White Paper

Intel® Debugger 9.1

G. Anthony Saladino
Debugger Consultant
Intel Corporation

Introduction

This white paper provides a high-level overview of Intel® Debugger 9.1 for technical decision makers. It describes the features and benefits, and it examines specific features in more detail.

Overview

The Intel® Debugger is a component of the Intel® compilers. It is a full-featured symbolic source-code application debugger that helps programmers locate run-time programming errors (i.e., bugs) in their code. It provides extensive debugging support for the following languages: C, C++, and Fortran (including later standards). It also provides a choice of control from the command line or a graphical user interface (GUI).

The command-line interface provides two types of command syntax:

- DBX*-like (default)
- GNU Debugger (GDB)-like (optional)

The GUI supports all Intel Debugger command-line functionality. The main window includes the basic debugging and convenience features. Other features are provided in optional windows, various pop-up menus, and dialog boxes. There is also a command-entry prompt.

The Intel Debugger works with the following compilers:

- Intel® C++ Compilers and Intel® Fortran Compilers
- gcc, g++, and g77 compilers
- Microsoft C and C++ compilers

Intel Debugger works on the following desktop and server platforms:

- IA-32 running Linux*, Windows*, and Mac OS*
- Systems using Intel® Extended Memory 64 Technology (Intel® EM64T), running Linux and Windows
- Itanium® 2-based systems running Linux and Windows

See the Intel Debugger Manual for Linux Systems[1] and the Intel Debugger Manual for Windows Systems[2] for additional information. Please also see the release notes for platform-specific restrictions and known limitations of this release.

Rationale

Intel compilers enable effective debugging on the platforms they support.

Intel compilers work with native debuggers, the Intel Debugger, and selected third-party debuggers. Customers are encouraged to use the native debuggers on Intel® processor-based platforms in cases where those debuggers provide satisfactory debugging functionality. In cases where the native debuggers do not provide satisfactory support on Intel processor-based platforms, customers are encouraged to use Intel Debugger.

The value provided by Intel Debugger includes the following:

- Excellent Intel® hardware support (especially for early adopters)
- Robust performance on Intel processor-based platforms (built and thoroughly tested using Intel compilers)
- Superior language-feature support, including C++ templates, user-defined operators, and modern Fortran dialects (with Fortran module support)
- Progressive support for the Intel compiler-specific features such as optimized code debugging and OpenMP*

- Special features, such as support for huge applications, regular-expression search of the symbol table, reflexes that track shared libraries as they are dynamically remapped throughout an address space, and support for applications that are distributed via MPI-1
- Interface options, such as DBX- and GDB-compatible command lines, access from Emacs, DDD, and Streamline DDT*; integration with Eclipse* CDT and a built-in GUI
- Support for native compilers (both gcc and Microsoft), as well as Intel compilers. This support gives customers a high degree of flexibility for mixing and matching development tools to meet the unique demands of their specific development environments
- Command compatibility across operating systems (currently Linux, Windows, and Mac OS)
- A solid roadmap for continuing improvements
- Intel® Premier Support

In addition, customers are referred to selected third-party premium debuggers in cases where they need features that are not yet provided by the Intel Debugger. Those debuggers are:

- The Streamline DDT Debugger[3], which can make use of either native debuggers or Intel Debugger as its underlying debugger engine; it adds both sophisticated Data Display and MPI aggregation technology.
- The Etnus TotalView* Debugger*[4], which brings an extraordinary level of professional and contemporary debugging capabilities to Intel processor-based platforms.

General Features

Intel Debugger provides the basic capabilities most users expect from a modern symbolic source-code application debugger:

- Attaches to (and detaches from) a running process and debugs the corresponding program
- Loads a program into (and unloads a program from) the debugger, automatically creating and deleting corresponding processes as necessary
- Supports multiple-process debugging, where the processes may be associated with the same program or with multiple other programs
  - Actively run one process at a time
  - Switch focus between processes
  - See processes and examine detailed process state
  - Set breakpoints for a specific process
- Supports remote debugging of applications on embedded Intel® architecture (using the GDB remote agent)
- Debugs programs with shared libraries
- Debugs core files
- Provides language-specific command-expression evaluation
- Provides ability to "call" functions in a target process from within a command expression
- Catches/ignores unaligned access
- Displays the source listing of a program
- Sets breakpoints to stop program execution when specified sections of program code are executed

- Sets watchpoints to stop program execution when a specified area of memory or specified program variable is written
- Adds conditions to breakpoints and watchpoints so that program execution will only stop at the specified break or watch event when the condition is true
- Steps into or over calls to routines
- Steps through the execution of a program one source line or one machine instruction at a time
- Examines the stack of currently active functions
- Examines and changes program variables and data structure values in same or in different scopes
- Examines and changes the contents of memory in various formats (including international character strings)
- Disassembles and examines machine code
- Examines and changes machine register values
- Supports mixed-language applications, C++ templates, C++ user-defined operators, and Fortran modules
- Provides a customizable debugging environment by using environment variables, initialization files, sourced scripts, aliases (i.e., parameterized macros), and debugger variables for commands and command sequences
- Supports line editing (simple Emacs binding)
- Can be used by Emacs GUD
- Can be used by DDD
- Can be used by Eclipse CDT

These are solid, basic debugging features that allow you to debug at both the source level and machine level in a single session by using a variety of interface options, and they can be customized according to your preferences.

Advanced Features

Intel Debugger also provides advanced capabilities such as the following:

- Regular expression searches of the symbol table
- Tracks breakpoints and watchpoints in shared libraries across program calls to DLOPEN and DLCLOSE
- Debugs optimized code
  - In-lined instances of functions (show in backtrace and selectable for current focus)
  - Registerized variables
  - Semantic stepping
  - PC-to-source column mapping (for multi-statement lines)
- Supports multi-core architecture
  - Debugs thread-parallel applications that make use of PTHREADS and OpenMP
  - Stops all threads when one is stopped, and restarts all when one is restarted
  - Switches focus between threads
  - Sees all threads or individual threads, and examines detailed thread state
  - Sets breakpoints for all threads or for a subset of all threads
- Supports cluster architecture
  - Debugs cluster-parallel applications that make use of MPI-1
  - Offers built-in cluster aggregation network
  - Includes user defined process sets

These advanced capabilities extend the effectiveness of Intel Debugger well into problem areas that are traditionally difficult to debug.

Graphical User Interface

The GUI provides the following features:

- A complete debugger command set available through the communications view
- Logical and easy access to the most common debugging functions
- Convenient call-stack maneuvering via a call-stack pane
- Convenient display of program execution through the source, communication, and breakpoint displays
- Paned windows that minimize the use of screen area
- Intelligent word selection from the source pane for convenient command composition
- Source navigation through the source viewer
- A push-button panel that contains the most common debugging functions

This is a basic GUI that enables users to quickly make use of the most common capabilities of Intel Debugger, without the need to learn detailed command syntax.

Conclusions

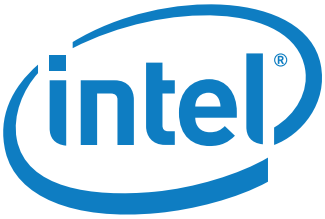
Intel Debugger is a high-quality alternative to the native debuggers running on Intel processor-based platforms. As a component of Intel compilers, its value spans the range of solid, basic debugging capabilities with extensive language support and interface options, advanced features, and a graphical user interface.

Where to find it

Intel Debugger is a component of the Intel compilers. Intel compilers can be purchased, or an evaluation copy can be downloaded, from the Intel® Software Development Products Web pages (http://www.intel.com/software/products/).

Customers have access to product updates and product support through the following Web pages:

- Intel Support and Downloads: http://www.intel.com/support/
- Intel Software Development Products Support: http://www.intel.com/software/products/support/
- Intel Software Development Products Self Help: http://www.intel.com/support/performancetools/index.htm
- Intel® Software Network Discussion Forums: http://softwareforums.intel.com/ids
- Intel Premier Support: http://premier.intel.com/



For product and purchase information visit:
www.intel.com/software/products

0506/DAM/ITF/PP/500 312568-001

---

[1] http://developer.intel.com/software/products/compilers/clin/docs/manuals.htm

[2] http://developer.intel.com/software/products/compilers/cwin/docs/manuals.htm

[3] http://www.streamline-computing.com/

[4] http://www.etnus.com/