MAXAda for RedHawk Linux

Version 3.4-Beta Release Notes

October 2003

0898537-3.4-Beta

READ ME BEFORE INSTALLING THIS PRODUCT

CONCURRENT COMPUTER CORPORATION

Contents

1.0. Introduction

MAXAda™ for RedHawk™ Linux® supports development of Ada95 programs running under Concurrent Computer Corporation's RedHawk Linux real-time operating system. MAXAda for RedHawk Linux processes the Ada language as specified by the *Reference Manual for the Ada Programming Language, ANSI/ISO/IEC-8652:1995*, referred to in this document as the *Ada 95 Reference Manual* or RM.

MAXAda for RedHawk Linux 3.4-Beta is a pre-production release. See "Known Issues" on page 8 and "Additional Considerations" on page 10 for details.

NOTE

Please contact the Concurrent Software Support Center as described in "Direct Software Support" on page 13 with respect to any problems encountered with this release. Any and all feedback would be greatly appreciated.

MAXAda for RedHawk Linux 3.4-Beta is based on MAXAda 3.1 which was certified using Version 2.1 of the Ada Conformity Assessment Test Suite (certificate #A981215E2.1-047).

In addition, MAXAda for RedHawk Linux 3.4-Beta includes POSIX® 1003.5, a complete implementation of the Institute of Electrical and Electronic Engineers (IEEE) standard IEEE-Std-1003.5-1992.

2.0.   Documentation

Table 2-1 lists the MAXAda for RedHawk Linux 3.4-Beta documentation available from Concurrent.

Table 2-1. MAXAda for RedHawk Linux Version 3.4-Beta Documentation

Manual Name	Pub. Number
MAXAda Reference Manual	0890516-100
MAXAda for RedHawk Linux Version 3.4-Beta Release Notes	0898537-3.4-Beta

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822 or 1-305-931-2408.

Additionally, the manuals listed above are available:

- online using the RedHawk Linux utility, **nhelp**

- in PDF format in the **documentation** directory of the MAXAda for RedHawk Linux 3.4-Beta Installation CD

- on the Concurrent Computer Corporation web site at www.ccur.com

3.0.   Prerequisites

Prerequisites for MAXAda for RedHawk Linux Version 3.4-Beta for both the host system and target system are as follows:

3.1.   Host System

3.1.1.   Software

- RedHawk Linux *or* Red Hat® 8.0[*]

3.1.2.   Hardware

- an Intel®-based PC - 300 MHz or higher (recommended minimum configuration)
- 64MB physical memory (recommended minimum configuration)

3.2.   Target System

3.2.1.   Software

- RedHawk Linux 1.4

3.2.2.   Hardware

- any Concurrent iHawk system

---

[*] This product has been tested on RedHawk Linux 1.4 and Red Hat Linux 8.0. However, this product has not been tested with versions of Linux supplied by other vendors or on other versions of Red Hat Linux.

4.0. System Installation

In addition to the RPMs for the MAXAda for RedHawk Linux product, additional RPMs are required to upgrade your system for use with MAXAda. Specifically, this release includes enhancements to the following products:

- NightView™

- NightTrace™

- NightBench™

Normally, these products are shipped separately.

Since this is a Beta installation of MAXAda, the MAXAda for RedHawk Linux 3.4-Beta CD includes all new RPMs for those additional products, as well as the base RPMs for operation of the NightStar™ Tools. It does not include the RedHawk 1.4 operating system.

RedHawk 1.4 is required for proper execution and debugging of MAXAda-built programs. You can compile and link programs on a Red Hat system, but execution of those programs requires RedHawk. Specifically, RedHawk 1.4 contains bug fixes and enhancements for correct execution of Ada tasking programs. Non-tasking programs may execute without detected problems on earlier versions of RedHawk.

A single command installs (or uninstalls) all of the RPMs needed to support MAXAda for RedHawk Linux 3.4-Beta, but does not address the installation of RedHawk 1.4. The following table shows the RPMs that will be installed.

Item	RPM
Élan License Manager	`ccur-elanlm-5.0-7.i386.rpm`

MAXAda	`ccur-MAXAda-invoker-3.4-4.i386.rpm`
	`ccur-MAXAda-i86_3.4_beta-001-7.i386.rpm`
	`ccur-MAXAda-rm-3.4-1.i386.rpm`

NightBench	`ccur-nbench-2.3-901.i386.rpm`
	`ccur-nbench-ada-2.3-901.i386.rpm`
	`ccur-nbench-c-2.3-901.i386.rpm`

NightProbe	`ccur-nprobe-2.5.1-003.i386.rpm` `ccur-nprobeserv-2.5.1-003.i386.rpm`
NightSim	`ccur-nsim-3.3.1-005.i386.rpm` `ccur-nsimserver-3.3.1-005.i386.rpm`
NightStar	`ccur-nstar-1.2-005.i386.rpm` `ccur-nstar-tutorial-1.3-1.i386.rpm`
NightTrace	`ccur-ntrace-5.2-004.i386.rpm` `ccur-ntracelog-5.2-004.i386.rpm`
NightView	`ccur-NightView-5.6-915.i386.rpm` `ccur-Nviewp-5.6-915.i386.rpm`
Scripts	`ccur-HyperHelp-scripts-6.4.2-002.i386.rpm`

RedHawk libraries	**ccur-fbsched-1.4-1.i386.rpm**
	ccur-rt-1.4-1.i386.rpm
shmdefine	**ccur-shmdefine-1.1-002.i386.rpm**
Utilities	**ccur-x11progs-6.4.2-006.i386.rpm**

NOTE

The user must be root in order to use the **rpm** product instal-
lation mechanism on the Linux system.

To install MAXAda for RedHawk Linux 3.4-Beta, issue the following commands on your RedHawk
Linux system:

1. Insert the MAXAda for RedHawk Linux 3.4-Beta Installation CD in the
   CD-ROM drive

2. Mount the CD-ROM drive (assuming the standard mount entry for the CD-ROM
   device exists in **/etc/fstab**)

   mount /mnt/cdrom

3. Change the current working directory to the directory containing the MAXAda
   for RedHawk Linux 3.4-Beta installation scripts

   cd /mnt/cdrom

4. Invoke the MAXAda for RedHawk Linux installation script

   ./ccur-install

   You may see messages similar to the following during an install (or uninstall):

   ```
   failed to stat /nfsfilesystem: Stale NFS file handle
   ```

   where *nfsfilesystem* may be any NFS filesystem. These messages may be ignored.

5. Change the current working directory outside the **/mnt/cdrom** hierarchy

   cd /

6. Unmount the CD-ROM drive (otherwise, you will be unable to remove the
   MAXAda for RedHawk Linux 3.4-Beta Installation CD from the CD-ROM
   drive)

   umount /mnt/cdrom

To uninstall the MAXAda for RedHawk Linux RPMs, use the following command:

```
rpm -e ccur-MAXAda-rm \
      ccur-MAXAda-invoker \
      ccur-MAXAda-i86_3.4_beta
```

4.1.   Getting Started

MAXAda provides a command-line interface as well as graphical interface to the compilation process. Refer to "Using NightBench with Ada" section in the *NightBench Users Guide* (`nhelp nbench`) or the "Using MAXAda" section in the *MAXAda Reference Manual*(0890516) (`nhelp maxada`).

Note:

The sections mentioned above have not yet been updated to address capbilities and privilieges required for execution of Ada tasking programs. See "Capabilities" on page 11 for more information.

5.0. Overview of MAXAda for RedHawk Linux 3.4-Beta

MAXAda for RedHawk Linux 3.4-Beta is a pre-production release. See "Known Issues" on page 8 and "Additional Considerations" on page 10 for details.

NOTE

> Please contact the Concurrent Software Support Center as described in "Direct Software Support" on page 13 with respect to any problems encountered with this release. Any and all feedback would be greatly appreciated.

MAXAda for RedHawk Linux 3.4-Beta is based on MAXAda 3.1 which was certified using Version 2.1 of the Ada Conformity Assessment Test Suite (certificate #A981215E2.1-047).

MAXAda for RedHawk Linux 3.4-Beta supports the Ada95 standard, ANSI/ISO/IEC-8652:1995 as indicated in the following table:

Sections 1 - 13	SUPPORTED
Annex A - Predefined Language Environment	SUPPORTED
Annex B - Interfaces to Other Languages	SUPPORTED
Annex C - Systems Programming	SUPPORTED (*with exceptions**)
Annex D - Real-Time Systems	SUPPORTED (*with exceptions**)
Annex E - Distributed Systems	NOT SUPPORTED
Annex F - Information Systems	NOT SUPPORTED
Annex G - Numerics	NOT SUPPORTED
Annex H - Safety and Security	NOT SUPPORTED
Annex J - Obsolescent Features	SUPPORTED

* The following features are not supported by this implementation:

Feature	RM Reference
Recommended representation support for the following clauses: 13.1(22) - support of non-static constant expressions 13.3(19) - inhibit optimizations based on assumptions of no aliases 13.3(35) - page alignment of standalone library-level objects	C.2
Preelaboration requirements	C.4
Atomic objects are not always moved indivisibly	C.6(15)
Not all storage associated with attributes of a task is reclaimed upon task termination	C.7.2(17)
`Ada.Asynchronous_Task_Control` package not provided or supported	D.11

Details regarding support for Annex C, Annex D, and all implementation-dependent portions of the language can be found in Appendix M of the *MAXAda Reference Manual* (0890516).

5.1. Known Issues

MAXAda for RedHawk Linux 3.4-Beta lacks some features that are planned for subsequent releases. These include:

- The MAXAda Reference Manual has not been updated with RedHawk Linux specifics. The majority of the manual discusses MAXAda features common to both RedHawk Linux and PowerMAX OS. Specifically, Appendix M which discusses implementation-dependent features does not fully reflect MAXAda for RedHawk Linux.

- Position Independent Code - currently MAXAda for RedHawk Linux only supports statically linked Ada code. Support for Ada shared libraries is anticipated in a future release.

- GLOBAL (**-O2**) and MAXIMAL (**-O3**) optimization levels are not fully tested; their use should be reserved for fully tested modules with controlled unit testing.

- Specification of a machine address in pragma SHARED_PACKAGE is not supported; however, machine address clauses for other entities are supported

- MAXAda programs currently will not operate with the Linux threads library.

- Specification of "restricted interrupt handling" has no effect in this release

- Pragma FAST_INTERRUPT_TASK has no effect in this release

The following tools are still under development and are only partially functional in MAXAda for RedHawk Linux 3.4-Beta:

- **a.monitor**

The following tools are not available in MAXAda for RedHawk Linux 3.4-Beta:

- **a.rtm**

The following tools will are not currently planned for MAXAda for RedHawk Linux:

- **a.analyze**
- **a.report**
- **a.slinker**

The following packages are not available in MAXAda for RedHawk Linux 3.4-Beta:

- Userdma_Support

  RedHawk Linux does not currently support a **userdma(2)** service.

- User_Level_Interrupts

  RedHawk Linux does not currently support user-level interrupts.

- Real_Time_Data_Monitoring

  Since the Real_Time_Data_Monitoring package is not available in this beta release, the **rtdm** environment (which contains this package) will be empty.

- Sockets

  The Sockets package is not available in this release.

- `RT_Interface`

  The `RT_Interface` package is not available in this beta release. It will be provided in a future release as a deprecated package. It has been replaced with the package FBSched which is available in this release.

5.2.   Additional Considerations

5.2.1.   Priorities

The Ada95 language defines priorities in terms of the discrete subtypes defined in the package `System`. The subtype `any_priority` spans the entire priority range supported by the implementation while the subtypes `priority` and `interrupt_priority` divide that range into standard user-level priorities and interrupt priorities (those which require the blocking of one or more interrupts).

For Ada tasking programs, the default Task Dispatching Policy is `FIFO_Within_Priorities`. The Ada priority values of `system.priority'first` (1) .. `System.priority'past` (98), map directly to the RedHawk Linux SCHED_FIFO 1..98 priorities. `System.interrupt_priority'first` (99), maps directly to RedHawk Linux SCHED_FIFO priority 99.

Use of `System.interrupt_priority'last` (100) is reserved for Protected Actions. All external maskable machine interrupts are masked during such actions. Programs which use this priority value must lock their address space in memory (e.g. `pragma Pool_Lock_State(default,locked)`) and must exercise extreme care inside protected actions. Misuse of this priority value can cause system panics and/or have significant effects on system performance and determinism.

By default, as per the Ada95 language standard, in the absence of a `Priority` pragma or a non-default setting of the task dispatching policy, all Ada tasks execute at priority 49.

Non-tasking Ada programs do not specifically set their priority in any way; they inherit the priority and scheduling policy of the invoking program, typically the shell.

Two additional task dispatching policies are available for Ada tasking programs: `Round_Robin_Within_Priorities`, and `Round_Robin_Adjustable_Priorities`. The former makes use of the SCHED_RR policy (mapping `system.priority` values 1..98 directly to SCHED_RR values 1..98), whereas the latter evenly distributes the system.priority range over the SCHED_OTHER policy using the **nice(1)** value to initially select scheduling urgency. As the name implies, tasks within `Round_Robin_Adjustable_Priorities` have their priority adjusted by the operating system based on CPU utilization. Use of Ada's Protected Objects requires selection of the `FIFO_Within_Priorities` dispatching policy.

5.2.2.   Capabilities

RedHawk Linux provides a means to grant otherwise unprivileged users the authority to perform certain privileged operations. The **pam_capability(8)** (Pluggable Authentication Module) is used to manage sets of capabilities, called roles, required for various activities.

RedHawk systems should be configured with an `adauser` role which provides the capabilities required by MAXAda for RedHawk Linux. In order to run MAXAda tasking programs on a RedHawk target, each MAXAda for RedHawk Linux user must be configured to use (at a minimum) the capabilities specified below. In addition, the **/etc/pam.d** configuration files associated with the **rsh** and **login** services must be modified.

To configure user capabilities, you must first become `root`. Edit the **/etc/pam.d/rsh** and **/etc/pam.d/login** files, adding the following line to each, if it is not already present:

```
session    required  /lib/security/pam_capability.so
```

Then edit **/etc/security/capability.conf** and define the adauser role (if it is not already defined) in the "ROLES" section:

```
role  adauser  cap_sys_admin  cap_sys_nice  cap_sys_rawio  cap_ipc_lock
```

and, for each MAXAda for RedHawk Linux user on the target system, add the following line at the end of the file:

```
user   username     adauser
```

where *username* is the login name of the user.

If the user requires capabilities not defined in the `adauser` role, add a new role which contains `adauser` and the additional capabilities needed, and substitute the new role name for `adauser` in the text above.

In order for the above changes to take effect, the user should log off and log back onto the target system.

Note:

> The **/etc/pam.d/rsh** and **/etc/pam.d/login** files, if edited as shown above, will allow capabilities to be granted to users who log into the system via **telnet**, **rlogin**, and **rsh**. Other methods of accessing the system may require that additional files in **/etc/pam.d** have similar modifications. For example, **/etc/pam.d/gdm**, **/etc/pam.d/kde**, and **/etc/pam.d/ssh**. To check to see if you have been granted capabilities, issue the following command: **cat /proc/self/status**. The last three lines labelled CapInh, CapPrm, and CapEff should have non-zero values if you have been granted capabilities.

5.2.3. Bit Numbering

Bit numbering on Pentium systems is different from that on PowerPC systems due to endian differences. These facts must be taken into consideration when interfacing to devices or using `Ada.Unchecked_Conversion`:

- `'Bit_Order` is `Low_Order_First`
- `System.Default_Bit_Order` is `Low_Order_First`
- In representation clauses (RM 13.5.1 & 13.5.2), the `first_bit` is the low-order bit and the `last_bit` is the high-order bit.
- In packed arrays, the low-order bit of a component appears in a lower byte number, or in the same byte number but lower bit number, than the component's high-order bit (assuming a component larger than 1 bit).

5.2.4. 'alignment max

`'alignment max` is 4 bytes for stack objects

5.2.5. Math Functions

- Exceptions raised within transcendental math functions (e.g. `sin`, `sqrt`, `exp`) may not be propagated correctly to handlers in calling frames, instead appearing to be unhandled exceptions
- Transcendental math functions may fail to raise appropriate exceptions for values which are out-of-range for the operation, instead producing odd results such as `+inf` or `-inf`

- Exponentiation with a left argument of 1.0 and a very large right value results in an exception being raised instead of 1.0 being returned

5.2.6.   Debug Level

The name of the debug level associated with the compile option **−g2** is now `simple`, not `full`.

In addition, the *debug_level* parameter `FULL` is not valid when using pragma `DEBUG`. The user should use *debug_level* `SIMPLE` instead.

6.0.   Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at 1-800-245-6453. Our customers outside the continental United States can contact us directly at 1-954-283-1822 or 1-305-931-2408. The Software Support Center operates Monday through Friday from 8 a.m. to 7 p.m., Eastern Standard time.

Calling the Software Support Center gives you immediate access to a broad range of skilled personnel and guarantees you a prompt response from the person most qualified to assist you. If you have a question requiring on-site assistance or consultation, the Software Support Center staff will arrange for a field analyst to return your call and schedule a visit.