Sun Visual WorkShop C++ Overview

THE NETWORK IS THE COMPUTER™

Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
901 San Antonio Road Palo Alto, CA 94303-4900 USA
650 960-1300
fax 650 969-9131

Part No.: 805-4964
Revision A, February 1999

Please
Recycle

Adobe PostScript™

Contents

Preface

This manual describes the capabilities of Sun™ Visual WorkShop™ C++ and guides you to the documentation you need for further information. Some of these products you already use, but others you may not use or even know about.

Who Should Use This Book

This document addresses application developers who are new users of Sun Visual WorkShop C++. It assumes that you have a working knowledge of the Solaris™ operating environment and UNIX® commands.

How This Book Is Organized

This overview is organized as follows:

Chapter 1, "Welcome to Sun Visual WorkShop C++," is an overview of the products included in Sun Visual WorkShop C++ and how they fit together.

Chapter 2, "Building and Compiling," discusses building and compiling processes.

Chapter 3, "Editing and Browsing," describes editors and benefits of the source browser.

Chapter 4, "Debugging and Tuning," reviews debugging and performance tools.

Chapter 5, "GUI Building," discusses the Sun™ WorkShop™ Visual GUI designer.

Chapter 6, "Managing Source Code," sketches the benefits of Sun™ WorkShop™ TeamWare code management tools.

Glossary is a list of words and phrases found in this book and their definitions.

Multiplatform Release

Note – The name of the latest Solaris operating environment release is Solaris 7 but code and path or package path names may use Solaris 2.7 or SunOS 5.7.

The Sun™ WorkShop™ documentation applies to Solaris 2.5.1, Solaris 2.6, and Solaris 7 operating environments on:

- The SPARC™ platform
- The x86 platform, where x86 refers to the Intel implementation of one of the following: Intel 80386, Intel 80486, Pentium, or the equivalent

Note – The term “x86” refers to the Intel 8086 family of microprocessor chips, including the Pentium, Pentium Pro, and Pentium II processors and compatible microprocessor chips made by AMD and Cyrix. In this document, the term “x86” refers to the overall platform architecture. Features described in this book that are particular to a specific platform are differentiated by the terms “SPARC” and “x86” in the text.

Sun Visual WorkShop C++ Related Books

This overview briefly describes the components of the Sun Visual WorkShop C++. The following sections describe other books included with this workshop.

Other Programming Books

- *C++ User’s Guide* provides information on command-line options and how to use the compiler.
- *C++ Programming Guide* discusses issues relating to the use of templates, exception handling, and interfacing with FORTRAN 77.

- *C++ Migration Guide* describes migrations between compiler releases.
- *C++ Library Reference* explains the `iostream` libraries.
- *Tools.h++ User's Guide* provides details on the `Tools.h++` class library.
- *Tools.h++ Class Library Reference* discusses use of the C++ classes for enhancing the efficiency of your programs.
- *Numerical Computation Guide* details floating-point computation numerical accuracy issues.
- *Standard C++ Library User's Guide* describes how to use the Standard C++ Library.
- *Standard C++ Class Library Reference* provides detail on the Standard C++ Library.

Other Sun WorkShop Books

- *Sun WorkShop Quick Install* provides installation instructions.
- *Sun WorkShop Installation and Licensing Reference* provides supporting installation and licensing information.
- *Using Sun WorkShop* gives information on performing development operations through Sun WorkShop.
- *Debugging a Program With dbx* provides information on using `dbx` commands to debug a program.
- *Analyzing Program Performance With Sun WorkShop* describes the profiling tools; the LoopTool, LoopReport, and LockLint utilities; and use of the Sampling Analyzer to enhance program performance.
- *Sun WorkShop TeamWare User's Guide* describes how to use the Sun WorkShop TeamWare code management tools.
- *Sun WorkShop Performance Library Reference Manual* discusses the library of subroutines and functions to perform useful operations in computational linear algebra and Fourier transforms.

Solaris Books

The following Solaris manuals and guides provide additional useful information:

- The Solaris *Linker and Libraries Guide* gives information on linking and libraries.
- The Solaris *Programming Utilities Guide* provides information for developers about the special built-in programming tools available in the SunOS™ system.

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of the SunExpress™ Internet site at `http://www.sun.com/sunexpress`.

Accessing Sun Documents Online

Sun WorkShop documentation is available online from several sources:

- The `docs.sun.com` Web site
- AnswerBook2 collections
- HTML documents
- Online help and release notes

Using the `docs.sun.com` Web site

The `docs.sun.com` Web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Accessing AnswerBook2 Collections

The Sun WorkShop documentation is also available using AnswerBook2 software. To access the AnswerBook2 collections, your system administrator must have installed the AnswerBook2 documents during the installation process (if the documents are not installed, see your system administrator or Chapter 3 of *Sun WorkShop Quick Install* for installation instructions). For information about accessing AnswerBook2 documents, see Chapter 6 of *Sun WorkShop Quick Install*, Solaris installation documentation, or your system administrator.

Note – To access AnswerBook2 documents, Solaris 2.5.1 users must first download AnswerBook2 documentation server software from a Sun Web page. For more information, see Chapter 6 of *Sun WorkShop Quick Install*.

Accessing HTML Documents

The following Sun Workshop documents are available online only in HTML format:

- *Tools.h++ Class Library Reference*
- *Tools.h++ User's Guide*
- *Numerical Computation Guide*
- *Standard C++ Library User's Guide*
- *Standard C++ Class Library Reference*
- *Sun WorkShop Performance Library Reference Manual*
- *Sun WorkShop Visual User's Guide*
- *Sun WorkShop Memory Monitor User's Manual*

To access these HTML documents:

1. **Open the following file through your HTML browser:**

   install-directory`/SUNWspro/DOC5.0/lib/locale/C/html/index.html`

   Replace *install-directory* with the name of the directory where your Sun WorkShop software is installed (the default is `/opt`).

   The browser displays an index of the HTML documents for the Sun WorkShop products that are installed.

2. **Open a document in the index by clicking the document's title.**

Accessing Sun WorkShop Online Help and Release Notes

This release of Sun WorkShop includes an online help system as well as online manuals. To find out more see:

- Online Help. A help system containing extensive task-oriented, context-sensitive help. To access the help, choose Help ⇒ Help Contents. Help menus are available in all Sun WorkShop windows.
- Release Notes. The Release Notes contain general information about Sun WorkShop and specific information about software limitations and bugs. To access the Release Notes, choose Help ⇒ Release Notes.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER 1

Welcome to Sun Visual WorkShop C++

Sun Visual WorkShop C++ is a tightly integrated environment designed to speed software development for you and your team. This package simplifies the tasks you perform most often: compiling, building, browsing, editing, debugging, and performance tuning.

Sun Visual WorkShop C++ includes the following major components: Sun WorkShop, Sun WorkShop Compilers C++ and C, multithreaded tools, Sun WorkShop TeamWare, and Sun WorkShop Visual GUI designer.

Take Advantage of an Integrated Environment: Sun WorkShop

Sun WorkShop supplies an integrated programming environment with a full set of graphical tools to create and maintain your C++ and C applications.

In Figure 1-1, these tools are shown on the outside edge of the “Sun WorkShop” block. The tasks these tools perform are shown inside the circle. Employ these tools separately or in combination:

- A GUI-based debugger that lets you run your programs in a controlled manner and inspect the state of a stopped program
- An analyzer that measures, records, and helps you improve the performance of your application program
- A source browser that helps you understand the hierarchy of your code and lets you search the code base
- A file-merging tool that loads and displays two text files side-by-side for easy comparison

- A choice of three integrated editors (XEmacs, Emacs, vi) to write and alter code
- A `make` utility that aids code compilation by automating and distributing compilation of source files that have changed

Build High-Performance Applications With Sun WorkShop Compiler C++

Sun WorkShop Compiler C++ helps you build high-performance C++ applications with:

- Incremental linker (`ild`) to reduce link time
- `Tools.h++` 7.0.7 and 8.0 class libraries
- Sun WorkShop Memory Monitor to solve memory problems

The C++ language features offer improved support for the ISO C++ standard, including:

- Standard C++ Library
- Namespaces
- Bool type
- Koenig lookup
- Mutable members
- Typename
- Template default parameters

Parallelize With Multithreaded Development Tools

With multithreaded tools you can take advantage of the parallelism inherent in your programs to analyze and improve performance. (Multithreaded tools are available on SPARC platforms only.) Use these tools to

- Have access to an autoparallelizing C compiler
- Analyze parallel loop performance
- Analyze ANSI C source code to detect data races and deadlocks

Manage Code With Sun WorkShop TeamWare

Sun WorkShop TeamWare code management tools extend the Sun WorkShop tools and C++ language systems with a suite of graphical tools for source code version control, release integration, and release management. These speed software development by helping your team to work together.

Build GUIs Quickly With WorkShop Visual

Use the Sun WorkShop Visual GUI builder to build sophisticated applications quickly and easily. View your GUI design as you build and make changes. When your design is complete, Sun WorkShop Visual automatically generates portable, object-oriented C++, C, or Java™ code.

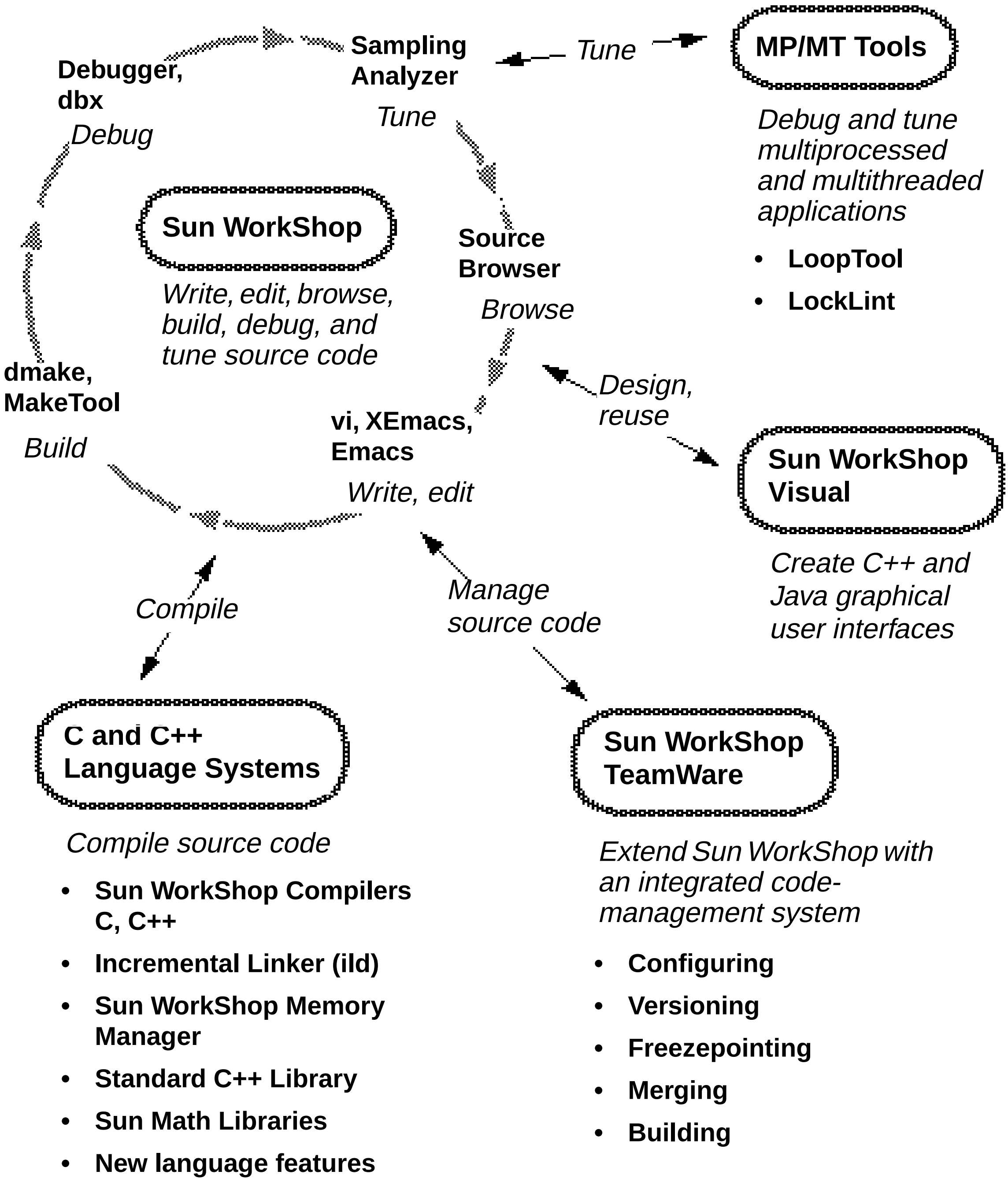

FIGURE 1-1 Sun Visual WorkShop C++

CHAPTER 2

Building and Compiling

With Sun Visual WorkShop C++, you can build and compile your programs efficiently, starting with the use of a single graphical interface as a control panel.

Program in an Integrated Environment

With Sun Visual WorkShop C++, you can manage and execute a complete suite of development tools through a single graphical user interface.

Since developers tend to have their favorite editors, Sun WorkShop offers three of the most popular: vi, Emacs, and XEmacs. When using Sun WorkShop, you can choose your editor from the Text Editor Options dialog box.

Simplify Building With Sun WorkShop Building Window

Your makefile describes explicitly how to build each module of your program as well as the final executable. Sun WorkShop Building is a graphical interface to the UNIX `make` utility, the utility that oversees program compilation and ensures that your programs are compiled from the newest sources. After you initiate a program build in the Building window, the build process reads the makefile for the program and executes the commands it finds there.

Use Building to expand the makefile rules and macros, increase productivity, and reduce errors. Sun WorkShop Building gives you the following advantages over issuing `make` commands from the command line:

- Storage of high-level makefile target names in a menu for easy access
- Visual feedback about the progress of your build
- Links from the build errors listed in the Building window that open the source files containing the errors in the text editor window
- Dialog boxes for changing the make options, makefile macros, and environment variables passed into the build (Sun WorkShop does not generate a makefile for you
- A choice of three build modes:
  - Execute one job at a time on the local host
  - Execute multiple build jobs concurrently on the local host
  - Execute multiple build jobs over several build servers

Improve Performance With the C++ Compiler

Take advantage of the Sun C++ Compiler, an optimizing, native compiler that closely follows the emerging ANSI C++ language definition. Make significant improvements in your program execution speed by using the variety of techniques the C++ compiler offers.

In addition to fast compilation and execution of program code, the Sun WorkShop Compiler C++ gives you the following features:

- Native C++ compiler
- Incremental linker (`ild`) to reduce link time
- Sun WorkShop Memory Monitor to solve memory management problems
- Exception handling for improved error handling
- Fast template instantiation
- `Tools.h++` class libraries

Test Changes Faster

Once a debugging session is completed, you can rebuild your application using the incremental linker. You can test the changes to your application more quickly because the incremental linker reduces the time necessary to relink an application. Only the changed `.o` files are linked into the previous executable to create a new executable file. Using the incremental linker, the time it takes to rebuild an application after a change is proportional to the magnitude of the change rather than to the total size of the application.

Prevent Memory Leaks

A significant addition to this release is the Sun WorkShop Memory Monitor, an automatic memory manager for C++ and C. With this feature, you can prevent memory leaks from occurring simply by relinking your program.

Sun WorkShop Memory Monitor automatically frees up leaked memory. Using this tool, you can write new programs without calling `free` or `delete`, and you can automatically fix memory errors in existing programs, even those that use third-party libraries.

You can also monitor your memory usage from within your browser and identify how much memory each part of your program is using at any time.

Sun WorkShop Memory Monitor also has a web interface that allows remote debugging. You can debug programs that exhibit memory problems in the field, but cannot be reproduced in house.

Control Optimization Levels

Take advantage of the wide range of options offered with Sun C++ compilers to manage code and application performance. Specify the `–O` option to optimize and improve the execution of your program. Balance performance gain, code size, and compilation time through efficient use of `–O` levels.

Reuse Code

C++ class libraries are modular components of reusable code. Use class libraries to integrate blocks of code that have been previously built and tested. The C++ class libraries include:

- Task library
- Complex library
- Standard C++ Library
- Multithreading-compatible iostream library (`libC`)

Save Time With the `Tools.h++` Class Library

Save programming time with `Tools.h++`, the C++ foundation class library, a set of classes that are useful for essentially any programming task. `Tools.h++` includes:

- Full set of reusable classes
- Full set of template classes

- Precondition checking for development time
- Persistent store

As a de facto industry standard, `Tools.h++` has been included with a wide variety of compiler products. Consequently, `Tools.h++` is available on all UNIX platforms.

Enhance Performance With the C Compiler

The C compilation system offers a compiler, assembler, and link editor. Enhance the performance of your C programs with the following C Compiler features:

- Incremental linking
- Support for automatic and explicit parallelization of loops
- Ability to turn off specific warning messages
- Support for UltraSPARC™ code generation
- Year 2000 support

CHAPTER 3

Editing and Browsing

Ease of editing and browsing are critical to efficient program development and maintenance. With Sun WorkShop editors and the source browser you can navigate quickly through large bodies of code.

Choose Your Editor

In Sun Visual WorkShop C++, you can write and edit source code more productively with your preferred editor: XEmacs, Emacs, or vi. Link to your source code line in the editor window to perform error and source browsing, and referencing. With these editors, you can perform many development functions and share task information with the other components of Sun WorkShop.

Try Out the Source Browser

Use the source browser to help you develop and maintain software systems, especially large ones. During the course of a programming project, you might join a new programming team to enhance, maintain, or port code. Before you become a productive team member, you must fully understand the code that you will modify. The source browser is a powerful tool for browsing large programs.

The source browser uses a “What you see is what you browse” paradigm. The source code you edit is the same source code the source browser uses in its searches.

Search for Patterns

You can use the source browser in pattern search mode to search for regular expressions or simple text strings. Pattern searching uses standard `grep` syntax to find all source lines that match the search string.

Search by Query

You can the use source browser in source browser mode to issue queries. A query instructs the source browser to locate all occurrences of, definitions of, uses of, or assignments to the symbol, string, or search pattern you specify. You then view the occurrences and surrounding source code.

Search Local or Network Files

You can search and browse source files residing locally or on the network. You can make global changes and pinpoint problem spots, even on large, distributed development projects.

Search Multiple-Language Programs

When you browse a program that uses more than one language in the implementation, the source browser automatically determines the language in which each source file is written. The browsing operations do not change from one language to another.

Experience Flexible Browsing

You can use the source browser in either the Sun WorkShop Browsing window (both modes) or a command-line environment (source browsing mode only).

You can edit source code using your favorite editor within the source browser.

Use Static Analysis to Examine Source Code

Use the source browser to examine your source code in an organized fashion. You can gain better understanding of a piece of software by:

- Observing the structure of a program, as defined by its source code
- Identifying:
  - Called and calling functions
  - Graphs of class genealogy
  - Organization of header files and defining statements

CHAPTER **4**

Debugging and Tuning

Many variables affect the performance of an application program. The one over which you have the most influence is the design of your program. With debugging and performance tuning, you can make your program efficient, reliable, and fast. Sun Visual WorkShop C++ includes a variety of debugging and performance tools to analyze source code, isolate problems, and provide you with the information you need to finely tune your applications for maximum performance.

Debugging

For your debugging tasks, use one of the two closely related debugging tools contained in Sun Visual WorkShop: `dbx` or the debugger.

For Command-Line Debugging, Use `dbx`

The `dbx` debugger is an interactive, source-level, command-line tool. You can use it to run a program in a controlled manner and to inspect the state of a stopped program. `dbx` gives you complete control of the dynamic execution of a program, including the collection of performance data.

Make use of the multithreaded features that are built into the standard `dbx`. You can identify all known threads, including their current state, base functions, and current functions. You can also examine thread stack traces. To ensure proper execution, you can debug threads by stepping through or over a thread, navigating between threads, and then resuming execution at any time.

Note – If you wish to debug your multithreaded application with `dbx`, you must include the `–lthread` option at link time.

If You Prefer a GUI, Use the Debugger

Should you favor a graphical interface to dbx, try the debugger (the Sun WorkShop Debugging window). During program execution, dbx obtains detailed information about program behavior and delivers this information to the debugger by a communications protocol. You can debug more easily because you can enter most commands by clicking redefinable buttons in the GUI.

You can also edit your programs with your favorite editor from the debugger and minimize the need to change tools.

Fix and Continue

With the Fix and Continue feature of the debugger you can modify source code, recompile the file, and continue program execution, without leaving the debugger. When you use this feature, you eliminate relinking and reloading the program.

Runtime Checking

Use Runtime Checking (RTC) to find elusive memory access violations and memory leaks in both single-threaded and multithreaded applications. With runtime checking, you can detect runtime errors in an application during the development phase. As errors are detected the debugger interrupts program execution and displays the relevant source code so you can fix bugs as they are found.

Tuning

After you have successfully debugged your program, you can evaluate its performance with the Sampling Analyzer, a program designed to help you tune application performance, including memory allocation. The Sampling Analyzer measures and graphically displays your application’s performance profile and suggests ways to improve performance. Its special data collection instrumentation eliminates the need to continually compile and link an application—any program that has been compiled can be analyzed.

Analyze a Variety of Performance Data

The performance data you can examine in the Sampling Analyzer include:

- User time. Time spent executing user program instruction
- Fault time. Time required to service fault-driven memory activities, classified into text and data page faults
- I/O time. Time the operating system spent waiting for input/output operations, such as writing to a disk or tape
- System time. Time the operating system spent executing system calls
- Trap time. Time spent in executing traps (automatic exceptions or memory faults)
- Lock wait time. Time spent waiting for lightweight process locks
- Sleep time. Time the program spent inactive, waiting for a wake up signal
- Suspend time. Time spent temporarily halted (includes time spent in the debugger during breakpoint and the time used by the Sampling Collector to gather data)
- Idle time. Time spent waiting to run while the system was busy
- Function sizes. Sizes of functions in the program
- Module sizes. Sizes of modules in the program
- Segment sizes. Sizes of segments in the program
- Memory usage. Memory page reference and modification data
- Resource usage. Information about the system resources that are used by the program, including major and minor page faults, process swaps, number input and output blocks, number of messages sent and received, number of signals handled, number of voluntary and involuntary context switches, number of system calls, number of characters of input/output, and number of working set memory pages

Control Your Analysis

The debugger serves as the data-gathering front end for the Sampling Analyzer. You can control the data collection process with the Sampling Collector window in dbx or the debugger while your program is running. You can collect data only between breakpoints, or you can limit data collection to a particular part of the program. The program run in which you collect data is known as an experiment, and the data file created by the Collector is called the experiment record. You then use the Sampling Analyzer to identify performance bottlenecks in the collected data.

Note – Performance tuning and runtime checking are mutually exclusive processes. You can perform only one or the other at a time. The information you receive from tuning your application can be adversely affected if you try to perform runtime checking simultaneously.

Focus on Problems

Test your hypotheses about a program's behavior by focusing on the areas where performance problems occur. To rebuild your programs with improved performance, use the Sampling Analyzer to identify areas where you can improve ordering for loading functions into the program's address space. In some cases, the Sampling Analyzer can improve performance automatically by creating a mapfile that instructs the linker to remap functions in memory more efficiently.

Find Which Modules Do the Calling

Performance analysis tools provide a range of analysis levels, from simple timing of a command to a statement-by-statement analysis of a program. While a flat profile can provide valuable data for performance improvements, sometimes the data is not sufficient to point out exactly where improvements can be made. You can obtain a more detailed analysis by using the call graph profile to identify which modules are called by other modules, and which modules call other modules.

Multiprocessing and Multithreading

Multiprocessing (MP) is the hardware technology that supports tightly coupled multi-CPU systems with shared memory. Multiple CPUs provide more power to drive application performance.

Multithreading (MT) is the software technology that enables the development of parallel applications, whether on single- or multiple-processor systems. Independent threads of execution can be scheduled on multiple CPUs in a multiprocessor system, but they share resources such as memory and files, allowing single applications to execute code in parallel. Threads share resources, synchronize, and communicate with each other through the use of mutual exclusion (mutex) locks provided by the operating system. Multiprocessing and multithreading together give you a scalable solution for higher application performance on the SPARC platform.

Take Advantage of Parallelism

If your applications use parallelism, use the new multiprocessing systems and multithreaded operating environments to improve performance, responsiveness, and flexibility. With multithreading you can:

- Increase performance on multiprocessor systems

- Increase performance on uniprocessor systems
- Use resources more efficiently
- Identify all known threads
- Examine thread stack traces

Speed Error Detection With Multithreaded Development Tools

Use multithreaded development tools to extend the C++ compiler (and multiprocessing C compiler) for multiprocessing optimizations. The multiprocessing/multithreading toolset includes multithreaded extensions to the Sun WorkShop debugger and `dbx`, and two additional tools: LockLint and LoopTool.

Find Inconsistent Lock Use With LockLint

Use LockLint to do static analysis of the use of mutex and read/write locks. In searching for inconsistent lock use, LockLint detects the most common causes of data races and deadlocks.

Analyze Loop Information

Take advantage of LoopTool and LoopReport, performance analysis tools used with the multiprocessing C++ and C compilers. The compilers automatically parallelize loops when they determine that it is safe and profitable to do so. With LoopTool you can:

- Browse parallel and sequential loops and view a graph of loop runtimes
- View displays of performance data showing which loops were parallelized
- Obtain compiler hints about why a loop was not parallelized
- Go directly from the graphical display to the source code for any loop, simplifying source code editing

Use the LoopReport command-line tool to create a summary table of all loop runtimes correlated with compiler hints about why a loop was not parallelized.

CHAPTER 5

GUI Building

User interface code can amount to seventy percent of your application’s source code. A significant portion of this code is considered standard—all applications include something similar. The graphical user interface builder in the Sun WorkShop environment generates this common code automatically, significantly reducing application development time and increasing code reliability and quality.

Visually Design Your GUI

Sun WorkShop Visual is an interactive tool that facilitates the development of a Motif or Microsoft Windows graphical user interface (GUI). You can see what the interface looks like and how it behaves while you are building it.

Create your user interface by clicking through a Motif widget hierarchy from an easily accessible icon palette and assembling the widgets in a construction area on your screen. You can add custom and third-party widgets to the palette as you need them, and take advantage of existing code.

Some important Sun WorkShop Visual features are:

- A simple, easy-to-use GUI designer
- Layout and pixmap editors
- Font and color choosers
- Full access to the Motif API
- Protection against invalid settings
- Code generation for C, C++, and UIL code
- Extensible widget palette
- Support for code reuse

- Support for developing Microsoft Windows GUIs using Microsoft Foundation Classes (available separately as an option)

Automatically Generate C++ and C Code

When you have finished testing your user interface, you can automatically generate highly portable code in traditional Kernighan and Ritchie (K&R) C or standard ANSI C, as well as structured code, using C++ classes.

Reuse Code

Use the structured code generation features of Sun WorkShop Visual to create reusable definitions and instances. A definition is a hierarchy of widgets whose root widget has been designated a class. When you create a definition, its hierarchy is frozen and an icon that creates instances of the class is added to the widget palette.

Develop Across Platforms

You can also use Sun WorkShop Visual to generate code for Microsoft Windows, using Microsoft Foundation Classes. Although the Motif and X toolkits are similar in appearance to the Microsoft Windows toolkit, the use of the toolkits is very different. Handle these differences in Sun WorkShop Visual using Windows mode. Use Sun WorkShop Visual's color cuing to set restrictions on generating Microsoft Windows code for a design. A design that conforms to the restrictions is said to be *Windows-compliant*, and C++ code can be generated for the Microsoft Foundation Classes.

Note – This feature is available as an option.

CHAPTER 6

Managing Source Code

Large development projects require complex coordination of the work of many programmers using common and interdependent files. Keeping track of the complex information under the Source Code Control System (SCCS) can be time-consuming and frustrating. The Sun WorkShop TeamWare source management tools provide a solution to this problem.

Extend the Sun WorkShop Compilers C++ and C language systems and Sun Visual WorkShop C++ products with Sun WorkShop TeamWare, a suite of graphical tools for source code version control, release integration, and release management.

Use Sun WorkShop TeamWare code management tools to:

- Graphically track and integrate software projects and releases
- Track versions of ASCII and binary files
- Track complete assemblies of source files
- Dramatically cut the time required to build large projects by executing build jobs in parallel on a single Solaris host

Coordinate Large Software Projects

Simplify the complex task of coordinating common and interdependent files with the Configuring tool. Choose a command-line interface (CLI) or a graphical user interface (GUI). Then copy project files from a central workspace into your own workspace, notify other team members that you're using files, make changes to files, and copy your changes back to the central workspace.

Use the Configuring tool to:

- Speed critical code management tasks, track versions of source files, integrate and build releases, and recreate prior releases using simple graphical tools

- Graphically navigate development workspaces, map development, test and release workspaces in a hierarchical format
- Speed project integration, transfer source files between workspaces by drag and drop
- Work on multiple releases simultaneously, and begin work on a new project without disturbing ongoing work
- Respond to changing project conditions, reorganize the workspace hierarchy at any time

Compare and Merge Files

Use the Merging tool to merge two versions of the same source file, with or without reference to a common earlier version. You can work on the same files and compare respective versions of the files without introducing discrepancies. The Merging tool displays both versions of a file side-by-side, highlighting any differences in color. You can then automatically or selectively merge your changes into a new file, and view them in an editable window.

With the Merging tool, you can also merge entire directories or lists of files. This feature is useful when two versions of a program have diverged significantly and must be made to converge.

Coordinate File Access Easily

Take advantage of the Versioning tool, a graphical user interface to SCCS, to manipulate files and perform SCCS functions. Even if you don't know SCCS commands, this intuitive method lets you check files in and out, as well as display and move through history branches.

Use the Versioning tool to:

- Check in files under SCCS
- Check out and lock a version of the file for editing
- Retrieve copies of any version of the file from SCCS history
- Peruse the branches of an SCCS history file
- Back out changes to a checked-out copy
- Inquire about the availability of a file for editing

- Inquire about differences between selected versions using the Merging tool
- Display the version log summarizing executed commands

Establish Milestones

During the software development process, you might want to create “freezepoints” of your work at various key junctures. These points serve as snapshots of the project that enable you to later recreate the state of the project at key development points. The Freezepointing tool creates a file that registers all source files for a given software release.

Put Freezepointing to work to:

- Create a bill of materials for a workspace
- Recreate prior versions of source files
- Track and warn users of renamed files

Glossary

AnswerBook Online documentation tools that display this manual and most other Sun WorkShop manuals.

browse To inspect source code for all or selected occurrences of a symbol, string constant, or search pattern that you specify.

Configuring The Sun WorkShop TeamWare utility used to create and display workspace hierarchies. The Configuring window is the main window of Sun WorkShop TeamWare.

experiment A set of samples taken on the target application program.

experiment record The pointer file and hidden directory for an experiment.

Freezepointing The Sun WorkShop TeamWare utility used to make snapshots of workspaces (or portions of them) at important junctures or “freezepoints.”

identifier A symbol for which the source browser searches. The identifier can be a variable name, constant name, or user-defined type name.

integration workspace A workspace to which multiple developers put back (merge) their work.

merge To produce a single version of a file from two conflicting files (deltas). Usually accomplished with the assistance of the Merging tool.

Merging The Sun WorkShop TeamWare utility used to merge deltas during Resolve transactions.

performance tuning The steps taken to make a program more efficient, reliable, and fast. These steps estimate the performance of a program, identify the bottlenecks limiting performance, and identify where the code spends most of its time.

query An instruction to the source browser to find a specified symbol.

Sampling Analyzer A tool used to examine performance data collected by the Sampling Collector. The Sampling Analyzer processes the collected data into displays of your choice. You then use these displays to examine the collected performance data.

Sampling Collector A tool used to set up an application for performance data collection. You can collect performance data on an application of your choice.

SCCS history file The file that contains a given file's delta history; also referred to as an "s-dot-file." All SCCS history files must be located in a directory named `SCCS`, which is located in the same directory as the g-file.

Versioning The Sun WorkShop TeamWare utility that provides a graphical interface to SCCS.

workspace A specially designated (but standard) directory and its subdirectory hierarchy. Usually all developers on a project work in their own workspace concurrently with other developers programming in other workspaces. The Sun WorkShop TeamWare Configuring utility lets you "intelligently" copy files from workspace to workspace.

Index