

Intel® Inspector XE 2011 Release Notes for Linux* OS

Installation Guide and Release Notes

Document number: 323609-001US

23 December 2010

Contents:

[Introduction](#)

[What's New](#)

[System Requirements](#)

[Installation Notes](#)

[Issues and Limitations](#)

[Attributions](#)

[Disclaimer and Legal Information](#)

1 Introduction

The Intel® Inspector XE 2011 helps developers identify and resolve memory and threading correctness issues in their C, C++ and Fortran programs.

The Intel Inspector XE is a static and dynamic error checking tool for developing multithreaded applications on Windows* or Linux* operating systems. The Intel Inspector XE maximizes code quality and reliability by quickly detecting memory, threading, and source code security errors during the development cycle. When combined with the Intel® Composer XE 2011 in the Intel® Parallel Studio XE 2011, you can perform Static Security Analysis with the Intel Composer XE 2011 and then view and manage the results in the Intel Inspector XE. The Intel Inspector XE is an easy, comprehensive solution that delivers rapid results in isolating memory and multithreading errors.

The Intel Inspector XE has a standalone graphical user interface (GUI) as well as a command-line interface (CLI).

This document provides system requirements, installation instructions, issues and limitations, and legal information.

Use the Getting Started tutorial and reference documentation to learn more about the Intel Inspector XE. For documentation, open the `documentation_inspector_xe.htm` file in the following directory: `<install-dir>\documentation\<locale>\`. For example, if you

choose the default installation path, you can find the `documentation_inspector_xe.htm` file in the `/opt/intel/inspector_xe_2011/documentation/en` directory.

If you did not register this product during installation, please do so at the Intel® Software Development Products Registration Center (<http://irc.intel.com>). Registration entitles you to free technical support, product updates and upgrades for the duration of the support term.

For Technical Support, Product Updates, User Forums, FAQs, tips and tricks, and other support information, please visit <http://www.intel.com/software/products/support>. **Note:** If your distributor provides technical support for this product, please contact them for support rather than Intel.

2 What's New

Intel® Inspector XE 2011 Update 1:

- Bug fixes

Intel® Inspector XE 2011:

This is the first release of the Intel Inspector XE product which is a successor to the Intel® Thread Checker. The following are some key new features in this next generation product:

- Find memory errors in addition to threading errors
- Find memory and threading errors in source code using the Static Security Analysis feature of Intel® Compiler XE and then view, filter and process the results in the Intel Inspector XE when both the products are installed as part of Intel® Parallel Studio XE
- New dynamic instrumentation implementation to accelerate analysis (only executed code is instrumented)
- New standalone GUI on all supported Operating Systems
- New grouping of related diagnostics, diagnostic suppression and a powerful results filtering interface

3 System Requirements

Supported Architectures and Terminology

The Intel Inspector XE supports the following architectures:

- **IA-32 Architecture** refers to systems based on 32-bit processors generally compatible with the Intel® Pentium® processors (such as, Intel® Pentium® 4 processor or Intel® Xeon® processor), or processors from other manufacturers supporting the same instruction set, running a 32-bit operating system.

- **Intel® 64 Architecture** refers to systems based on IA-32 architecture processors that have 64-bit architectural extensions (such as, Intel® Core™2 processor family), running a 64-bit operating system. If the system is running a 32-bit operating system, then IA-32 architecture applies instead. Processors from other manufacturers supporting the same instruction set and running a 64-bit operating system are also supported.

Minimum System Requirements

- A system based on an IA-32 or Intel® 64 architecture processor supporting the Intel® Streaming SIMD Extensions 2 (Intel® SSE2) instructions (Intel® Pentium® 4 processor or later, or compatible non-Intel processor)
 - The Intel Inspector XE requires specific knowledge of assembly-level instructions. Its analysis may not operate correctly if a target executable contains instructions not supported by IA-32 or Intel® 64 architectures. In this case, run the analysis with a target executable that contains only supported instructions. After you finish using the Intel Inspector XE, you can revert to using unsupported instructions in the target executable.
 - For the best experience, a multi-core or multi-processor system is recommended.
- 2GB RAM
- 4GB free disk space for all tool features and architectures
- Supported operating systems:
 - Red Hat* Enterprise Linux* 4 Update 7, 8
 - Red Hat* Enterprise Linux* 5 Update 4 and 5
 - Red Hat* Enterprise Linux* 6
 - Red Hat* Fedora* 12, 13
 - SUSE* Linux Enterprise Server* 10 Service Pack 2 and 3
 - SUSE* Linux Enterprise Server* 11
 - CentOS* 5.5
 - Asianux* 3.0
 - Debian* 4, 5
- Supported compilers:
 - Intel® C/C++ Compiler 11 and higher
 - Intel® Fortran Compiler 11 and higher
 - GNU* C/C++ Compiler 3.4.6 and higher
- Application coding requirements:
 - Supported programming languages:
 - Fortran
 - C
 - C++
 - Supported threading methodologies:
 - Intel® Threading Building Blocks (Intel® TBB)
 - POSIX* Threads on Linux* OS
 - OpenMP*[1]
 - Intel C/C++ parallel language extensions

- Adobe* Reader* 7.0 or later needed to read installed documentation
- GNU* Debugger versions 6.3 or later with working remote debugging support (there are known issues with gdb-7.1-xx.fc13 versions)

Note:

The Intel Inspector XE supports analysis of applications built with the Intel® Fortran Compiler Professional Edition version 11 or higher, the Intel® C++ Compiler Professional Edition version 11.0 or higher, and the GNU* C/C++ Compiler 3.4.6. Applications that use OpenMP* technology and are built with the GNU* compiler must link to the OpenMP* compatibility library as supplied by an Intel Compiler.

4 Installation Notes

This product package can be used to install the software on both IA-32 systems and Intel® 64 systems. The installer determines the system architecture and installs the appropriate files. Both 32-bit and 64-bit versions of the software are automatically installed on an Intel® 64 system.

The installation of the Intel Inspector XE removes any earlier installed minor version of this product (with the same major version number). Different major versions can co-exist with each other.

If you are installing the product for the first time, you will need to have the product serial number or a valid license file to activate the product.

To install, perform the following steps:

1. Uncompress the package: `gunzip <package-name>.tar.gz`
2. Extract the files: `tar xf <package-name>.tar`
3. Start the installation (Note: For successful installation you should have read and write permissions for the /tmp directory)

- To install on a local system enter the following:

```
cd <package-name>
```

- `./install.sh`

Note: If you want to install the software for use by any user, you must do this as the root user. To install to a network-mounted drive or shared file system for multiple users, become the root user then:

```
cd <package-name>
```

```
./install.sh --SHARED_INSTALL
```

4. Follow the prompts to complete the installation and activation of the software

Default Installation Folders

The default top-level installation folder for this product is:

```
/opt/intel/inspector_xe_2011
```

A different top-level folder name can be chosen during a custom installation.

Installing Collectors On Remote Systems

You can install just the data collection support features of the product on remote systems where run-time license checking is not feasible. The results of any data collection that is run on the remote system must then be copied to the system where the regular install was done for analysis, viewing, and reporting.

To do this:

1. Copy the `CLI_install` folder (found at the top level in the untarred product install package) to the remote machine.
2. Execute the `./install.sh` script file (this file is located inside the `CLI_install` folder). Activation is not required.

How To Activate Your Evaluation Software After Purchasing

Users of evaluation versions of Intel Developer Products have a new tool that allows converting evaluation-licensed products to fully licensed products once the product is purchased and a serial number is obtained. The “Activation Tool” is a utility that allows users of evaluation products to enter a valid product Serial Number to convert the product to fully licensed status.

Run the `/opt/intel/ActivationTool/Activate` script and provide your purchased product serial number, either as an argument to the program or when prompted. For example:

```
/opt/intel/ActivationTool/Activate ABCD-123AB45C
```

Be sure to login or “su” to root if you want the product license to be available to all system users.

Removing The Product

To remove the product, execute the following commands:

Note: Replace `/opt/intel` with the chosen install folder name if the default folder was not used.

1. `cd /opt/intel/inspector_xe_2011`
2. `uninstall.sh` (as a root user or the same user who performed the install)

5 Issues and Limitations

General Issues

- Intel does not guarantee this software tool will detect or report every memory and threading error in an application.
 - Not all logic errors are detectable.
 - Heuristics used to eliminate false positives may hide real issues.
 - Highly correlated events are grouped into a single problem.
- You can use the Intel Inspector XE to analyze applications in debug and release modes. To learn more about options necessary to produce the most accurate, complete results, refer to the following related resources:
 - Memory error analysis: <http://software.intel.com/en-us/articles/compiler-settings-for-memory-error-analysis-in-intel-inspector-xe>
 - Threading error analysis: <http://software.intel.com/en-us/articles/compiler-settings-for-threading-error-analysis-in-intel-inspector-xe>
- If no symbols are found for a module in which a problem is detected, the Intel Inspector XE displays the call stack and observation source code of the first location where it can find symbols. If it cannot find any location in the call stack with symbols, it displays the module name and relative virtual address (RVA) for the location.
- Applications that crash when run outside the Intel Inspector XE may crash or hang the Intel Inspector XE runtime analysis engine. For example, a corrupt return address on an application call stack crashes the runtime analysis engine. If a crash occurs, problems detected prior to that time can be viewed, but memory leaks will not be reported.
Recommendation: Review these problems – it is likely one of them caused the crash.
- The Intel Inspector XE uses a socket to communicate between the graphical user interface and the runtime analysis engine. Preventing an application from opening a socket prevents an analysis of the application from being started by the graphical user interface. The command line user interface can be used to run an analysis in this case and the results can subsequently be viewed using the graphical interface.
- The Intel Inspector XE may report an incorrect call stack following an interruption of normal call flow, such as when an exception is thrown and caught. While the Intel Inspector XE recognizes and attempts to correct result data when this situation occurs, it is possible for a threading or memory problem to be reported before the call stack is fully corrected.
- If the Intel Inspector XE reports “insufficient memory” errors while analyzing OpenMP* applications, try setting OMP_NUM_THREADS to limit the number of OpenMP* threads. In most cases, OMP_NUM_THREADS=2 is sufficient.
- If the Intel Inspector XE reports “insufficient memory” errors during analysis, try analyzing your application multiple times and exclude a different set of modules during each analysis.
Note: The Intel Inspector XE does not detect or report issues in excluded modules.
- 200142169: The product may miss threads created when you parallelize your Linux* OS program with the OpenMP* threading methodology. This can manifest in timeline view where only main thread is present (no OMP threads are present).
- 200151347: The product cannot load results if special symbols (such as &) are used in user supplied data (thread names, for example).
- 200159955: On Ubuntu* 10.04 systems with Ambience and Radiance themes, the error message “CRITICAL **: murrine_style_draw_box: assertion `width >= -1` failed” is

printed in console for `inspxe-gui`.

Recommendation: Please switch to another theme to avoid the issue. See <https://bugs.launchpad.net/ubuntu/+source/light-themes/+bug/538499> for more information on this issue.

- 200159955: On Ubuntu* 10.10 systems, Standalone GUI silently disappears when opening the Intel Inspector XE results.
Recommendation: Switch visual theme to "New wave" or switch to another window manager (for example, KDE* manager).
- When configuring an analysis, there is an option to break into debug mode when a memory or threading error is detected. This option is still in rudimentary form in this release, with the following limitations:
 - Deadlocks are detected and reported, but will not stop execution if the `Break application for debug (command line knob appdebug)` option was selected when running a threading error analysis. Only data race errors will result in a break in the target application execution for threading analysis. All memory errors detected during a memory error analysis will result in a break in execution. (Note that memory leaks are determined after the target application execution has completed and therefore will not cause a break in execution.)
 - A GNU* debugger window (GDB) must be started manually and connected to the running analysis using the given target remote command.
 - When `Break into debugger when application starts execution` is selected (`appdebug=on-start`), GDB connection information is written to the terminal which launched `inspxe-gui`, and not to any window within the Intel Inspector XE GUI itself. When `Break into debugger when application error detected` is selected (`appdebug=on-error`), the GDB connection information is written to the "Collector Messages" window in the Intel Inspector XE GUI.
 - Memory error and threading error breakpoints can only be turned on/off by error types. Enter the command `monitor help` at the GDB prompt for a list of command options. To execute any of the commands listed by that help you must preface it with "monitor".
 - To avoid stopping at memory or threading errors that are not of interest, consider setting a source code breakpoint near the error(s) of interest and turn off all the analysis error breakpoints (i.e., `monitor nobrkAll`), then continue execution. After hitting the code breakpoint, turn on all the analysis error breakpoints again (i.e., `monitor brkAll`), or only the error type of interest (for example, `monitor brkIR` will break execution only when an invalid read is detected.)

Memory Error Analysis Issues

- When doing Memory Error Analysis on applications that use fibers or user-level threads, the Intel Inspector XE may not work properly and/or results may be incorrect in some cases. For such an application, if the "analyze stack accesses" feature is turned on, the application will in general not work properly and/or data collection will fail. If the "analyze stack accesses" feature is not turned on, then in some cases, incorrect call stacks may be reported. Intel® Cilk™ Plus uses fibers or user-level threads, and as such, this caveat applies to any software that uses Intel® Cilk™ Plus.

- The Intel Inspector XE may report false positives when the analyzed application uses custom memory allocators. Use of `_itt_notify` to annotate your source code can reduce these false positives.
- The behavior of the Intel Inspector XE is unknown and could lead to abnormal analysis termination if the semantics of standard C runtime allocators is changed. For instance, if the application is using non-standard versions of these allocators where the memory returned by the allocator is initialized when it would normally be uninitialized.
- 200154305: First (narrow) level of Memory Error analysis may be slow for binaries that do not contain binary search table in the `.eh_frame_hdr` section.
- 200154002: First (narrow) level of Memory Error analysis captures child processes created using `libc` `exec` family of functions only.

Threading Error Analysis Issues

- The Intel Inspector XE does not capture the main thread creation site if the binary is built without debug symbol information.
- The Intel Inspector XE does not detect deadlocks or potential deadlocks created with:
 - Some types of locks via Intel® C/C++ parallel extension (`__critical`) provided by the Intel® Compiler Professional Edition 11.0 or higher.
 - Some types of locks in Intel® TBB (`spin_mutex`, `spin_rw_mutex`)
 - Non-exclusive ownership synchronization objects involved, for example, condition variables, semaphores and events and reader/writer locks.
- The Intel Inspector XE may not detect threading issues on data accessed in the C runtime library (like `memmove` and `memcpy`).
- The Intel Inspector XE does not detect inter-process data races or deadlock/potential deadlocks.
- The Intel Inspector XE may report false positives for analyzed applications using customized synchronization primitives. Use of `_itt_notify` to annotate your source code can reduce these false positives.
- The Intel Inspector XE is not a replacement for a traditional debugger, such as the Microsoft Visual Studio* debugger on Windows* operating systems or GNU* GDB debugger on Linux* operating systems. If an application crashes inside/outside the Intel Inspector XE, try running the application inside the debugger to reproduce and fix the crash.
- The Intel Inspector XE may report false positives if you have `libc` or `libpthread` statically linked in the application.
Recommendation: Build your application with `libc` and `libpthread` dynamically linked in.
- To enable correct analysis of Intel® TBB based applications, set the following required macros before compiling::
 - **TBB_USE_DEBUG** (which sets **TBB_USE_THREADING_TOOLS**) if you use Intel® TBB debug libraries
 - **TBB_USE_THREADING_TOOLS** if you use Intel® TBB release libraries
- The Intel Inspector XE analyzes only the first instance of a forked child process.

6 Attributions

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2009 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

wxWindows Library

This tool includes wxWindows software which can be downloaded from <http://www.wxwidgets.org/downloads>.
wxWindows Library Licence, Version 3.1

=====
Copyright (C) 1998-2005 Julian Smart, Robert Roebing et al
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

WXWINDOWS LIBRARY LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public Licence for more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.
2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.
3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.
4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly

Boost Software License – Version 1.0 – August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Libxml2

Except where otherwise noted in the source code (e.g. the files hash.c,list.c and the trio files, which are covered by a similar license but with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

7 Disclaimer and Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL(R) TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE

INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm> for details.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Cilk, Core Inside, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, skool, the skool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

* Other names and brands may be claimed as the property of others.

Microsoft, Windows, Visual Studio, Visual C++, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Portions Copyright (C) 2001, Hewlett-Packard Development Company, L.P.

Copyright © 2010, Intel Corporation. All rights reserved.