

Intel® Inspector 2017 for Windows*

Release Notes

Installation Guide and Release Notes

15 September 2016

Contents:

[Introduction](#)

[What's New](#)

[System Requirements](#)

[Installation Notes](#)

[Issues and Limitations](#)

[Attributions](#)

[Disclaimer and Legal Information](#)

1 Introduction

Intel® Inspector 2017 helps developers identify and resolve memory and threading correctness issues in their unmanaged C, C++ and Fortran programs as well as in the unmanaged portion of mixed managed and unmanaged programs. Additionally the tool identifies threading correctness issues in managed .NET* C# programs.

Intel Inspector is a dynamic error checking tool for developing multithreaded applications on Windows* or Linux* operating systems. Intel Inspector maximizes code quality and reliability by quickly detecting memory, threading, and source code security errors during the development cycle. Intel Inspector is an easy, comprehensive solution that delivers rapid results by isolating memory and multithreading errors.

Intel Inspector has a standalone graphical user interface (GUI) as well as a command line interface (CLI). In addition, the Intel Inspector integrates into the Microsoft Visual Studio* 2010 and later versions.

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. For documentation, open the `documentation_inspector.htm` file.. For example, if you choose the default installation path, you can find the `documentation_inspector.htm` file in the `C:\Program Files (x86)\IntelSWTools\documentation_2017\en\Inspector` directory. You can now

access the product help in a web browser by opening the index.htm in the documentation help directory.

If you did not register this product during installation, do so at the Intel® Software Development Products Registration Center (<https://registrationcenter.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, visit <http://www.intel.com/software/products/support/>. **Note:** If your distributor provides technical support for this product, contact them for support rather than Intel Corporation.

2 What's New

Intel® Inspector 2017 Update 1

- Bug fixes

Intel® Inspector 2017

- Fix for suppression file usage when run in command line mode
- Added support for C++11 synchronization primitives during threading analysis
- Fixes for analyzing MPI applications
- Variable name detection for threading analysis (global, static and stack variables)
- Bug fixes

3 System Requirements

Supported Architectures and Terminology

Intel Inspector 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 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)
 - Intel Inspector 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, you can revert to using unsupported instructions in the target executable.
 - For the best experience, a multi-core or multiprocessor system is recommended.
- 4GB RAM
- 350M free disk space for all tool features and architectures
- Supported Microsoft* operating systems (embedded editions not supported):
 - Microsoft Windows 7* SP1
 - Microsoft Windows Server* 2008 SP2 and 2008 R2 SP1
 - Microsoft Windows HPC Server 2008 and 2008 R2
 - Microsoft Windows 8* operating system
 - Microsoft Windows 8.1* operating system
 - Microsoft Windows 10* operating system
 - Microsoft Windows Server* 2012
- Supported Microsoft Visual Studio* versions:
 - Microsoft Visual Studio* 2012
 - Microsoft Visual Studio* 2013
 - Microsoft Visual Studio* 2015
- Recommended compilers:
 - Intel® C/C++ Compiler XE 12.0 or higher
 - Intel® Visual Fortran Compiler XE 12.0 or higher
- Native Application coding requirements
 - Supported programming languages:
 - Fortran
 - C
 - C++
 - Supported threading methodologies:
 - Intel® Threading Building Blocks (Intel® TBB)
 - Win32* Threads on Windows* OS
 - OpenMP* (*via Intel® OpenMP compatibility library*)
 - Intel® C++ Compiler parallel language extensions
 - Intel® Cilk™ Plus
 - Microsoft PPL* (*via Intel TBB*)
- Managed Application coding requirements

- Supported programming languages:
 - C# (full support for .NET* versions between 2.0 to 3.5 inclusive)
 - C# (limited support for .NET 4.0 – see note below).
- Adobe* Reader* 7.0 or later for reading installed documentation

Note:

Intel Inspector has limited support for .NET 4.0:

- The Microsoft * Task Parallel Library classes and algorithms in the System.Threading.Tasks namespace are not supported.
- The following synchronization class in the System.Threading namespace (new in .NET 4.0) is not supported: Barrier.

4 Installation Notes

The installation of the Intel® Inspector removes any earlier installed minor version of this product (with the same major version number). Different major versions can co-exist with each other, however, only one version can be integrated with any one particular Microsoft Visual Studio* installation at a given time.

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

The product installation package is a self-extracting compressed executable file with one IA-32 package that you can install on either a 32-bit or 64-bit system. If downloaded, this compressed file may be located at: `C:\Users\<username>\Download`

To begin the installation, double-click the compressed self-extracting `<package_name>_setup.exe` file as a user with administrative privileges. Select the full installation option to use default install settings and get the complete set of user interfaces (includes a GUI front-end for using the Intel Inspector as well as Visual Studio integration). Activation is required.

Activation

You must activate the product to allow installation to proceed. There are several methods for product activation:

- Activate using a serial number. Internet connection is required.
- Activate remotely using a serial number. Use when your computer is not connected to the Internet. You can use another computer with Internet access.
- Activate using a license file.
- Activate using a license server.

You can also evaluate the product for 31 days.

Floating licensing service requirements

Intel® Software License Manager version 2.3, available on the [Intel® Registration Center](#) (choose the right OS and platform; only users with floating license registrations have access to this link).

For more details please refer to the <https://software.intel.com/en-us/articles/intel-software-license-manager-getting-started-tutorial>

Intel® Software Manager

Intel® Software Manager is a utility that lets you:

- Download and install updates for your Intel® Software Development Products.
- Manage subscription status of installed software.
- Activate serial numbers.
- Find out about the latest news for Intel Software Development Products.

Intel Software Manager requires an Internet connection to connect to a remote server for information and updates.

Intel Software Manager installs with Intel Software Development Products on Windows*, Linux*, and Mac OS* X operating systems.

To obtain more information about the Intel Software Manager, please refer to the <https://registrationcenter-ssl.intel.com/Docs/ism.htm> web page.

Default Installation Folders

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

```
C:\Program Files (x86)\IntelSWTools\Inspector 2017
```

You can select a different top-level folder name as part of a custom installation.

Activating Your Evaluation Software After Purchase

If you installed an evaluation version of Intel Developer Products and then decided to purchase the product, you can use the Intel Software Manager to enter a valid product serial number to convert your evaluation license to fully licensed status.

You can run Intel Software Manager from `C:\Program Files (x86)\Common Files\Intel\Intel Software Manager\ism.exe`. Click **Activate**, and supply a valid product serial number to convert your evaluation software to a fully licensed product.

Changing, Updating or Removing the Product

If you want to add or remove components from an installation, open the **Control Panel**, select the **Add or Remove Programs** applet, select **Intel Inspector 2017 Update X**, and click **Change/Remove** (or **Uninstall/Change** in the Microsoft Windows 7* operating system).

Installing Silently (Non-interactive)

Silent installation allows you to install, repair or remove the product without requiring your input (batch mode) during the execution of the install. Options typed on the command line allow you to activate the product with a license file or serial number and customize the installation location.

To install the Intel Inspector in silent mode, invoke the setup using the following command:

```
<package_name>_setup.exe --s --a install --eula=accept [{--  
license=<path-to-license-file> | --sn=<serial number>}] --  
output=<report-file>
```

With this command, the compressed `<package_name>_setup.exe` self-extracting file silently uncompresses to `%ProgramFiles%\Intel\Download\Inspector_2017_setup` (the default extract location) and executes `setup.exe`, passing all arguments following option `--a`. In turn, the uncompressed application `setup.exe` installs the product into a default directory using the specified license file or serial number for activation.

- The following options are **required** after the `--a` option (they are passed to the uncompressed `setup.exe`): The use of `--eula=accept` means you accept all terms and conditions of the End User License Agreement.
- Unless you already have an existing license, specify either `--sn` or `--license`. If you have a serial number, choose the `--sn` option. If you have a license file, choose the `--license` option. If you omit both options, the install will look for an existing license.
- The `--output` option writes log information reported during the install into the file specified. View this file to see installation progress and any warnings or errors encountered.

To specify a non-default installation directory, type the `--installdir` option.

To get more help on options for the self-extracting compressed file, type the following command:

```
<package_name>_setup.exe -help
```

To get more help on uncompressed `setup.exe` options, type the following command:

```
setup.exe --help
```

Installing Collectors on Remote Systems

You can install the Intel Inspector collectors on a remote system after installing and activating the full product on one system. This is intended for use on systems where space or license checking is an issue. No product activation is required to install and run the collectors on a remote system, but the analysis results may not be viewed on the remote system. Result folders must be copied or made available through the network to the system where the full product was installed and activated, since the visualization of results and command-line reporting option require an activation check.

To install the Intel Inspector collectors on a remote system:

- 1) Run `<package_name>_setup.exe -x` to extract the installer files into the default location: `%ProgramFiles%\Intel\Download\Inspector_XE_2017_setup`. If you need another location, use the command `<package_name>_setup.exe -x -f <location>`. **Note:** You can skip this step if you know the extraction location for the original installation.
- 2) Locate and copy the folder `Installs\cs_he_cli` to the remote machine.
- 3) On a remote machine, run `Installs\cs_he_cli\Inspector_XE.msi` and follow the installer instructions.
- 4) To install collectors silently, use:
`msiexec.exe -i Installs\cs_he_cli\Inspector_XE.msi -qn -log <log-file>`

IMPORTANT: While no activation is required to install and run collectors on a remote machine, you are still bound by the terms and conditions of the End User License Agreement in the original install and included with the collectors remote install after you install the collectors.

Setting up the Intel Inspector Command Line Environment

To easily access the command line interface, use a command prompt window with the proper environment settings. You can open such a window from the **Start** menu, using a shortcut under **All Programs > Intel Parallel Studio XE 2017 > Analyzers > Command Prompt**.

Alternatively, you can open a command prompt window and use the following command to setup the proper environment:

```
C:\Program Files (x86)\IntelSWTools\Inspector 2017\inspxe-  
vars.bat
```

Known Installation and Configuration Issues

- In very rare situations, the installer may hang during the phase *Configuring integration with Microsoft* Visual Studio 20xx software...* phase. If the installer hangs for more than 20 minutes, locate process `devenv.exe` and shut down the process. Make sure that you do not have any running instances of Visual Studio software. Reboot the machine and then install the product again.

- When installing the Intel Inspector for the first time on a Visual Studio 2010 system, you may be asked to initialize the Local Store for documentation. Follow **Help Library Manager** installation wizard instructions to register and install the Intel Inspector Help documentation. You do not need to re-register the Intel Inspector Help documentation when you install future Intel Inspector updates. For more information, see <http://msdn.microsoft.com/en-us/library/dd264831.aspx>. If you encounter problems viewing the Intel Inspector (local) help: Choose **Help > Manage Help Settings > Settings**, and check **I want to use local help**.
- By default, Visual Studio 2012 software sets the **Launch in Browser** option to display documentation for integrated products. To view the Intel Inspector local help documentation: Choose **Help > Intel Inspector 2017 > Intel Inspector 2017 Help** or use context-sensitive help. If you still encounter problems viewing the help, Choose **Help > Set Help Preferences > Launch in Help Viewer**.
- If you have problems viewing Microsoft Visual Studio 2012 help pages, check the settings for Internet Explorer in **Tools > Internet Options > Security**. To allow correct display of help, in the **Internet zone** enable **MIME Sniffing** and **Active scripting**. Alternatively, from **Tools > Internet Options > Security** add “**about:internet**” to the list of trusted sites.

5 Issues and Limitations

General Issues

- On some hardware, the Intel Inspector may report data race problems on simple operations (such as variable load or store operations) assumed to be atomic operations instead of explicit atomic instructions. In particular, the Intel Inspector reports false data race issues on C++ 11, `std::atomic` class operations compiled with the GNU gcc* compiler. Intel and Microsoft* compilers use explicit atomic instructions; therefore, code generated with these compilers is not impacted.
- Your target application might crash or freeze in case when you are using one of the Intel® Inspector debug analysis options in Microsoft Visual Studio* IDE running on systems with Intel(R) Xeon(R) processor E5 v2 and v3 family.
- Intel® Inspector does not support development of new Windows 8* Store Apps and use of WinRT APIs.
- Intel Inspector supports the .NET 4.0 Runtime with the exception of the new Task Parallel Library classes.
- Intel Inspector may report false positives when analyzing applications that use the Microsoft C++ runtime in Visual Studio 2012.
- Intel Inspector collected results may not be visible after closing and re-opening of the solution (.sln file) during a single Visual Studio 2012 work session. This is due to a known bug in Visual Studio 2012.
Recommendation: Restart Visual Studio 2012.
- Some errors in the new C++11 extensions supported in Visual Studio 2012 are not currently detectable.

- Intel® Inspector does not support processing of results generated by the legacy Intel® Thread Checker product. However, such Intel® Thread Checker generated results can still be processed and converted by Intel® Inspector XE 2011 product (either Standalone or Visual Studio integrated version of the product).
- Intel does not guarantee that the Intel Inspector 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.
- In the Intel Inspector, the Code complexity metrics feature isn't available for Fortran code.
- You can use the Intel Inspector 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 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 may crash or hang the Intel 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 will not be reported.
Recommendation: Review the reported problems – it is likely one of them caused the crash.
- Intel Inspector 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 interface can be used to run an analysis in this case and the results can subsequently be viewed using the graphical interface.
- Intel 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 Intel 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.
- If the Intel Inspector 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 reports insufficient memory errors during analysis, try analyzing your application in sections by running several analyses and excluding a different set of modules for each run.
Note: The Intel Inspector does not detect or report issues in excluded modules.
- To analyze applications using Boost* on a Windows* operating system, you must compile your application and Boost libraries (for example, `boost_thread-vc80-mt-gd-1_42.dll` etc.) with debugging information enabled (`/Zi`), optimization disabled (`/Od`), and the C runtime library dynamically linked in. To make sure you have

debugging information enabled and optimization disabled, you can download the Boost source and build the libraries yourself. The `.pdb` files need to be placed next to the binaries or at the location of `pdb` paths coded in the binaries. To learn more, see: <http://software.intel.com/en-us/articles/building-boost-for-applications-analyzed-by-intel-inspector-xe/>.

- If a child application is selected for analysis, the Intel Inspector analyzes only the first instance of that application, even if multiple instances of that application occur.
- If the Intel Inspector is integrated with Visual Studio* software, it is not possible to remove the last module from the include/exclude module list in the Target tab of the Project Properties dialog box. [200217183]
Recommendation: Deselect the “Inherit system environment variables” check box on the same tab to enable removal of the last module from the list.
- To debug an application that is running an Intel Inspector analysis you must choose one of the Intel Inspector debug analysis options when starting the application. Using the Visual Studio Debugger “attach” command after starting an application under analysis does not have the same effect and will result in debugging the Inspector analysis tool and not the application. When starting a debug analysis session in the documented manner, the Intel Inspector uses a probe in the Visual Studio process to intercept and modify the Visual Studio Debugger behavior such that it controls execution and displays information about the target application rather than the analysis tool.
- When taking advantage of the Intel Inspector custom filter by source, be aware that this filter does not persist when reloading results.
- Detaching an active Intel Inspector debug session is not supported on Visual Studio* IDEs and will cause the debug session and collection to terminate. [200233639]
- On-demand leak detection will not report memory leaks until the end of the basic block in which they occur, as there is still an internal reference in the program to that memory. If the baseline for leak detection is not reset, these leaks will be reported at the next request [200348282]
- The definition of `std::cv_status` changed in Microsoft Visual Studio* 2013 in a way that makes it impossible for the Intel Inspector to support running Visual Studio* 2012 and Visual Studio* 2013 simultaneously. By default, the Intel Inspector supports the 2013 version. If you need to support the 2012 version, set `INSPXE_USE_CPP11_VS2012` in your environment before performing an analysis.
- The message detected an attempt to suspend an internal thread... happens when the target application attempts to suspend all threads, causing Inspector thread suspension as well. To avoid this error:
 - Disable the Enable collection progress information checkbox in the Target tab of the Project Properties dialog box.
 - Disable the Enable interactive memory growth detection checkbox in the Analysis Type pane.
 - Rerun the analysis.
 - Also avoid running an interactive debugging session during analysis. [200238394]
- Inspector XE 2015 Update 1 only has limited support for Windows* fibers.

Memory Error Analysis Issues

- An Intel® Cilk™ Plus program that does not run to completion when serialized will not run successfully under memory analysis. Intel Inspector memory analysis works with Intel Cilk Plus programs that have well defined semantics. These semantics are defined with respect to program serialization (see the following resource for more details: http://software.intel.com/sites/products/documentation/hpc/composerxe/en-us/cpp/mac/cref_cls/common/cilk_serial.htm). [200174735]
- Intel Inspector 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.
- If the semantics of standard C runtime allocators are changed, the behavior of the Intel Inspector is unknown and could lead to abnormal analysis termination. 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.
- Memory error analysis of applications that use Qt* APIs may report false positives.
- If you turn on `Analyze stack accesses` for a memory analysis, then make sure the application being analyzed is not compiled with the default Microsoft Visual C++* setting `/RTC1` or with `/RTCs`. These options cause a basic runtime check for stacks which uses a fill technique to detect uninitialized accesses that hides errors from the Intel Inspector's memory analysis.
- Intel Inspector memory analysis may miss memory releases in `atexit()` registered functions with analysis type `mi2` and above and therefore report leaks of the memory released in these functions. [200233761]

Threading Error Analysis Issues

- Intel Inspector is not a replacement for a traditional debugger, such as the Visual Studio* debugger on Windows* operating systems or GNU gdb* debugger on Linux* operating systems. If an application crashes inside/outside the Intel Inspector, try running the application inside the debugger to reproduce and fix the crash.
- Intel Inspector does not capture the main thread creation site if:
 - The binary is built without debug symbol information.
 - 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.
- Intel Inspector 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.
 - Some types of locks in Intel® Threading Building Blocks (Intel® TBB) (`spin_mutex`, `spin_rw_mutex`)
 - Non-exclusive ownership synchronization objects involved, for example, condition variables, semaphores and events and reader/writer locks.
- Intel Inspector does not detect inter-process data races or deadlocks.
- Intel Inspector does not detect data races or deadlocks in system modules (say modules residing in `windows\system32` directory). For example, DX9 libraries and memory accessed by DX libraries installed in Windows system directory are not analyzed. To enable detection of the issues, move the modules to a non-system directory (say to the Windows* Program Files directory). [200139450]

- Intel Inspector 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.
- Intel Inspector may report false positives when analyzing applications that use the Microsoft* Concurrency Runtime framework (Parallel Patterns Library, Asynchronous Agents Library, etc.).
- Intel Inspector may report false positives if you have `Microsoft C runtime` statically linked in the application.
Recommendation: Build your application with `Microsoft C Runtime` dynamically linked in.
- To enable correct analysis of Intel TBB 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
- Using multiple versions of the Microsoft .NET* framework in the same application may cause the Intel Inspector to hang.

6 Attributions

The following are licenses for third party software that was used to develop the Intel Inspector XE. These licenses are listed due to attribution requirements in these license agreements. For the avoidance of doubt, the Intel Inspector XE is solely governed by the terms and conditions of the End User License Agreement for Intel® Software Development Product that accompanies the Intel Inspector XE.

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 Roebling 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 licensed 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 NON-INFRINGEMENT.

IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHERIN 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.

Libpng

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.2.6, August 15, 2004, through 1.5.11, June 14, 2012, are Copyright (c) 2004, 2006-2012 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors

Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux
Eric S. Raymond
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are

distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996, 1997 Andreas Dilger
Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any

source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png_get_copyright" function is available, for convenient use in "about" boxes and the like:

```
printf("%s",png_get_copyright(NULL));
```

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
June 14, 2012

Libjpeg

We welcome the use of this software as a component of commercial products. No royalty is required, but we do ask for an acknowledgement in product documentation, as described under LEGAL ISSUES.

LEGAL ISSUES

=====

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you,

its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by

patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

Libtiff

Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache

Apache License - Version 2.0 – January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this

definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the

Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole

responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

7 Disclaimer and Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Cilk, Intel Xeon Phi, VTune, and Xeon are trademarks of Intel Corporation in the U.S. and/or 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.

© 2016 Intel Corporation.