



StrongARM^{**} SA-1100 WebPhone Reference Design

User's Guide

September 1998



Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The SA-1100 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Copyright © Intel Corporation, 1998

*Third-party brands and names are the property of their respective owners.

**ARM and StrongARM are trademarks of Advanced RISC Machines, Ltd.



Contents

1	Hardware Description.....	1-1
1.1	Document Organization.....	1-1
1.2	Conventions	1-1
1.2.1	Caution	1-1
1.2.2	Data Units.....	1-1
1.2.3	Note.....	1-2
1.2.4	Numbering.....	1-2
1.2.5	Power Supply Voltages	1-2
1.2.6	Signal Names	1-2
1.2.7	Warning.....	1-2
1.3	Overview	1-3
1.3.1	Features	1-3
1.3.1.1	Telephone Features	1-3
1.3.1.2	Digital Answering Machine Features	1-3
1.3.1.3	Internet Features	1-4
1.3.1.4	Miscellaneous Features	1-4
1.4	Hardware Component and Interface Description	1-4
1.4.1	SA-1100 Microprocessor.....	1-4
1.4.2	Memory System	1-5
1.4.3	Clocks.....	1-5
1.4.4	Telephone-Answering Device (TAD).....	1-5
1.4.5	Modem	1-6
1.4.6	Touch Screen	1-6
1.4.7	Keyboard.....	1-6
1.4.8	LCD	1-6
1.4.9	Infrared Transceiver	1-6
1.4.10	Touch Pad.....	1-6
1.4.11	Minislide Switches	1-7
1.4.12	Jumpers.....	1-7
1.4.13	LEDs.....	1-8
1.4.14	Trimpots	1-9
1.4.15	External Connectors.....	1-9
1.4.16	Power Supply and Power Regulation.....	1-9
1.5	Signal List.....	1-10
1.5.1	GPIO Port Assignments	1-10
1.5.2	Chip Selects	1-11
1.5.3	Serial Port Assignment.....	1-11
1.5.4	I ² C Signals	1-12
1.6	Physical.....	1-12
1.6.1	Sockets.....	1-12
2	Test Applications.....	2-1
2.1	Requirements	2-1
2.2	Setup	2-1
2.2.1	Memory Test	2-2
2.2.2	Initialization.....	2-2



2.2.3	I ² C Test	2-2
2.3	Menu-Driven Tests	2-3
2.3.1	Keyboard Test	2-3
2.3.2	LCD Test	2-3
2.3.3	TAD Test	2-3
2.3.4	Telephone Features Verification Tests	2-4
2.3.5	UCB1200 Interface Test	2-5
2.3.6	Touch Screen Test	2-5
3	WebPhone Inferno Software Overview	3-1
3.1	Inferno OS and WebPhone Software	3-1
3.1.1	Architecture	3-1
3.1.2	Applications	3-2
3.1.2.1	Application Toolbar Configuration	3-2
3.1.3	WebPhone API	3-3
3.1.3.1	Inferno Telephony Interface	3-3
3.1.3.2	Inferno Reference API	3-3
3.1.4	Dis Virtual Machine	3-4
3.1.5	Inferno Kernel	3-4
3.1.6	Device Drivers	3-4
3.2	WebPhone Device Drivers	3-5
3.2.1	LCD Display	3-5
3.2.2	Touch Screen	3-5
3.2.3	Keyboard	3-5
3.2.4	Flash ROM	3-5
3.2.5	Serial Port	3-5
3.2.6	Soft Modem	3-5
3.2.7	Telephony	3-6
3.3	WebPhone Applications	3-6
3.3.1	Application Toolbar	3-6
3.3.2	Telephone	3-7
3.3.3	Web Browser	3-7
3.3.4	E-mail	3-8
3.3.5	Configure	3-8
3.3.6	Address Book	3-8
3.3.7	Shell	3-8
3.4	WebPhone API	3-8
3.4.1	Inferno Telephony Interface	3-8
3.4.1.1	Status Inquiry	3-9
3.4.1.2	Event Detection	3-9
3.4.1.3	Console Control	3-10
3.4.1.4	Call Control	3-10
4	Getting Started	4-1
4.1	Quick Start	4-1
4.1.1	Setting Up the Hardware	4-1
4.2	Installation and Setup for Development	4-2
4.2.1	Inferno Device Driver Kit Package	4-2
4.2.2	Inferno Device Driver Kit Installation	4-3
4.2.2.1	Installing Inferno from the CD-ROM	4-3
4.2.2.2	Installing the DDK File System	4-3



4.3	WebPhone Applications	4-5
4.3.1	Using Telephone	4-6
4.3.2	Running Web Browser	4-7
4.3.3	Using E-mail	4-8
4.3.4	Using Configure.....	4-12
4.3.5	Running Address Book.....	4-14
4.3.6	Using the Shell	4-16

Figures

3-1	Software Architecture	3-2
3-2	Application Toolbar	3-6
4-1	DDK Mash Window	4-4
4-2	Toolbar Buttons	4-5
4-3	Virtual Phonepad Window	4-6
4-4	Volume Control Window	4-7
4-5	Browser Start Page	4-8
4-6	Mail Menu	4-9
4-7	Mail Information Window	4-9
4-8	New Message Menu	4-11
4-9	Keyboard Option	4-12
4-10	Configuration Window	4-13
4-11	Address Book Window	4-14
4-12	Record Dialog	4-15
4-13	Shell Window	4-16

Tables

1-1	Data Unit Terminology	1-1
1-2	Signal Names	1-2
1-3	Memory Map	1-5
1-4	Minislide Switch Settings	1-7
1-5	Jumper Settings (Sheet 1 of 2)	1-7
1-6	LED Operation	1-8
1-7	Trimpot Operation	1-9
1-8	External Connector Operation	1-9
1-9	GPIO Port Assignments by Function	1-10
1-10	GPIO Port Signals in Numerical Order	1-11

This document describes the StrongARM** SA-1100 WebPhone Reference Design (WebPhone).

For more information about the StrongARM** SA-1100 microprocessor (SA-1100), refer to the *StrongARM** SA-1100 Microprocessor Technical Reference Manual*.

1.1 Document Organization

This document contains the following chapters:

- Chapter 1, “Hardware Description”, lists the WebPhone features and describes the major hardware components and interfaces.
- Chapter 2, “Test Applications”, describes the WebPhone components test and sample telephony applications.
- Chapter 3, “WebPhone Inferno Software Overview”, describes the WebPhone software architecture and applications.
- Chapter 4, “Getting Started”, describes how to set up the hardware and software to build a working WebPhone.

1.2 Conventions

The following terminology and conventions are used in this document.

1.2.1 Caution

Cautions provide information to prevent damage to equipment or loss of data.

1.2.2 Data Units

The following data unit terminology is used throughout this document.

Table 1-1. Data Unit Terminology

Term	Quantity			Other Terms
	Words	Bytes	Bits	
Byte	½	1	8	—
Word ¹	1	2	16	—
Dword	2	4	32	Longword (LW)

1. A 32-bit quantity in some SA-1100 documentation.

1.2.3 Note

Notes emphasize important information.

1.2.4 Numbering

All numbers are decimal or hexadecimal unless otherwise indicated. For example, 19 is decimal, but 19h and 19A are hexadecimal. In ambiguous cases, a subscript indicates the radix of nondecimal numbers.

1.2.5 Power Supply Voltages

The plus sign (+) is omitted from all positive power supply voltages.

1.2.6 Signal Names

All signal names are printed in plain and boldface type, and in lowercase, uppercase, or mixed case, as follows:

Table 1-2. Signal Names

Example	Signal Type
NAME	SA-1100, high-active
nNAME	SA-1100, low-active (n prefix indicates low-active)
name	PCI and nonSA-1100, high-active
name#	PCI and nonSA-1100, low-active (# suffix indicates low-active)
NAME	Board or third-party device, high-active
NAME_L	Board or third-party device, low-active

1.2.7 Warning

Warning indicates the potential for personal injury.

1.3 Overview

WebPhone is a complete turnkey reference design based on the SA-1100 microprocessor and the Inferno* operating system by Lucent Technology.. The WebPhone reference design provides an evaluation board and software development environment for a simple, low-cost appliance that gives users:

- All the functionality of a telephone
- The capability to send and receive e-mail
- Access to the Internet through a web browser

Two separate phone lines are used to access telephone and Internet services. The user interface can include a QWERTY keyboard, and monochrome or color LCD display with a touch screen feature (mouse and touch screen are standard).

1.3.1 Features

WebPhone includes the following telephone, answering machine, Internet, and miscellaneous features:

1.3.1.1 Telephone Features

- Make calls
- Receive calls
- Call waiting
- Caller ID
- Three-way conference
- Full-duplex speakerphone
- Store phone numbers
- Speed dial
- Hold, mute, and flash buttons
- Volume control

1.3.1.2 Digital Answering Machine Features

- Record messages with date/time stamp
- Play messages with date/time stamp
- Fast forward messages
- Rewind messages
- Save messages
- Delete messages
- Record prompts/phrases
- Play prompts/phrases

- New messages indication

1.3.1.3 Internet Features

- World Wide Web browser
- Receive e-mail
- Send e-mail
- Family white board (BBS)
- Automatic subscription to some Internet service providers

1.3.1.4 Miscellaneous Features

- Calendar
- Directory address book
- Notepad
- Small footprint

1.4 Hardware Component and Interface Description

This section describes the major hardware components and interfaces of the WebPhone.

The WebPhone reference design is based on an SA-1100 microprocessor operating at 160 MHz to 220 MHz. Main memory includes 4 Mbyte of Flash ROM and 8 Mbyte of DRAM. The peripheral devices include:

- Keyboard
- Monochrome LCD display
- Touch screen
- Infrared digital-to-analog transceiver (IrDA)
- RS-232
- Modem
- Digital telephone-answering device (TAD)
- Two external telephone lines

1.4.1 SA-1100 Microprocessor

The SA-1100 microprocessor is a high-performance, low-power integrated processor configured to run at speeds ranging from 133 MHz to 220 MHz. The SA-1100 package is a 208-pin TQFP. The SA-1100 requires two separate supply voltages: 1.5 V-core and 3.3 V-I/O. For more information see the *StrongARM** SA-1100 Microprocessor Technical Reference Manual*.

1.4.2 Memory System

The memory system provided on the WebPhone reference design contains:

- 8 Mbyte EDO DRAM SIMM
- 4 Mbyte Flash ROM. This includes the boot block. Write operations into the boot block are controlled by signal FLASH_WP_L from GP[14].

Note: Care should be taken to avoid overwriting this boot block.

Table 1-3 lists the memory address allocation.

Table 1-3. Memory Map

Boot ROM and Flash ROM (4 Mbyte)	0x0000 0000 to 0x03FF FFFF
TAD (low byte)	0x0A00 0000
TAD (high byte)	0x0A00 0004
DRAM (8 Mbyte)	0xC000 0000 to 0xC7FF FFFF

1.4.3 Clocks

The WebPhone reference design uses the following clocks:

- 32.768-kHz crystal oscillator
- 3.6864-MHz crystal oscillator
- 36.864-MHz crystal oscillator

Note: 32.768-kHz and 3.6864-MHz crystal oscillators use resistor dividers to ensure the voltage level of the clock driven into the SA-1100 does not exceed 1 V. The voltage level of any clock from an external clock source also must *not* exceed 1 V, to prevent problems with the operation of the 1.5 V-CPU core.

The WebPhone reference design also accepts clock sources from test equipment (for example, clock generators).

1.4.4 Telephone-Answering Device (TAD)

The digital telephone-answering device function is based on the DSP Group 6417C* TAD chip and National Semiconductor TP3054* codec. The on-board microphone/speaker and handset are connected to the TAD codec, and the TP3054* codec is connected to the line through the telephone-line interface circuit, also known as the data access arrangement (DAA).

The TAD is mapped to 0x0A00 0000. Address line A2 is toggled to access the low and high byte; that is, a read or write to 0x0A00 0000 accesses the low byte and 0x0A00 0004 accesses the high byte.

4 Mbyte of NAND Flash and PROM are connected to the TAD. Only the TAD can access this memory. This is not a standard Flash ROM with address and data bits.

A Cermetek CH1837C* DAA connects to the telephone line. This DAA includes the caller ID (CID) option, 2-wire to 4-wire conversion, and voltage isolation required for the public switched telephone network (PSTN) connection.

The ring-detect, off-hook, and caller-ID enables are connected to GPIO pins controlled by the SA-1100 (see Table 1-9).

1.4.5 Modem

The Philips UCB1200* advanced modem/audio analog front end is used as a front end. It is connected to the telephone line through the DAA. Here, the XE0071 DAA module is used, as 2-wire to 4-wire circuitry is available inside the UCB1200 so the DAA need not have this feature. The caller ID feature is not provided on this line but the UCB1200 does have interfaces to a speaker or microphone.

1.4.6 Touch Screen

The WebPhone board touch-screen panel is used for data input. It is connected to the UCB1200.

1.4.7 Keyboard

The Fujitsu Microelectronics FKB1406* keyboard is controlled through the serial peripheral interface (SPI). The SA-1100 SPI protocol interfaces to a USAR Systems UR5HCSPI-06-FB* keyboard controller. The controller interprets input from the keyboard.

1.4.8 LCD

The LCD is a Sharp Electronics LM64P11*. This is monochrome, 4-bit, dual-panel, 640 × 480 VGA display.

1.4.9 Infrared Transceiver

The TEMIC Semiconductors TFDS6000D* is a dual-speed Infrared Digital-to-Analog (IrDA) transceiver (115 kb/s or 4 Mb/s) connected to serial port 2 of the SA-1100. The transceiver receives digital data output from the SA-1100 and converts the data to infrared pulses, or receives infrared pulses and converts them to digital data for the SA-1100. The SD pin is connected to **GP[21]**.

Note: This device is currently not tested.

1.4.10 Touch Pad

This module is included but not tested.

1.4.11 Minislide Switches

There are 10 switches on the WebPhone. These switches are single-pole double through (SPDT) minislide switches that have a setting showing either a DOT or NO DOT. When the switch is in one of its two positions, a DOT is visible on the physical switch; a NO DOT is visible when the slide is in the other position. Table 1-4 shows the function and default position of the minislide switches.

Table 1-4. Minislide Switch Settings

Switch	Function	Default Setting
S1	Battery fault	DOT
S2	Reset (push button)	Not installed
S3	VDD fault	DOT
S4	GPIO5 Daa-OH	NO DOT
S5	Boot DEBG-normal	NO DOT
S6	Reset	DOT
S7	GPIO0	DOT
S8	Keyboard xSW	NO DOT
S9	Handset off hook	NO DOT
S10	Lid closed	NO DOT

1.4.12 Jumpers

There are 27 jumper connectors on the board. The jumpers can be used for debugging and for special evaluation tests. Table 1-5 shows the function and the settings of the jumpers.

Table 1-5. Jumper Settings (Sheet 1 of 2)

Jumper	Function	Setting
JP1	-5.0 V	Installed
JP2	—	Not used
JP3	LCD_VEE Labeled J3 on board	Installed
JP4	VCC3 Labeled J4 on board	Installed
JP5	—	Not used
JP6	VCC3 (3.3 V)	Installed vertically on all pins
JP7	VCCP (1.5 V)	Installed vertically on all pins
JP8	VCC (5 V)	Installed vertically on all pins
JP9	—	Not used
JP10	—	Not used
JP11	—	Not used
JP12	—	Not used
JP13	—	Not used
JP14	—	Not used

Table 1-5. Jumper Settings (Sheet 2 of 2)

Jumper	Function	Setting
JP15	—	Not used
JP16	SPKR OUT	Installed on bottom two pins
JP17	—	Not used
JP18	—	Not used
JP19	TST CLK_BY P	Installed on left two pins
JP20	V REF	Installed on bottom two pins
JP21	ROM WIDTH	Installed on bottom two pins
JP22	16-bit or 32-bit boot ROM	Installed on bottom two pins
JP23	—	Not used
JP24	—	Not used
JP25	—	Not used
JP26	—	Not used
JP27	Keypad	Not used

1.4.13 LEDs

There are six LEDs on the board. Table 1-6 describes LED operation.

Table 1-6. LED Operation

LED	Function	Color	Programmable
D4	VCC5	Red	No
D6	VCC3	Green	No
D7	Wall in + 12 V	Red	No
D10	VCCP	Yellow	No
D22	GPIO 27	Green	Yes
D23	GPIO 26	Red	Yes

1.4.14 Trimpots

There are six trimpots on the board. Table 1-7 describes trimpot operation.

Table 1-7. Trimpot Operation

Trimpot	Function
R60	LCD contrast.
R64	LCD brightness adjust.
R72	VR1 adjust
R83	Speaker adjust.
R90	VREF adjust. Not used.
R99	Microphone adjust.

1.4.15 External Connectors

There are 10 external connectors on the board. Table 1-8 describes the function of external connectors.

Table 1-8. External Connector Operation

Connector	Function
P1	Serial.
J1	Touch screen.
J2	LCD interface.
J3	Wall in + 12 V.
J4	Labeled JP4. Smart battery connector. Not used.
J5	LCD power.
J6	Phone in voice on.
J7	UCB in modem.
P8	Phone hand set in.
J10	Keyboard.
J11	Keyboard.

1.4.16 Power Supply and Power Regulation

System power is provided by an external power transformer (wall unit) that supplies 12-V dc at 1 A to the power supply. Rechargeable battery backup is supported.

The power supply generates the following voltages and currents:

V	mA	V	mA
1.5	500	3.3	1500
5.0	1500	24.0	100
-5.0	100	1000.0	25

1.5 Signal List

The following sections list General-Purpose I/O (GPIO) port and various signal assignments.

1.5.1 GPIO Port Assignments

Table 1-9 lists the GPIO port assignments in functional groups

Table 1-9. GPIO Port Assignments by Function

Port	Signal
Configuration Switches	
GP[0] GP[1] GP[20]	SWITCH 1 HANDSET OFF_HOOK DEBUG_NOR
Battery Thermal Condition from the Smart Battery	
GP[3]	BATT_THRM
TAD Line Control Signals (such as RING, OFF HOOK, CALLER ID, ACK for Host R/W)	
GP[2] GP[4] GP[5] GP[15]	SPKRDAACID SPKRDAARI_L SPKRDAAOH TADACK_L
I²C Control Signals	
GP[6] GP[7]	I ² C_SDA I ² C_SCL
Flash ROM Write Protect	
GP[14]	FLASH_WP_L
IrDA and UCB1200 Signals	
GP[21] GP[22]	IRDA_SD UCB_IRQ
Keyboard Signals	
GP[10] GP[11] GP[12] GP[13] GP[23] GP[24] GP[25] GP[26] GP[27]	KBC_MOSI KBC_MISO KBC_SCLK KBC_SFRM KBC_WKUP_L KBC_WUKO KBC_ATN_L DEBUG_LED DEBUG_LED

Table 1-10 lists the GPIO port assignments in numerical order.

Table 1-10. GPIO Port Signals in Numerical Order

Port	Signal	Port	Signal
GP[0]	SWITCH 1	GP[14]	FLASH_WP_L
GP[1]	HANDSET OFF_HOOK	GP[15]	TAD_ACK_L
GP[2]	SPKRDAACID	GP[16]	Keypad C1
GP[3]	BATT_THRM	GP[17]	Keypad C2
GP[4]	SPKRDAARI_L	GP[18]	Keypad C3
GP[5]	SPKRDAAOH	GP[19]	Keypad C4
GP[6]	I2C_SDA	GP[20]	DEBUG_NOR
GP[7]	I2C_SCL	GP[21]	IRDA_SD
GP[8]	Not used.	GP[22]	UCB_IRQ
GP[9]	Not used.	GP[23]	KBC_WKUP_L
GP[10]	KBC_MOSI	GP[24]	KBC_WUKO
GP[11]	KBC_MISO	GP[25]	KBC_ATN_L
GP[12]	KBC_SCLK	GP[26]	DEBUG_LED
GP[13]	KBC_SFRM	GP[27]	DEBUG_LED

1.5.2 Chip Selects

The SA-1100 chip select signals are assigned as follows:

nCS0	Boot ROM
nCS1	TAD
nCS2	Not used
nCS3	Keypad

1.5.3 Serial Port Assignment

The SA-1100 serial ports are assigned as follows:

Port 0	Not used
Port 1	Not used
Port 2	Infrared
Port 3	RS-232 serial
Port 4	UCB1200

1.5.4 I²C Signals

The I²C bus provides the following signals:

MUTE SPEAKER

MUTE HANDSET

LCD BACK-LITE ENABLE

LCD VEE ENABLE

CHARGE 1

CHARGE 2

TAD FLASH ROM WRITE PROTECT SIGNAL

RS-232 ON OR OFF

1.6 Physical

The WebPhone board measures 19.0 × 24.8 cm (7.5 × 9.75 in). The inverter has cutouts. Mounting holes are located at all four corners.

1.6.1 Sockets

Sockets are provided for the system Flash ROM (56-pin TSOP), PAL (28-pin PLCC), DRAM SIMM (72-pin) and PROM (32-pin PLCC, not populated). All other components are mounted directly on the PCB.

The test applications test the StrongARM** SA-1100 WebPhone Reference Design (WebPhone) hardware. Visit the following web site and download the latest software used to run on the WebPhone:

<http://developer.intel.com/design/strong/sa-eval.htm>

2.1 Requirements

Sample applications require the ARM** SDT version 2.1 or later and run on the Angel* debug monitor version 1.02 or later. The Angel* version replaces the Damon* debug monitor. The Angel* version can be found in the ANGEL variant of the project. Use this variant to build the project and debug with the Angel* debug monitor.

Warning: High voltage is used to drive the LCD display and caution should be used when operating the WebPhone. Stickers indicating the location of the high voltage are placed on the platform.

2.2 Setup

To run the test applications:

1. Connect a null modem serial cable between the PC serial port and the RS-232 port (P1) on the board.
2. Connect the LCD data and backlight cables, touch screen cable, and keyboard to the board.
3. Ensure that the board switches are correctly set. See Table 1-4 for more information about the switch settings.
4. Ensure that the Flash ROM and DRAM SIMM are correctly inserted.
5. Connect power to the board and turn it on. On power up, green LEDs D22 and D23 are lighted if Damon* is running or D22 is lighted and D23 is off if Angel* is running.
6. Run the debug monitor with the correct port and baud rate setting of 38400.
7. Download the AXF file from the directory in your project and run the program.
8. By default, the program has a breakpoint at “main”.
9. Run or step through the tests.

The test program begins with the memory test option.

2.2.1 Memory Test

To select the memory test, press **Y** on the keyboard. This program runs five tests on the DRAM, first on 1 Mbyte to 7 Mbyte. The lower 1 Mbyte and upper 1 Mbyte are tested separately because code resides in the lower 1 Mbyte and the upper memory is used as stack.

Note: Test 5 on the upper 1 Mbyte may give an error message. Ignore this message. The Angel* debug monitor uses this space for the stack.

The following tests are run:

- Test 1: Write the address of the memory at that location and read it back.
- Test 2: Write the complement of the address of the memory and read it back.
- Test 3: Write 0x55555555 and read it back.
- Test 4: Write 0xAAAAAAAA and read it back.
- Test 5: Write a 1 and 0, individually one bit at a time, and read it back.

If all the tests pass, the following message displays in the console window:

```
No memory errors in DRAM test
```

If any test fails (that is, the value read from the memory location does not match the value written), the memory location, the expected value, and the actual value read display in the following format:

```
Memory error at ADDR wanted DATA1 got DATA2
```

The number of errors are counted and the following message displays when the test is completed:

```
Memory errors - Total error count = COUNT
```

2.2.2 Initialization

The program runs the initialization routine to initialize all the devices.

Note: There is no interrupt setup in the Angel* debug monitor.

2.2.3 I²C Test

This test is run after initialization. It is not menu driven because its only purpose is to ensure that the I²C signals are correctly generated. Because this platform uses the I²C only for generating control signals, the test generates a signal on each microprocessor. On microprocessor 1, the RS232_ON_OFF and TAD_FL_WP signals are generated and on microprocessor 2 the LCD_BACK_EN signal is generated and then canceled.

If the test failed to generate the signals, either or both of the following messages display in the console window:

```
Error 03: I2C Chip 1 interface failed
Error 04: I2C Chip 2 interface failed
```

If the test program successfully generated the signals, the following message displays in the console window:

```
I2C Chip 1 interface passed
I2C Chip 2 interface passed
```

2.3 Menu-Driven Tests

Following the I²C test, the test menu displays. The menu-driven tests are described in the following sections.

2.3.1 Keyboard Test

The keyboard test initializes the synchronous serial port (SSP), reads characters from the keyboard, and displays the encoded value in the console window. The test program remains in the loop of reading a character from the keyboard and displaying it in the console window until uppercase **E** (**Shift +E**) is pressed. The console window displays the following:

```
Character Recvd.: CHAR
(CHAR is the character received)
```

If the keyboard is not working correctly, nothing is displayed; the only way to exit the keyboard-test loop is to reset the board and close the debugger.

2.3.2 LCD Test

There is no separate LCD test. The debugger console, menu, other messages, and the characters typed on the keyboard are displayed on the LCD. This is sufficient to determine whether the LCD is working correctly.

2.3.3 TAD Test

The telephone-answering-device (TAD) test reads the status word issued by the TAD component on power up, writes a command, and reads the status returned by the TAD. The TAD component sets the ACK pin low when it has a status word to be read. When the StrongARM^{**} SA-1100 microprocessor (SA-1100) reads the status word properly, the TAD component sets the ACK pin high. The TAD interface is verified by verifying that the signal transition occurs on pin **GP[15]** pin, which is connected to the TAD ACK pin. If verification is successful, the following message is displayed:

```
TAD interface test passed. Cold Start
```

for a cold start with the TAD Flash not initialized, or:

```
TAD interface test passed. Warm Start
```

for a warm start with the TAD Flash initialized.

The program will then go through flash initialization and codec selection. If successful, the messages on the screen will display the size of the flash and the codec selected.

If any of the tests fails, the following message is displayed:

```
TAD interface test failed
```

At the completion of this test, the TAD runs an initialization routine, to initialize the codec and the Flash.

2.3.4 Telephone Features Verification Tests

The telephone features tests are performed when the keyboard, LCD, and TAD tests are successfully executed.

Note: The tests are executed using the ARM** debugger to download the executable to the SWORD board. All key inputs are from the SWORD board keyboard.

Follow this procedure to perform the tests:

1. In the main test menu, select 5 for the Telephone Feature Tests. The Telephone Feature Tests menu is displayed. Tests can be selected in any order.
2. Select 1 for the TAD Initialization Test. Although TAD initialization has been performed in the TAD test from the main menu, run the test again to reinitialize the TAD component. (See Section 2.3.3, “TAD Test” on page 2-3.) The TAD Initialization Test:
 - Configures the TAD to use one internal voice prompt with a Samsung* Flash memory.
 - Initializes the TAD Flash memory. It first attempts to initialize 16 Mbytes. If it fails, it reconfigures the TAD and initializes 4 Mbytes. If that fails, no other telephone feature test can be performed.
 - Selects fixed mode codec with μ -law interface. Codec 0 is used for TAD data I/O.

Note: If the TAD acknowledgment pin fails to go low after the TAD receives a command or the pin does not go high after the host has read the TAD status, the test will hang.

3. Select 2 for the Incoming Call Test.
 - a. Connect the SWORD board to an analog phone line jack.
 - b. Dial the phone number; the ringing tone should be audible from the board.

If there is no ringing tone and the message “Switch offhook to pick up the call” is not displayed, the board failed to detect the ring signal from the line. Switch to offhook, then switch to onhook. When the test menu displays, start the test again. It can be a phone system problem.
 - c. Switch to offhook to pickup the call. The talk path is established.
 - d. Switch to onhook to end the call.
4. Select 3 for the Outgoing Call Test.
 - a. Connect the SWORD board to an analog phone line jack.
 - b. Press **D** to make the call.
 - c. Dial the number. The called number should ring. If the call does not succeed, press **D** again to end the test, and then try again.
 - d. Answer the phone at the called number. The talk path should be established.
 - e. Press **D** to end the test.
5. Select 4 for the Record Message Test.
 - a. Use the microphone and start recording following the tone.
 - b. Press **B** to stop recording. The recording stops after a long silence. A tone sounds when the recording has stopped.
 - c. If “Memory Full Message” is displayed, run the TAD Initialization Test to reinitialize the TAD.

6. Select 5 for the Play Message Test.
 - The last recorded message is played following the tone.
 - A tone sounds when playback is finished.
7. Select 6 for Caller ID Test
 - a. Connect the SWORD board to an analog phone line jack.
 - b. Dial the phone number; the ringing tone should be audible from the board. (A ringing tone is not generated when collecting a caller ID message.)

If there is no ringing tone and the message “Start collect Caller ID ” is not displayed, the board failed to detect the ring signal from the line. Switch to offhook, then switch to onhook. When the test menu displays, start the test again. It can be a phone system problem.
 - c. Switch to offhook to pickup the call. The talk path should be established.
 - d. Switch to onhook to end the call.
8. Select 0 to exit the Telephone Feature Tests.

2.3.5 UCB1200 Interface Test

This test checks the hardware interface between the SA-1100 and the UCB1200. The program reads the status register from the UCB and displays the following information:

```
Version number = 4
Device ID = 0
Supplier ID = 1
```

2.3.6 Touch Screen Test

The touch screen test is integrated with the UCB1200 interface test and is executed only once; it is skipped if subsequently selected. This test also tests writing to the UCB1200 registers (reading from UCB1200 registers is tested in the UCB1200 interface test).

To perform the touch screen test:

1. Verify the touch panel bias by writing to the appropriate registers to ground TSMX and TSPX and power TSMY and TSPY.
2. Measure the voltages on all four pins of the touch-panel header to verify the bias.
3. After verifying the bias, press **C** on the WebPhone keyboard to continue.
4. Verify the touch panel bias by writing to the appropriate registers to power TSMX and TSPX and ground TSMY and TSPY.
5. Verify voltage and voltage variations with pen down on the touch panel header pins. The voltage TSMX and TSPX should fall from 3.3 V to approximately 1.5 V (± 0.5 V).
6. Press **C** on the WebPhone keyboard to continue.

WebPhone Inferno Software Overview 3

This chapter describes the StrongARM^{**} SA-1100 WebPhone Reference Design (WebPhone) software architecture, applications, device drivers and application programming interfaces (APIs).

The WebPhone software is composed of:

- The Inferno^{*} operating system (Inferno OS^{*})
- Device drivers that provide the interface between the Inferno OS^{*} kernel and the hardware devices
- Applications written in the Limbo^{*} programming language
- An API for advanced telephony applications and the complete Inferno Reference API^{*}

Contact the Lucent-Inferno World Wide Web Internet site to obtain software for installation:
<http://www.lucent-inferno.com>

3.1 Inferno OS and WebPhone Software

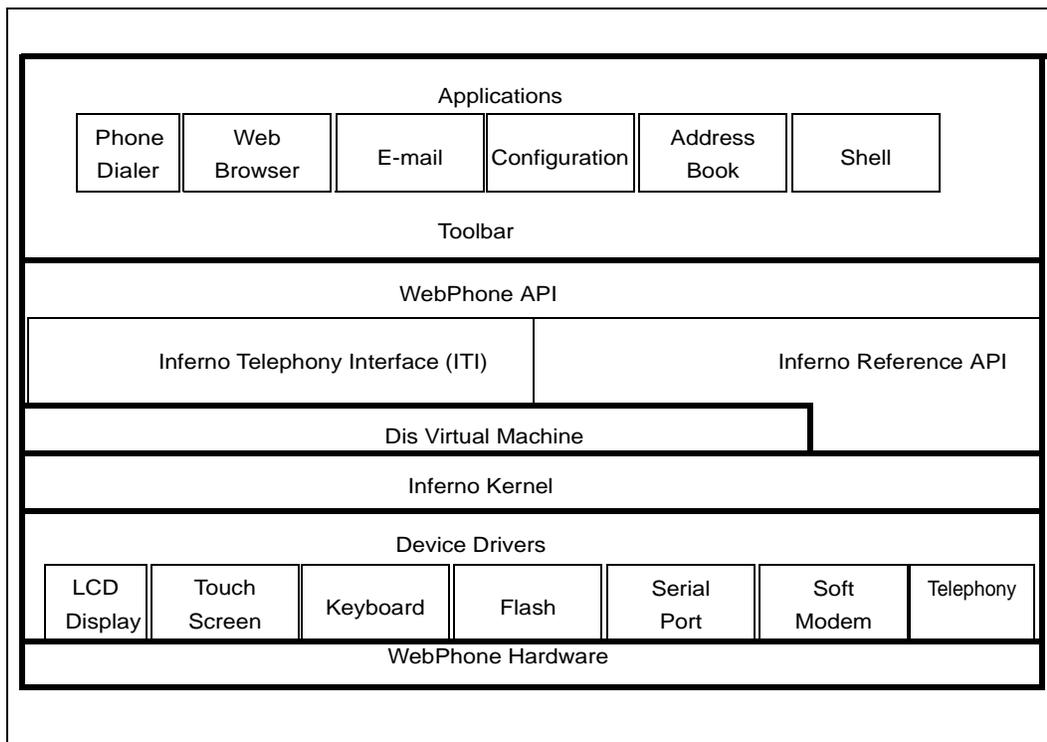
At the heart of the software for the WebPhone is the Inferno^{*} operating system (Inferno OS^{*}). Inferno OS^{*} provides a software infrastructure for distributed, network applications supporting diverse systems in a variety of network environments, such as advanced telephones, handheld devices, TV set-top boxes, Internet telephony, and network switching.

3.1.1 Architecture

Figure 3-1 shows the WebPhone software architecture built around the Inferno OS^{*}.

S

Figure 3-1. Software Architecture



3.1.2 Applications

The applications layer includes the Inferno^{*} demonstration applications:

- Telephone
- Web Browser
- E-mail
- Configuration
- Address Book
- Shell

The applications are invoked from buttons on the configurable Application Toolbar that displays at the bottom of the screen. These applications are written entirely in Limbo^{*}, the native programming language of Inferno^{*}.

3.1.2.1 Application Toolbar Configuration

The Application Toolbar configuration is kept in a file, *tbsetup*. This file defines the format of the Application Toolbar and is stored in each user's directory (for example, */usr/inferno/tbsetup*).

See Section 3.3, "WebPhone Applications" on page 3-6 for more information about the Application Toolbar and applications.

3.1.3 WebPhone API

The API layer provides an interface between the applications and the devices.

3.1.3.1 Inferno Telephony Interface

Inferno Telephony Interface* (ITI) supports telephony functionality. It provides a well-defined interface for applications to access telephony features and hide the hardware dependence from application developers.

See Section 3.4, “WebPhone API” on page 3-8 for more information about the ITI.

3.1.3.2 Inferno Reference API

The Inferno Reference API* is the standard interface for applications written for Inferno*. This provides the interface to the lower levels of the Inferno* system for access to system resources.

The Inferno Reference API* is a set of Limbo* modules that are shareable and dynamic. They are identical on any machine running Inferno*, as either the native operating system, such as the WebPhone, or as an emulated development environment, such as on a Solaris* or Microsoft NT* workstation.

The primary API modules include:

- **System**
System calls for basic string and file manipulation, primitive printing functions and network functions.
- **Security**
Security functions for mutual authentication, message digesting and encryption.
- **Math**
Primitives for scientific computing, including control of the floating point environment, elementary functions of applied mathematics and linear algebra operations.
- **Draw**
Basic raster graphics and windowing functions; provide primitives for graphics functions in Prefab* and Limbo/Tk*.
- **Prefab**
Graphics functions specifically designed for use on non-traditional computing devices, such as with interactive TV applications, including support for infrared remote control devices.
- **Limbo/Tk***
Graphics functions for conventional desktop GUI applications; similar to the Tk Toolkit* defined by John Ousterhout, including menus, buttons, and other widgets.

Other modules provide interfaces to buffered input/output functions, database support, regular expressions and the MPEG device driver.

See the *Inferno Reference Manual* for more information about the Inferno Reference API*.

3.1.4 Dis Virtual Machine

The Dis^{*} virtual machine in Inferno^{*}, provides the execution environment for the applications. The Dis^{*} virtual machine abstracts the hardware from the applications, enabling true portability of the applications.

The Dis^{*} virtual machine models a CISC-like, three-operand, memory-to-memory architecture. Inferno^{*} applications are compiled to a hardware-independent binary image (byte code) for the Dis^{*} virtual machine architecture.

See the System Overview chapter in the *Inferno Reference Manual* for more information about the Dis^{*} virtual machine.

3.1.5 Inferno Kernel

The Inferno^{*} kernel provides the following services:

- Process management
- Memory management
- Namespace management
- Data streaming
- Networking

The kernel is responsible for I/O and hardware connections via device drivers.

See the System Overview chapter in the *Inferno Reference Manual* for more information about the Inferno^{*} kernel.

3.1.6 Device Drivers

Device drivers provide the interface to the hardware devices. The devices for the WebPhone include: an LCD, a touch screen, a keyboard, a Flash ROM, a serial port, a soft modem, and telephony.

In Inferno^{*}, device drivers fit into the model of an hierarchical file system. This differs from UNIX systems where drivers are modeled after regular files. In UNIX, there must be driver routines to satisfy requests to open, read, write, and close the driver file. The file system model for Inferno^{*} device drivers offers interesting possibilities to device driver developers. Complex driver services can be provided by a series of simple interface files, rather than a single heavy-weight interface.

For example, the standard IP driver provides separate file interfaces for data, control, and status for each active connection. By convention, devices are controlled by simply writing a text string to the appropriate control file. The IP data file provides access to the media. Reading the data file returns data received across the network, blocking the file until data is available. Data to be sent across the network is written to the data file. The data is then automatically prefixed with a header and queued for transmission.

3.2 WebPhone Device Drivers

This section describes the WebPhone device drivers.

3.2.1 LCD Display

The LCD supplied with the WebPhone is a monochrome, 4-bit, dual-panel, 640x480 VGA display.

The display driver provides the interface for the output to the LCD according to the VGA specification.

3.2.2 Touch Screen

The WebPhone is supplied with a touch-screen panel for data input. The touch-screen driver maps the input from the panel to the screen.

3.2.3 Keyboard

The keyboard supplied with the WebPhone is controlled through the serial peripheral interface (SPI).

The keyboard driver interfaces with the keyboard controller to interpret input from the keyboard.

3.2.4 Flash ROM

The Flash ROM on WebPhone provides local non-volatile storage. Flash ROM can be written using a special software tools provided with the Inferno Device Driver Kit (DDK)*. The Inferno* boot image and the file system are stored on the Flash.

The Flash driver provided with the WebPhone enables the Flash ROM to be accessed in much the same way as a hard disk, except that the Flash cannot be overwritten unless the Inferno DDK* tools are used.

The WebPhone has 4 Mbyte of Flash ROM. A 128-K area is reserved on the Flash for user-specific data such as configuration information and address book entries.

3.2.5 Serial Port

The serial port on the WebPhone provides RS-232 serial communication with the board.

3.2.6 Soft Modem

The modem is implemented as a 'soft modem' (software modem). It interfaces with the Philips* UCB1200 advanced modem/audio analog front-end connected to the telephone line through the DAA. There is no caller ID on this line and there is no speaker or microphone interface.

The soft modem supports V.34 with speeds up to 33.6 kb/s.

3.2.7 Telephony

The telephony driver includes the following features:

- Ring detection
- Speakerphone and handset volume control
- Full-duplex speakerphone with near-end echo cancellation
- DTMF generation
- Programmable rings
- Line control
- Mute and hold functionality

3.3 WebPhone Applications

The following sections describe the Inferno* applications included with the WebPhone. These applications are written entirely in Limbo*, the native programming language of Inferno*.

3.3.1 Application Toolbar

For the default user interface, the Application Toolbar displays across the lower portion of the screen. Each of the applications on the Application Toolbar launches when the corresponding button on the Application Toolbar is selected.

Figure 3-2. Application Toolbar



The format of the Application Toolbar is set in the *tbsetup* file that is stored in each user's directory (for example, */usr/inferno/tbsetup*). The default *tbsetup* file contains the following entries:

```
#
#This file defines the format of the toolbar
#
#The entries are 2 ":" separated fields
#<Button Label/Bitmap>:<application path>
!EXEC=/dis/lib/plumb.dis
!EXEC=/dis/svc/cs/cs.dis
!EXEC=/dis/wm/digitalInit.dis
!NELEMS=6
!HEIGHT=44
phone:Virtual Phonepad:/dis/wm/phoneGUI.dis
www:HTML Browser:/dis/wm/ct1R2.dis -h414 -w632
mail:Mail:/dis/wm/mailtoolR2.dis
config:Configuration:/dis/wm/swordconfig.dis
addr:PIM:/dis/wm/addrbook.dis
shell:Shell:/dis/wm/sh.dis
```

There are entries for specifying the number of elements (!NELEMS) and height (!HEIGHT). The number of elements are the number of buttons on the toolbar, and height is the height of the toolbar specified in pixels.

The buttons are configured by entries that contain two fields, separated by a colon (:). The first field specifies the button label or the bitmap image (relative to the */icons* directory). The second field specifies the application, and the third field specifies the application to be launched (relative to the */dis* directory).

3.3.2 Telephone

The telephone application allows the user to make and receive phone calls using the handset or hands free speakerphone. The telephone application includes the following features:

- Handset, speaker, and ringer volume controls
- Mute and speaker buttons
- Hold, flash, hang up, and redial buttons
- Status and call timer windows

3.3.3 Web Browser

After connected to an Internet Service Provider (ISP), the web browser is used to view information on the Internet and World Wide Web. The web browser supports the following features:

- HTML 3.2/HTTP 1.0 support
- GIF and JPEG image support
- Cookies (limited)
- Frames (limited)
- Bookmarks—add, view, delete, and save up to 50
- Navigation—back, forward, reload, and home
- History—list of the last 20 sites visited
- Fonts—one fixed and one proportional

Note: The web browser does not support plug-ins, CGI* scripts, or Java* applets.

3.3.4 E-mail

The E-mail application retrieves and sends messages from and to the mail server using the POP3 and SMTP standard protocols. It includes options to reply to and forward messages and send or copy to a single recipient or a group of recipients.

Note: Support for attachments depends on memory size. Currently, attachments are not supported.

3.3.5 Configure

The Configure application sets up the necessary configuration information for making the connection to an Internet Service Provider (ISP) including: dial-up telephone number, account user name and password, primary and secondary DNS IP addresses, and an ISP connection script. The Configure application includes a screen that can be recalibrated by selecting **config** on the Application Toolbar and then selecting **Touch Screen**.

3.3.6 Address Book

The Address Book stores contact information, including name, address, phone numbers, company, and e-mail address. It has options to add, change, delete and search entries. The entries are stored on the WebPhone. When the user selects a telephone number entry in the address book, it is dialed.

3.3.7 Shell

The shell is a command-line interface for entering Inferno* commands.

3.4 WebPhone API

This section describes the Application Programming Interface (API) for advanced telephony applications and the complete Inferno Reference API*.

3.4.1 Inferno Telephony Interface

This section describes the basic functionality supported by the Inferno Telephony Interface* (ITI) version 1.0 for the WebPhone. The functionality is divided into the following categories:

- Status inquiry
- Event detection
- Console control
- Call control

3.4.1.1 Status Inquiry

The ITI* provides status information for lines, devices, and tones.

- Line status

Provides line status indication and detection of a parallel set pick up. Line status indicates if the line is in use, idle, or dead. When a line is connected, the destination phone number is provided, if known.

Note: In a household environment, there can be parallel phone sets in different rooms that share the same phone number.

- Device status

Provides the current status of the hookswitch, speaker, ringer, and handset, and their corresponding volume levels.

- Device capabilities

Provides an interface for applications to query whether certain actions may be performed. Query on a specific line type returns all valid commands and its descriptions for that line type. Query on a specific telephony device returns all valid commands and its descriptions on that device.

- Tones

A query on Tones Supported returns all special tones the device supports. It returns tone index, name, and its description.

3.4.1.2 Event Detection

The ITI* supports the following event detection features.

- Ring detection

Alerts the specified application of the new incoming call. The ITI* reports the ring count and the ringing state change. Ring detection is active whenever the WebPhone is on hook.

- Distinctive ring detection

On POTS lines, multiple addresses are made possible by various systems, such as DID (Direct Inward Dialing) or “distinctive ring,” which are extra-fee services provided by the telephony company. Distinctive ringing detection support is optional.

- Speaker on or off

Reports to the calling application that the speaker is turned off or on.

- Handset on or off hook

Reports to the calling application that the handset is taken off or on hook.

3.4.1.3 Console Control

The ITI* provides control to the physical telephony console if one exists. The calling applications are notified when keys and buttons are pressed. The calling applications can turn on or off LEDs through the ITI*.

- **Dial pad key press**
Reports that the dial pad key is pressed and returns the key position.

Note: The dial pad key press can be interpreted as dialing string as well as number input to an application. It depends on which state the phone is in: when the phone is off-hook, the key press is taken as dialing string; when the phone is on-hook, the dial pad key press is taken as normal number input.

- **Button press**
Reports when a button is pressed and returns the button name. Standard button names include Flash, Forward, Hold, Mute, Speaker, and Line.
- **LED on or off control**
Interface to turn an LED on or off and display a specified pattern (frequency). The state information of the LED is returned.

3.4.1.4 Call Control

The ITI* supports the following call control features.

- **Tone generation**
Generates a specified tone to a certain destination. All tones, including Dual Tone Multi-Frequency (DTMF) tones, supported by a specified device can be obtained through query Tones Supported interface. Tone generation is device-dependent; an application should query the tones supported before it sends the tone generation request.
- **Line control**
Takes the specified line on or off hook. When trying to bring a line off-hook and the line is already off-hook, an error message returns. If a line is taken off-hook while the phone is in RINGING state, the incoming call is answered. When a line is taken off-hook when the phone is in IDLE state, the phone is ready to make an out-going call. When a line is put back to on-hook when the phone is in ACTIVE state, the call is disconnected.
- **Dial**
Provides an interface to dial a dialable address. The dial interface provides the options for pulse or DTMF dialing. Dial supports special dialing characters such as pause, wait and hook flash. Dial returns the result, which can be *connected*, *remote end busy*, or *failed*.
- **Volume control**
Sets the volume of the handset, speakerphone, or ringer, to a specified level.

This chapter describes how to install and set up the hardware and software to build a working StrongARM** SA-1100 WebPhone Reference Design (WebPhone) for demonstration and evaluation purposes. It also describes how to install and use the Inferno Device Driver Kit* (DDK) to develop additional applications and device drivers and to update the Inferno* kernel for the WebPhone.

- For instructions to get you started quickly if your WebPhone board has the Flash pre-installed with the software, see Section 4.1, “Quick Start” on page 4-1.
- For detailed instructions on setting up and installing development tools to write device drivers or to develop applications, see Section 4.2, “Installation and Setup for Development” on page 4-2.
- For instructions on using the applications delivered with the WebPhone, see Section 4.3, “WebPhone Applications” on page 4-5 .

4.1 Quick Start

The following procedure describes how to set up the WebPhone hardware to get you started using the phone and the WebPhone applications that are included in the WebPhone reference design. This procedure assumes the Flash ROM has the software installed on it.

4.1.1 Setting Up the Hardware

To set up the hardware, follow these steps.

1. Connect the appropriate end of the display board’s cable to the board and the other end to the display. The cable end that connects to the display is very fragile so use extreme care.
 - a. While the display board remains face down, slide the display board to the right until two inches of the board is outside the casing.
 - b. Carefully open the black tabs on the right side of the tan connector. Do not remove the black tabs.
 - c. With the silver side up, carefully slide the silver end of the orange plastic tab that has embedded wires into the tan connector.
 - d. Lightly push the black tabs in until the orange plastic tab is snug.
 - e. Slide the display board back into position.
2. If the display is not connected to the board, plug the red and black set of wires on the left side of the display into the board.

Warning: Exercise caution. This is a high-voltage connection.

3. Plug the blue touch panel connector to the black wire aligning the arrows, and plug the other end of the black wire to the board.
4. If not already connected, connect the keyboard ribbon cables to the board.

5. Connect the two phone lines to the ports in the rear of the board. The left connector is for the telephone; the right connector is for the modem.
6. Connect the ac adaptor to the board.
7. Plug the ac adaptor to the wall outlet.

Note: Use only 110 voltage.

8. When booting, the touch calibration routine is started. A series of square blocks displays in the corners. Use the stylus to touch each block. It may take several touches to set the values.

After a successful booting and logon, the Application Toolbar displays.

See Section 4.3, “WebPhone Applications” on page 4-5 for more information about using the software on the WebPhone.

Note: Wait 10 to 15 seconds after saving any configuration information, address book entries, or web browser bookmarks before powering off your screenphone or you will lose all user data.

For some board configurations, if the board loses power you must wait 10 to 15 seconds before reapplying power to the board for the reset to be complete.

4.2 Installation and Setup for Development

The Inferno Device Driver Kit* (DDK) is required for you to use the development tools to write a device driver or to install an application on the reference platform. The Inferno 2.0 SDK* portion of the DDK* package can be used to develop applications for installation on the reference platform.

The following sections describe how to install and set up the Inferno Device Driver Kit* (DDK).

Note: The Inferno Device Driver Kit* (DDK) is not included with the WebPhone reference design. However, these procedures require the DDK* utilities and that the DDK* has been installed on the host system.

The source code for the WebPhone applications is available from Inferno* and are not included with either the WebPhone reference design or the DDK*.

4.2.1 Inferno Device Driver Kit Package

The Inferno Device Drive Kit* (DDK) includes the DDK* tools and documentation and the Inferno 2.0 SDK*.

Because the DDK* CD-ROM includes a version of the Inferno* emulator for use with the DDK* and the tools and object files for building the Inferno* native kernel, you do not need to install or use the separate Inferno 2.0 SDK* when you are writing device drivers and not applications.

Note: Install the host version of Inferno* found on the DDK* CD-ROM and the version of Inferno* that comes with the Inferno 2.0 SDK* in different directories. The Inferno 2.0 SDK* should not be used as the host version of Inferno* for DDK*.

The Inferno Device Driver Kit* (DDK) includes the following documentation:

- *DDK Cookbook*
Installation and build procedures for the DDK*.
- *Inferno Device Driver Kit, Volume 1: Writing Device Drivers*
General information about Inferno* device drivers including a description of the file interface model and case studies of device drivers.
- *Inferno Device Driver Kit, Volume 2: DDK Tools*
Manual pages for functions used in device drivers and documents about tools used to build, install, and debug Inferno* native kernels.

4.2.2 Inferno Device Driver Kit Installation

The Inferno Device Driver Kit* (DDK) CD-ROM includes the files necessary to install the Inferno* operating system (hosted on Solaris/sparc*) as well as the files necessary to build an Inferno* kernel for a native platform.

To install the DDK* you must

- Install Inferno* from the CD-ROM (inferno.tar)
- Install the DDK* file system (ddk.fs)

4.2.2.1 Installing Inferno from the CD-ROM

To install Inferno* from the CD-ROM, run the shell script setup.sh from the host command line.

```
cd {DDKPATH}
setup.sh {INFROOT}
```

Note: {INFROOT} is the directory to install into.
{DDKPATH} is the path to the DDK* CD-ROM.

4.2.2.2 Installing the DDK File System

To install the DDK* file system, follow these steps:

1. From the host command line, copy the DDK* distribution file to a directory that will be visible from Inferno*.

```
host% cd {INFROOT}/inferno
host% cp {DDKPATH}/ddk.fs tmp
```

Note: {INFROOT} is the directory to install into.
{DDKPATH} is the path to the DDK* CD-ROM.
Host commands are preceded by host%.
Inferno* commands are preceded by inferno\$.

2. Start Inferno* and install the DDK* file system.

```
host% ./Solaris/sparc/bin/emu -r. [-7]
inferno$ bind -c '#U' /n/client
inferno$ disk/mkext -vd /n/client < /tmp/ddk.fs
```

Note: If you are using an X-terminal, you may need to restrict emu to a 7-bit color map.

3. Read `/dev/user` to discover the current *user*.

```
inferno$ cat /dev/user; echo ` `
user
inferno$
```

4. Create a directory for the current *user* by copying the `/usr/ddk` directory. You can set up directories for other users in the same way. When you set up directories for other users, make sure that you change permissions on the files; otherwise the files will be owned by the original *user*.

```
inferno$ cp -r /usr/ddk /usr/user
```

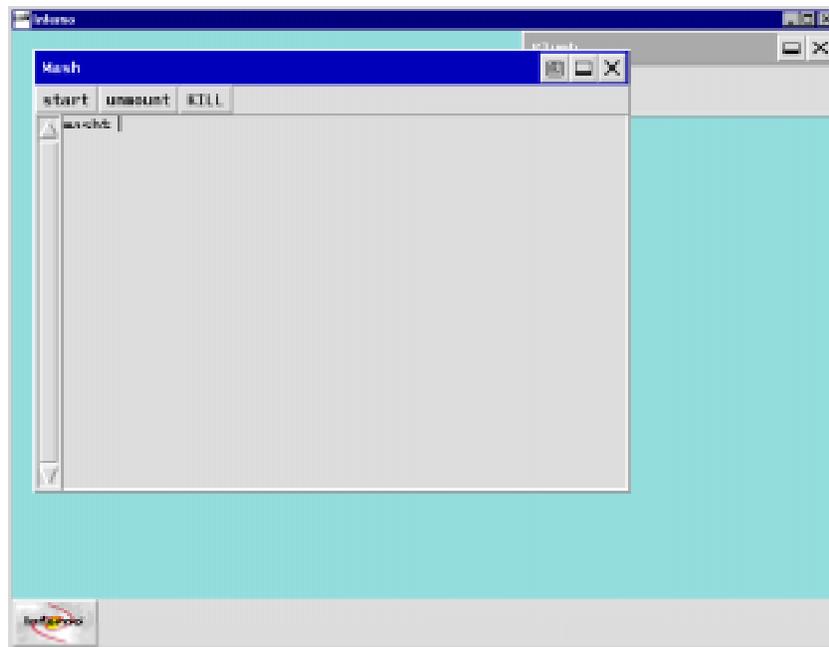
Note: When you use DDK* to communicate using the host serial port, the Inferno* *user* should be identical to the host *user*. This *user* also must have permission to open and write to the serial port

5. Start the Inferno* window manager and log on as *user*.

```
inferno$ wm/logon
```

6. When the Inferno* window manager appears, log on as *user* and accept the license agreement (only appears the first time you log on). The DDK* mash window appears. (Figure 4-1)

Figure 4-1. DDK Mash Window



The mash initialization file (`/usr/user/lib/mashinit`) contains a mash script that is run when mash starts. This file defines the following command and three buttons:

- `sb`
Echoes arguments to `/n/rdbg/sbootconsole` and then reads `/n/rdbg/sbootconsole` for the result
- `start`
Executes the command `lib/rdbgsvr /n/rdbg`
- `unmount`
Executes the command `unmount /n/rdbg`
- `KILL`
Kills any `RDbgSrv` threads

You can customize the Inferno* mash window by adding buttons and built-in commands.

```
7. Exit Inferno* .  
inferno$ shutdown -h
```

4.3 WebPhone Applications

The WebPhone reference design includes a set of applications. See Section 3.3, “WebPhone Applications” on page 3-6 for a general list of application features. This section provides brief instructions for starting and using each of the WebPhone demonstration applications.

Figure 4-2. Toolbar Buttons



The applications can be launched by selecting the corresponding button on the Application Toolbar. (Figure 4-2):

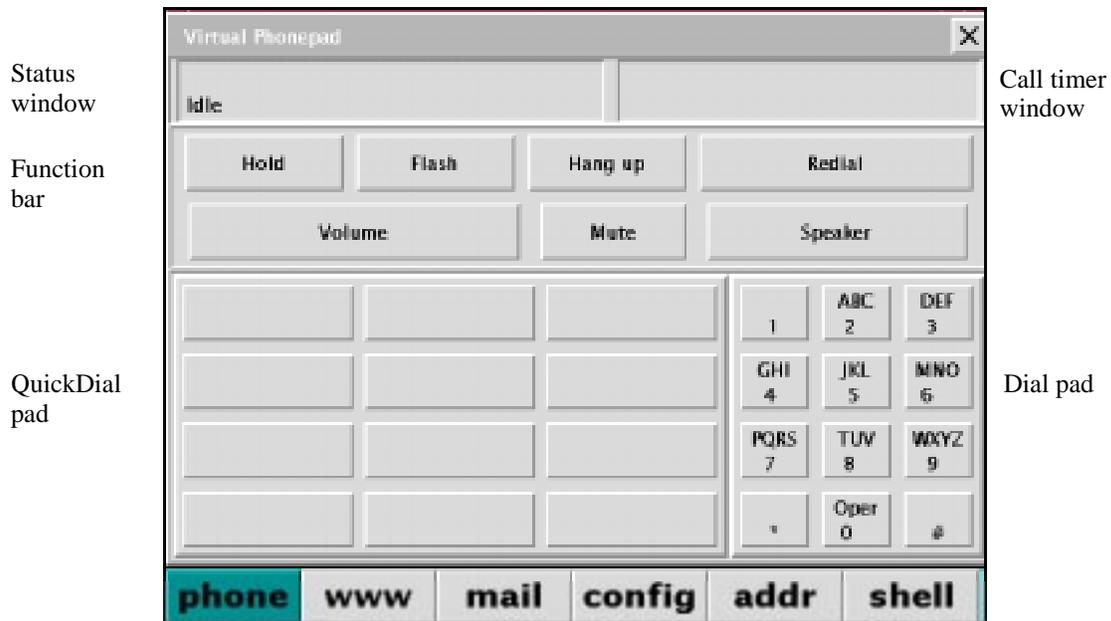
- Telephone (phone)
- Web Browser (www)
- E-mail (mail)
- Configure (config)
- Address Book (addr)
- Shell (shell)

4.3.1 Using Telephone

To use Telephone to display the Virtual Phonepad window and dial a telephone number, follow these steps:

1. Select **phone** on the Application Toolbar to display the Virtual Phonepad window. The **phone** button darkens and the Virtual Phonepad window displays. (Figure 4-3)

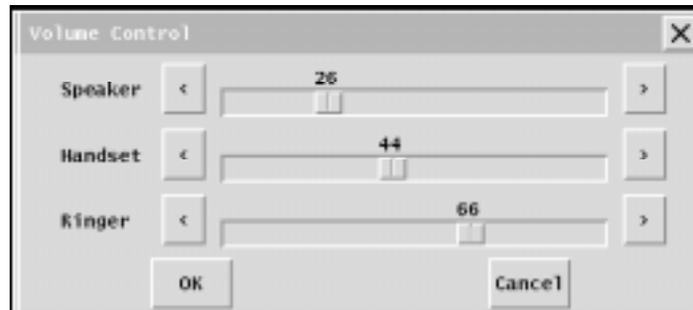
Figure 4-3. Virtual Phonepad Window



The following information is displayed:

- Status of the telephone displayed in the Status window
 - Connect time displayed in the Call Time window (when the handset is lifted)
 - Telephone number dialed on the **Redial** button for later user
2. Select the **Volume** button to adjust the volume of the speaker, the handset, and the ringer using the sliders in the Volume Control window. (Figure 4-4)

Figure 4-4. Volume Control Window



3. Dial a telephone number by using the on-screen keypad or by selecting one of the twelve buttons that has been programmed from the address book. See Section 4.3.5, “Running Address Book” on page 4-14.
4. When the call is complete, select **Hang up** to release the line.
5. Select the **Close** button in the titlebar to return to the Application Toolbar.

4.3.2 Running Web Browser

To use Web Browser, follow these steps.

1. Make sure you have made a connection to the Internet using an Internet Service Provider (ISP). See Section 4.3.4, “Using Configure” on page 4-12 for information about configuring the WebPhone to connect to an ISP.
2. Select **www** on the Application Toolbar to launch the web browser.
The web browser starts with a blank page (Figure 4-5).

Figure 4-5. Browser Start Page


3. You can enter the URL of a new page in the **Location** entry box followed by pressing **Enter** on the keyboard. You can also select **File>>Open Location**, then enter the URL (including the *http://* or the *file://* identifier) in the **Open Location** dialog box.
4. From the **Bookmarks** menu, you can select **Show bookmarks** or **Add bookmarks** to open the **Bookmarks** window. Select an item in the list and press **Go** to go to it, **Cancel** to close the window, or **Edit** to edit the title and/or the URL.
5. From the **Options** menu, you can set the **General** and **Network Preferences**, and specify showing the location and/or status bar, and loading images.

You customize the appearance of the browser in the **General Preferences** window. This includes the options to show the location and status bar and to select the automatic loading images. You can specify an URL for your Home page. Select **OK** to save your preferences that will be effective the next time the browser is started.

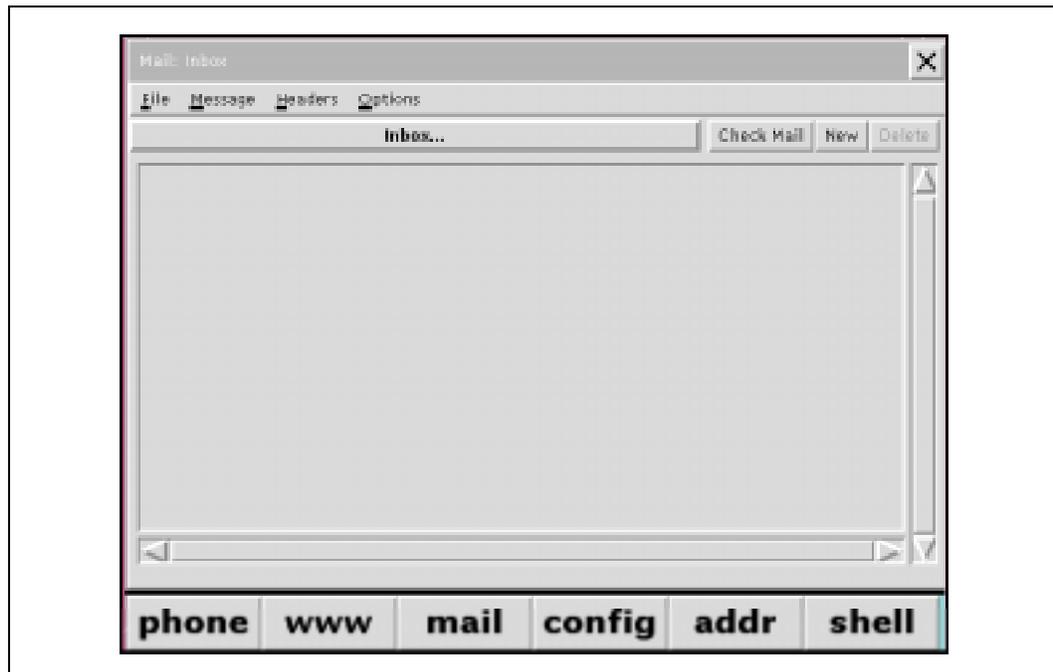
The **Network Preferences** window allows you to specify proxy configuration, if required, and the image memory cache.

4.3.3 Using E-mail

To use E-mail, follow these steps.

1. Make sure you have made a connection to the Internet using an Internet Service Provider (ISP). See Section 4.3.4, “Using Configure” on page 4-12 for information about configuring the WebPhone to connect to an ISP.
2. Select **mail** on the toolbar (Figure 4-6).

Figure 4-6. Mail Menu



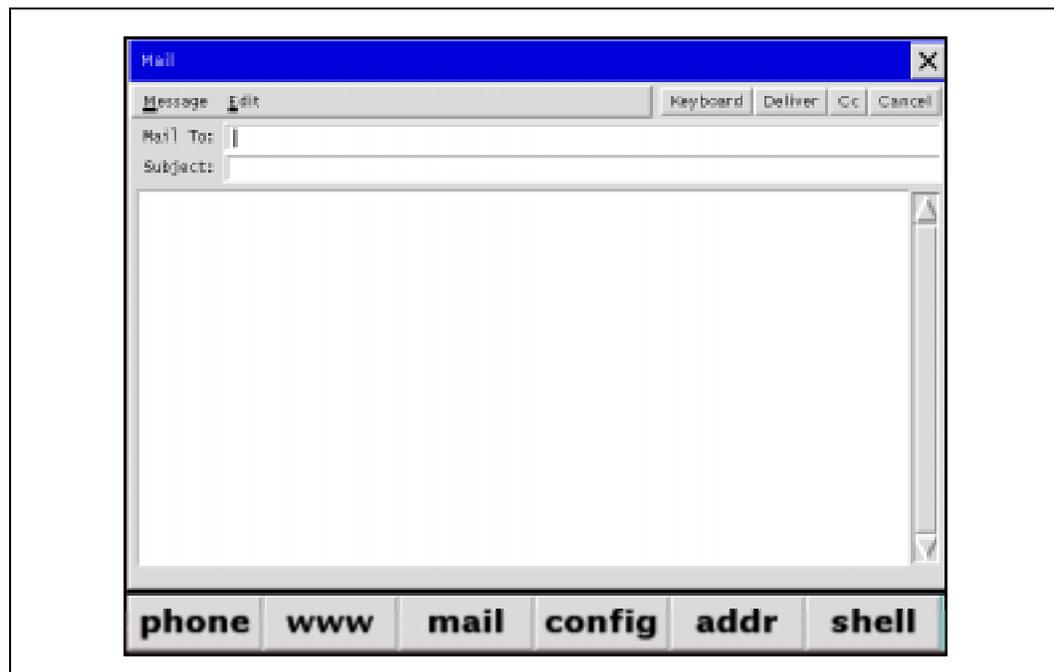
3. Select the **Check Mail** button to connect to your mail server. The first time you make this connection, you are prompted to enter your mail account information. (Figure 4-7)

Figure 4-7. Mail Information Window



4. Enter the requested information. Select **Save** to save this to the *mail.cfg* file in the */usr/inferno* directory. (You can change these options later through the **Options>>Server** and **Options>>User** menus.)
5. Select **Save** to save your mail information and then **OK** to connect to the mail server. If any new mail is on the server, it is downloaded and the message headers display.
6. Select the message header twice in short succession to display the message.
From the **Message** window:
 - Select **Prev** to display the previous message. **Prev** is not active if this is the first message.
 - Select **Next** to display the next message. **Next** is not active if this is the last message.
 - Select **New** to compose a new message.
 - Select **Re:** to reply to the sender of the message.
 - Select **All** to reply to the sender and all recipients of the message.
 - Select **Fwd** to forward the message.
 - Select **Show** to toggle on and off the display of the full message header information.
 - Select **Delete** to delete this message.
 - Select **Close** to close this message.
 - The **Message** menu has options to display the previous and next message, toggle the header information, save this message to a mail folder, delete this message, and to close the window.
 - The **Edit** menu has options to copy selected text and to select the entire message for copying.
 - The **Compose** menu has options to create a new message, reply to the sender of this message, reply to the sender and all recipients, and to forward the message.
7. You can select the **New** button on the **Mail** window to compose a new message. (Figure 4-8)

Figure 4-8. New Message Menu



From this window:

- Enter the recipient's E-mail address in the **Mail to:** field. Separate multiple recipient addresses with a comma. Press the **Enter** key on the keyboard to move to the next field.
- Enter the message subject in the **Subject** field. Press the **Enter** key on the keyboard to move to the message body window.
- Select **CC** to toggle a copy-to field, where you can enter copy-to recipients.
- Select **Deliver** to deliver this message to your outgoing message (SMTP) server.
- Select **Cancel** to cancel this message.
- Select **Keyboard** to toggle the display of an on-screen keyboard that you can alternatively use to "type" the message. (Figure 4-9)

Figure 4-9. Keyboard Option



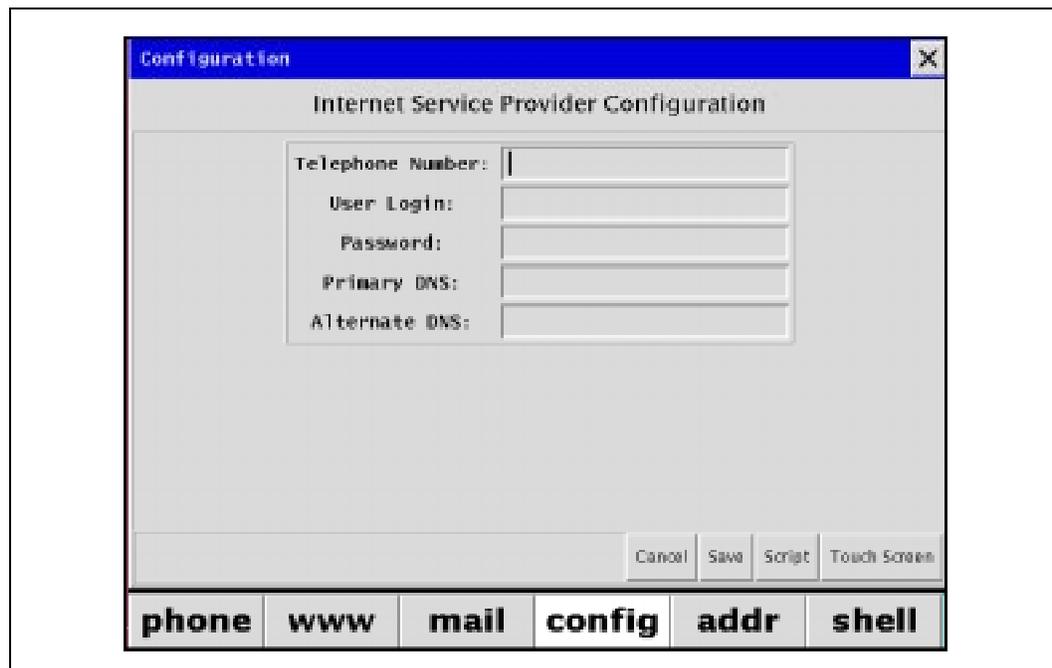
- The **Message** menu has options to deliver this message, add copy-to recipients, toggle the display of an on-screen keyboard, and to cancel this message.
- The **Edit** menu has options to cut, copy and paste text, and to select all the text in the message.
- Select the **Close** button in the titlebar to return to the Application Toolbar.

4.3.4 Using Configure

To use Config, follow these steps.

1. Select **config** on the Application Toolbar. (Figure 4-10)

Figure 4-10. Configuration Window



2. Select **Touch Screen** to calibrate your screen. A grid appears with a small box that includes a plus sign (+) in the upper left-hand corner of the screen. When you select the plus sign (+), the four small grid boxes darken and the plus sign (+) moves to the upper right-hand corner of the screen. The plus sign (+) moves to each of the screen corners until the touch screen is calibrated. The Configuration window displays again.
3. Enter the following configuration information about your account:
 - a. Telephone number of your Internet Service Provider (ISP)
 - b. User login name
 - c. Login password
 - d. Primary DNS IP address
 - e. Alternate DNS IP address
4. If you need to use a script file to connect to your ISP, select **Script** to display the Script Configuration Window.
Use the keyboard to enter your connection script. Select the **Use Connection Script** box to darken the box and then select **Done** to close the Script Configuration Window.
5. Select **Save** to save the ISP configuration information.

The WebPhone is configured for you to use the web browser and E-mail applications. When an application attempts to connect to the ISP, a status bar and status messages display. A connection should be established within 60 seconds.

4.3.5 Running Address Book

To run Address Book, follow these steps.

1. Select **addr** on the toolbar to launch the Address Book. (Figure 4-11)

Figure 4-11. Address Book Window



In the Address Book window you can review your address book entries and add, edit or delete address book entries that can include a URL. You can search for a particular entry by entering the name in the search bar.

2. Select an entry and the summary information displays in the Summary window.
3. Select a telephone number shown in the Summary window and the phone application starts and dials the number.
4. Select the **Exit** button to close your address book.

5. If you select the **Add** or **Edit** buttons, a Record dialog displays for you to enter information for a specific address book entry. (Figure 4-12)

Figure 4-12. Record Dialog

The Record dialog box contains the following elements:

- Title bar: Record
- Toolbar: Close, Save, Clear, Cut, Copy, Paste, QuickDial (checkbox)
- Fields: LastName, MiddleName, FirstName
- Phone fields: Business, Home, Fax, Pager
- Address field
- City, State, Zip fields
- Company field
- Email field with Launch mail button
- WebSite field with Launch www button
- Note field
- Footer bar: phone, www, mail, config, addr, shell

6. Select the **QuickDial** check button to have your address book entry appear on one of the twelve buttons on your Virtual Phonepad. Ignore this step to add the entry to your address book.
7. Select a telephone number button (**Business**, **Home**, **Fax**, or **Pager**) and the phone application starts and dials the number.
8. Select the **Save** button to save your changes and the **Close** button in the titlebar to return to the Application Toolbar.

4.3.6 Using the Shell

To use the Shell, follow these steps.

1. Select **shell** to display the Inferno* shell. (Figure 4-13)

Figure 4-13. Shell Window



2. Enter any Inferno* command at the shell prompt to give you command-line access to the entire file system available to the WebPhone.
3. Select the **Close** button in the title bar to return to the Application Toolbar.

See the Inferno Command Line Utilities chapter in the *Inferno User's Guide* for more information about Inferno commands.

Note: The Shell is not a supported application.

Support, Products, and Documentation

If you need technical support, a *Product Catalog*, or help deciding which documentation best meets your needs, visit the Intel World Wide Web Internet site:

<http://www.intel.com>

Copies of documents that have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling **1-800-332-2717** or by visiting Intel's website for developers at:

<http://developer.intel.com>

You can also contact the Intel Massachusetts Information Line or the Intel Massachusetts Customer Technology Center. Please use the following information lines for support:

For documentation and general information:	
Intel Massachusetts Information Line	
United States:	1-800-332-2717
Outside United States:	1-303-675-2148
Electronic mail address:	techdoc@intel.com

For technical support:	
Intel Massachusetts Customer Technology Center	
Phone (U.S. and international):	1-978-568-7474
Fax:	1-978-568-6698
Electronic mail address:	techsup@intel.com



