iMessenger, MultiMail, MyPalm, Palm, the Palm logo, the Palm trade dress, Palm Computing, Palm OS, Palm Powered, PalmConnect, PalmGear, PalmGlove, PalmModem, Palm Pack, PalmPak, PalmPix, PalmPower, PalmPrint, Palm.Net, Palm Reader, Palm Talk, Simply Palm and ThinAir are trademarks of PalmSource, Inc. or its 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 3129-001

July 7, 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
 2 Using Palm OS Cobalt Simulator	 5
Installing Palm OS Cobalt Simulator	5
Starting Palm OS Cobalt Simulator	6
Specifying Command Line Arguments	7
Using the Initialization File	14
Loading ROM Images	16
Running Palm OS Cobalt Simulator	17
Palm OS Cobalt Simulator Display	17
Displaying Menu Items	17
Entering Data	18
Installing Applications	18
Using AutoRun, AutoLoad, and AutoRunAndQuit	19
Using AppCreator to Start an Application	20
Starting Simulator with a Storage Snapshot File	20
Using Communication Functions	20
Testing Infrared Communication (Beaming)	20
Performing a HotSync Operation	21
Using External Debug Tools with Palm OS Cobalt Simulator	22
Using Palm OS Cobalt Simulator with 68K Debuggers	22

3 Using Palm Reporter	27
About Palm Reporter	27
Palm Reporter Features	28
Installing Palm Reporter	28
Palm Reporter Package Files	28
Installing Palm Reporter	28
Adding Trace Calls to Your Application	29
Tracing ARM-Based Applications	29
Trace Manager Functions in a Code Sample	30
Tracing 68K-Based Applications	30
Specifying Trace Strings	32
Displaying Trace Information in Palm Reporter	33
Starting a Palm Reporter Session	34
Filtering Information in a Palm Reporter Session	35
Using the Palm Reporter Toolbar	36
Troubleshooting Palm Reporter	37
 4 Palm OS Cobalt Simulator User Interface Reference	 39
Menu Reference Summary	39
Displaying the Palm OS Cobalt Simulator Menu Items	40
Reset Menu	40
Soft	40
Hard	40
No Notify	40
View Menu	41
Databases	41
Heaps	42
Events	43
Install Menu	44
Settings Menu	45
Display>Color Depth	45
Display>Magnification	45
Display>Always on Top	45
Display>Resolution	45
Communication>Communication Ports	46

Communication>Redirect SocketLib Calls to Host TCP/IP	47
Communication > Use Host Ethernet Adapter	47
Dock Status	48
Memory>RAM Size	48
Battery	48
HotSync Name.	49
Enable Sound	49
Storage Menu	50
Load	50
Save	51
About Menu	51
Exit Menu	51
Keyboard Equivalents Reference	52
Hardware Buttons	52
Additional Keyboard Functions	52

Index

55

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 6.0.1.

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 5 provides task information describing how to use Palm OS Cobalt Simulator functions.
- [Chapter 3, “Using Palm Reporter,”](#) on page 27 describes how you can use Palm Reporter with Palm OS Cobalt Simulator to do trace analysis of your application.
- [Chapter 4, “Palm OS Cobalt Simulator User Interface Reference,”](#) on page 39 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>

About This Book

Additional Resources

- 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® 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

What Is Palm OS Cobalt Simulator?

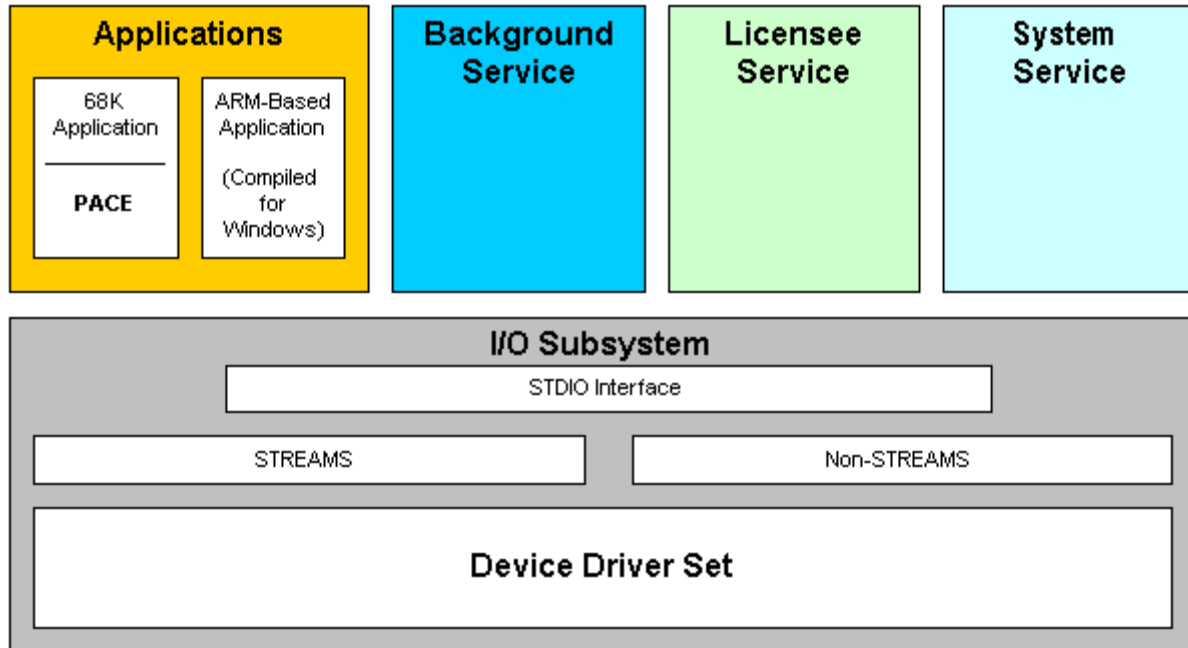
Palm OS Cobalt Simulator is Palm OS recompiled for a desktop machine processor. Palm OS Cobalt Simulator combines all of the Palm OS® 6 architecture components into a single execution environment.

- 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 18.

Palm OS System Code

Palm OS Cobalt Simulator includes all of the Palm OS 6 system code, compiled to run on Windows.

Palm OS Runtime Services

Palm OS Cobalt Simulator implements the Palm OS 6 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 SocketLib and Telephony components. Palm OS Cobalt Simulator can optionally redirect SocketLib 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 6. For example, 68K applications read and write data in big-endian mode, but Palm OS 6 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 6.

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 CodeWarrior, Visual C++, and Eclipse).

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 5
- [“Starting Palm OS Cobalt Simulator”](#) on page 6
- [“Specifying Command Line Arguments”](#) on page 7
- [“Using the Initialization File”](#) on page 14
- [“Loading ROM Images”](#) on page 16
- [“Running Palm OS Cobalt Simulator”](#) on page 17
- [“Using Communication Functions”](#) on page 20
- [“Using External Debug Tools with Palm OS Cobalt Simulator”](#) on page 22

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 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 Reporter, see [Chapter 3, "Using Palm Reporter,"](#) on page 27.

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 7.

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 8 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>Communication Ports” on page 46.</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>Communication Ports” on page 46.</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 style="text-align: center;"><i>(portName, portType, 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>Communication Ports” on page 46.</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 45.</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 20.</p>
<code>-autoload:[filename directory]</code>	<p><i>filename</i> - The PRC or PDB file that you want Palm OS Cobalt Simulator to automatically load.</p> <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 19.</p>
<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 AutoRunAndQuit, see “Using AutoRun, AutoLoad, and AutoRunAndQuit” on page 19.</p>

Table 2.1 Command line options (continued)

Option Syntax	Parameter Values
<code>-autorun: [<i>filename</i> <i>directory</i>]</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 AutoRun, see “Using AutoRun, AutoLoad, and AutoRunAndQuit” on page 19.</p>
<code>-bitdepth: [4 8 16]</code>	<p>Sets the color depth for the display setting. The default is 16.</p> <p>For more information, see “Display>Color Depth” on page 45.</p>
<code>-cradleport: COMx</code>	<p>COMx - The communications port used to talk to the FFUART port.</p> <p>For more information, see “Communication>Communication Ports” on page 46.</p>
<code>-dllpath: <i>directories</i></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 PALMSOURCE_SIM_PATH environment variable (described in “Installing Applications” on page 18).</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>

Using Palm OS Cobalt Simulator

Specifying Command Line Arguments

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-infraredport:[COMx None]</code>	<p><i>COMx</i> - The communications port used for infrared.</p> <p>For more information, see “Communication>Communication Ports” on page 46.</p>
<code>-infraredporttype: <i>type</i></code>	<p><i>type</i> - The type of port used for infrared communication.</p> <p>For more information, see “Communication>Communication Ports” on page 46.</p>
<code>-preferredLocale: <i>llCC</i></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>
<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 default is 32 MB.</p> <p>For more information, see “Memory>RAM Size” on page 48.</p>
<code>-redirectsocketlibcalls: [on <u>off</u>]</code>	<p>Indicates whether you want to redirect SocketLib calls to the host machine’s TCP/IP stack. The default is <code>off</code>.</p> <p>For more information, see “Communication>Redirect SocketLib Calls to Host TCP/IP” on page 47.</p>

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-rom:romname</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 <u>off</u>]</code>	<p>Activates or deactivates sound output. The default is <code>off</code>.</p> <p>For more information, see “Enable Sound” on page 49.</p>
<code>-storagesnapshotfile:ssfname</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 50.</p>
<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.</p>

Using Palm OS Cobalt Simulator

Using the Initialization File

Table 2.1 Command line options (*continued*)

Option Syntax	Parameter Values
<code>-usehostethernetadapter:</code> <code>[on off]</code>	Indicates whether Palm OS Cobalt Simulator can use the host ethernet adapter. The default is <code>off</code> . For more information, see “Communication > Use Host Ethernet Adapter” on page 47.
<code>-windoworiginx:integer</code>	<i>integer</i> - Specifies Simulator’s horizontal distance from the left of the screen when the window is opened. The default is 0.
<code>-windoworiginy:integer</code>	<i>integer</i> - Specifies Simulator’s vertical distance from the top of the screen when the window is opened. The default is 0.
<code>-zoom: [1 2 3 4]</code>	Magnification level. The default is 1. For more information, see “Display>Magnification” on page 45.

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 8.)

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.

Listing 2.1 Sample palmsim.ini File

```
[Settings]
ROM=C:\PalmOS6Simulator\SDK\ROM\Simulator_Full_EFIGS_Release.
rom
Sound=0
AlwaysOnTop=0
```

```
Zoom=1
BitDepth=16
RedirectSocketLibCalls=0
CradlePort=
CradlePortType=Standard RS-232
UseHostBatteryInfo=1
BatteryPluggedIn=1
BatteryPercent=75
BatteryKind=2
DockStatus=0
RAM=32768
LastSilkScreen=
InfraredPort=localhost:6417
InfraredPortType=TCP/IP
AppCreator=
68KDebuggerPort=localhost:2000
68KDebuggerPortType=TCP/IP
VirtualPhonePort=localhost:6416
VirtualPhonePortType=TCP/IP
AdditionalPorts=
LastStorageSnapshot=storage snapshot.ssf
StorageSnapshotFile=mynsnapshot.ssf
ScreenDensity=144
ScreenWidth=320
ScreenHeight=480
HasSilkScreen=0
TraceTarget=tcp:localhost:25998
ConsoleVisible=0
EventsVisible=0
ThreadsVisible=0
HeapsVisible=0
EventsFilter=
AutoStartReporter=0
UseHostEthernetAdapter=0
Window0Left=0
Window0Right=326
Window0Top=0
Window0Bottom=505
```

[Table 2.2](#) describes additional options that you can set in the initialization file. The options that you can also set on the command line are described in [Table 2.1](#) on page 8.

Using Palm OS Cobalt Simulator

Loading ROM Images

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`.

Table 2.2 Initialization file options

Option Syntax	Parameter Values
<code>CradlePortType=type</code>	<p><i>type</i> - The type of port used for communication with the FFUART port. The default is Standard RS-232.</p> <p>For more information on communication port settings, see “Communication>Communication Ports” on page 46.</p>
<code>UseHostBatteryInfo=[1 0]</code>	<p>Indicates whether the changes in the host machine should be indicated in the battery state of the handheld (for example, when you want to have the battery state of a laptop computer mapped to the Palm OS Cobalt Simulator display). The default is 0 for <code>off</code>.</p> <p>For more information, see “Battery” on page 48.</p>

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.

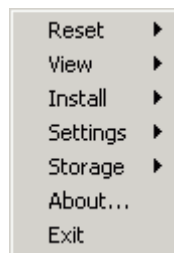
Palm OS Cobalt Simulator Display

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



Using Palm OS Cobalt Simulator

Running Palm OS Cobalt Simulator

For more information about the Palm OS Cobalt Simulator menu items, see “[Menu Reference Summary](#)” on page 39.

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 52.

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.

For information about the command line options, see [Table 2.1](#) on page 8.

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 7 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 7 for more information about command line arguments. For more information about storage snapshot files, see "[Storage Menu](#)" on page 50.

Using Communication Functions

Palm OS Cobalt Simulator supports the following communication functions:

- "[Testing Infrared Communication \(Beaming\)](#)" on page 20
- "[Performing a HotSync Operation](#)" on page 21

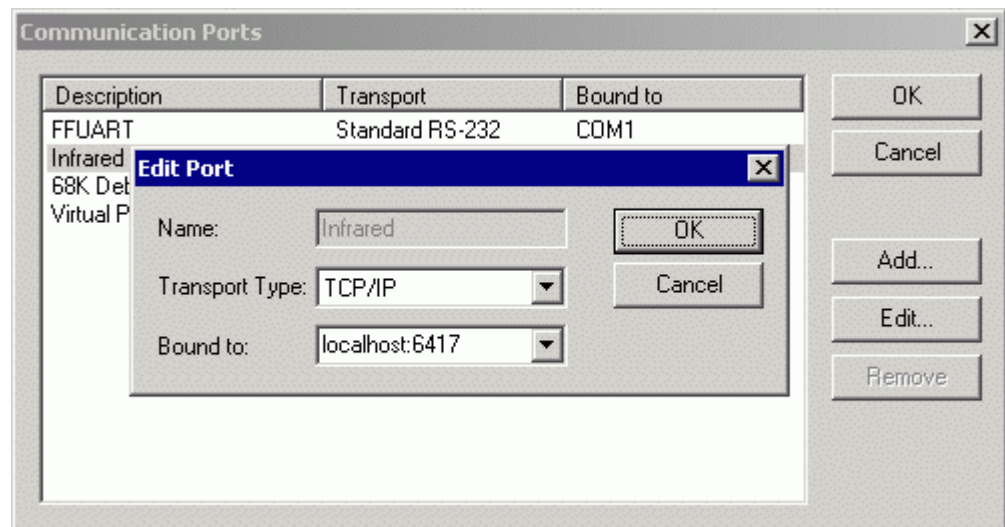
Testing Infrared Communication (Beaming)

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

1. Select [Communication>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 21.

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 must 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 > Communication Ports**, set the FFUART port to Standard RS232 bound to the other serial communication port.

Using Palm OS Cobalt Simulator

Using External Debug Tools with Palm OS Cobalt Simulator

HotSync Operation with One Serial Port

1. Select the Simulator menu **Settings>Communication>Redirect SocketLib Calls to Host TCP/IP** in order to redirect the SocketLib calls to the host machine's TCP/IP stack.
2. Using the Simulator menu **Settings>Communication>Communication Ports**, set the FFUART port **Transport Type** to TCP/IP and for **Bound to** enter `localhost:9`.
3. In the host computer's HotSync settings, enable **Network Hotsync**.
4. In Palm OS Cobalt Simulator, tap the HotSync application icon in the Launcher. Tap **Network** to select Network HotSync. Then tap the HotSync icon to start the HotSync operation with `localhost`.

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.

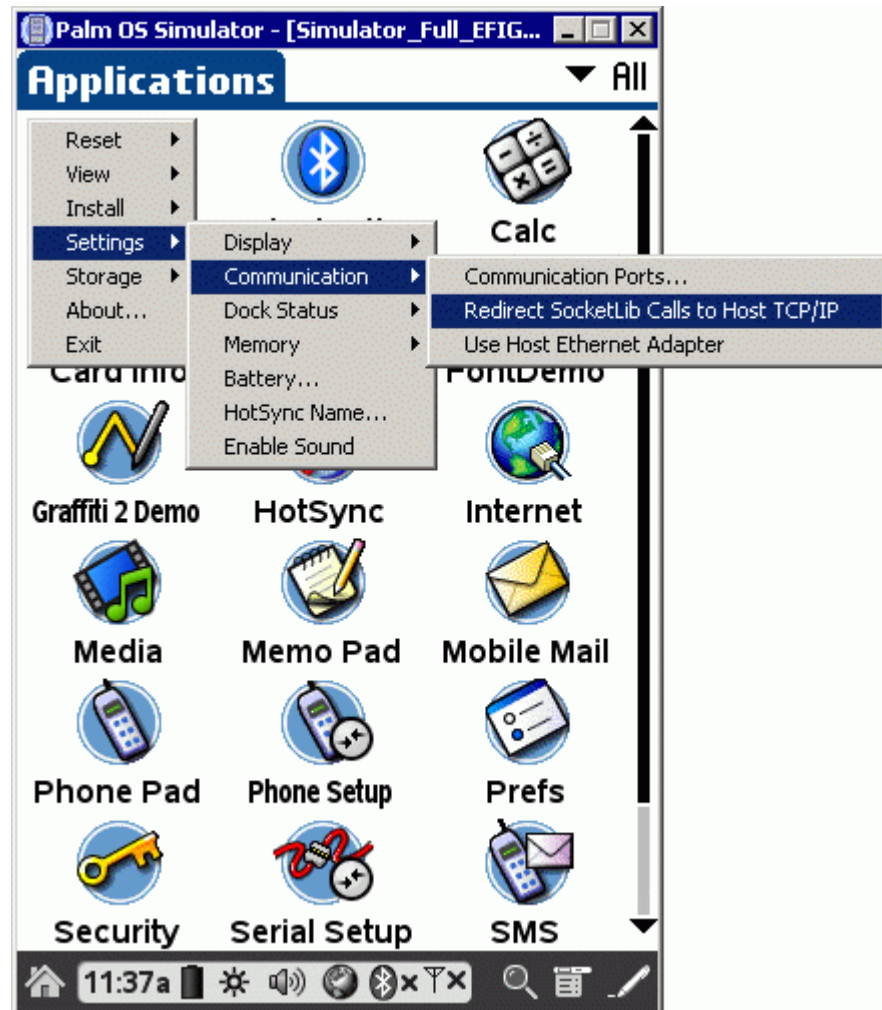
Using Metrowerks CodeWarrior Debugger

For example, you can use Palm OS Cobalt Simulator with Metrowerks CodeWarrior by following these steps:

- In Palm OS Cobalt Simulator, select **Settings>Communication>Redirect SocketLib Calls to Host TCP/IP**, as shown in [Figure 2.4](#) on page 23, in order to

redirect the SocketLib calls to the host machine's TCP/IP stack.

Figure 2.4 Redirect SocketLib Calls to Host TCP/IP menu

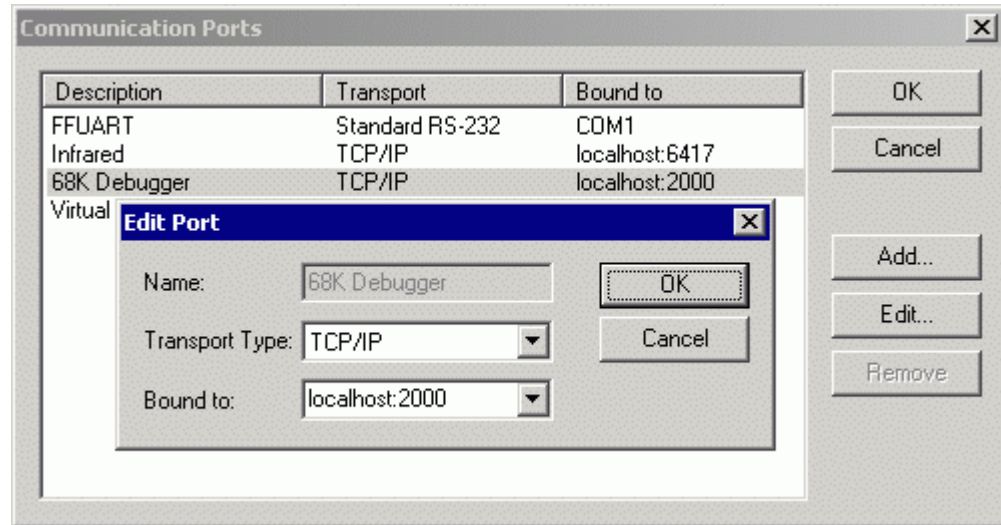


- Also in Palm OS Cobalt Simulator, select **Settings>Communication>Communication Ports** to bind the 68K debugger transport to `localhost:2000`, as shown in [Figure 2.5](#).

Using Palm OS Cobalt Simulator

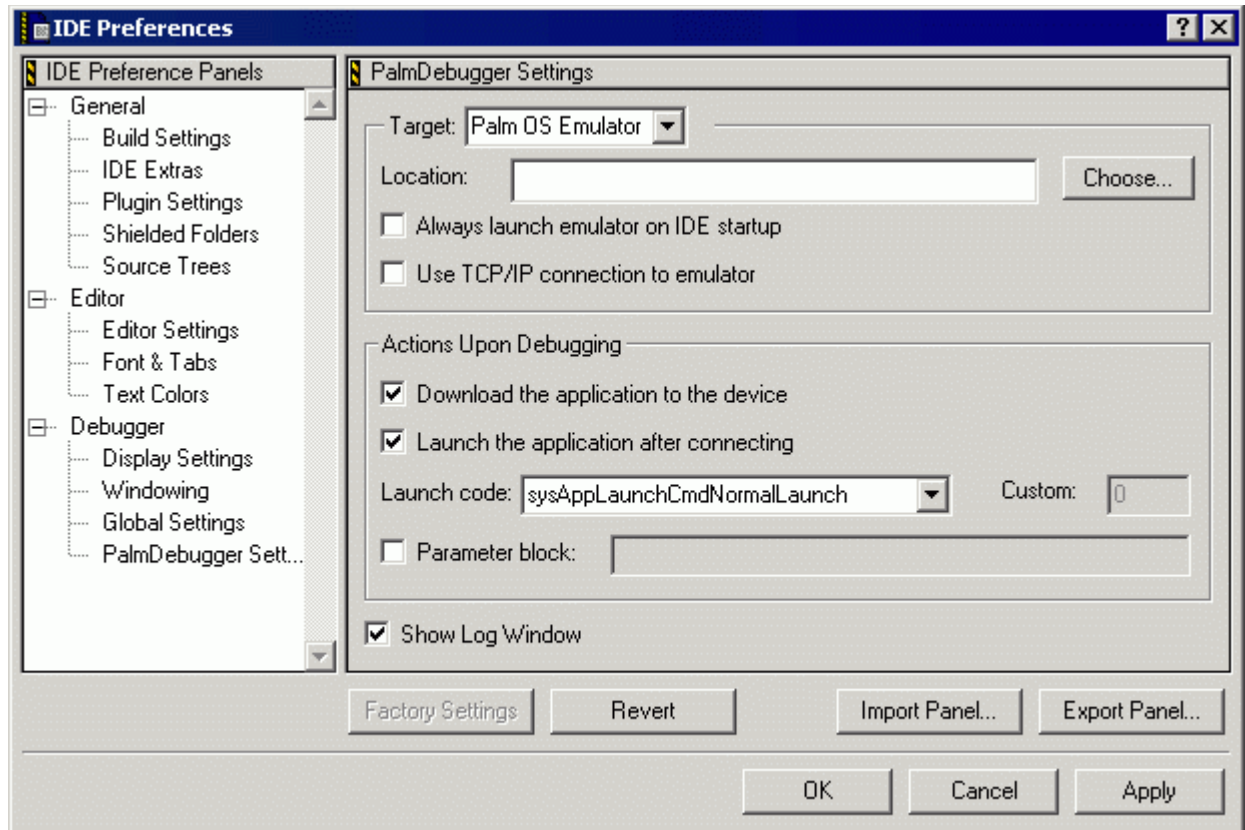
Using External Debug Tools with Palm OS Cobalt Simulator

Figure 2.5 Setting the 68K Debugger transport values



- In CodeWarrior, select **Edit>Preferences** to display the IDE Preferences dialog box, shown in [Figure 2.6](#) on page 25.

Figure 2.6 IDE Preferences dialog box

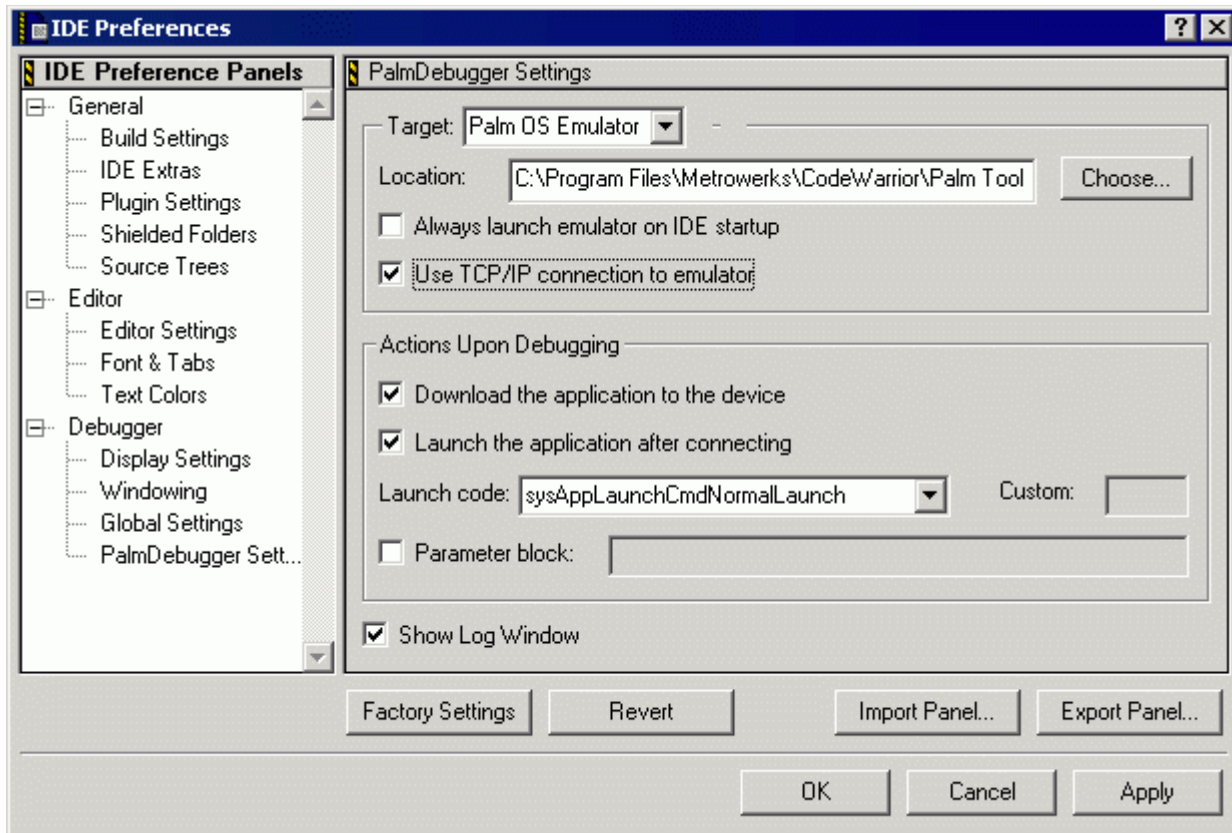


- Click **PalmDebugger Settings** in the IDE Preference Panels tree to display the PalmDebugger Settings panel.
- Set the Target selection to be Palm OS Emulator.
- Click **Choose** next to the Location entry field.
- Use the Choose File dialog box to select the PalmSim.exe executable.
- Select the setting **Use TCP/IP connection to emulator**.

Using Palm OS Cobalt Simulator

Using External Debug Tools with Palm OS Cobalt Simulator

Figure 2.7 Setting the CodeWarrior IDE Preferences



- Click **Apply** or **OK** to set your new values so that Palm OS Cobalt Simulator is used to run your application rather than Palm OS Emulator.

NOTE: Simulator supports Metrowerks CodeWarrior for Palm OS Version 8 and later.

Using Palm Reporter

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

- “[About Palm Reporter](#)” - An introduction to Palm Reporter concepts
- “[Installing Palm Reporter](#)” on page 28 - How to download and install the Palm Reporter package
- “[Adding Trace Calls to Your Application](#)” on page 29 - How to add Trace Manager calls to your ARM application or Host Control trace calls to your 68K application.
- “[Displaying Trace Information in Palm Reporter](#)” on page 33 - How to open a Palm Reporter session to view the trace information
- “[Troubleshooting Palm Reporter](#)” on page 37 - How to make sure Palm Reporter is running correctly

About Palm Reporter

Palm 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 your application on Palm OS Cobalt Simulator.

Palm Reporter Features

Palm 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 Reporter

Palm Reporter is a component of the Palm OS 6 developer tools package. When you install the developer tools package, you install Palm Reporter.

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

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

Palm Reporter Package Files

The Palm Reporter package includes the following files:

Table 3.1 Files included in the Palm Reporter package

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

Installing Palm Reporter

Palm Reporter requires either Palm OS Cobalt Simulator or Palm OS Emulator. When you installed Palm 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 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 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 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 Reporter

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

- Add trace calls to your application and build your application.
- Start a Palm Reporter session.
- Start a Palm OS Cobalt Simulator session.
 - Redirect the Simulator’s SocketLib calls to the host machine’s TCP/IP stack. Select the Simulator menu

Using Palm Reporter

Displaying Trace Information in Palm Reporter

Settings>Communications> Redirect SocketLib Calls to Host TCP/IP.

- Install your trace-enabled application in the Simulator session.
- Run your trace-enabled application in the Simulator session.

Starting a Palm Reporter Session

To start a Palm Reporter session, run the `Reporter.exe` file. After starting Palm 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 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 Reporter

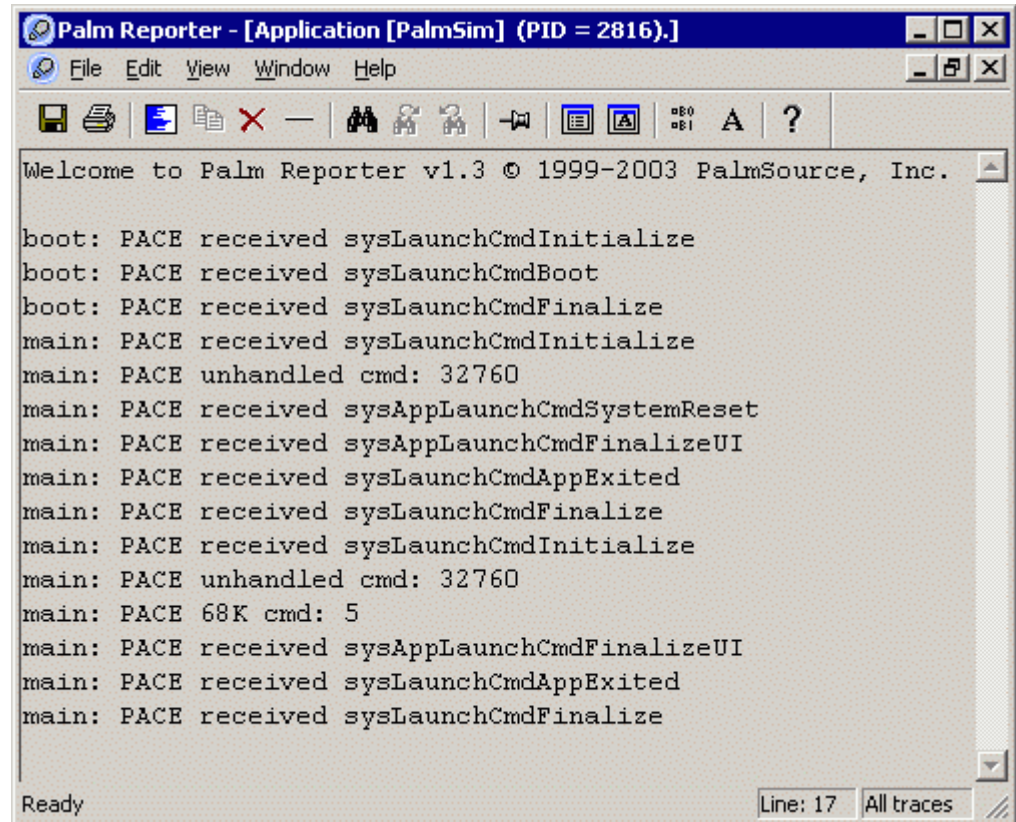
A new trace window is created for each `HostTraceInit` to `HostTraceClose` sequence in your trace-enabled application.

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 3.1](#) shows a Palm Reporter session window.

Figure 3.1 Palm Reporter session window



Filtering Information in a Palm Reporter Session















You can control the type of trace information Palm 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 Reporter session.

Using Palm Reporter

Displaying Trace Information in Palm Reporter

Using the Palm Reporter Toolbar

Palm 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 Reporter

Table 3.2 How to solve possible Palm 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 Reporter session window.	Make sure that: <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 3.3 Palm 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 40
- “[View Menu](#)” on page 41
- “[Install Menu](#)” on page 44
- “[Settings Menu](#)” on page 45
- “[Storage Menu](#)” on page 50
- “[About Menu](#)” on page 51
- “[Exit Menu](#)” on page 51

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 4.1 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.

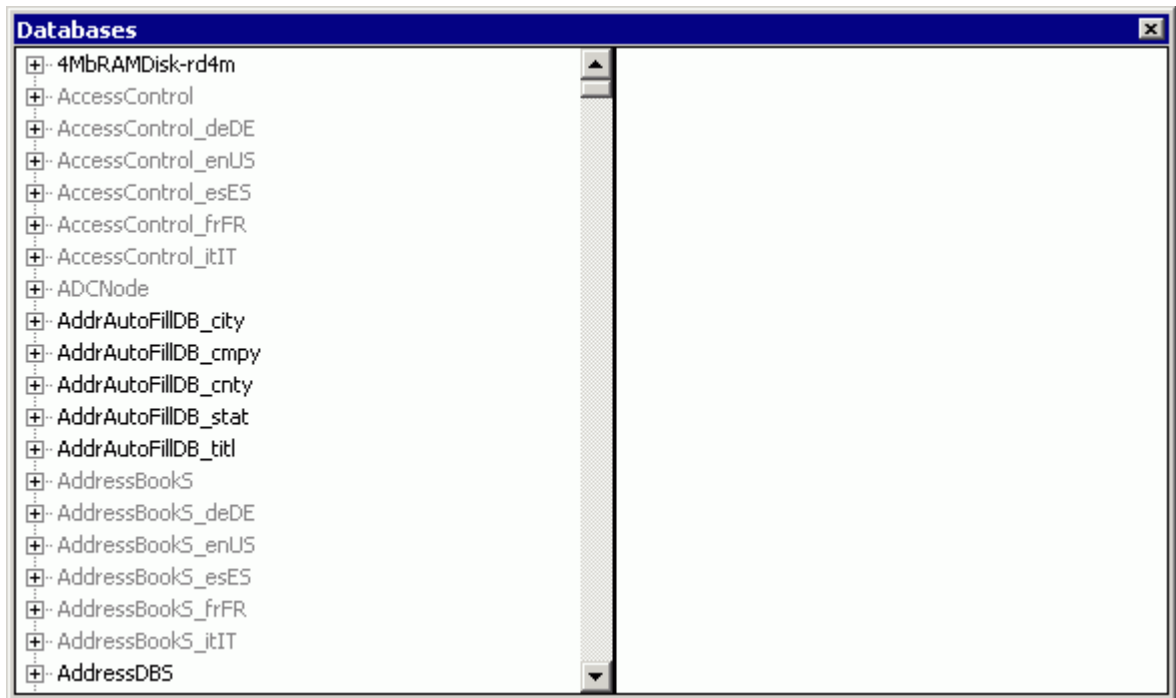
View Menu

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 4.2](#), which lists the databases included in the current simulator session.

Figure 4.2 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.

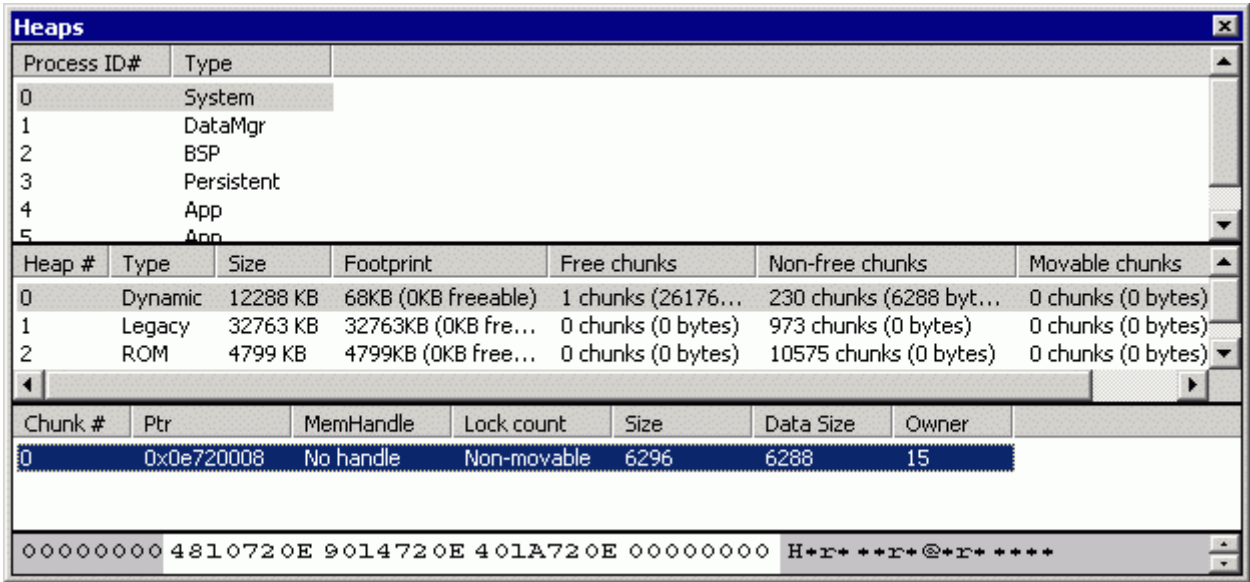
- Right-click a database (use mouse button 2) to open the Export Database dialog box and save the contents of the database.

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

Heaps

Opens the Heaps dialog box, shown in [Figure 4.3](#), which displays the contents of the heaps for the current simulator session.

Figure 4.3 View>Heaps dialog box



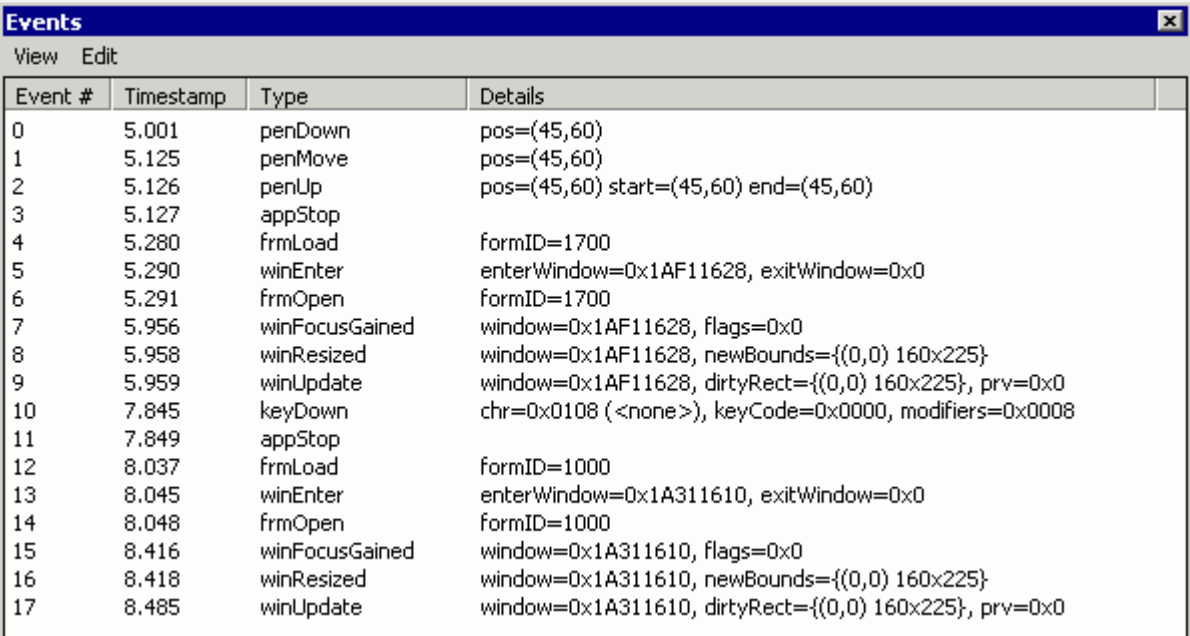
Using the Heaps Dialog Box

- 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 4.4](#), which displays the user interface events for the current simulator session.

Figure 4.4 View>Events dialog box



Event #	Timestamp	Type	Details
0	5.001	penDown	pos=(45,60)
1	5.125	penMove	pos=(45,60)
2	5.126	penUp	pos=(45,60) start=(45,60) end=(45,60)
3	5.127	appStop	
4	5.280	frmLoad	formID=1700
5	5.290	winEnter	enterWindow=0x1AF11628, exitWindow=0x0
6	5.291	frmOpen	formID=1700
7	5.956	winFocusGained	window=0x1AF11628, flags=0x0
8	5.958	winResized	window=0x1AF11628, newBounds={{0,0} 160x225}
9	5.959	winUpdate	window=0x1AF11628, dirtyRect={{0,0} 160x225}, prv=0x0
10	7.845	keyDown	chr=0x0108 (<none>), keyCode=0x0000, modifiers=0x0008
11	7.849	appStop	
12	8.037	frmLoad	formID=1000
13	8.045	winEnter	enterWindow=0x1A311610, exitWindow=0x0
14	8.048	frmOpen	formID=1000
15	8.416	winFocusGained	window=0x1A311610, flags=0x0
16	8.418	winResized	window=0x1A311610, newBounds={{0,0} 160x225}
17	8.485	winUpdate	window=0x1A311610, dirtyRect={{0,0} 160x225}, prv=0x0

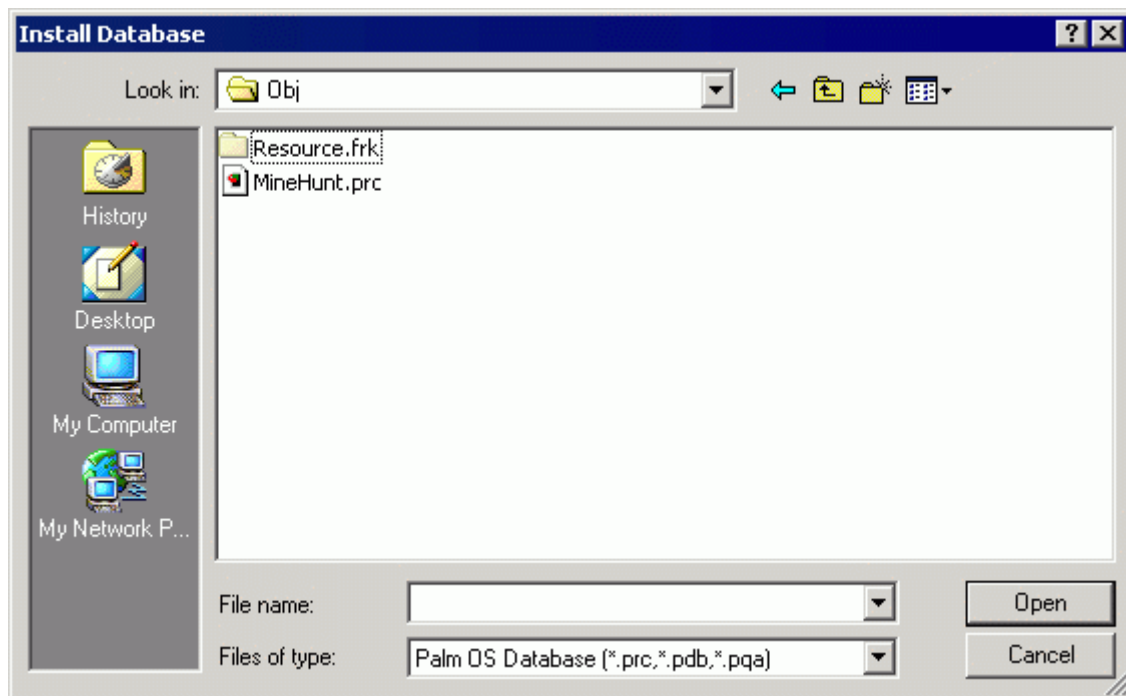
Using the Events Dialog Box

- 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>Nil Events** menu to allow nil events to be displayed in the event viewer. If this menu item is not selected, then nil events are filtered out of the list of events.
- Use the **Edit > Clear** menu to clear the events information that is displayed.
- Use CTRL-A to select all the events information.
- 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 4.5](#).

Figure 4.5 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.

Display>Color Depth

Set the color depth for this simulation session:

- 16 Gray Levels
- 256 Gray Levels
- 65536 Colors

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>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.

Display>Resolution

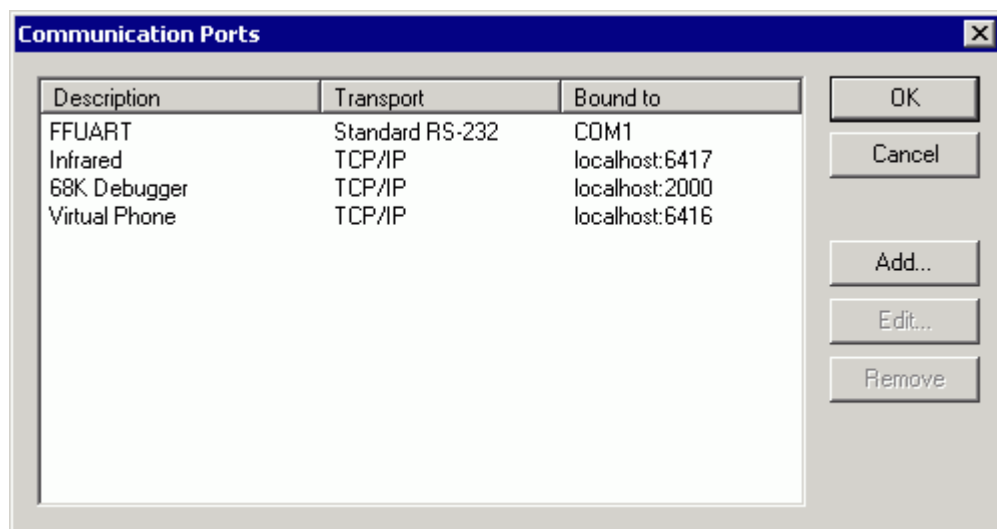
Sets the resolution for the simulation session display:

- 320 x 320
- 320 x 480

Communication>Communication Ports

Displays the Communication Ports dialog box, shown in [Figure 4.6](#), 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 20.

Figure 4.6 Communication ports dialog box



Using the Communication Ports Dialog Box

FFUART:

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

The FFUART 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 20.

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 22.

Virtual Phone:

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

Communication>Redirect SocketLib Calls to Host TCP/IP

Indicates whether SocketLib calls should be redirected to the host TCP/IP communication stack. (This setting does not affect the TCP/IP communication between Palm OS Cobalt Simulator and the 68K debugger.)

Communication > Use Host Ethernet Adapter

Indicates that you want Palm OS Cobalt Simulator to use your PC's ethernet card to simulate an ethernet connection.

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>

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.

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>RAM Size

Sets the RAM size for this simulation session:

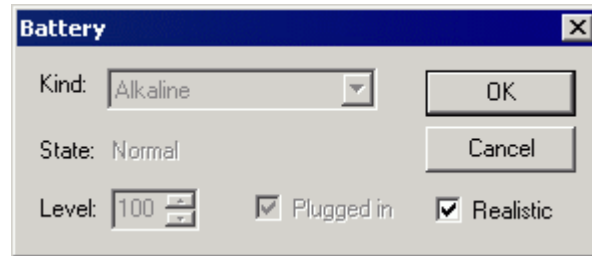
- 16 MB
- 32 MB
- 64 MB
- 128 MB
- 256 MB

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

Battery

Displays the Battery dialog box, shown in [Figure 4.7](#) on page 49, so that you can change the simulated battery settings.

Figure 4.7 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.

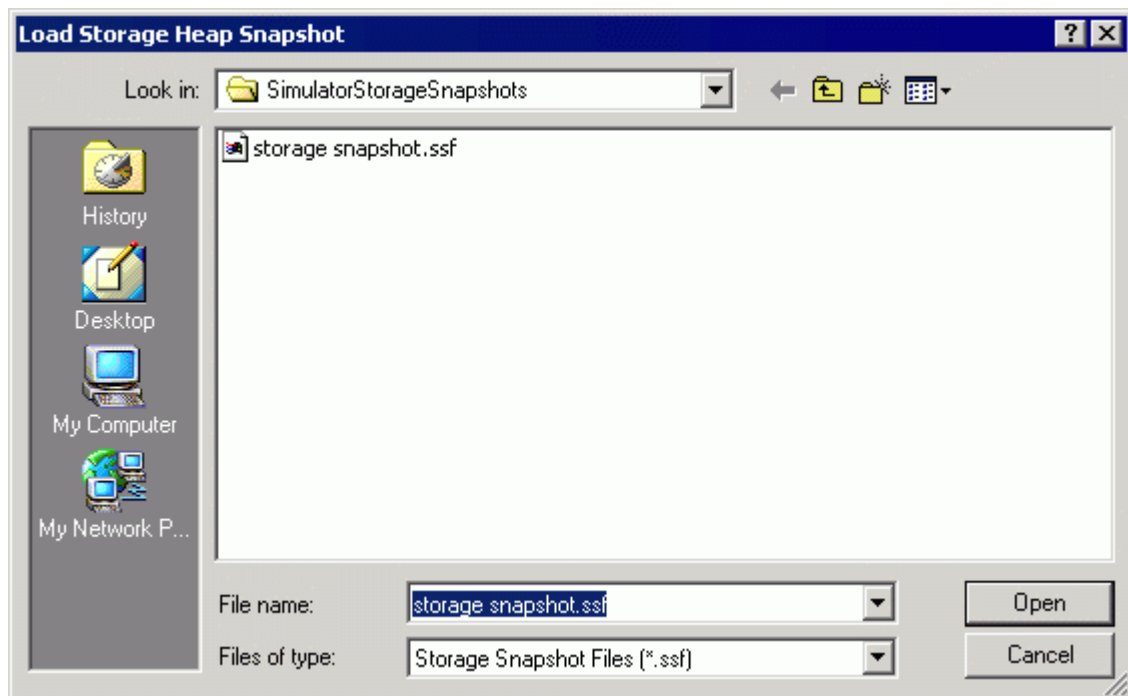
Storage Menu

Use the **Storage** menu to load and save a storage snapshot file (SSF file).

Load

Opens the Load Storage Heap Snapshot dialog box, shown in [Figure 4.8](#), so that you can load the storage heap with contents that were previously saved to a storage snapshot file (SSF file).

Figure 4.8 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 4.9](#), so that you can save the current contents of the storage heap to a storage snapshot file (SSF file).

Figure 4.9 Save Storage Heap Snapshot 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 14.

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 4.1 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 4.2 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.	

Palm OS Cobalt Simulator User Interface Reference

Keyboard Equivalents Reference

Index

Numerics

320 x 320 display 45
320 x 480 display 45

A

about menu 51
adding trace calls 29
alkaline battery 49
always on top menu 45
AppCreator 20
AutoLoad 19
AutoRun 19
AutoRunAndQuit 19

B

battery
 dialog box 49
 menu 48
 settings 48

C

changing the RAM size setting 48
CodeWarrior Debugger 22
color depth menu 45
command line arguments 7
communication
 functions 20
 ports dialog box 46
 ports menu 46
 settings 47
 stacks 3
communication>communication ports menu 46
communication>redirect socketlib calls to host
 TCP/IP menu 47
critical battery state 49

D

databases 41
 dialog box 41
debugging 22, 53
display>always on top menu 45
display>color depth menu 45
display>magnification 45

display>resolution 45
displaying trace information 34
drag and drop 19
driver set 3

E

enable sound menu 49
entering
 data 18
 the debug tool 53
events 43
 dialog box 43
EXE file 6
exit menu 51
external debug tools 22

F

file
 DLL 2
 EXE 6
 INI 14
 PalmSim.exe 6
 palmsim.ini 14
 PDB 18
 PRC 18
 ROM 5, 13
 SSF 13, 50
 storage snapshot 13
filtering trace information 35
fixing reporter problems 37

H

hard reset 40, 53
hardware buttons 52
heaps 42
 dialog box 42
host control
 adding trace calls 29
 using reporter 29
host TCP/IP 47
HotSync operation 21

I

INI file 14

initialization file 14

install

- database dialog box 44
- menu 44

install>database menu 44

installing

- applications 18
- databases 44
- Palm OS Cobalt Simulator 5
- reporter 28

K

keyboard

- equivalents 52
- functions 52

kind of battery 49

L

liion battery 49

liion1400 battery 49

load

- menu 50
- storage heap 50
- storage heap snapshot dialog box 50

loading StorageSnapshotFile 50

low battery state 49

M

magnification 45

main window 7

memory RAM size 48

memory>RAM size menu 48

menu items 17

menus 39

- about 51
- displaying 40
- exit 51
- reset 40
- settings 45
- storage 50
- view 41

Metrowerks CodeWarrior Debugger 22

N

next field 53

nicad battery 49

nimh battery 49

normal battery state 49

O

overview 1

P

PACE 3

Palm Application Compatibility Environment 3

Palm Debugger 22

Palm OS

- applications 2
- communication stacks 3
- runtime services 2

Palm OS Cobalt Simulator 4

- command line arguments 7
- communication functions 20
- components 2
- debugging 22
- display 17
- entering data 18
- initialization file 14
- installing 5
- installing applications 18
- main window 7
- menu items 17
- overview 1
- performing a HotSync operation 21
- prerequisites 4
- ROM images 16
- running 17
- starting 6

Palm Reporter 27

palmsim.ini 14

PDB file 18

performing

- a hard reset 40
- a soft reset 40

PRC file 18

previous field 53

R

- RAM size menu 48
- rechalk battery 49
- redirect SocketLib calls 47
- redirect SocketLib calls to host TCP/IP menu 47
- reporter 27
 - about 27
 - adding trace calls 29
 - features 28
 - filtering information 35
 - functions 36
 - installation errors 37
 - installing 28
 - package file contents 28
 - sample code 30, 31
 - session window 35
 - toolbar 36
 - trace session 34
 - trace strings 32
 - troubleshooting 37
 - using simulator 34
- reset menu 40
- reset>hard menu 40
- reset>soft menu 40
- resolution 45
- ROM
 - file 5, 13
 - images 16
- running Palm OS Cobalt Simulator 17
- runtime services 2

S

- save
 - menu 51
 - storage heap 51
 - storage heap snapshot dialog box 51
- saving StorageSnapshotFile 51
- scaling size 45
- screen size 45
- setting
 - the color depth 45
 - the communication ports 46
 - the scaling size 45
 - the screen magnification 45
 - the screen resolution 45

- settings menu 45
- settings>battery menu 48
- settings>communication menu 46
- settings>display menu 45
- settings>enable sound menu 49
- settings>memory menu 48
- showing
 - heap contents 42
 - installed databases 41
 - user interface events 43
- shutdown battery state 49
- simulator
 - using reporter 34
- SocketLib calls 47
- soft reset 40, 53
- sound settings 49
- SSF file 13, 50
- starting Palm OS Cobalt simulator 6
- stay on top 45
- storage
 - heap contents 50
 - menu 50
 - snapshot file 50
- storage>load menu 50
- storage>save menu 51
- StorageSnapshotFile 20, 50

T

- tabbing 53
- TCP/IP communication 47
- testing 68K applications 3
- trace
 - analysis 27
 - strings 32
- tracing
 - applications 27
 - sample code 30, 31

U

- user interface events 43
- using
 - a mouse 18
 - AppCreator 20
 - AutoRun, AutoLoad, and AutoQuit 19

- drag and drop 19
- external debug tools 22
- reporter 27
- StorageSnapshotFile 20
- the initialization file 14

V

- view menu 41
- view>databases menu 41

- view>events menu 43
- view>heaps menu 42
- viewing
 - databases 41
 - events 43
 - heaps 42

W

- window size 45