

Ada

 DDC-I





Flexible Solutions Saving Time and Money

- **The Only Ada Solution that Supports the RTX® RTOS, as well as VxWorks, LynxOS and Bare Boards**
- **Full Ada Library Management Allows Groups to Maintain Consistency w/o Bottlenecks**
- **A Configurable Ada Run-time System Removes Non-Deterministic Behavior from Your Application**
- **Editable Debugger Window Saves Time and Effort**
- **User Customizable Menu Offers Flexibility**
- **Produces EABI Compliant Code in ELF Format**

Ada Compilers

- Support ANSI/ISO/IEC 8652:1995 standard version of the programming language Ada, as amended in 2000 (Ada95)
- Interrupt handling using protected procedures
- Pragma Profile (ARINC_653_Safe) restricts Ada programs to the safe subset for an ARINC 653-compliant RTOS.
- Support language annexes:
 - Annex A - Predefined Language Environment
 - Annex B - Interface to Other Languages
 - Annex C - Systems Programming
 - Annex D - Real-Time Systems
 - Annex G - Numerics
 - Annex H - Safety and Security
- Validated using ACATS test suite
- Compiler Switch to Support Legacy Ada 83 Applications

Targets:

- PowerPC Architecture
- Intel 80x86 Architecture
- MIL-STD 1750A
- TMS320C4x
- SPARC

Run-Time Systems

SCORE® supports four categories of run-time systems that will enable the developer to select the run-time environment suitable for the need. All of the run-time environments support C, Embedded C++, Ada, and mixed language applications. The run-time consists primarily of the root Ada program library, the Ada run-time library, libc, and libcpp.

- A Bare run-time provides the smallest and fastest run-time environment. There is no dependency upon an operating system - all language features are supported within the run-time. These run-times are:
 - Scalable - only the components needed by the application are included, (e.g. Ada tasking can be excluded)
 - Configurable to the hardware using the Board Support Package.
 - Certifiable to DO-178B level A.
 - Designed to be placed in ROM.
- A small step up from the Bare run-time is the simulator/emulator run-time. These are very similar to the Bare run-times, but execute in a simulated or emulated environment. Examples are the PowerPC Simulator (PSIM) and the 80x86 PC emulator (PC/Bare).
- The run-time can be linked for a native operating system such as Windows or Solaris. All of the OS features are available to the application.



SCOREAda is available as an Eclipse plug-in or with its own graphical user interface (GUI). The SCORE® GUI integrates all of the tools into a common integrated development environment. It provides a clean, efficient working environment and an object-oriented menu bar that does not contain superfluous bells and whistles. The features provided are robust tools typical developers will use everyday.

Additional functionality:

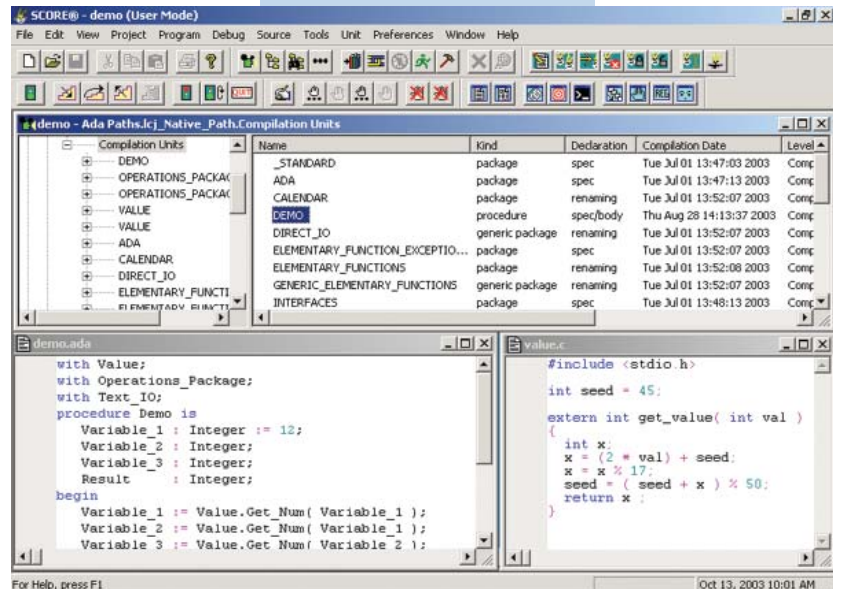
- Project management support is built-in through the definition of project-level options that are consistently applied throughout a project.
- A built-in source editor which provides color coding of syntax tokens and automatically indents program text. The same editor is used by the debugger when displaying source so edits can be made while debugging.
- Build and make capabilities are provided for programs and source files.

Symbolic Mixed-Language Debugger

The SCORE® Mixed-Language Debugger (MLD) is a full-featured, multi-windowed, non-intrusive debugging environment. It is offered both as an integral part of the Graphical User Interface (GUI) and as a command-line tool suitable for running standalone tests and debug scripts. The MLD seamlessly and automatically transitions between all of the languages found in the application.

Key features include:

- Enabling debug does not change the generated code - debug the final application
- Non-intrusive debugging while the application executes - even in ROM
- Debug optimized code efficiently - no need to disable optimizations to debug
- Recognizes C/C++, Ada, and Fortran syntax and expressions
- Customizable via startup files, initialization files, paths files, debugger abbreviations, scripts, and macros
- Log and replay debug sessions even when executed via the SCORE® GUI
- Debug at the high-level source level or the machine-code level
- View objects, expressions, call chains, execution traces, interspersed machine code, machine registers, program stack, etc.
- Supports full Ada-level debugging including constraints, attributes, tasking, break on exceptions and tasking events, and raise exceptions
- Communicates with remote Windows XP embedded target with RTX® support
- Supports Macraigor and Abatron JTAG Probes

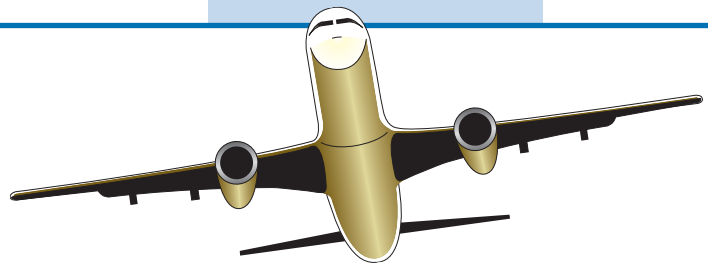


Customization

Many of the SCORE® tools are customizable by the user. The run-time environment is targeted to specific hardware by adaption of the board support package. The GUI features user-customizable menu items. The structure of the Ada program library is specifiable.

Additional Tools

- Assemblers
- Disassemblers
- Object tools
- Stand-alone ethernet and serial downloaders
- User-configurable target linkers



SCORE®Ada is used on the Aircraft Condition Analysis and Management System (ACAMS) program, an aircraft diagnosis and prognosis system designed to automatically diagnose and predict faults in complex aircraft subsystems, including flight subsystems, landing gear, and structural elements.

LegacyAda

DDC-I Ada Compiler System (DACS)

Ada 83

The DDC-I Ada Compiler System (DACS) has been designed to fully utilize the features of each processor, with special emphasis on the needs of true real-time embedded systems. Continuously validated since 1984, DACS are mature development systems, field proven by hundreds of applications.

Hosted on PC, Sun, or VAX, DACS cross compiler targets include Pentium, 80486, 80386, 80386EX, 80286, 80186, 8086 and Bare PC. Native compilers include DACS-PC for Windows and LynxOS.

- FAA Certification available (also tasking)
- Advanced symbolic Ada debugger
- Superior run-time performance
- Generates COFF compatible code
- High speed compilation
- Automatic recompilation and other program library tools
- Selective linking
- Stack analysis (static and dynamic)
- Real, protected and unsegmented flat modes
- Very compact and efficient ROMable code
- Easily configurable to your hardware
- CIFO and RTS entrypoint packages
- Supports rate monotonic scheduling
- Fast ethernet download

Ada Compiler

- Extensive optimizations
- Optimal Register Allocation
- Pragma interface to Assembler and C
- Representation clauses
- Address clauses
- Machine code insertions
- Extensive diagnostic messages

Library Management

- Tree-structured sublibraries
- Supports separate development
- Automatic recompilation
- Each library is protected by host file system
- Interactive utility for library management

Ada Linker

- Checks consistency of the Ada program
- User control of the final linking
- RTS control through link options

Ada Run-time System

- Preemptive priority scheduling
- Round-robin time slicing
- Super fast scheduling
- Semaphore operations available from Ada

DACS-Network Bridge software allows the DACS user to manage multiple targets from a remote location.

- Setup serial cables once
- Allows targets to be reserved for an individual user
- Allows debugging from remote terminals
- Allows querying of which targets are currently available, and in use

Symbolic Ada Debugger

- Program interrupt and restart
- Debugging session can be logged and played
- Command procedures may be defined
- Supports full Ada:
 - Set breakpoints at any statement or declaration
 - Break on exceptions, accepting an entry, leaving block/body containing tasks
 - Select breakpoint to effect a task or set of tasks
 - Debug fully optimized code including inline expansions and generics
 - Display Ada objects and their layout - variables, records, arrays, etc.
 - Display call chain, task structure and status
 - Modify variables, suspend/resume tasks and raise exceptions
- Supports machine level debugging
 - Display and modify machine code, registers and memory

DACS-PC/Bare turns your everyday PC into a bare machine for embedded development. The DDC-I run-time system replaces DOS and offers the same real-time behavior you would expect from a single board computer. Use any personal computer as a preliminary target board to develop the greater part of any embedded application. Development teams are able to test algorithms, timing and actual code and debug the greater part of an embedded application before the actual target hardware becomes available.

DACS-MAPP (Multi-Application Programming with Paging) provides key real-time operating system benefits supporting the development of a multi-application system, which consists of a system application and one or more user applications:

- Kernel using paging
- Break into a running system
- Application loading and unloading
- Call other applications
- Target output
- Withdrawing

Other Tools

- Assembler tools
- Disassembler merges Ada source with code
- Stack analyzer for proper stack size configuration
- Refather tool for configuration of libraries
- Make tool
- Stand-alone ether and serial downloader
- Multibus II communications packages
- TCP/IP
- Rate Monotonic Scheduling
- In Circuit Emulator
- 386EX Support



LegacyAda TADS Ada Development System (TADS) 83

TADS generates the fastest, most compact code available for the MIL-SPEC-1750A, Motorola 68xxx (68020, 68030, 68040), and Intel 960 (mcpm & mcsvm) target processors. TADS combines a highly optimized compiler with a modular run-time system and selective linking to produce the high performance code required by safety-critical real-time embedded applications.

The TADS Run-Time System (RTS) is a real-time, deterministic run-time executive designed strictly for real-time embedded applications. The TADS AdaScope debugger provides full source and machine-level debugging via the Run-Time in a window-based or command line environment.

Compiler

- Highly efficient code generation through target specific, Ada specific, and classical optimizations.
- Five optimization levels that permit the proper optimization strategy at each point in the development cycle.
- Representation specifications and support for address clauses that provide more control over generated code format.
- Predefined pragmas, implementation-defined pragmas, and support for unchecked programming provide control of compilation characteristics and run-time performance.

Librarian

- Tree-structured sublibraries isolate experimental code and allow development of alternate implementations.
- Layered system design facility allows the user to reflect and track and design or implementation changes.
- Concurrent user access and structured object code sharing facilitate team development of large applications.
- Compilations can be done in parallel.
- Automatic compilation and recompilation of units enhance programmer productivity.

Run-Time System (RTS)

- Modular structure allows all unnecessary RTS functionality to be excluded from the executable image.
- A priority driven pre-emptive scheduler provides real-time multitasking.
- The ARTClient package can be used to perform time-slice scheduling.
- A rendezvous accelerator speeds handling of simple rendezvous cases.
- Multiple options for handling interrupts: attachment to task entry, Ada procedure, or assembly routine.

AdaScope Debugger

The AdaScope debugger simultaneously displays multiple windows: command execution, source code, program input/output, scoreboarding of automatically updated data, disassembled machine code, source/assembly interspersed code, and on-line help. This flexible and comprehensive interface makes debugging convenient and productive.

- Full symbolic debugging at source or machine code levels.
- Full point-and-click access to pull-down menus on Windows host.
- Full tasking and exception support.
- Dynamic code patching of the executing program via built-in Ada subset interpreter.

Board Retargeting Kit: Allows the TADS Kernel Implementation (TKI) start-up code and debug monitor to be customized for and supported processor board.

Linker

- Precisely controls program layout in memory.
- Accepts commands from a highly customizable linker control file.
- Automatically selects required modules from libraries.
- Detects and eliminates unused code and data sections.

Generates a customizable linker map listing.

Tools and Utilities

AdaRef: Provides static analysis of an object file or executable image including the following:

- Subprogram Call Graph
- Compilation Unit Dependency Graph
- Cross Reference Listing
- Permuted Index Listing

AdaList: Generates an annotated description of an object file or executable image, optionally including interspersed source/assembly, symbolic references, absolute and relative addresses, and linker labels.

Checksum: Provides checksum support, memory gap fill, and binary listing.

Object File Librarian: Combines object files into an object file library for faster linking and easier maintenance.

Utilities: Convert TADS Object File Format (TOFF) to and from other formats; ReOrder bytes for exchange between hosts.

Packages

Standard Packages: Predefined Ada packages including Calendar, System, and Input/Output functions.

Run-Time Client Interface: Provides dynamic control of tasking operations, callable from the application and the debugger.

Intrinsics Package: Permits access directly from the Ada source to target specific hardware capabilities.

Machine Code Package: Allows the coding of in-line target specific assembly instructions in the Ada source code.

Math Package: Elementary transcendental functions compliant with specifications of the SIGAda Numerics Working Group (NumWG), the Ada Europe Working Group (A-ENVWG), and the ISO-IEC/JTC1/SC22/WG9 Numerics Working Group (NRG) - 68xxx and 960 only.

Optional Components

- Stand-Alone AdaScope
- Run-Time Modification Kit
- AdaScope Link Retargeting Kit
- AdaScope Protocol Retargeting Kit (1750A and 68xxx only)
- Assembler (1750A and 960 only)
- Math Package (1750A only)
- Target Simulator (1750A only)
- Prototype Development Unit (PDU) Support (1750A only)



Contact DDC-I, Inc.

With over 30 years of experience supplying complex COTS and custom embedded solutions, DDC-I provides long-term, strategic advantages to an ever changing safety critical industry. Boasting founding and committed management, our staff offers a level of experience and service that is unmatched in the industry. Our customers have direct access to DDC-I's highly experienced engineers, which ensures the success of our customer's development efforts.

For additional information about DDC-I's experienced, industry leading, safety critical development systems and solutions please contact:

DDC-I, Inc.
4600 E. Shea Blvd. #102
Phoenix, AZ 85028
USA
Tel: (602) 275-7172
Fax: (602) 252-6054
Email: sales@ddci.com
<http://www.ddci.com>

Distributor:



DDC-I

Safety Critical Software Solutions
for Mission Critical Systems

