

MAXAda for Linux

Version 3.5.1 Release Notes

August 2006

0898537-3.5.1



Copyright

Copyright 2006 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.

SUSE is a registered trademark of Novell, 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 RedHawk Linux Targets	3
3.2.1.1 Software	3
3.2.1.2 Hardware	4
3.2.2 SUSE Linux Targets	4
3.2.2.1 Software	4
3.2.2.2 Hardware	4
4.0 System Installation	5
4.1 Kernel Configuration	7
4.1.1 Shared Memory Limits	7
4.1.2 Randomization of Address Space	7
5.0 Getting Started	8
6.0 Overview of	
MAXAda for Linux 3.5.1	9
6.1 a.monitor	11
6.2 Open Motif Libraries	12
6.3 Implementation-Dependent Issues	13
6.3.1 Priorities	13
6.3.2 Capabilities	13
6.3.3 Bit Numbering	15
6.3.4 'Alignment maximum	15
6.3.5 Deprecated Math Package	15
6.4 Compatibility Issues for RedHawk Systems	16
6.4.1 Binary Incompatibility	16
6.4.2 Source Incompatibility	16
6.5 Known Issues	17
7.0 Direct Software Support	18

1.0. Introduction

MAXAda™ for Linux® supports development of Ada95 programs running under selected Linux distributions. MAXAda for 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 Linux 3.5.1 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 Linux 3.5.1 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 13 and “Known Issues” on page 17 for more information.

2.0. Documentation

Table 2-1 lists the MAXAda for Linux 3.5.1 documentation available from Concurrent.

Table 2-1. MAXAda for Linux Version 3.5.1 Documentation

Manual Name	Pub. Number
<i>MAXAda for Linux Reference Manual</i>	0898537-140
<i>MAXAda for Linux Version 3.5.1 Release Notes</i>	0898537-3.5.1

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 utility, **nhelp**
- in PDF format in the **documentation** directory of the *MAXAda for Linux 3.5.1 Installation CD*
- on the Concurrent Computer Corporation web site at **www.ccur.com**

3.0. Prerequisites

Prerequisites for MAXAda for Linux Version 3.5.1 for both the host system and target system are as follows:

3.1. Host System

3.1.1. Software

Any of the following operating systems:

- RedHawk Linux Version 1.4 or later
- Red Hat Linux Versions 8.0, Enterprise 3, or Enterprise 4
- SUSE Linux Enterprise Real Time Version 10
- SUSE Linux Enterprise Desktop 10
- SUSE Linux Enterprise Server 10
- SUSE Linux 10.1

3.1.2. Hardware

- Any Pentium (Pentium III or newer), EMT64 or AMD64 system supported by the host operating system

3.2. Target System

MAXAda supports self-hosted targets as well as cross compilation between compatible systems.

When cross-compiling for target systems, the host system operating system type and version must match the underlying target system operating type and version. For example, a system running Red Hat Enterprise Linux 3.0 is an appropriate host for a RedHawk Linux 2.3 target system., whereas a system running SUSE Linux Enterprise Desktop 10 is an appropriate host for a target system running SUSE Linux Enterprise Real Time 10.

When cross-linking, any required libraries specific to the target system must be present on (or accessible from) the host system.

For example:

```
> mkdir -p /redhawk-root/usr/lib
> scp redhawk:/usr/lib/libccur_rt.* /redhawk-root/usr/lib
> a.partition -default -oappend -L/redhawk-root/usr/lib
```

The former two commands setup and populate a local directory with a library from the target system.

The latter command sets the default link options for the current environment (assuming the MAXAda environment already exists) to search **/redhawk-root/usr/lib** when searching for libraries.

3.2.1. RedHawk Linux Targets

3.2.1.1. Software

- RedHawk Linux 1.4 or later

3.2.1.2. Hardware

- Any Concurrent iHawk Series system

3.2.2. SUSE Linux Targets

3.2.2.1. Software

- SUSE Linux Enterprise Real Time 10

3.2.2.2. Hardware

- Any Pentium (Pentium III or newer), EMT64 or AMD64 system supported by the target operating system

4.0. System Installation

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

NightView 6.1 or later is required in order to debug Ada programs built with this release of MAXAda. NightView is a part of the NightStar RT product.

A single command installs (or uninstalls) all of the RPMs needed to support MAXAda for Linux 3.5.1, but does not address the installation of Linux.

The following table shows the RPMs that will be installed.

Item	RPM
MAXAda	<i>Pentium:</i> <code>ccur-MAXAda-suite-3.5.1-3.i386.rpm</code> <code>ccur-MAXAda-invoker-3.5.1-3.i386.rpm</code> <code>ccur-MAXAda-rm-3.5.1-3.i386.rpm</code> <code>ccur-MAXAda-i86_3.5.1-0-11.i386.rpm</code> <code>ccur-MAXAda-open-i86_3.5-1-3.i386.rpm</code> <i>AMD64:</i> <code>ccur-MAXAda-suite-3.5.1-3.x86_64.rpm</code> <code>ccur-MAXAda-invoker-3.5.1-3.x86_64.rpm</code> <code>ccur-MAXAda-rm-3.5.1-3.x86_64.rpm</code> <code>ccur-MAXAda-amd64_3.5.1-0-11.x86_64.rpm</code> <code>ccur-MAXAda-open-amd64_3.5.1-1-3.x86_64.rpm</code>
NightBench	<code>ccur-nbench-2.3.2-3.i386.rpm</code> <code>ccur-nbench-ada-2.3.2-3.i386.rpm</code>
Scripts	<code>ccur-HyperHelp-scripts-6.4.2-004.i386.rpm</code>
Utilities	<code>ccur-x11progs-6.4.2-19.i386.rpm</code> - or - <code>ccur-su-x11progs-6.4.2-19.i386.rpm</code>

NOTE

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

To install MAXAda for Linux 3.5.1, issue the following commands on your Linux system:

1. Insert the *MAXAda for Linux 3.5.1 Installation CD* in the CD-ROM drive
2. Mount the CD-ROM drive (assuming the standard mount entry for the CD-ROM device in `/etc/fstab` is `/mnt/cdrom`)

```
mount /mnt/cdrom
```

NOTE:

Many systems automatically mount the CD when it is inserted. Some systems use `/media/cdrom` as the mount point for the CD device.

3. Change the current working directory to the directory containing the MAXAda for Linux 3.5.1 installation scripts

```
cd /mnt/cdrom
```

4. Invoke the MAXAda for Linux installation script

```
./install-maxada
```

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 Linux 3.5.1 Installation CD* from the CD-ROM drive)

```
umount /mnt/cdrom
```

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

```
./uninstall-maxada
```

from the `/mnt/cdrom` directory.

4.1. Kernel Configuration

Both RedHawk Linux and Concurrent Real-Time Extensions Powered by SUSE Linux come with default kernel configurations which are sufficient for most Ada programs.

Changes to kernel configuration parameters may benefit or impair some Ada programs. Consider the following issues:

- Shared Memory Limits
- Randomization of Address Space

See `sysctl(8)` for more information on kernel configuration parameters.

4.1.1. Shared Memory Limits

The following kernel configuration parameters may need to be increased for Ada programs using large numbers of shared packages (Pragma `shared_package`) or shared packages with large size requirements.

- `kernel.shmni`
- `kernel.shmall`
- `kernel.shmmax`

4.1.2. Randomization of Address Space

Implementation of the following pragmas and clauses requires an understanding of how the kernel forms a process's address space:

- Pragma `memory_pool`
- Pragma `pool_lock_state`
- Pragma `shared_package`
- Address clauses (with machine address specifications)

The `kernel.randomize_va_space` kernel configuration parameter should remain set to zero to prevent the Linux kernel from randomly defining the virtual memory layout of a process. Such randomization may conflict with decisions made by MAXAda utilities in order to implement the above pragmas and clauses.

NOTE:

Randomization of address space is a recent feature in some kernels. Your system may not have this feature.

5.0. 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
```

6.0. Overview of MAXAda for Linux 3.5.1

This release is a maintenance release which contains all updates to MAXAda 3.5 as well as support for RedHawk 4.1 and Concurrent Real-Time Extensions Powered by SUSE Linux -- Version 10.0.

MAXAda for Linux 3.5.1 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 Linux 3.5.1 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).

6.1. a.monitor

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

As such, the executable file:

```
/usr/ada/rel/bin/a.monitor-x
```

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

NOTE

rel is either **i86_3.5.1** (*Intel*) or **amd64_3.5.1** (*AMD64*)

All other files released with MAXAda for Linux (except for Open Motif libraries) 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/rel/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.

6.2. Open Motif Libraries

On some Red Hat AMD64 systems, the 32-bit compatible Open Motif libraries are not installed by default. The required shared libraries are installed automatically during MAXAda installation if required.

OpenMotif is distributed under the Open Group Public License which can be viewed at <http://www.opengroup.org/openmotif/license>.

6.3. Implementation-Dependent Issues

6.3.1. 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 Linux `SCHED_FIFO 1..98` priorities.

`System.interrupt_priority'first (99)`, maps directly to 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* (0890537) for more information on priorities and scheduling classes.

6.3.2. Capabilities

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

Linux systems should be configured with an `adauser` role which provides the capabilities required by MAXAda for Linux. In order to run MAXAda tasking programs on a target, each MAXAda for Linux user must be configured to use (at a minimum) the capabilities specified below.

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
```

Additionally, for each MAXAda for 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 addition to registering your login name in `/etc/security/capability.conf`, files under the `/etc/pam.d` directory must also be configured to allow capabilities to be activated.

To activate capabilities, add the following line to the end of selected files in `/etc/pam.d` if it is not already present:

session required pam_capability.so

The list of files to modify is dependent on the list of methods that will be used to access the system. The following table presents a recommended configuration that will grant capabilities to users of the services most commonly employed in accessing a system.

/etc/pam.d File	Affected Services	Comment
remote	telnet rlogin rsh (when used <u>w/o</u> a command)	Depending on your system, the remote file may not exist. Do not create the remote file, but edit it if it is present.
login	local login (e.g. console) telnet* rlogin* rsh* (when used <u>w/o</u> a command)	*On some versions of Linux, the presence of the remote file limits the scope of the login file to local logins. In such cases, the other services listed here with login are then affected solely by the remote configuration file.
rsh	rsh (when used <u>with</u> a command)	e.g. rsh system_name a.out
sshd	ssh	You must also edit /etc/ssh/sshd_config and ensure that the following line is present: UsePrivilegeSeparation no
gdm	gnome sessions	
kde	kde sessions	

If you modify **/etc/pam.d/sshd** or **/etc/ssh/sshd_config**, you must restart the **sshd** service for the changes to take effect:

On RedHawk systems: **service sshd restart**

On SUSE systems: **bash /etc/init.d/sshd restart**

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

NOTE

To verify that you have been granted capabilities, issue the following command:

```
/usr/sbin/getpcaps $$
```

The output from that command will list the roles currently assigned to you.

See the section titled “Capabilities” in the “Introduction to MAXAda” chapter of the *MAXAda Reference Manual* (0898537) for more information.

6.3.3. 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).

6.3.4. 'Alignment maximum

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

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

6.3.5. 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_3.5.1 (Intel)
```

or:

```
amd64_3.5.1 (AMD64)
```

6.4. Compatibility Issues for RedHawk Systems

MAXAda programs built with version 3.5.1 are generally upward compatible with RedHawk systems starting with RedHawk 2.1 - RedHawk 4.1. Thus programs built with MAXAda 3.5.1 on a RedHawk 2.1 system will generally operate correctly on all RedHawk versions from 2.1 through 4.1.

Backward compatibility is undefined -- generally you should not expect programs built on a more recent operating system to run on an older operating system.

Due to changes in the Red Hat Enterprise Linux 4 environment, some unusual incompatibilities have arisen, as discussed below in the following two subsections.

6.4.1. Binary Incompatibility

MAXAda programs built with version 3.5 for AMD64 systems are incompatible with the RedHawk 4.1 (Red Hat Enterprise Linux 4) C library. Unless your program was statically linked, you will need to rebuild it using MAXAda 3.5.1. The incompatibility involves any service requiring a `stat` (`stat(2)`), `lstat(2)`, or `fstat(2)` action, which includes all programs which require `ada.text_io`, `posix_files`, or `posix_file_status` or programs which use one of the `stat` functions defined in `posix_1003_1`.

6.4.2. Source Incompatibility

The RedHawk 4.1 (Red Hat Enterprise 4) libraries have changed the set of reserved signals. This change affects `SIGRTMIN`, which is no longer a static value.

The MAXAda-supplied package, `posix_1003_1`, previously included a static constant called `SIGRTMIN`. `SIGRTMIN` has been changed to be a function which returns the appropriate value of `SIGRTMIN`, based on the platform and library set in use.

This could cause user declarations to fail to compile if `posix_1003_1.SIGRTMIN` was used as an initial value when a static initial value was required.

In previous releases of Red Hat, `SIGRTMIN` had the value 32 even though a few signals starting at that value were reserved for use by the threads library.

Depending on your library usage, under RedHawk 4.1, `SIGRTMIN` may now have a different value and typically all signals starting at `SIGRTMIN` through `SIGRTMAX` are now available for general use.

Note:

The MAXAda runtime system continues to reserve signal 47 for its use.

6.5. Known Issues

MAXAda for Linux 3.5.1 lacks some features that historically have been available on other MAXAda-supported platforms. These include:

- SUSE NFS Support

Some versions of SUSE do not support file locking on NFS-mounted file systems which is a requirement for MAXAda. Therefore you may not be able to create or operate on MAXAda environments on NFS-mounted partitions associated with SUSE systems (host or target).

- The **strace** facility in **a.monitor** is not functional in RedHawk versions prior to RedHawk Linux 2.3.
- Position Independent Code - currently MAXAda for Linux only supports statically linked Ada code although MAXAda programs may utilize system shared libraries
- 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 Linux 3.5.1. 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 Linux:

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

The following packages are not available in MAXAda for Linux 3.5.1:

- Userdma_Support

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

- User_Level_Interrupts

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.

7.0. 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 also 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 or by sending an email to support@ccur.com.