Introduction................................................................................................................................. 8
Overview ............................................................................................................................................ 3
Servers and Workstations ................................................................................................................. 4
  Support for Linux*, Mac OS X*, and Windows*............................................................................. 4
  Support for Linux* and Mac OS* Only ......................................................................................... 4
  Linux*-Only Support .................................................................................................................... 5
  Mac OS*-Only Support ............................................................................................................... 5
  Windows*-Only Support .............................................................................................................. 6
Low-Power IA-32 and Consumer Electronics .................................................................................... 6
  Embedded Platforms ...................................................................................................................... 6
  Intel® Atom™ Technology and Consumer Electronics ...................................................................... 6
  Linux*-Only Support .................................................................................................................... 6
  Windows*-Only Support ............................................................................................................... 7
Conclusions ........................................................................................................................................ 8
Where to find them ............................................................................................................................ 8
References .......................................................................................................................................... 10
Introduction

This white paper provides a high-level overview of debugging solutions for Intel® processor-based platforms from Intel as well as other debug solution providers for technical decision makers. It catalogues some of the relevant debuggers and points out some of the features and benefits of each with references to additional information.

Overview

Intel® architecture and its many variants have over the years encouraged and fostered a rich ecosystem of debug solutions for all the arenas that Intel® architecture based platforms can be found. This includes the very small starting with cellular, handheld and Mobile Internet Device as well as location service applications. It also includes the very large like many-core clusters used in high-performance computing and server applications. Finally it also includes everything in between. This rich ecosystem of development tools solutions and debug solutions is part of what led to the huge number of developers comfortable and familiar with Intel® architecture today.

With the advent of more and more “multi”- and “many” core processors requires parallel programming models are being used more as well. Using more advanced parallel programming techniques makes it however also more difficult to keep track of all active threads and their possible data interdependencies. This is where the feature sets and user interfaces of modern debug solutions come in. Advanced parallel programming techniques call for advanced parallel debugging techniques to keep the executions flow and the analysis of runtime problems straight forward and allow for speedy error corrections.

Additionally Intel® architecture finds its way into an increasing number of new usage models. The new generation of low-power IA-32 based processors gain popularity in such diverse places as in-vehicle infotainment, internet- and location service centric devices.

The Intel® Compilers strive to not only provide rich and complete symbol information and work seamlessly together with our Intel® Debugger technology, but also to support the wide ecosystem of debug solutions provided by the wide software ecosystem.

To find out more about the Intel® Debug Solution please have a look at the following articles and whitepapers:

Intel® Parallel Debugger Extension for Microsoft Visual Studio*

Intel® Debugger for Linux*
The Intel® Debugger works with native compilers, the Intel® C++ Compiler, the Intel® Fortran Compiler and selected third party compilers. This range of capabilities provides a great deal of flexibility in mixing and matching development tools to suit various development environments.

In this paper we would like to introduce some of the debug solutions from across the Intel® architecture targeted software ecosystem and 3rd party vendors.

### Servers and Workstations

This section discusses debugging functionality for platforms based on IA-32, Intel® 64 and IA-64 processors, running Linux®, Mac OS X®, or Windows® operating systems.

**Support for Linux®, Mac OS X®, and Windows®**

The Intel® Debug solutions provide quality debugging features across all supported platforms for C/C++ as well as Fortran source level debugging. Specific details can be found in the [Intel® Debugger white paper](http://software.intel.com/en-us/articles/idb-linux/) and the [Intel® Parallel Debugger Extension whitepaper](http://software.intel.com/en-us/articles/parallel-debugger-extension/).

Currently for Max OS X® only command line debug support via the Intel® Debugger is available.

**Support for Linux® and Mac OS® Only**

**GDB**

([http://www.gnu.org/software/gdb/gdb.html](http://www.gnu.org/software/gdb/gdb.html)), the GNU Project Debugger, allows you to see what is going on `inside` another program while it executes—or what another program was doing at the moment it crashed. GDB can perform four primary functions (plus other functions in support of these) to help you catch bugs in the act:

- Start your program, specifying anything that might affect its behavior.
- Make your program stop on specified conditions.
- Examine what has happened, when your program has stopped.
- Change things in your program, so you can experiment with correcting the effects of one bug and go on to learn about another. The program being debugged can be written in C, C++, Pascal, Objective-C, or many other languages. Those programs might be executing on the same machine as GDB (native) or on another machine (remote). GDB can run on most popular UNIX and Microsoft Windows variants.
The **TotalView* debugger**

(http://www.totalviewtech.com/) is a powerful, sophisticated, and programmable tool that simplifies the debugging and analysis of serial, multi-threaded, or multi-process applications.

The **MemoryScape* memory debugger**

also at (http://www.totalviewtech.com/) is an easy to use tool for manual analysis of memory usage and automated detection of memory errors such as leaks and bounds errors in serial, multi-threaded or multi-process applications.

**Linux*-Only Support**

Allinea DDT*

(Distributed Debugging Tool) (http://www.allinea.com/index.php?page=48) is an advanced and comprehensive graphical debugger for scalar, multi-threaded and large-scale parallel applications that are written in C, C++, or Fortran.

The **ReplayEngine* add on to the TotalView* debugger**

discussed above, (http://www.totalviewtech.com/) simplifies troubleshooting and root cause analysis in serial, multi-threaded and multi-process applications. It records program execution, capturing hard to reproduce error scenarios and allowing the developer to follow clues backwards through execution history from the failure to the cause.

**Mac OS*-Only Support**

Xcode* 3.0


is a fast way for developers to create Mac OS X* applications and an easy way to take advantage of new Apple technologies. Xcode* 3.0 brings together the ease of use of Mac OS X*, the power of UNIX, and a mixture of high-performance development technologies. Xcode* works with numerous other tools available with Mac OS X*, such as the integrated user interface construction application, compilers such as GCC, javac, and Jikes*, and debuggers such as GDB*.

It’s a challenge to debug full-screen applications such as games. Where do you put the debug window? With Xcode* 3.0’s graphical remote debugging, all of Xcode*’s powerful debugging features, such as data formatters, Fix, and Continue, are available for debugging applications on remote machines. Xcode* even helps keep your data secure by using an encrypted connection to the remote debugging session.
Windows*-Only Support


Low-Power IA-32 and Consumer Electronics

Debugging applications running on embedded platforms and platforms based on Intel® Atom™ technology and on the new generation of Intel® Media Processors are included in this section.

Embedded Platforms


The QNX Momentics* Tool Suite (http://www.qnx.com/products/tools/) provides a comprehensive toolset to build and optimize applications.

The Wind River VxWorks Workbench* IDE (http://www.windriver.com/products/workbench/) provides an open, unifying framework for use by project teams.

The MontaVista Linux DevRocket* IDE (http://www.mvista.com/devrocket/) provides the tools and functionality you need to develop and deploy system software and applications.

Support for debugging on Windows XP* Embedded systems is provided by the Microsoft Embedded Studio* (http://www.microsoft.com/windows/embedded/xpe/deven.mspx).

Intel® Atom™ Technology and Consumer Electronics

Linux*-Only Support

Intel® Debugger - For platforms based on the Intel® Atom™ processor and the Intel® Media Processors you can rely on Intel’s product offering with the Intel® C++ Software Development Tool Suite for Linux* OS Supporting Mobile Internet Devices. (http://www.intel.com/software/products/mid)
This is a complete tool suite containing an **Intel® C++ Application Debugger** targeting TCP/IP based application cross development between a Linux* host and an Intel® Atom™ processor based small form-factor target device. More details can be found at

(http://software.intel.com/en-us/articles/cross-application-debugging)

It also includes an **Intel® C++ JTAG Debugger** targeting device manufacturers (OEMs and ODMs) to assist them with OS kernel customization and driver development. This debugger is JTAG standard based and thus only tends to connect to early reference platforms. It does however allow OS level debugging on even very early hardware with a very incomplete software stack.

It even permits reflashing of system BIOS and will in upcoming releases support NAND flashing of bootloader and OS image.

More information can be found at


Both debuggers included in the discussed tools suite come with the new Eclipse Rich Client Platform based rich, convenient, and user friendly graphical user interface.

For more traditional development models targeting not Intel® Media Processor based platforms or Mobile Internet Devices, but rather Netbooks or similar slightly larger form factors it is recommended to use the standard Intel® Debugger as included with the Intel® Fortran Compiler or Intel® C++ Compiler.

(http://software.intel.com/en-us/articles/idb-linux)

**GDB***

(http://www.gnu.org/software/gdb/gdb.html), the GNU Project Debugger, is a widely available debugger that is part of most any Linux* OS and GNU build environment installation. It is the de facto standard for Linux* debugging and its features have been outlined above when discussing Linux* application debugging on workstation and personal computer platforms.

What is additionally interesting about GDB* in the Mobile Internet Device context is it’s support for TCP/IP based remote debug concepts using an application called gdbserver.

There are also a multitude of open-source solutions looking to provide basic graphical user interfaces (CDT*, DDD*) for GDB* as well as system debug adaptations from various vendors.

**Windows*-Only Support**

Since Windows* XP and Windows* 7 based Intel® Atom™ processor designs will in general be Netbooks or Ultra-Mobile PCs we are looking at a development methodology similar to regular notebook or laptop targeted applications. If you are using concurrent programming techniques and parallelism Intel’s **Intel® Parallel Debugger Extension for Microsoft Visual Studio***

(http://software.intel.com/en-us/articles/parallel-debugger-extension/)
Additionally Microsoft* offers the following debug solutions:

The **Windbg***
(http://www.microsoft.com/whdc/devtools/debugging/default.mspx)

and **Microsoft Visual Studio***
(http://msdn.microsoft.com/vstudio/)


**Conclusions**

Intel enables effective debugging on its platforms. There are a variety of quality debuggers to choose from that work well with most compilers. This variety provides a great deal of flexibility in mixing and matching development tools to suit any given development environment.

**Where to find them**

References throughout this white paper give sources to obtain additional information about the featured debuggers, as well as to obtain the debuggers themselves.

Native debuggers are typically provided with the host operating system (or as a component of the compilers or other development tools intended for use on those systems).

The Intel® Debugger and Intel® Parallel Debugger extension are value-added components of the Intel® Compilers. The 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:
White Paper  Debugging on Intel® Platforms

- Intel Software Network Articles
  http://software.intel.com/en-us/articles/
- Intel® Premier Support:
  http://premier.intel.com/
References

1. Intel® Debugger white papers:
Intel® Parallel Debugger Extension for Microsoft Visual Studio*
Intel® Debugger for Linux*
2. GNU Project Debugger*:
http://www.gnu.org/software/gdb/gdb.html
4. Allinea DDT* (Distributed Debugging Tools):
http://www.allinea.com
debugging/default.mspx
windows/embedded/xpe/deven.mspx