

FEATURES

• TotalView

- Built-in memory debugger
- Advanced thread debugger
- Outstanding data analysis feature
- Scalable, robust reliable
- Advanced C++ debugging features
- Leader in MPI and OpenMP
- Worlds leading FORTRAN debugger

• TaskView

- Processes drawn in tree diagram,
- Several process connection styles
- Tree branches can be hidden, outlined
- Active processes are decorated with "LED" indicators showing status, load, priority
- Leader bars indicate group, session leaders
- Mouse based interaction; no typing required
- Real time tree display updates.
- Processes color coded by User-Group ID, effective User-Group...
- Externally driven color coding
- Data mining metaphor used to "drill down"
- Selectable polling rate
- Hierarchical display
- Separate dialog for color legend

• strace

- Trace system calls

• ltrace

- Trace library calls

• ddd

- Supports multiple languages
- Supports multiple debuggers
- Data plot
- Attach to a running process
- Recompile from GUI
- Easy to use interface
- Same GUI for multiple debuggers
- Updates readily available

InSight Tool Kit for Compro's RTE and RT-Lightning

Overview

Compro's **InSight** tool kit is a complete software suite specifically tailored to augment real time application development, and is compatible with Compro's highly deterministic Real Time Environment (RTE) and RT-Lightning hardware. With InSight, programmers can debug, trace and monitor applications locally or on a remote PC.

InSight is user-friendly and comprehensive, maximizing programmer productivity. Resultant applications are fast, efficient and bug-free.

The Compro InSight tool kit consists of:

- **TotalView**
- **TaskView**
- **strace**
- **ltrace**
- **ddd**

TotalView

The *TotalView* debugger from Etnus, LLC is a powerful programmable tool that debugs, analyzes and tunes complex serial, multiprocessor, and multi-threaded programs.

TotalView uses principles found in other debuggers, thereby shortening its learning-curve. However, its superior GUI, extensive command set, and multi-process/multi-threaded support differentiates it from lesser products.

TotalView offers built-in memory leak detection and heap allocation debugging. These functions are built into the debugger, providing an interactive, minimally-intrusive tool.

Compatible with C, C++, FORTRAN, OpenMP, MPI, and mixed language codes, the debugger detects memory leaks, monitors program memory size, and tracks allocated and de-allocated memory heaps. Significantly more convenient than post-mortem reports, the memory debugger displays status and leaks at any time. Programmers can interactively stop execution, check for problems, test fixes, and continue debugging.

TaskView

Compro's *TaskView* is a powerful GUI-based tool providing simple, intuitive process viewing and management. When launched, *TaskView* displays a graphical tree of foreground and background processes. These processes are arranged by invocation hierarchy (that is, "who spawned who"). The init process is always displayed indicating a common ancestry and tree "root."

Mousing over any process displays a "process tip" that indicates key values. Selecting a process opens a "process details dialog." This dialog contains the most commonly requested process-related values.

Any process detail can be extracted to the main tree by selecting its label. Left-clicking on any process displays a detailed pop-up dialogue. Right-clicking on a process evokes a menu listing possible operations. Actions include process killing, signaling, or opening a man page (if one exists). You may also start a generalized application and pass its PID and process_name, thereby creating a custom action.

The processes displayed in the main tree are color-coded. By default, the initial color coding is determined by *effective user ID*. This immediately shows which processes are running *setuid* (that is, running with another user ID). Other color-coding options are available to suit the programmer's style.

The display is updated in real time. *TaskView* repetitively checks process status and updates tree display and detail information. The update rate is controllable via programmable timers.

A menu bar and button bar are provided for convenient operation. The button bar may be removed by modifying a X resource.

Any part of the tree may be hidden or revealed, permitting focus upon selected processes. Any subtree can be outlined (that is, shown as a reduced color-filled box). The outline function toggles processes from outlined to normal, providing an effective method of selective viewing.

A *user-group* tree dialog may be used to view user-group relationships and isolate users or groups of interest. Each user and group is color-coded based upon X resource file definitions.

strace (System Call Trace)

Compro's *strace* intercepts and records *system calls* (called by a process), and *signals* (received by a process). It runs a specified application until it exits. The *name* of each system call, its *arguments*, and its *return value* are displayed upon standard error. Signals are displayed as a *signal symbol* and a *signal string*. All output may be directed to a specified file.

In legacy environments, application source code is sometimes unavailable. Even in this situation, *strace* is effective because recompilation is not required. From an educational/exploratory perspective, *strace* is useful with any program. Since system calls and signals occur at the user/kernel interface, closely examining this boundary assists bug isolation, "sanity" checking, and race condition detection.

ltrace (Library Call Trace)

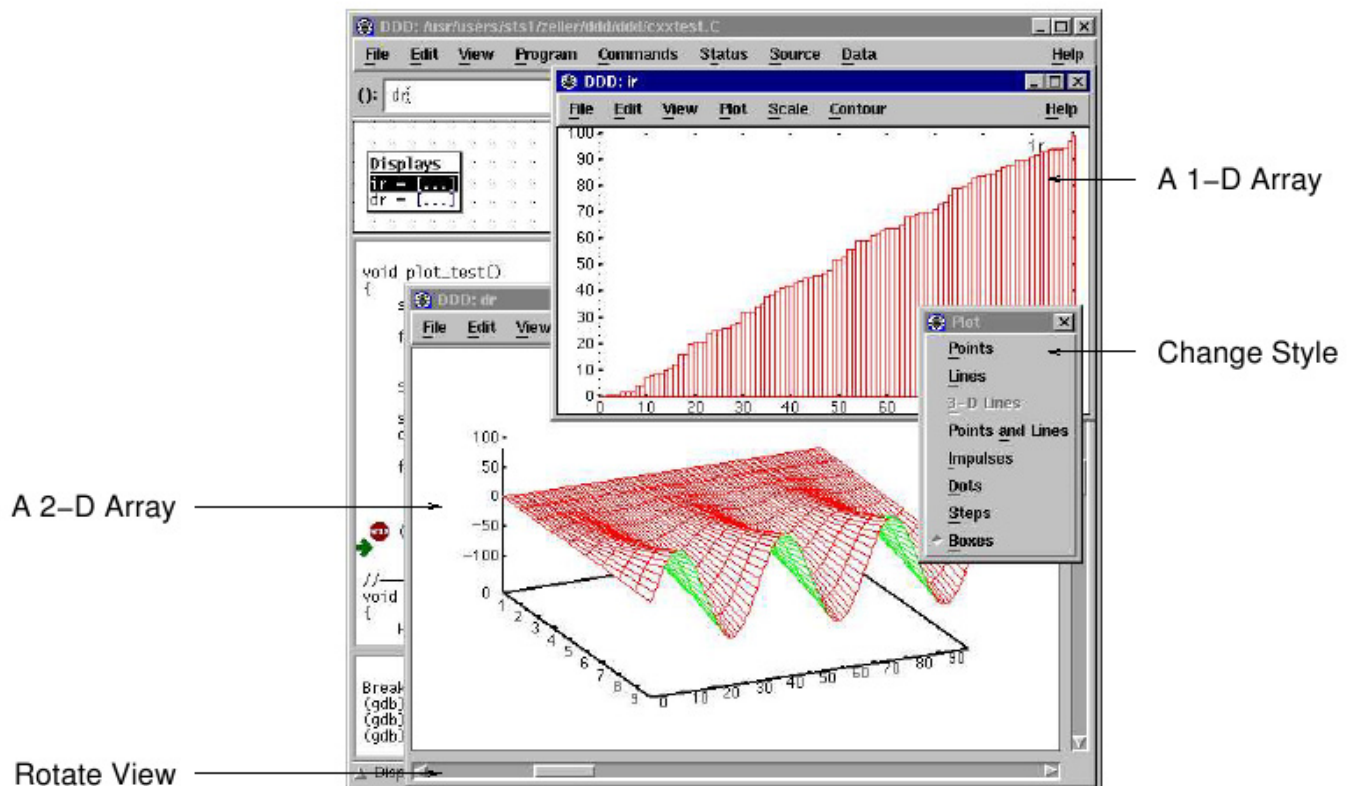
Compro's *ltrace* intercepts and records dynamic library calls (called by a process), and signals (received by a process). It runs a specified application until it exits. *ltrace* can also intercept and display system calls executed by a process.

ddd Debugger

In general, a debugger allows a programmer to see what is going on “inside” a program during execution or at the moment it crashed. Compro’s *ddd* has four primary functions:

- **Start a program specifying anything that might affect its behavior**
- **Stop a program based upon specified conditions**
- **Examine the outcome when a program stops**
- **Experiment with program changes**

ddd is a convenient graphical front-end for popular command-line debuggers. Besides typical features such as viewing source text, *ddd*’s power lies in its interactive graphical display, where data structures are represented as graphs:





Corporate Headquarters

Compro Computer Services, Inc.
105 East Drive
Melbourne, Florida 32904
U.S.A.

Telephone: (800) 936-2673

WWW URL: <http://www.compro.net>

Email: info@compro.net

International

Belgium
Brazil
England
France
Germany
Italy
Japan
Spain

Supported command-line debuggers include:

- **gdb**
- **dbx**
- **wdb**
- **ladebug**
- **jdb**
- **xdb**
- **perl**
- **bash**
- **python**

Compro's *ddd* provides valuable insight into process execution, speeding the application development cycle.

Summary

Compro's InSight tool kit offers programmers a comprehensive, consolidated suite of outstanding software development aids. The tools have been carefully selected to work seamlessly with our Real Time Environment, and emphasis upon GUI interaction accommodates programmers of all skill levels. InSight is the only product needed to completely develop, debug, tune and maintain real time applications.

Compro, the Compro logo, and other branded items are trademarks or registered trademarks of Compro Computer Services, Inc.

Linux is a registered trademark of Linus Torvalds. All other product, service, and company names are trademarks or registered trademarks of their respective owners.

Compro products are subject to a continuing program of enhancement and refinement, and the specifications contained herein are therefore subject to change without notice.

©2007 Compro Computer Services, Inc.
Pub. No. 204-316-03