

MAXAda for RedHawk Linux

Version 3.5 Release Notes

March 2005

0898537-3.5



Copyright

Copyright 2005 by Concurrent Computer Corporation. All rights reserved. This publication or any part thereof is intended for use with Concurrent Computer Corporation products by Concurrent Computer Corporation personnel, customers, and end-users. It may not be reproduced in any form without the written permission of the publisher.

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

MAXAda, NightTrace, NightView, NightBench, NightStar, RedHawk, and iHawk are trademarks of Concurrent Computer Corporation.

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat, Inc.

POSIX is a registered trademark of the IEEE Inc.

Pentium is a registered trademark of Intel Corporation.

AMD is a trademark of Advanced Micro Devices, Inc.

Contents

1.0 Introduction	1
2.0 Documentation	2
3.0 Prerequisites	3
3.1 Host System	3
3.1.1 Software	3
3.1.2 Hardware	3
3.2 Target System	3
3.2.1 Software	3
3.2.2 Hardware	3
4.0 System Installation	4
4.1 Getting Started	6
5.0 Overview of MAXAda for RedHawk Linux 3.5	7
5.1 Changes in this Release	8
5.1.1 Default Link Rule	8
5.1.2 -f77version Link Option	8
5.1.3 -ld[:order] Link Option	8
5.1.4 Interface to Other Languages	8
5.1.5 Debug Information	9
5.1.6 Pragma IMPORT_AUX	9
5.1.7 Byte Swapping	9
5.1.8 RedHawk 1.4 and libpthread	9
5.1.9 Reorganization of Implementation-Defined Units	9
5.1.10 Reorganization of POSIX Interfaces	11
5.1.11 Size of Common Types (AMD64 only)	11
5.1.12 System.addresses Utility Functions (AMD64 only)	12
5.1.13 a.monitor	12
5.1.14 a.monitor and RedHawk 1.x	13
5.2 Implementation-Dependent Issues	14
5.2.1 Priorities	14
5.2.2 Capabilities	14
5.2.3 Bit Numbering	15
5.2.4 'Alignment maximum	15
5.2.5 Deprecated Math Package	15
5.3 Known Issues	17
6.0 Direct Software Support	18

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 3.5 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 3.5 includes POSIX® 1003.5, a complete implementation of the Institute of Electrical and Electronic Engineers (IEEE) standard IEEE-Std-1003.5-1992.

See “Implementation-Dependent Issues” on page 14 and “Known Issues” on page 17 for more information.

2.0. Documentation

Table 2-1 lists the MAXAda for RedHawk Linux 3.5 documentation available from Concurrent.

Table 2-1. MAXAda for RedHawk Linux Version 3.5 Documentation

Manual Name	Pub. Number
<i>MAXAda for RedHawk Linux Reference Manual</i>	0898537-130
<i>MAXAda for RedHawk Linux Version 3.5 Release Notes</i>	0898537-3.5

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.

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 3.5 Installation CD*
- on the Concurrent Computer Corporation web site at **www.ccur.com**

3.0. Prerequisites

Prerequisites for MAXAda for RedHawk Linux Version 3.5 for both the host system and target system are as follows:

3.1. Host System

3.1.1. Software

- RedHawk Linux, Red Hat™ Linux 8.0, Red Hat Linux 9.0, or Red Hat® Enterprise Linux WS 3.0

3.1.2. Hardware

- any Concurrent iHawk™ system

3.2. Target System

3.2.1. Software

- RedHawk Linux 1.4 or later (*Pentium*)
- RedHawk Linux 2.2 or later (*AMD64*)

3.2.2. Hardware

- any Concurrent iHawk system

4.0. System Installation

The *MAXAda for RedHawk Linux Installation CD* contains all RPMs specifically required for operation of the MAXAda compiler, linker, and utilities.

NightView 6.1 is required in order to debug Ada programs built with this release of MAXAda. NightView is a part of the RedHawk NightStar Tools.

RedHawk Linux 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.

A single command installs (or uninstalls) all of the RPMs needed to support MAXAda for RedHawk Linux 3.5, but does not address the installation of RedHawk Linux.

The following table shows the RPMs that will be installed.

Item	RPM
MAXAda	<i>Pentium:</i> <code>ccur-MAXAda-invoker-3.5-2.i386.rpm</code> <code>ccur-MAXAda-rm-3.5-2.i386.rpm</code> <code>ccur-MAXAda-i86_3.5-000-7.i386.rpm</code> <code>ccur-MAXAda-open-i86_3.5-000-4.i386.rpm</code> <i>AMD64:</i> <code>ccur-MAXAda-invoker-3.5-2.x86_64.rpm</code> <code>ccur-MAXAda-rm-3.5-2.x86_64.rpm</code> <code>ccur-MAXAda-amd64_3.5-000-7.x86_64.rpm</code> <code>ccur-MAXAda-open-amd64_3.5-000-5.x86_64.rpm</code>
NightBench	<code>ccur-nbench-2.3.1-000.i386.rpm</code> <code>ccur-nbench-ada-2.3.1-000.i386.rpm</code>
Scripts	<code>ccur-HyperHelp-scripts-6.4.2-002.i386.rpm</code>
Utilities	<code>ccur-x11progs-6.4.2-009.i386.rpm</code> <code>ccur-elanlm-5.0-9.i386.rpm</code> (<i>Pentium</i>) <code>ccur-elanlm-5.0-9.x86_64.rpm</code> (<i>AMD64</i>)

NOTE

The user must be `root` in order to use the `rpm` product installation mechanism on the Linux system.

To install MAXAda for RedHawk Linux 3.5, issue the following commands on your RedHawk Linux system:

1. Insert the *MAXAda for RedHawk Linux 3.5 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 3.5 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 3.5 Installation CD* from the CD-ROM drive)

```
umount /mnt/cdrom
```

To uninstall the MAXAda for RedHawk Linux RPMs, follow the steps above to mount the CD and issue the following command:

```
./ccur-uninstall
```

from the `/mnt/cdrom` directory.

4.1. Getting Started

MAXAda provides a command-line interface as well as graphical interface to the compilation process. Refer to “Using MAXAda” section in the *MAXAda Reference Manual* (0890516) (**nhelp maxada**) or the “Using NightBench with Ada” section in the *NightBench User’s Guide* (0890514) (**nhelp nbench**).

To utilize the command-line interface, add the following to your `PATH`:

```
/usr/ada/bin
```

5.0. Overview of MAXAda for RedHawk Linux 3.5

MAXAda for RedHawk Linux 3.5 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 3.5 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 (<i>with exceptions</i> *)
Annex D - Real-Time Systems	SUPPORTED (<i>with exceptions</i> *)
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. Changes in this Release

5.1.1. Default Link Rule

The default link rule changed from:

```
object,archive,shared_object
```

to:

```
object,archive-system,shared_object
```

This allows system libraries to be linked as shared objects, instead of statically. Certain `glibc` and `pthread` functions require the use of shared objects, and use of shared objects for other system libraries results in smaller executables.

5.1.2. `-f77version` Link Option

To interface to Concurrent Fortran 77 on systems which have multiple Fortran versions installed, the new `-f77version` is supplied. It takes an argument which specifies the version to which to interface. Its use is not necessary on systems with a single version, or where `c.release` indicates a default version.

5.1.3. `-ld[:order]` Link Option

The `-ld` link option was improved to allow specification of the place of the following argument among the other arguments specified. The *order* is a number which specifies the place, with higher-numbered orders appearing after lower-numbered orders. See the section titled “ld Argument” in the “MAXAda Utilities” chapter of the *MAXAda for RedHawk Linux Reference Manual* (0898537) for a list of the predefined orders of arguments supplied automatically by `a.link`.

5.1.4. Interface to Other Languages

The *MAXAda for RedHawk Linux Reference Manual* now contains descriptions of how to link code written in other languages into MAXAda programs, and how to link MAXAda code into programs written in other languages. See the section titled “Interface to Other Languages” in the “MAXAda Utilities” chapter of the *MAXAda for RedHawk Linux Reference Manual* (0898537).

Also, example programs are included in:

```
/usr/ada/rel/sup/examples
```

where *rel* is either:

```
i86_3.5 (Intel)
```

or:

```
amd64_3.5 (AMD64)
```

To allow interfacing with GNU Fortran, the new `Restricted_Gnu_Fortran` convention was added.

5.1.5. Debug Information

MAXAda supports improved symbolic debug information that can keep track of multiple locations for a given variable, and knows when each location is valid. This is generally useful, but especially so when debugging optimized code.

This improved debug information is enabled with the `-g` option, or explicitly with the `-g3` compile option or `pragma Debug (full)`, which specify the new "full" debug level.

Information similar to that available in previous versions is available with the `-g2` compile option or `pragma Debug (simple)`, which specify the "simple" debug level.

5.1.6. Pragma IMPORT_AUX

Pragma `IMPORT_AUX` is a new pragma which should be used when interfacing to C/C++ functions which have ellipses in their declarations. See the section titled “Pragma `IMPORT_AUX`” in the “Implementation-Defined Characteristics” chapter of the *MAXAda for RedHawk Linux Reference Manual* (0898537) for usage instructions.

5.1.7. Byte Swapping

The new `ccur.unchecked_byte_swap` generic function implements byte swapping operations that are optimized for the target. Generally, they are used when transforming data between a Big Endian device or target and a Little Endian device or target. See the section titled “Byte Swapping” in the “Real-Time Extensions” chapter of the *MAXAda for RedHawk Linux Reference Manual* (0898537) for usage instructions.

5.1.8. RedHawk 1.4 and libpthread

MAXAda now supports linking with `libpthread` on RedHawk 1.4 systems. In fact, it always links with `libpthread` for programs which use tasks or other real-time features. A side effect of this is that the following signals are reserved on RedHawk 1.4 in programs with tasks or other real-time features:

- `SIGRTMIN+1` (33) (if tasking, real-time features or `libpthread` is used)
- `SIGRTMIN+2` (34) (if tasking, real-time features or `libpthread` is used)
- `SIGRTMIN+3` (35) (if tasking, real-time features or `libpthread` is used)

5.1.9. Reorganization of Implementation-Defined Units

The implementation-defined units shipped with MAXAda for RedHawk Linux 3.5 underwent a reorganization to eliminate namespace pollution. All Concurrent-defined units have become children of the new package `ccur`. For backward compatibility, the old names still exist as renames of the new units, so existing customers should not be affected (but see the WARNING below).

This reorganization allows new or existing customers to override the old unit names (e.g. `bit_ops`) without interfering with dependencies in the Concurrent-supplied environments, and without losing access to the Concurrent-supplied units which will be available as descendants of the `ccur` package (e.g. `ccur.bit_ops`).

The sole exception to this reorganization is the deprecated `default_handler` package, which cannot be made a rename. However, customers may override this package, and still obtain functionality equivalent to the `default_handler` functionality using `pragma task_handler`.

The following packages were renamed in the new organization:

```
real_time_data_monitoring => ccur.rtdm
test_and_set_package      => ccur.test_and_set_operations
```

However, for backward compatibility, the renames retain the original names, so existing customers should not be affected.

WARNING

To support this reorganization, the package `ccur` is new in MAXAda. Customer code must not use this package name.

The specific name changes are detailed below:

```
a_strings                => ccur.a_strings
binary_semaphores       => ccur.binary_semaphores
bit_ops                 => ccur.bit_ops
bit_ops.long           => ccur.bit_ops.long
c_strings               => ccur.c_strings
c_to_ada_types         => ccur.c_to_ada_types
character_type          => ccur.character_type
client_server_services => ccur.client_server_services
complex_arith           => ccur.complex_arith
current_exception      => ccur.current_exception
curses                 => ccur.curses
cyclic_scheduler       => ccur.cyclic_scheduler
default_handler         => default_handler           (UNCHANGED)
distrib_services       => ccur.distrib_services
enum_16_io             => ccur.enum_16_io
enum_8_io              => ccur.enum_8_io
eti_control            => ccur.eti_control
eti_services           => ccur.eti_services
fbsched                => ccur.fbsched
float_io               => ccur.float_io
generic_math           => ccur.generic_math
indivisible_operations => ccur.indivisible_operations
integer_io             => ccur.integer_io
interrupt_entry        => ccur.interrupt_entry
interval_timer         => ccur.interval_timer
long_float_io         => ccur.long_float_io
math                   => ccur.math
night_trace_bindings  => ccur.night_trace_bindings
null_workload          => ccur.null_workload
qsort                  => ccur.qsort
random_support         => ccur.random_support
real_time_data_monitoring => ccur.rtdm                 (RENAMED)
rescheduling_control  => ccur.rescheduling_control
rt_interface           => ccur.rt_interface
rtc_control            => ccur.rtc_control
rtc_services          => ccur.rtc_services
runtime_configuration => ccur.runtime_configuration
shared_memory_support => ccur.shared_memory_support
short_integer_io       => ccur.short_integer_io
sockets                => ccur.sockets
spin_locks            => ccur.spin_locks
```

task_attributes	=> ccur.task_attributes
task_synchronization	=> ccur.task_synchronization
tasking_semaphores	=> ccur.tasking_semaphores
test_and_set_package	=> ccur.test_and_set_operations (RENAME)
timers	=> ccur.timers
tiny_integer_io	=> ccur.tiny_integer_io
u_env	=> ccur.u_env
unix_prcls	=> ccur.unix_prcls
unsigned_support	=> ccur.unsigned_support
user_trace	=> ccur.user_trace
user_trace.raw	=> ccur.user_trace.raw
usermap_support	=> ccur.usermap_support

5.1.10. Reorganization of POSIX Interfaces

The POSIX interfaces underwent reorganization. All packages with the prefix `posix_now` are children of the package `posix` instead.

For full compliance with IEEE Std 1003.5b-1996 (POSIX 1003.5b), the `posix_names` are renames of the new children of `posix`.

The specific name changes are detailed below:

<code>posix_1003_1</code>	=> <code>ccur.posix_1003_1</code>
<code>posix_calendar</code>	=> <code>posix.calendar</code>
<code>posix_configurable_file_limits</code>	=> <code>posix.configurable_file_limits</code>
<code>posix_configurable_system_limits</code>	=> <code>posix.configurable_system_limits</code>
<code>posix_file_locking</code>	=> <code>posix.file_locking</code>
<code>posix_file_status</code>	=> <code>posix.file_status</code>
<code>posix_files</code>	=> <code>posix.files</code>
<code>posix_group_database</code>	=> <code>posix.group_database</code>
<code>posix_io</code>	=> <code>posix.io</code>
<code>posix_local_signals</code>	=> <code>posix.local_signals</code>
<code>posix_permissions</code>	=> <code>posix.permissions</code>
<code>posix_process_environment</code>	=> <code>posix.process_environment</code>
<code>posix_process_identification</code>	=> <code>posix.process_identification</code>
<code>posix_process_primitives</code>	=> <code>posix.process_primitives</code>
<code>posix_process_primitives.local</code>	=> <code>posix.process_primitives.local</code>
<code>posix_process_times</code>	=> <code>posix.process_times</code>
<code>posix_signals</code>	=> <code>posix.signals</code>
<code>posix_supplement_to_ada_io</code>	=> <code>posix.supplement_to_ada_io</code>
<code>posix_terminal_functions</code>	=> <code>posix.terminal_functions</code>
<code>posix_unsafe_process_primitives</code>	=> <code>posix.unsafe_process_primitives</code>
<code>posix_user_database</code>	=> <code>posix.user_database</code>

5.1.11. Size of Common Types (AMD64 only)

On the AMD64 architecture, types involving addresses and access types as well as some commonly-used types have increased in size:

<code>long_integer</code>	64 bits
<code>system.address</code>	64 bits
<code>duration</code>	64 bits

<i>access-to-object</i>	64 bits
<i>access-to-subprogram, protected</i>	128 bits
<i>access-to-subprogram, not protected</i>	192 bits

5.1.12. System.addresses Utility Functions (*AMD64 only*)

On the AMD64 architecture, the interface for the "+" and "-" operations which used to involve `system.address` and an `integer` addend in **system.addresses** has changed such that the addend must now be `long_integer` instead of `integer`.

For example,

```
function "+" (left : in address; right : in integer)
return system.address
```

changes to

```
function "+" (left : in address; right : in long_integer)
return system.address
```

Similarly, on the AMD64 architecture, the instantiations of `unchecked_conversion` in the package **system.addresses** between `system.address` and `integer` have been changed to `system.address` and `long_integer`.

5.1.13. a.monitor

The graphical interface to **a.monitor** has been redesigned using the Qt graphical toolkit.

As such, the executable file:

```
/usr/ada/i86_3.5/bin/a.monitor-x
```

is released in accordance with Q Public License v1.0 and the GNU GPL.

All other files released with MAXAda for RedHawk Linux are licensed solely in accordance with the Concurrent Computer Corporation Software License Agreement which can be found at <http://www.ccur.com/info/stc.html>.

The files in:

```
/usr/ada/i86_3.5/sup/license
```

describe these licenses in detail.

a.monitor operates with or without the GUI interface; issue:

```
a.monitor -H
```

for more information.

NOTE

Use of **a.monitor-x** during the development and debug of applications with MAXAda does not affect the licensing of such applications. Such applications are not derived works from **a.monitor-x** or the Qt toolkit, thus the Q Public License and the GNU GPL do not apply to such applications unless they specifically incorporate use of **a.monitor-x** as part of their functionality.

5.1.14. a.monitor and RedHawk 1.x

The graphical interface to **a.monitor** is not supported on RedHawk 1.x systems. The ASCII interface is supported, however.

5.2. Implementation-Dependent Issues

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.

See the section titled “Priorities” in the “Run-Time Concepts” chapter of the *MAXAda Reference Manual* (0890516) for more information on priorities and scheduling classes.

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, edit the `/etc/pam.d/rsh` and `/etc/pam.d/login` files as `root`, 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.

See the section titled “Capabilities” in the “Introduction to MAXAda” chapter of the *MAXAda Reference Manual* (0890516) for more information.

5.2.3. Bit Numbering

Bit numbering on Pentium and AMD64 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 maximum

On Pentium, `'Alignment` maximum is 4 bytes for stack objects.

On AMD64, `'Alignment` maximum is 16 bytes for stack objects.

5.2.5. Deprecated Math Package

The deprecated `math` package's handling of error conditions is different on Linux than on PowerMAX OS. For many errors, an exception is raised immediately while performing the math operation instead of merely setting `errno` and postponing the exception until `math.check_errno` is called.

For more information, see the comments in the specification of the `math` package in:

`/usr/ada/rel/deprecated/math.a`

where *rel* is either:

`i86_3.5` (*Intel*)

or:

`amd64_3.5` (*AMD64*)

5.3. Known Issues

MAXAda for RedHawk Linux 3.5 lacks some features that are planned for subsequent releases. These include:

- The **strace** facility in **a.monitor** is not currently functional due to a defect in the RedHawk operating system; it will be corrected in RedHawk Linux 2.3.
- 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.
- Specification of "restricted interrupt handling" has no effect in this release.
- Pragma `FAST_INTERRUPT_TASK` has no effect in this release.
- **a.rtm** is not available in MAXAda for RedHawk Linux 3.5. The NightProbe tool (**nprobe**) is recommended in its place. See the *NightProbe User's Guide* (0890465) for more information.

The following tools 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 3.5:

- `Userdma_Support`

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

- `User_Level_Interrupts`

RedHawk Linux does not currently support user-level interrupts.

- `RT_Interface`

The `RT_Interface` package has been moved to the **deprecated** environment. It has been replaced in this release with the package `FBSched` which is available in the **vendorlib** environment.

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 our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time. You may submit a request for assistance at any time by using the Concurrent Computer Corporation web site at http://www.ccur.com/isd_support_contact.asp.

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.

