SONY

OPEN-R SDK

Installation Guide

115-01

Notes on This Document

Notes on Using This Document

- The contents provided by this document (PDF files) are intended only for supplying information.
- The contents provided by this document (PDF files) are subject to change without notice.
- We are not responsible for errors or omissions in technical or editorial aspects concerning the contents described in this document. We also are not responsible for technical measures, correspondence, execution according to this document, as well as for the results occurred by them such as inevitable, indirect or incidental damages.

Notes on Copyright

About Trademarks

Index

Chapter1 Introduction

1.1 Preparation

The OPEN-R SDK runs on the following platforms.

- Windows 2000 professional or Windows XP

Notes
The OPEN-R SDK runs on not only Windows, but also other operating systems. The following environment is needed for using the OPEN-R SDK.

Standard UNIXcommands such as sh, cp, perl
GNU make
GNU binutils, gcc and newlib (built for a MIPS cross-development environment)

These tools are compatible with normal UNIX environments. If you are using the Windows cygwin environment, you can download the binary MIPS cross-compilation tools for this environment from our web site. However, if you need to run these tools in another UNIX based environment, please refer to [Notes on 2.2 gcc]. On non-Windows/cygwin platforms, you can use a shell script that we provide to automatically build tool binaries from source code. We confirmed that this script and the OPEN-R SDK tools work on Linux.

The following hardware is necessary to perform software development with the OPEN-R SDK.

- PC
  CPU:Pentium 233MHz or above
  Memory: 64MB or more
  Free space on a hard disk drive: more than 200MB
- Memory Stick reader/writer (an internal or external device)
- AIBO ERS-7, ERS-210, ERS-220, ERS-210A, or ERS-220A
- Wireless LAN environment
  PC : You can choose either (a) or (b):

  (a) An IEEE802.11b-compliant wireless LAN card
  (b) A wired LAN card and an IEEE802.11b-compliant access point

  AIBO: ERS-210 and ERS-220 series need an AIBO wireless LAN card (ERA-201D1) installed; ERS-7 does not require this because ERS-7 contains a built-in wireless LAN card.
- AIBO Programming Memory Stick

Notes
Only AIBO Programming Memory Sticks can be used to run programs built with the OPEN-R SDK.

It is assumed that the user of the OPEN-R SDK is familiar with the following.

- C++ Programming
- How to use GNU development tools
  gcc/g++, ld, make, etc
- How to use Cygwin tools
  Shells, UNIX-like commands, correspondence between UNIX-style paths and Windows-style paths, etc

Notes
It is recommended that you have a tool for reading text files with the UNIX-style newline code (^J only), because all text files in the OPEN-R SDK, including sample programs and header files, have UNIX-style newlines.

1.2 Download files

The following is the list of files for downloading

- OPEN-R SDK

OPEN_R_SDK-1.1.5-r1.tar.gz	OPEN-R SDK
OPEN_R_SDK-sample-1.1.5-r1.tar.gz	Sample programs
OPEN_R_SDK-docE-1.1.5-r1.tar.gz	Manuals

For windows platforms

cygwin-packages-1.5.5-bin.exe	Cygwin binaries
mipsel-devtools-3.3.2.bin-r1.tar.gz	MIPS cross development tools

- For ERS-210 users

upgrade-OPEN_R-1.1.3-r2.tar.gz	FlashUpdater for ERS-210
upgrade-OPEN_R-1.1.3-r2.exe	(same as above)

- Source files

cygwin-packages-1.5.5-src.tar.gz	Source files of Cygwin
binutils-2.14.tar.gz	Source files of binutils
gcc-3.3.2.tar.gz	Source files of gcc
newlib-1.10.0.tar.gz	Source files of newlib
cygipc-2.00-src.tar.gz	Source files of cygIPC

- Script

build-devtools--3.3.2-r1.sh	Shell script for building the binutils, gcc, newlib
build-devtools--3.3.2-macosx-r1.sh	Shell script for building tools on Mac OS X

Chapter2 Installation

2.1 Cygwin

If you are using a Windows environment, you should first install Cygwin. Cygwin is the UNIX environment that works on Windows. Install Cygwin with the following steps.

1 Double-click cygwin-packages-1.5.5-bin.exe.

2 Specify the folder to Unzip to, and click [Unzip]. This process creates the directory cygwin-packages-1.5.5. Execute setup.exe in this new directory.

3 The version number of setup.exe will be displayed. Click [Next->].

4 Choose [Install from Local Directory]. Click [Next->].

5 Specify the directory where Cygwin will be installed. You should leave this as "C:\cygwin", unless you have a need to install it elsewhere. Select [Unix] for the text file type and click [Next->].

6 Specify the directory where setup.exe exists. Click [Next->].

7 The window for selecting packages will be displayed. Leave it as the default and click [Next->]. Then, installation starts.

> **Notes**
> This package only includes the minimum set of tools necessary for using the OPEN-R SDK. You can install additional tools by re-executing setup.exe and choosing [Install from Internet] at step 4. However, please note that if you choose to update existing packages, the OPEN-R SDK may not be fully compatible with these updated packages.

In the rest of this document, we will unpack packages in cygwin's path by using its tar command. In this explanation, whenever you see cygwin's path referred to (for example, /usr/local), it indicates cygwin's /usr/local, NOT c:\usr\local (from the root of your hard drive).

2.2 gcc

Install gcc with the following steps.

1 Unpack the package. (/xxx is your directory where the downloaded file is placed)

```
cd /usr/local
tar zxvf /xxx/mipsel-devtools-3.3.2-bin-r1.tar.gz
```

A /usr/local/OPEN_R_SDK directory is created. The following tools and libraries targeting 'mipsel-linux' are installed under the directory OPEN_R_SDK. (Linux is a target name here, but this does not mean that Linux works on an AIBO.)

GNU binutils-2.14
GNU gcc-3.3.2
newlib-1.10.10

Notes

For non-Windows platforms, follow this procedure instead:
Place the following files into the same directory as the one that build-devtools-3.2-r1.sh is in, and execute build-devtools-3.2-r1.sh in this directory.

binutils-2.14.tar.gz
gcc-3.3.2.tar.gz
newlib-1.10.10.tar.gz

2.3 OPEN-R SDK

Install the OPEN-R SDK with the following steps.

1 Unpack the package. (/xxx is your directory where the downloaded file is placed)

```
cd /usr/local
tar zxvf /xxx/OPEN_R_SDK-1.1.5-r1.tar.gz
```

The following directories will be created.

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7   (for ERS-7)
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS200  (for ERS-210/220)
/usr/local/OPEN_R_SDK/RP_OPEN_R
       (for Remote Processing OPEN-R)
```

Notes

(1)MS_ERS7 directory is only to be used for ERS_7; MS_ERS200 directory is only to be used for ERS-210/220. MS_ERS7 directory is used as a typical example in the chapter3 and chapter4. You can confirm that you are using the correct directory by checking /OPEN-R/VERSION.txt.

(2)If you set an environmental variable called OPENRSDK_ROOT, you can install the OPEN-R SDK in a directory other than the default (which is /usr/local/OPEN_R_SDK).

(Example of .bashrc)
export OPENRSDK_ROOT=/home/aibo/OPEN_R_SDK

2 To install Remote Processing OPEN-R, run the setup-rp-openr script.

```
/usr/local/OPEN_R_SDK/RP_OPEN_R/bin/setup-rp-openr
```

2.4 Sample programs

Install the sample programs with the following steps.

1 Unpack the package. (/mydir is a given directory, and /xxx is your directory where the downloaded file is placed)

```
cd /mydir
tar zxvf /xxx/OPEN_R_SDK-sample-1.1.5-r1.tar.gz
```

The directory sample will be created and the sample programs are installed in the directory.

2.5 AIBO Built-in Flash ROM Upgrade

To run your program created with the OPEN-R SDK, it is necessary for AIBO ERS-210 to update the version of the flash ROM.
However, upgrading the flash ROM is not necessary for ERS-220, ERS-210A, and ERS-220A. In case a sticker indicating "OPEN-R Ver 1.1.2" or "OPEN-R Ver 1.1.3" is attached to the core unit, this procedure is not necessary.

Upgrade the flash ROM with the following steps.

1 Move to a given directory and unpack the package. (/mydir is a given directory, and /xxx is your directory where the downloaded file is placed)

```
cd /mydir
tar zxvf /xxx/upgrade-OPEN_R-1.1.3-r2.tar.gz
```

The directory upgrade will be created.

2 The steps of upgrading the flash ROM is described in README_E.txt under the directory upgrade.

Chapter3 Building & running HelloWorld

You should now build and run one of the included sample programs to confirm the success of the installation. “HelloWorld” consists of OPEN-R objects, which correspond to HELLO.BIN and POWERMON.BIN, and simply prints the message “HelloWorld” to the wireless console. (Note: Whenever you see the word “object” in this manual, it indicates an “OPEN-R object”.) The “HelloWorld” directory contains the following sub-directories:

- HelloWorld/HelloWorld
  It contains the source code of the “HelloWorld” object.

- HelloWorld/MS
  It contains files that should be placed on the AIBO Programming Memory Stick. Initially, only a file OBJECT.CFG exists later, the executable files will be copied here.

3.1 Building

Build the sample program with the following steps.

1 Build the executable file. (/mydir is the directory where you installed the sample programs.)

```
cd /mydir/sample/common/HelloWorld
make
make install
```

The executable files are created and copied to the directory MS.

Notes
The file name must comply 8+3 format (xxxxxxxx.xxx).
The file is also gzipped.

3.2 Running

3.2.1 Execution in a Wireless LAN environment

Set up the wireless LAN environment with the following steps.

1 Copy the following two OPEN-R directories to a blank AIBO Programming Memory Stick. (/mydir is the directory where you installed the sample programs.)

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/memprot/OPEN-R
/mydir/sample/common/HelloWorld/MS/OPEN-R
```

2 Edit WLANCONF.TXT on the AIBO Programming Memory Stick and setup the wireless LAN parameters for AIBO. Refer to [3.2.2 How to set up WLANCONF.TXT] for details.

3 Insert an AIBO wireless LAN card and an AIBO Programming Memory Stick into AIBO. Then boot AIBO.

4 To connect your PC to AIBO's wireless console, start the following telnet program on your PC. The AIBO wireless console uses a TCP port number 59000 and telnet protocol. (You can alternatively use other third-party telnet programs if you prefer.)

```
> telnet (IP address of AIBO) 59000
```

If the characters [!!! Hello World!!!] are displayed after a series of system information is displayed, your AIBO is working normally in the OPEN-R SDK environment.

5 Press the pause button to shutdown AIBO.

Notes

There are two ways to connect your PC to AIBO: with a wireless access point, or with an additional wireless LAN adapter in ad-hoc mode. To make sure AIBO is successfully connected to your network, you can execute the following console command after booting AIBO

```
ping (IP address of AIBO)
```

3.2.2 How to set up WLANCONF.TXT

There are two wireless LAN configuration files for AIBO, WLANDFLT.TXT and WLANCONF.TXT. WLANDFLT.TXT contains the default configuration for AIBO's wireless LAN card, and should not be edited. WLANCONF.TXT is a file that contains your custom configuration. When AIBO boots, it looks for WLANCONF.TXT first and uses the information in that file if it is found. If not, it uses the information in WLANDFLT.TXT instead.

Copy WLANDFLT.TXT in /OPEN-R/SYSTEM/CONF/ of an AIBO Programming Memory Stick to WLANCONF.TXT, and edit it. (The following is the default setting of WLANCONF.TXT.)

```
HOSTNAME = AIBO
ETHER_IP = 10.0.1.100
ETHER_NETMASK = 255.255.255.0
IP_GATEWAY = 10.0.1.1
ESSID = AIBONET
WEPENABLE = 1
WEPKEY = AIBO2
APMODE = 2
CHANNEL = 3
#DNS_SERVER_1=10.0.1.1
#DNS_DEFDNAME=example.net
#USEDHCP = 1
```

Here are the descriptions of the above items.

HOSTNAME	Defines the name AIBO uses on the wireless network. This name can be up to 8 alphanumeric characters and must contain at least one letter.
ETHER_IP	Set AIBO's IP address. This parameter must be specified when the USE_DHCP value is set to 0.
ETHER_NETMASK	Sets the IP subnet mask. This parameter must be Specified when the USE_DHCP value is set to 0.
IP_GATEWAY	Sets the gateway IP address. If a gateway is not present on the network, set this value to be the same as the ETHER_IP address. This parameter must be specified when the USE_DHCP value is set to 0.
ESSID	The name of the wireless network. Up to 32 alphanumeric characters and symbols are allowed.
WEPENABLE	Sets whether to use WEP(wireless encryption system). Set this value to 0 if disabled, and 1 if enabled.
WEPKEY	Sets the WEP key(wireless passkey). Alphanumeric passkeys must be 5 or 13 characters. Hexadecimal passkeys must be 10 or 26 characters(0-9,A-F,a-f). (Supports WEP64 (40 bit) and WEP128(104 bit). If using WEP64, use 5 alphanumeric characters, or "0x" followed by 10 hexadecimal characters. If using WEP128, use 13 alphanumeric characters, or "0x" followed by 26 hexadecimal characters. Example: For WEP64(40bit) String: WEPKEY=AIBO2 Hexadecimal: WEPKEY=0x414924f32
APMODE	Specify an AIBO wireless LAN mode. If you connect AIBO using Ad Hoc Demo Mode, use 0 (zero). If you connect AIBO with Infrastructure Mode, use 1. If you want AIBO to automatically detect the

	correct setting, use 2. Note: if both types of network are present, AIBO defaults to Infrastructure Mode.
CHANNEL	Specify the channel when Ad Hoc Demo Mode is used. Use 1 to 11 for this.
DNS_SERVER_1	Sets the DNS server IP address. This parameter must be specified when the USE_DHCP is set to 0.
DNS_DEFDNAME	Specify default domain name.
USE_DHCP	Defines whether to use DHCP(Dynamic Host Configuration Protocol) to automatically assign AIBO IP address. Set this value to 0 if disabled, and 1 if enabled.

Notes

1 In case you directly connect your PC with a wireless LAN card inserted to AIBO without an access point (IBSS Peer-to-peer Mode or Ad Hoc Demo Mode),

Set the values in ESSID, WEPENABLE and WEPKEY to the same ones in you PC. In case you connect with IBSS Peer-to-Peer Mode, set APMODE = 1. In case you connect with Ad Hoc Demo Mode, set APMODE = 0 and set the value in CHANNEL to the same one in your PC.

2 In case you connect your PC to AIBO via an access point, set each value in ESSID and WEPKEY to the same values as those in the access point. Set APMODE = 1 to work with Infrastructure Mode.

Chapter4 Remote Processing OPEN-R

Notes

1 For communication between AIBO and remote host through a wireless LAN, it is necessary to boot AIBO before accessing it from the remote host.

2 Remote Processing OPEN-R runs on cygwin and Linux.

4.1 Building & running ObjectComm

4.1.1 Running on AIBO

1 Build the executable file. (/mydir is the directory where you installed the sample programs)

```
cd /mydir/sample/common/ObjectComm
make install
```

2 Copy these two OPEN-R directories to a blank AIBO Programming Memory Stick.

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/nomemprot/OPEN-R
/mydir/sample/common/ObjectComm/MS/OPEN-R
```

3 Insert the AIBO Programming Memory Stick into AIBO, then boot it. Subsequent operations (usage of wireless console, how to shutdown AIBO, etc.) are the same as before.

4.1.2 Running on host

1 Run ipc-daemon.

```
/usr/bin/ipc-daemon2
```

2 Build the executable file.

```
cd /mydir/sample/common/ObjectComm/RP/host
make install
```

Run start-rp-openr

```
/usr/local/OPEN_R_SDK/RP_OPEN_R/bin/start-rp-openr
[pid:29444,msqid:196610,oid:0x00030002] oserviceManager
[pid:29445,msqid:229379,oid:0x00038003] tcpGateway
[pid:29446,msqid:262148,oid:0x00040004]
        MS/OPEN-R/MW/OBJS/SUBJECT.BIN
[pid:29447,msqid:294917,oid:0x00048005]
        MS/OPEN-R/MW/OBJS/OBSERVER.BIN
SampleSubject::Ready() : ASSERT READY
SampleObserver::Notify() !!! Hello world !!!
SampleSubject::Ready() : ASSERT READY
```

```
SampleObserver::Notify() !!! Hello world again !!!
SampleSubject::Ready() : ASSERT READY
```

3 Type ctrl-c to terminate the program.

4.2 Building & running LMasterRSlave

4.2.1 Running on AIBO.

1 Build the executable file.

```
cd /mydir/sample/ers7/LmasterRSlave7
make install
```

2 Copy the following two OPEN-R directories to a blank AIBO Programming Memory Stick.

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/nomemprot/OPEN-R
/mydir/sample/ers7/LmasterRSlave7/MS/OPEN-R
```

3 Insert the AIBO Programming Memory Stick into AIBO. Then boot AIBO. Subsequent operations (usage of wireless console, how to shutdown AIBO, etc.) are the same as before.

4.2.2 Distributed execution on AIBO and host

0 Run ipc-daemon (for Cygwin only)

```
/usr/bin/ipc-daemon2
```

Procedures for AIBO

1 Build the executable file.

```
cd /mydir/sample/ers7/LmasterRSlave7/RP/robot
make install
```

2 Copy the following two OPEN-R directories to a blank AIBO Programming Memory Stick.

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/nomemprot/OPEN-R
/mydir/sample/ers7/LmasterRSlave7/RP/robot/MS/OPEN-R
```

3 Edit the following file according to your wireless setup.

```
/OPEN-R/SYSTEM/CONF/WLANCONF.TXT
```

4 insert the AIBO Programming Memory Stick into AIBO, then boot it. Subsequent operations (usage of wireless console, how to shutdown AIBO, etc.) are the same as before.

Procedures for host

1. Build the executable file.

```
cd /mydir/sample/ers7/LmasterRSlave7/RP/host
make install
```

2. Edit the following file.

```
MS/OPEN-R/MW/CONF/HOSTGW.CFG
```

Change 10.0.1.100 to the IP address of your AIBO.

```
!ROBOT_PROXY 59001 10.0.1.100
TCPGateway.Sensor.OSensorFrameVectorData.S 59001 10.0.1.100
TCPGateway.Effector.OCommandVectorData.O 59003 10.0.1.100
```

3. Run start-rp-openr

```
/usr/local/OPEN_R_SDK/RP_OPEN_R/bin/start-rp-openr
```

4. Type ctrl-c to terminate the program.

4.3 Limitations of Remote Processing OPEN-R

- The configuration of the system objects on AIBO is limited to 'nomemprot' when executing a program distributed between AIBO and a remote host. Therefore, please use the following directory as the OPEN-R directory to be copied to an AIBO Programming Memory Stick.

```
/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/nomemprot/OPEN-R
```

- Description using [RobotDesign] as shown below is not supported in OBJECT.CFG and CONNECT.CFG on the host.

```
#
# OBJECT.CFG
#
[ERS-210]
/MS/OPEN-R/MW/OBJS/ERS-210.BIN
[ERS-220]
/MS/OPEN-R/MW/OBJS/ERS-220.BIN
```

- Unavailable OPEN-R API for a host program

```
OPENR::ControlPrimitive()
OPENR::NewSoundVectorData()
OPENR::DeleteSoundVectorData()
```

```
OPENR::NewCdtVectorData()
OPENR::DeleteCdtVectorData()
OPENR::SetCdtVectorData()
OPENR::Shutdown()
OPENR::ObservePowerStatus()
OPENR::UnobservePowerStatus()
OPENR::FindDesignData()
OPENR::DeleteDesignData()
OPENR::Fatal()
```

- ANT (Aperios Network Toolkit) class library is not available for a host program.

Notes on Copyright

- Copyrights of newlib

The OPEN-R SDK uses newlib-1.9.0. It is a collection of software from several sources. Each has own copyrights:

(1) University of California, Berkeley

Copyright (c) 1990 The Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE MPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

(2) DJ Delorie

Copyright (C) 1991 DJ Delorie, 24 Kirsten Ave, Rochester NH 03867-2954

This file is distributed under the terms listed in the document "copying.dj", available from DJ Delorie at the address above.
A copy of "copying.dj" should accompany this file; if not, a copy should be available from where this file was obtained. This file may not be distributed without a verbatim copy of "copying.dj".

This file is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

(3) David M. Gay at AT&T

The author of this software is David M. Gay.
Copyright (c) 1991 by AT&T.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
(4) Advanced Micro Devices

Copyright 1989, 1990 Advanced Micro Devices, Inc.

This software is the property of Advanced Micro Devices, Inc (AMD) which specifically grants the user the right to modify, use and distribute this software

provided this notice is not removed or altered. All other rights are reserved by AMD.

AMD MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS SOFTWARE. IN NO EVENT SHALL AMD BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS SOFTWARE.

So that all may benefit from your experience, please report any problems or suggestions about this software to the 29K Technical Support Center at 800-29-29-AMD (800-292-9263) in the USA, or 0800-89-1131 in the UK, or 0031-11-1129 in Japan, toll free. The direct dial number is 512-462-4118.

Advanced Micro Devices, Inc.
29K Support Products
Mail Stop 573
5900 E. Ben White Blvd.
Austin, TX 78741
800-292-9263

(5) C.W. Sandmann

Copyright (C) 1993 C.W. Sandmann
This file may be freely distributed as long as the author's name remains.

(6) Eric Backus

(C) Copyright 1992 Eric Backus

This software may be used freely so long as this copyright notice is left intact. There is no warrantee on this software.

(7) Sun Microsystems

Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.

Developed at SunPro, a Sun Microsystems, Inc. business. Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

(8) Hewlett Packard

(c) Copyright 1986 HEWLETT-PACKARD COMPANY

To anyone who acknowledges that this file is provided "AS IS"
without any express or implied warranty:
permission to use, copy, modify, and distribute this file for any purpose is hereby granted without fee, provided that the above copyright notice and this notice appears in all copies, and that the name of Hewlett-Packard Company not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose.

(9) Cygnus Solutions

Copyright (c) 1994, 1997 Cygnus Solutions.
All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms

and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed at Cygnus Solutions. Cygnus Solutions may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE MPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.