

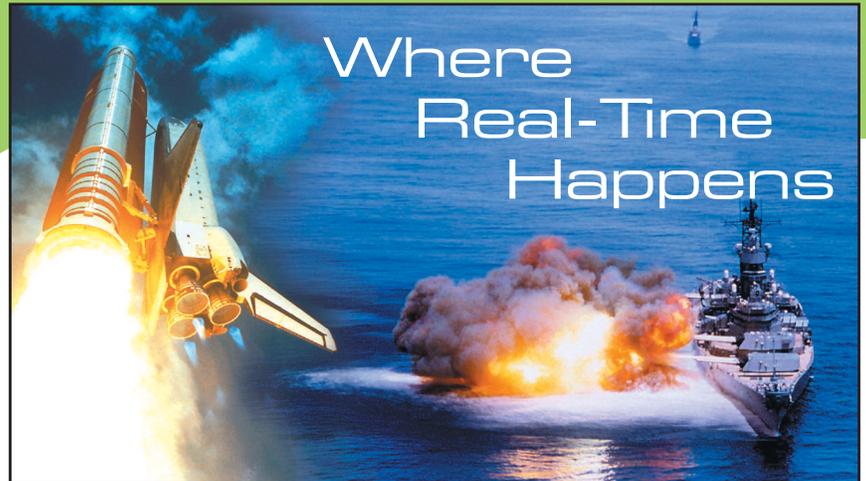
Features

- ANSI/ISO/IEC-8652:1995 compliant compiler
- Ada Real-Time Multiprocessing System (ARMS™)
 - Multiprocessor Ada tasking
 - Predictable task execution and communication
 - Hardware interrupt handling
- Supports team programming
 - Allows parallel development in multiple environments
 - Provides protection for baseline environments
- Fully integrated with Concurrent's NightStar™ software development tool set
 - NightView™ symbolic debugger
 - NightTrace™ analyzer
 - NightProbe™ data monitor
 - NightSim™ periodic scheduler
- Real-time tracing
- Real-time data monitoring
- Common software environment for all Concurrent platforms
- Aggressive code optimization
- Automatic and on-demand syntax checking
- Create, configure, and build shared libraries and archives in addition to executable programs
- Automatic build facility assures consistent program generation with persistent compile and link options
- Provides Ada packages and bindings, including:
 - POSIX.1 (IEEE-Std-1003.1)
 - POSIX.1b (IEEE-Std-1003.1b)
 - POSIX.5 (IEEE-Std-1003.5-1992)
 - AXI™ bindings to MOTIF™/Xt/Xlib (optional)
- Extensive graphical online help
 - MAXAda Reference Manual
 - NightBench User's Guide
 - Ada 95 Reference Manual
- Optional *Understand for Ada* Tools:
 - Reverse engineering
 - Automated documentation
 - Code navigation
 - Cross referencing



MAXAda™

Program Development Environment



Overview

Concurrent Computer Corporation's MAXAda™ is a high-performance system intended for the large-scale development of real-time Ada applications. MAXAda processes the Ada language as specified by the Ada 95 Reference Manual, ANSI/ISO/IEC-8652:1995.

MAXAda is available on Concurrent iHawk™ Xeon™-based multiprocessors running RedHawk™ Linux® as well as its PowerPC-based platforms running PowerMAX™ OS.

Concurrent is committed to a long-range strategy of creating tools necessary for generating production-quality Ada software. To this end, Concurrent has produced a sophisticated optimizing technology that meets the high-performance, real-time needs of today's deterministic Ada applications. In addition to a self-targeted compiler, MAXAda includes the advanced ARMS run-time system, the NightBench program development environment, and comprehensive real-time tools.

The MAXAda Compiler

The MAXAda compiler maximizes run-time efficiency, featuring multiple levels of optimized code generation.

The compiler incorporates state-of-the-art optimizations such as constraint propagation as well as other standard optimizations, such as constant folding, common sub-expression elimination, moving invariant code, reduction of operator strength, and peephole optimizations. The compiler utilizes table-driven code selection and register allocator algorithms, thereby providing rapid retargetability. MAXAda's efficient instruction scheduling is a key feature in compilation for state-of-the-art computer architectures.

Comprehensive programming support includes:

- extensive support of representation items, including enumeration and record representation clauses
- concentrated records and array packing
- virtual and physical address clauses
- interfacing to other languages
- multiple shared memory paradigms
- software and hardware interrupt handling
- user-level device drivers

Integrated Solutions... Real Benefits

ARMS Run-Time System

MAXAda includes the Ada Real-Time Multiprocessing System (ARMS), an Ada run-time executive designed for critical real-time tasking applications. ARMS implements all tasks as individual threads of execution that can be independently scheduled by the OS kernel, thereby providing true concurrence in a real-time multiprocessor/parallel execution environment.

For greater real-time control, ARMS provides a number of pragmas for configuration of the whole run-time system, including task execution and memory utilization. Explicit control of CPU assignment, task prioritization, and task time slice duration are examples of these pragmas.

MAXAda provides fast exception handling with comprehensive descriptions of exceptions, with specific references to the Ada 95 Reference Manual. In addition, MAXAda provides exception tracing, which describes the path of exception handling from point of raise to point of handling, giving the user a unique debugging advantage.

Furthermore, hardware interrupts attached to Ada-protected objects and tasks provide ultrafast interrupt response times.

NightBench Program Development Environment

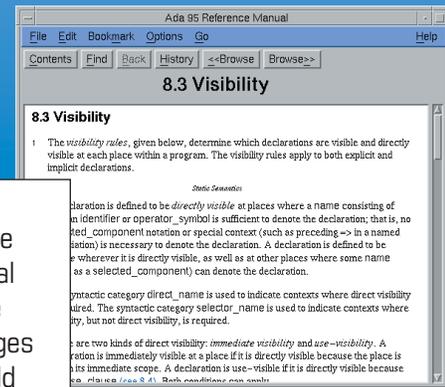
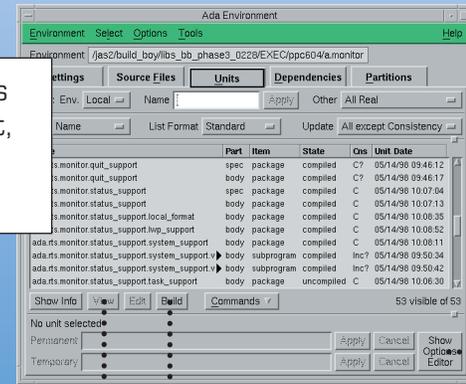
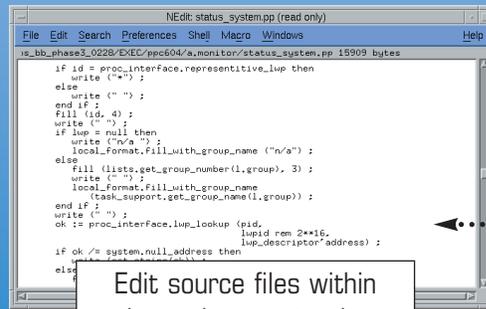
NightBench is a graphical user interface that provides a common work environment for the tools necessary for Ada program development.

NightBench supplies these features in a unified graphical user interface based on OSF/Motif™ and the X Window System™ standards, providing the user with a work environment in which to develop Ada programs quickly and easily.

NightBench organizes program development, keeping track of all relevant information

Edit source files within the environment using integrated text editor

Extensive online help including hyperlinks to the Ada 95 Reference Manual and MAXAda Reference Manual from error messages generated during the build



Organized Program Development

MAXAda organizes Ada program development by keeping track of information related to each environment. This information ranges from the source files and their corresponding compilation units to the programs that are defined within these environments. MAXAda handles the intricate details that are crucial to consistent program generation. Dependencies between units, their compilation states, and level of consistency with respect to the project are all maintained by MAXAda.

Automatic Build Utility

MAXAda provides a highly sophisticated build facility, which executes the required

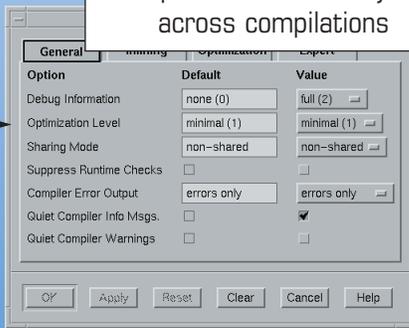
sequence of compilations and links to build executable programs, shared objects, or archives, importing required units from outside environments, if necessary.

Compile options, link options, and other preferences are persistent, providing consistency across compilations and builds. This architecture assures reproducibility of programs within a given build environment.

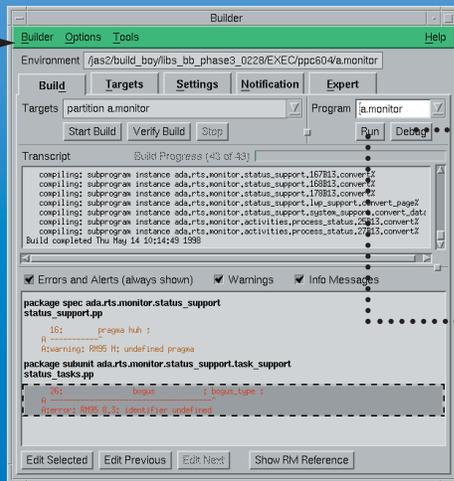
Protected Environments

Programmers can work without interference on local versions of individual program units while retrieving the remainder of the program from previously developed environments using MAXAda's Environment Search Path. In this way, baseline environments remain protected,

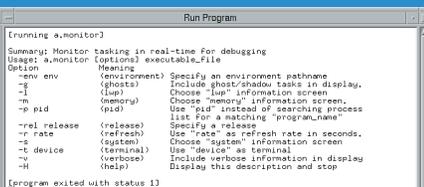
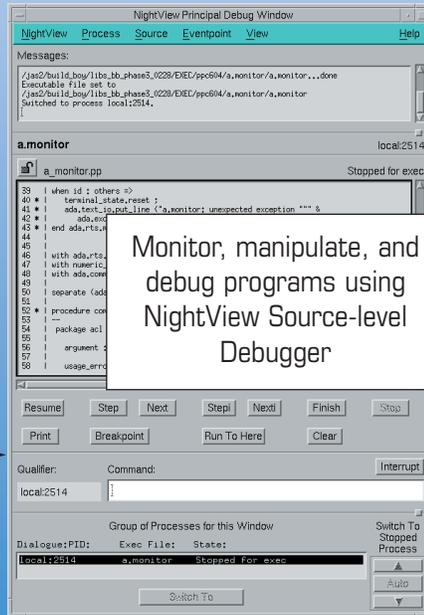
Persistent build options provide consistency across compilations



Build programs, shared libraries, and archives with the press of a button



Monitor, manipulate, and debug programs using NightView Source-level Debugger



Run programs directly from NightBench

standard breakpoint and stepping capabilities. Additional real-time debugging mechanisms provided by NightView include monitorpoints, tracepoints, patchpoints, and watchpoints.

Real-Time Tracing

MAXAda includes support for real-time debug and performance analysis via real-time trace points. MAXAda provides an Ada executive that has been instrumented with trace points describing task scheduling activities. The operating system has also been instrumented with trace points that describe activities such as system calls, exceptions and interrupts. Users may add additional tracepoints through MAXAda-supplied bindings or with NightView.

MAXAda also includes a trace tool, which provides raw and symbolic (ASCII) analysis of the trace data.

An optional product, NightTrace is available to provide graphical analysis of the trace data as well as user trace daemons and operating system support.

Real-Time Monitoring

MAXAda includes a symbolic real-time data monitoring utility that allows users to view and modify data from multiple programs running concurrently, in real-time.

In addition, a task monitoring tool can monitor an Ada program in real-time by displaying system utilization and the activities of selected Ada tasking threads of execution, allowing users to cyclically monitor task and system information.

Both monitoring tools operate non-intrusively and are stand-alone utilities.

Real-Time Bindings

MAXAda supplies a number of bindings to real-time services, extending the capabilities of the Ada language. These real-time features provide an interface to operating system utilities, which include Concurrent's Frequency-Based Scheduler, a cyclic scheduler, high-resolution timing interfaces, and mutual exclusion interfaces. In addition

while multiple development efforts may occur simultaneously without conflict.

Error Processing

Innovative error-recovery techniques minimize cascading errors while maximizing the effectiveness of error-processing by referencing the related paragraph of the Ada 95 Reference Manual for each actual error. In addition, MAXAda can be configured to embed error messages in the source file where they occur and to open the corresponding source file in the user's editor when the build completes.

NightBench simplifies error processing, allowing users to select errors from a list generated during the build. Selecting an error from this list opens the supplied editor to the appropriate position within the

corresponding source file. For additional references and analysis, hyperlinks are provided to the related sections of the online versions of the MAXAda and Ada 95 reference manuals.

NightView Debugger

MAXAda uses the NightView debugger for source-level monitoring and debugging.

This graphical debugger is specifically designed for real-time applications, allowing multiple programs executing across multiple processors or single-board computers to be simultaneously monitored, manipulated, and debugged.

In addition to providing access to Ada source code during a debugging session, NightView allows the user to control the execution of a MAXAda program using

to the task synchronization features defined by the language, MAXAda supplies additional mechanisms that provide extremely efficient synchronization of two or more tasks.

Support also includes low-level interfaces for rescheduling, preemption control and client-server services, as well as bindings to direct memory access.

Ada Bindings to POSIX

A complete implementation of IEEE-Std-1003.5-1992 (POSIX.5) is included with MAXAda. POSIX.5 is a high-level abstract binding to IEEE-Std-1003.1, which provides complete access to items covered by the associated C standard bindings. Thin direct bindings to IEEE-Std-1003.1 (POSIX.1) and IEEE-Std-1003.1b (POSIX.1b) are also supplied with MAXAda.

Ada Bindings to X/Motif™

MAXAda supports the Ada X Interface (AXI) bindings to Xlib, Xt, and Motif. These bindings grant MAXAda programmers full use of over 500 functions and procedures in the X Window System network display services and the Motif graphical user interface standard.

Using AXI, MAXAda users can utilize X and Motif while taking advantage of Ada's modularity and language features (such as generics, tasking, and exceptions) when writing GUI applications. AXI is available as a separate product from Concurrent.

Extensive Online Help

MAXAda provides a graphical, topic-based help system that contains both the MAXAda Reference Manual and NightBench User's Guide in their entirety, as well as the complete Ada 95 Reference Manual. These online manuals reference each other through hyperlinks, providing the user with all relevant information about a particular topic. In addition, system main pages that are referenced in the online help system can be displayed under the same graphical interface.

Tutorials provide specific examples on using MAXAda and the NightBench Program Development Environment. Context-sensitive help is also provided for more specific information.

Understand for Ada

Understand for Ada is an optional interactive development environment that provides for reverse engineering, automatic documentation, code navigation and comprehension, metrics, maintenance and cross referencing for Ada83 and Ada95 source code. It is designed to assist engineers who have inherited large amounts of Ada legacy code or whose Ada projects have grown to immense size or complexity.

Worldwide Support and Service

Concurrent offers a wide range of service and support programs for its products, as well as for hardware from other vendors. Concurrent's Professional Services group provides custom hardware engineering, software development, and third-party integration services.

MAXAda™ Environment

```

nterm
[Task] <Sys> <Mem> <Lwp> <ReFrsh> <Int> <Verbose> <Ghosts> <Addr> <+><-> <Quit>
(pid 26874)
-----
lwpid grp tcb_addr task_name task_status
-----
11 b004edd0 DEFAULT<courier> executing
0 b004ae50 signal_handler<shadow:signintr> interrupts
* 3 1 b004a130 e executing
0 1 b0043af8 d delay (ready to wakeup)
0 1 b003d4c0 c async select (calling d.call_me):
async select (calling a.sync):
delay wakeup in 73.195082s
0 1 b0036e88 b entry call calling c.sync
0 2 b0030850 a entry call calling b.call_me
10 b002a218 signal_handler accept at entry: signintr
8 b0022858 handle<intr_courier:1> executing
7 b001e758 handle<courier:1> executing
5 b001a1b8 <timer> timer set wakeup in 73.195012s
6 b0007b60 <admin> executing
1 b00014a8 <environment> executing

interrupt_id received delivered pending
Urto0c1 446 445 1
SIGINT 27 27 0

lwpid grp group_name server_status
-----
* 4 1 my_group <idle>
3 1 my_group serving "e"
2 2 <predefined> <idle>

```

MAXAda provides a task monitoring tool that displays system utilization of an Ada program in real-time, allowing users to cyclically monitor task and system information.



2881 Gateway Drive
Pompano Beach, Florida 33069
Phone: 1-800-666-4544 or 954-974-1700
Ext. 5067, Sales or Marketing Support
FAX: 954-973-5398
E-mail: isd.info@ccur.com • www.ccur.com

Information subject to change without notice. Concurrent Computer Corporation and its design are registered trademarks and MAXAda, NightBench, NightStar, NightView, NightTrace and ARMS are trademarks of Concurrent Computer Corporation. OSF/Motif and X Window System are trademarks of Open Software Foundation. UNIX is a registered trademark licensed through X/Open Company. All other trademarks are the property of their respective owners. © 2003 Concurrent Computer Corporation RTlit 0010-0803 02000