

## ORACLE SOLARIS STUDIO

# VISUALIZE AND DEBUG MULTITHREADED APPLICATIONS WITH `dbxtool`

### KEY FEATURES

- Ensures application stability with event handling and multithread support
- Quickly debugs both Oracle Solaris Studio and GCC-produced binaries
- Provides easy-to-use lightweight visual debugging solution

### BENEFITS

- Improves application stability and quality
- Enhances developer productivity

*Successfully debugging programs is more an art than a science. The Oracle Solaris Studio `dbxtool` combines the power of the time-tested `dbx` debugger with a GUI that reveals trouble spots and simplifies debugging complex applications.*

### Introduction

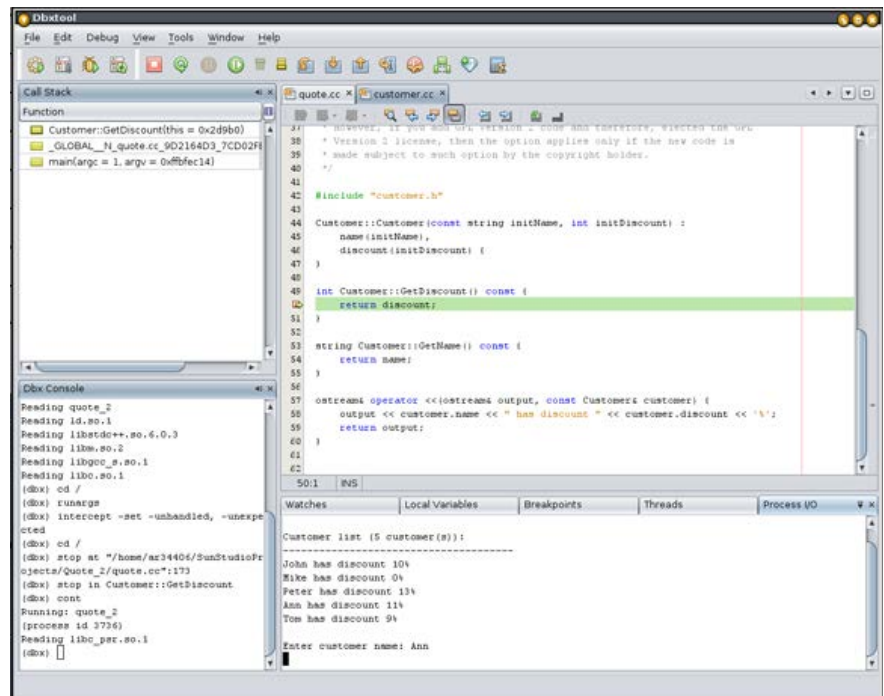
Debugging applications is a time-consuming task that requires a great deal of effort. Certain aspects of application development, such as memory management and parallelization, add complexity and can make debugging much more difficult for developers. Debugging tools that contain a graphical interface can help make the inner workings of an application visible and make it easier for developers to see where defects in a program are occurring.

Available as a standalone interactive graphical tool or as a component of the Oracle Solaris Studio suite of application development tools, Oracle Solaris Studio `dbxtool` simplifies the debugging process. Combining the debugging functionality of the Oracle Solaris `dbx` debugger with an easy-to-use graphical interface, `dbxtool` lets developers quickly and easily debug an executable or core file or attach to a running process.

Using `dbxtool`, developers can view and visit code, control program execution, examine call stacks, evaluate and display data, step through programs, and set breakpoints and traces. `dbxtool` also makes it possible to save trace output, manage events, detect runtime errors, modify source code, and save and re-run a debugging run. With support for Oracle Solaris and Linux operating systems and SPARC and x86 platforms for applications written using Oracle Solaris Studio or `gcc` compilers, developers can create and deploy programs on a wide variety of systems.

### Take Advantage of Powerful Debugging Features

The proven `dbxtool` interactive debugging features support running applications in a controlled fashion. Developers gain complete control of the dynamic execution of a program, including inspecting the state of stopped programs and collecting performance data.



**Figure 1: Oracle Solaris Studio dbxtool adds powerful visualization features to provide proven debugging functionality.**

The debugging features included in dbxtool can help developers to find and fix problems in a number of areas. The process of running a program, correcting a bug, and recompiling each time a change is made can be tedious. dbxtool supports "fix and continue," a feature that updates the running program with source file edits while the program is being debugged.

dbxtool starts up faster and consumes less memory than the full integrated development environment (IDE), and it can debug binaries directly without having to create a project first. Developers can invoke dbxtool from shell scripts and driver programs via the `ss_attach` command, or attach to a process as it starts to begin debugging immediately. dbxtool also offers developers the ability to run multiple debugging sessions simultaneously and supports remote debugging from Oracle Solaris or Linux clients accessing Oracle Solaris or Linux servers.

### Detect Elusive Memory Errors with Runtime Checking

Memory management often is noted as the cause of most program failures. Memory errors tend to be intermittent or difficult to find and fix, and they might not be detected during application testing. Memory leaks can cause increased consumption of virtual memory and memory fragmentation, resulting in poor application and system performance. The runtime checking (RTC) features in dbxtool automatically detect issues such as memory access errors and memory leaks. RTC checks whether an application accesses memory correctly by monitoring read, write, and memory free operations, and it determines whether or not those operations are valid. It also collects memory usage data and can check array bounds.

### Examine Application Code in Depth

Developers need to be able to gain greater insight into applications in order to determine the cause of application behavior. `dbxtool` provides the ability to thoroughly examine stack traces, register dumps, multiple thread execution states, and variables. With `dbxtool`, developers can insert breakpoints in a program. The program can be run up to a breakpoint or stepped line by line—or both actions can be combined—to find a bug. In addition, when a program is stopped, `dbxtool` lets developers navigate to any function or file that is part of the program.

### Leverage Unique Multithreading Debugging Features

Multithreaded applications add a dimension of complexity to debugging and require tools that are designed to handle multiple threads. `dbxtool` supports multithreaded debugging for applications that use either Oracle Solaris or POSIX threads. Developers can examine stack traces for each thread, resume all threads, step or next a specific thread, and navigate between threads.

Being able to rapidly produce reliable code free of costly defects is vital to application developers. With `dbxtool`, Oracle offers powerful debugging capabilities that can help application developers to complete program development more easily, with higher quality, in less time.

## Contact Us

For more information about Oracle Solaris Studio, visit [oracle.com/goto/solarisstudio](http://oracle.com/goto/solarisstudio) or call +1.800.ORACLE1 to speak to an Oracle representative.



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2005, 2010, 2011 Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0611

**Hardware and Software, Engineered to Work Together**