ObjectCenter

The Most Efficient Environment for Building C and C++ Applications

ObjectCenter™ is the only UNIX C/C++ programming environment that provides the interactivity you need to maximize all of the benefits of C++ and object-oriented programming.

Greater Efficiency in C++ Development

ObjectCenter enables professional C++ developers to increase their productivity, improve code quality, maximize reusability, and lower maintenance costs. Available on all leading UNIX workstations, ObjectCenter provides the best platform-independent programming environment for developing portable C++ components and applications. ObjectCenter even includes a unique, interpreted Interactive Workspace that provides the capabilities of classic object-oriented environments to enable users to interact with individual C++ components. This unit-based approach to developing C++ code, coupled with ObjectCenter's extensive run-time error checking, is the safest and fastest route to ensuring that you build quality into your applications from the first line of code.

Develop Better C++ Code, Faster

ObjectCenter gives you the rapid interactivity you need with the fastest edit-to-execute turnaround available. Use the editor of your choice to make changes. Vi and emacs are available at the click of an icon. ObjectCenter's integral CenterLine-C++ compiler skips the time-consuming step of recompiling header files and optimizes work with large class libraries. ObjectCenter also saves time by incrementally linking only the edited modules.

Automatically Detect Errors

ObjectCenter automatically checks every line of your code at load, compile, and run time—maximizing code quality and minimizing debugging time. Code is checked for more than 250 run-time and static errors, whether it is loaded as a module or entered through the C/C++ interpreter in the Workspace. The Error Browser records error messages in a folder for convenient viewing, and the Source Area subsequently loads suspect source code for debugging.

Key Features

- An interpreted, Interactive Workspace for component development, debugging, and rapid prototyping
- Process-and-component-level debugging modes for power and scalability
- Complete C++ templates support for builder reusable classes
- Automatic static and dynamic runtime detection of over 250 errors for optimal code quality
- Integrated graphical browsing tools for visualizing class hierarchies, finding and reusing classes, and understanding control flow
- API and open architecture for enhanced customization and integration
- CenterLine-C++ high-speed compiler and incremental linking for rapid edit-to-execute time
- Full C language support to protect existing C investments while easing the migration from C to C++

Execute Prototype Code Without Waiting to Link/Compile

Only ObjectCenter'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 ObjectCenter, you can write C and C++ fragments and execute them immediately. Development is incremental, allowing you to work and experiment with manageable units of code, such as classes, functions, and statements. ObjectCenter incrementally links your experimental code to any loaded code or data 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.

The Most Efficient Environment for Building C and C++ Applications

Interactive Workspace for C or C++

- Provides "on-the-fly" prototyping and experimentation
- Interpretively executes any C or C++ statement or block of statements, including templates
- Resolves external calls
- Automatically checks for run-time and static errors

Object-Oriented, Mixed-Mode Debugging

- Supports unit-based interpreted debugging mode (CDM)
- Supports process debugging mode (PDM) with integrated, dbx-style process debugger
- Lets the user mix interpreted source and object code in CDM mode
- Debugs any loaded module, whether source or object code, C or C++, including templates
- Accepts conditional breakpoints and watchpoints
- Extends breakpoints and watch points to execute temporary "actions" in the Interactive Workspace

Automatic Error Detection

- Checks automatically during module loading or execution and, during Interactive Workspace use, indicates error type and location in source code; collects all errors and warnings for later browsing
- 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, memory leaks, dangling pointers, illegal array indices, bad function arguments, type mismatches, and uninitialized variables
- Allows users to suppress uninteresting errors/ warnings

CenterLine-C++ Compilation

- Precompiles header files to significantly reduce recompilation time when using large class libraries
- Significantly reduces object code size for debugging
- Complies with ANSI standards

Incremental Linker

- Reloads modified files without relinking the entire program
- Keeps turnaround time short, even with large programs
- Allows mixed use of source and object modules within a programming environment

Browsers

- Project Browser for examining and controlling project modules and libraries
- Class Examiner and Inheritance Browser for viewing class and member information
- Cross-Reference Browser for viewing function and variable references
- Data Browser for graphically viewing and modifying data structures
- Manual Browser for online access to documentation
- Options Browser for controlling ObjectCenter options

Integration

- Fully supported and documented Application Programming Interface provides access to CenterLine Engine
- Easily integrates UNIX tools, e.g. make, vi, emacs
- Supports leading preprocessors, including for embedded SQL

The Most Efficient Environment for Building C and C++ Applications

Maximize Your Code Reuse with Full Template Support

Templates are an essential feature of the C++ language for working with reusable code libraries. Only ObjectCenter provides complete template support throughout the environment. To give you the full power of object-oriented development in C++, ObjectCenter lets you automatically instantiate, examine, debug, modify, run, and test C++ templates and gain easy access to their member functions and data.

Fully Comprehend C++ Classes

ObjectCenter's class browsers help you extract the information you need from complex C++ classes and reusable libraries. The Inheritance Browser graphically depicts all levels of the hierarchy and includes virtual classes. Click on "Examine Class" for a selected class, and the Class Examiner displays member functions or data. Unique filters let you separately explore inherited, public, protected, and/or private interfaces and selectively display static, virtual, constructor, or other member types. As always, with ObjectCenter editing is only a mouse-click away.

Interactively Explore Cross-References

ObjectCenter's Cross-Reference Browser helps you comprehend your code better through interactive exploration and multi-level viewing. This graphical browser helps you trace cross-references to any function or variable, including class member functions, constructors, overloaded functions, and virtual functions.

Improve Your Debugging with an Object-Oriented, Mixed-Mode Approach

Object-oriented programming requires a different level of debugging support than procedural programming. ObjectCenter’s integrated Data Browser can be used to inspect complex objects. Visual access to these objects via the browser is much simpler than with traditional command line debuggers, since the objects can be manipulated directly from within the browser. Data members can be filtered and pointers to other objects can be traversed interactively to make navigation through large structures easier. ObjectCenter's Source Area simplifies exploration and debugging of your program's execution. The Source Area lets you step through code line-by-line (including C++ templates), set breakpoints and watchpoints, and even conditionally branch to "actions" (temporary code) in the Interactive Workspace. The Data Browser allows continuous viewing and modification of objects and data values. ObjectCenter also dramatically reduces object file size, ensuring responsive debugging for today's large programs.

Enhance Top-Level Insight and Control

Project-level insight is essential for complex, real-world C++ applications intended for commercial production environments. The Project Browser provides top-level views of an entire program, showing all loaded modules. Click on "Contents" to view a file's functions, variables, headers, types, and typedefs, or click on an icon to edit. The Project Browser also provides a graphical interface for standard control systems such a s SCCS, RCS, and others, and provides a convenient launching point for editing and other tools.