

# Intel® Parallel Inspector 2011

## Release Notes

---

Installation Guide and Release Notes  
Document number: 320754-002US

16 March 2011

### Contents

[Introduction](#)

[What's New](#)

[System Requirements](#)

[Installation Notes](#)

[Issues and Limitations](#)

[Attributions](#)

[Disclaimer and Legal Information](#)

## 1 Introduction

Intel® Parallel Inspector 2011 is a serial and multithreading error checking analysis tool for Microsoft Visual Studio\* C/C++ developers. Inspector detects memory leaks and errors as well as threading data races and deadlock errors. This comprehensive developer productivity tool pinpoints errors and provides guidance to help ensure application reliability and quality.

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

To learn more about this product, see the **Inspector Documentation** at:

- **Start > All Programs > Intel Parallel Studio 2011 > Parallel Studio Documentation > Inspector Documentation.**
- Or `<install-dir>\documentation\<locale>\documentation_inspector.htm`. For example, if you install the product in the default installation path, you can find the documentation at: `C:\Program Files\Intel\Parallel Studio 2011\Inspector\documentation\en\documentation_inspector.htm`

For Technical support, including answers to questions not addressed in the installed tool, visit the technical support forum at: <http://software.intel.com/sites/support/>

Please remember to register your tool at <https://registrationcenter.intel.com/> by providing your email address. This helps Intel recognize you as a valued customer in the support forum.

## 2 What's New

Intel® Parallel Inspector 2011 Update 2:

- Improved GUI:
  - Simpler, more intuitive real-time analysis views, main result data view, and import view
  - Enhanced state management and problem filtering
  - New memory overhead gauge to help choose the optimal preset analysis configuration
- Updates for Operating System and IDE support
  - Added Microsoft Windows 7\* SP1
  - Added Microsoft Visual Studio\* 2010 SP1
- Added stability improvements

Intel® Parallel Inspector 2011 Update 1:

- Improved analysis configuration (The Collection dialog now contains three levels of analysis. Level of analysis formerly known as mi4/ti4 is now available as an additional option when you select mi3 or ti3 levels of analysis, respectively)
- New *ManagingSuppressions* tutorial
- Bug fixes

Intel® Parallel Inspector 2011:

- Microsoft Visual Studio\* 2010 support
- Resource leak detection
- Intel® Cilk™ Plus support
- Activation tool

See <http://software.intel.com/en-us/intel-parallel-inspector/> or the What's New section in the help.

## 3 System Requirements

For an explanation of architecture names, see <http://software.intel.com/en-us/articles/intel-architecture-platform-terminology/>

- A system with 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)

- Incompatible or proprietary instructions in non-Intel processors may cause the analysis capabilities of this tool to function incorrectly. Any attempt to analyze code not supported by Intel® processors may lead to failures in this tool.
- 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
- Software requirements
  - Operating system: Microsoft Windows 7\* SP1, Microsoft Windows XP\* SP3, Microsoft Windows Vista\* SP2, Microsoft Windows Server\* 2003, Microsoft Windows Server\* 2003 R2, Microsoft Windows Server\* 2008 SP2, 32-bit or x64 editions – embedded editions not supported
  - Microsoft Visual Studio\* 2005 SP1, 2008 SP1 or 2010 SP1 software with C++ component installed [0] – Microsoft Visual Studio\* Express Edition not supported
- Application coding requirements
  - Programming Language: C or C++ (native, not managed code)
  - Threading methodologies supported by the analysis tool:
    - Intel® Threading Building Blocks (Intel® TBB)
    - Win32\* Threads on Windows\*
    - OpenMP\* [1]
    - Intel's C/C++ Parallel Language Extensions
    - Intel® Cilk™ Plus
- Adobe\* Reader\* 7.0 or later to read installed documentation

Notes:

[0] Inspector supports analysis of applications built with the Intel® Parallel Composer, Intel® C++ Compiler Professional Edition version 10.0 or higher, and/or Microsoft Visual C++\* 2005 SP1, 2008 SP1 or 2010 SP1 software.

[1] Applications that use OpenMP\* technology and are built with the Microsoft\* compiler must link to the OpenMP\* compatibility library as supplied by an Intel® compiler.

## 4 Installation Notes

If you are installing the Inspector for the first time, please be sure to have the product serial number available so you can type it in during installation.

Inspector updates uninstall your currently installed Inspector version, and use the existing valid Inspector license on the system.

## Default Installation Folders

The default top-level installation folder for the Inspector is:

```
C:\Program Files\Intel\Parallel Studio 2011\Inspector
```

If you are installing on a system with a non-English language version of the Windows\* operating system, the name of the Program Files folder may be different. On Intel® 64 architecture systems, the folder name is `Program Files (x86)` or the equivalent.

## Changing, Updating and Removing the Tool

To remove, modify, or repair the Inspector:

1. Open the Control Panel.
2. Select the **Add or Remove Programs** applet.
3. Select **Intel Parallel Inspector 2011**.
4. Click the **Change** button.

## Converting Evaluation-licensed Products to Fully Licensed Products

To convert your evaluation software to a fully licensed product:

1. From the start menu, click **Start > All Programs > Intel Parallel Studio 2011 > Product Activation**
2. Supply a valid product serial number
3. Click **Activate**

## Inspector Documentation

Inspector documentation is automatically integrated into supported versions of Microsoft Visual Studio\*. If documentation integration does not work or disappears, follow these steps to restore documentation integration:

1. Click **Start > All Programs > Intel Parallel Studio 2011 > Command Prompt** and choose any shortcut (such as **IA-32 Visual Studio 2005 mode**).
2. Remove integration:
  - “`insp-vsreg -d 2005`” to remove the Inspector integration with VS2005
  - “`insp-vsreg -d 2008`” to remove the Inspector integration with VS2008
  - “`insp-vsreg -d 2010`” to remove the Inspector integration with VS2010
3. Restore integration:
  - “`insp-vsreg -i 2005`” to restore the Inspector integration with VS2005
  - “`insp-vsreg -i 2008`” to restore the Inspector integration with VS2008
  - “`insp-vsreg -i 2010`” to restore the Inspector integration with VS2010

If you still cannot access integrated Inspector documentation from the Microsoft Visual Studio\* Help menu, try accessing Inspector documentation from the Start menu (**Start > Intel Parallel Studio 2011 > Parallel Studio Documentation > Inspector Documentation**) or directly from the Inspector Documentation Index at `<install-dir>\documentation\<locale>\documentation_inspector.htm`.

Also, the Inspector Help may be unavailable in Microsoft Visual Studio\* software if the language for non-Unicode programs does not match the operating system language: for example, the Japanese Windows\* operating system with English language set for non-Unicode programs. Workaround: Configure the language for non-Unicode programs to match the operating system language (go to **Control Panel > Regional and Language Options > tab: Advanced**).

## 5 Issues and Limitations

### Installation

- Inspector may not install correctly if an installation of other software is in progress.
- If you have both Microsoft Visual Studio\* 2005 and 2008 integrated development environments (IDEs) installed on your system and integrate the Intel® Parallel Studio 2011 into both IDEs, removing the integration from one IDE can remove the integrated Intel® Parallel Studio documentation from both IDEs. To work around this problem, follow the instructions provided in *Installation Notes/Inspector Documentation* subsection. Follow only the steps for VS2005 and VS2008.

### General Issues

- Inspector 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 will be grouped into a single problem.
- You can use the Inspector to analyze applications in debug and release modes. To learn more about options necessary to produce the most accurate, complete results, please refer to the following two resources:
  - Memory error analysis: <http://software.intel.com/en-us/articles/compiler-settings-for-memory-error-analysis-in-intel-parallel-inspector/>
  - Threading error analysis: <http://software.intel.com/en-us/articles/compiler-settings-for-threading-error-analysis-in-intel-parrallel-inspector/>

- If no symbols are found for a module in which a problem is detected, the Inspector 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.
- Inspector analyzes only one process in an application: the initial process created by the execution of the targeted application. This means an application launched by a script results in analysis of the script, not the process the script starts.
- Applications that crash when run outside the Inspector may crash or hang the Inspector 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 are not reported.
- Inspector uses a socket to communicate between the graphical user interface and the runtime analysis engine. Preventing an application from opening a socket prevents the Inspector from analyzing the application.
- Inspector may report an incorrect call stack following an interruption of normal call flow, such as when an exception is thrown and caught. While the Inspector 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.
- You cannot obtain meaningful results if the application under analysis launches a debugger.
- Synchronization, function calls and memory loads/stores that occur before the Inspector takes control of the program are not visible to the Inspector. Missing these events may cause the tool to report false positives. This situation can occur if these constructs occur in `DllMain`.
- When using the Help Viewer in Visual Studio 2010 SP1, if the user clicks the *Where am I in the Workflow?* icon in the upper-right of some Inspector help topics, to resume reading the original topic:
  - Click the original tab (where the user clicked the *Where am I in the Workflow?* icon).
  - Click its Back button.

## Threading Error Analysis

- Inspector may report false positives and false negatives when analyzing applications that call Microsoft Windows\* `ThreadpoolWait`, `ThreadpoolTimer`, and `ThreadpoolIo` APIs (first introduced in the Microsoft Windows Vista\* operating system) or `User-`

Mode scheduling (UMS) APIs (first introduced in the Microsoft Windows 7\* operating system).

- If you use Intel® Threading Building Blocks (Intel® TBB), set the macro `TBB_USE_THREADING_TOOLS` at compilation time to enable correct analysis of Intel® TBB applications. Otherwise the Inspector may generate false positives during threading error analysis. If you use Intel® TBB debug libraries, do one of the following to set the macro `TBB_USE_THREADING_TOOLS`:
  - Use the `/MDd` switch to set the `_DEBUG` preprocessor symbol (recommended).
  - Set the macro `TBB_USE_DEBUG`.

If you use Intel® TBB release libraries, set `TBB_USE_THREADING_TOOLS` macro. See Intel® TBB documentation for more information.

- Inspector does not detect deadlocks or potential deadlocks created with:
  - Some types of locks via Intel's C/C++ parallel extension (`__critical`) provided by the Intel® Parallel Composer
  - 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 etc.
- Inspector may not detect threading issues on data accessed in the C runtime library (like `memmove` and `memcpy`).
- Inspector does not detect inter-processes data races or deadlock/potential deadlocks.
- Inspector does not capture the main thread creation site if the `.pdb` symbol file is not in the location specified within the `.exe` or `.dll` executable file, or in the location containing the `.exe` or `.dll` executable file.
- Inspector may report false positives for analyzed applications using customized synchronization primitives.

## Memory Error Analysis

- On the 64-bit version of the Windows 7\* operating system, the Inspector may show incorrect call stacks associated with memory leaks detected by the narrow (mi1) analysis setting. Any stack frames corresponding to functions in libraries/executables that call `LoadLibrary()` will be missing in call stacks associated with memory leaks. Workaround: Analyze your application using a wider memory analysis setting (mi2 and mi3).

- Inspector does not report memory leaks when using the narrow (mi1) analysis setting if the application under analysis circumvents the normal termination flow and does not call `ExitProcess()` (which is a call normally made by the runtime library when the application's main function ends). Workaround: Analyze your application using a wider memory analysis setting (mi2 and mi3).
- Inspector does not report memory as leaked if a pointer to the memory is available in the application memory space at the time the application exits, because the application has the ability to free this memory. For example, if an application allocates a block of memory and stores a pointer to the memory in a global variable, this memory is not included in a list of reported memory leaks. Only memory that has no pointer to it is considered as a leak.
- Inspector may report false positives when the analyzed application uses custom memory allocators.
- In some circumstances, the Inspector does not record the deallocation of memory freed during application shutdown. For example, the Inspector may not record the event if memory is freed from the destructor of an object that is located in global memory, and that destructor does not execute until late in the shutdown process. Such memory may be reported as a memory leak.
- If the semantics of standard C runtime allocators are changed (the application uses non-standard versions) such that the memory returned by the allocator is initialized, the behavior of the Inspector is unknown and could lead to abnormal analysis termination.
- Inspector may report mismatched allocation/deallocation for an array that appears correct with an allocation of `new type[]` and a matching `delete[]` if the code uses `#include <new.h>`. This occurs because the underlying implementation brought in by this include file may not actually use a matched deallocation to support backward compatibility. Applications that use `#include <new.h>` are non-conforming C++ applications. Workaround: Make the code conform by using `#include <new>` (which eliminates this problem), or suppress the code.
- Narrow memory error analysis setting (mi1) may not report leaks for the memory allocated with the operator `new` from `mfc90ud.dll` (`mfc90u.dll`). Workaround: Copy the corresponding pdb-file (`mfc90ud.i386.pdb` or `mfc90ud.AMD64.pdb`) from the `C:\WINDOWS\symbols\dll` directory to the directory where `mfc90ud.dll` is located.
- The behavior of Memory Leak Analysis level 1 (mi1) is undefined and could lead to abnormal analysis termination if the analyzed application links with the release version of `tbbmalloc.dll`. Workaround: Use the debug version of `tbbmalloc.dll`.
- When doing Memory Error Analysis on applications that use fibers or user-level threads, the Inspector 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 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.

## Command-line Interface

- Options put in a file and passed to the `insp-cl` command with the `-option-file` option cannot use the same syntax alternatives used when entering these options on the command line. The restrictions are as follows:
  - Put a newline character after the final line in the file, otherwise the final character is duplicated.
  - Use '=' between the option name and its value(s)

For more information, please refer to Technical Support.

## 6 Attributions

### 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

## 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.

## Boost

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.

## 7 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'S 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>

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](http://www.intel.com/products/processor_number) for details.

MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264, MP3, DV, VC-1, MJPEG, AC3, AAC, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.167, G.168, G.169, G.723.1, G.726, G.728, G.729, G.729.1, GSM AMR, GSM FR are international standards promoted by ISO, IEC, ITU, ETSI, 3GPP and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to  
<http://www.intel.com/performance>

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Core Inside, i960, Intel, the Intel logo, 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.

Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

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

Copyright © 2009-2011, Intel Corporation. All rights reserved.