



The Professional UNIX Development Tool for Building C Applications

CodeCenter™ is the proven, professional solution for prototyping, building, testing, debugging, enhancing, and maintaining UNIX C programs.

Greater Productivity in C Development

CodeCenter is the complete, "live-in" development environment for all stages of an application's development and maintenance. CodeCenter's functions support prototyping, editing, compiling, incremental linking, debugging, testing, file management, and interactive graphical browsing. CodeCenter delivers these functions in a multi-platform, open-architecture, standards-based package that ensures the highest return on an organization's investment in development tools.

Rapid Turnaround from Edit to Execute

CodeCenter gives you rapid turnaround in several ways. For prototyping and experimentation, CodeCenter's C interpreter in the Interactive Workspace provides immediate execution and testing. For program editing, CodeCenter lets you invoke vi or emacs with the click of a button, or use any other editor. CodeCenter compiles your edits using the CenterLine-C compiler or your system compiler and a unique, incremental linker then re-links only changed modules, saving time with every edit. Finally, CodeCenter lists any errors in the Error Browser and immediately displays suspect source code in the Source area for debugging.

Better Code Comprehension

The multi-level Cross-Reference Browser increases your code comprehension by letting you explore graphical maps of functions, global-variables, and their relationships. These interactive diagrams let you trace referenced functions or global-variables throughout your program, from the main level all the way down to the library level. Simply click on a function or variable name to examine its references. The cross-reference diagram can be output to a PostScript™ file and printed onto one or multiple pages.

Key Features

- A complete, tightly-integrated tool suite for faster delivery of superior quality code
- An interpreted, Interactive Workspace for component development, debugging, and rapid prototyping
- Incremental linker for fast turnaround on code changes
- Automatic run-time and static error checking
- Advanced integrated debugger for rapid debugging
- CenterLine-C compiler, for ANSI C and K&R C compliance
- Dynamic graphical browsers for understanding complex code
- API for enhanced customization and integration.
- Support for leading UNIX workstations, GUIs, and frameworks for maximum return on investment

Interactive Data Exploration

The Data Browser lets you examine and modify data structures in the same graphical way you think about them. It is invaluable for untangling complex data structures, including linked lists and tree structures. Choose a data identifier and the browser will show you its name, type, and value, all in a clear and intuitive format. Click on a pointer in a data structure, and the data browser instantly shows the referred data. At any time, you may edit values or change the data type. You can also visually distinguish lvalues from rvalues, check values of expressions at each break point, and check if a value is valid.





The Professional UNIX Development Tool for Building C Applications

Interactive Workspace

- Provides "on-the-fly" prototyping and experimentation
- Interpretively executes any C statement or block of statements
- Resolves external calls

Automatic Error Detection

- Checks automatically during loading or Workspace use
- Indicates error type and location in source code
- Finds more than 250 types of errors, including more than 80 run-time errors
- Identifies numerous errors at the source level when running object code
- Detects inter-module inconsistencies not reported by the UNIX linker
- Locates incorrect pointer values, illegal array indices, bad function arguments, type mismatches, and uninitialized variables
- Collects all errors and warnings for later browsing and correction

Advanced Integrated Debugging

- Accepts conditional breakpoints and watchpoints in source code
- Extends breakpoints and watchpoints to execute temporary C language "actions" in the Workspace
- Attaches to running processes and debugs linked executables
- Debugs threaded applications on Solaris workstations

Incremental Linker

- Reloads modified files without relinking the whole program
- Reduces turnaround time to a few seconds, even for large programs
- Links any mix of source and object modules

Browsers

- Cross-Reference Browser for tracing function calls and variable usage, plus PostScript printing of call graph
- Data Browser for examining and modifying data values and types
- Error Browser for reviewing run-time and static errors found during automatic error checking
- Project Browser for examining and controlling project modules and libraries
- Manual Browser for on-line access to documentation
- Options Browser for controlling CodeCenter options

Integration

- Fully supported and documented API
- Easily integrates with UNIX tools, e.g. make, vi, emacs
- Uses CenterLine-C, platform vendor's compiler, or gcc
- Uses make and existing makefiles to update and build programs
- Supports leading preprocessors, including preprocessors for embedded SQL
- emacs editor integration allows access to all CodeCenter functions from within emacs





The Professional UNIX Development Tool for Building C Applications

Rapid Prototyping

Only CodeCenter's unique, interpreted, Interactive Workspace provides programmers with the ability to try out a few lines of code without having to write a complete program or module. With CodeCenter, you can write C code fragments and execute them immediately. Development is incremental, allowing you to work and experiment with manageable units of code such as functions and structures. CodeCenter incrementally links your experimental code to any loaded code or libraries and thoroughly checks it for errors. This unit-based development approach offers the flexibility to build quality into your complex applications from the ground up by executing and testing each component as you go.

Advanced Debugging

CodeCenter's advanced, integrated, source-level debugger helps you find and fix bugs fast. For instance, if CodeCenter finds an error, it immediately displays the suspect source code in the Source Area for debugging. In the Source Area, you can not only trace code line-by-line and set breakpoints or watchpoints, but also conditionally branch to temporary C-language "actions" in the Interactive Workspace to help identify problems quickly without editing your program. Even the Data Browser is fully integrated so you can see data values continuously and dynamically modify them as you debug. For debugging full executables, CodeCenter gives you process debugger features to handle running processes and linked executables. The process debugger can debug threaded applications. A threads browser is also available for displaying all threads and for switching context between threads.

Automatic Testing

The Error Browser keeps you informed as CodeCenter automatically tests every line of code for over 250 run-time and static errors, including such common but elusive errors as incorrect pointer values, illegal array indices, bad function arguments, invalid pointer casts, type mismatches, and uninitialized variables. CodeCenter also detects inter-module inconsistencies missed by the UNIX linker. Testing is built into the normal CodeCenter programming cycle. Whether you load a single object, a few modules, or a full program—or even if you just try out a few lines in the Interactive Workspace—CodeCenter rapidly executes a battery of tests, storing error messages in a folder for convenient browsing. Just click on an error message to begin debugging the suspect source code in the Source area.

Project-Level Control

The Project Browser gives you control of your entire project, showing a list of all files and libraries currently loaded. This browser also displays which modules are loaded as source, object, or libraries, and allows you to switch quickly between source and object code. The Project Browser makes it easy to view headers, types, typedefs, and C definitions in a file; just click on a button to display or edit the file's contents. It also provides a customizable, graphical way to check files in or out through standard configuration control systems such as SCCS and RCS.



**Integrated Computer
Solutions Incorporated**

54 Middlesex Turnpike • Bedford, MA • 01730
www.ics.com • info@ics.com • 617.621.0060