# Intel® Parallel Composer Installation Guide and Release Notes

Document number: 321418-001US
21 October 2009

## Table of Contents

# 1 Introduction

This document provides a summary of new and changed product features, and notes about features and problems not described in the product documentation.

First-time users are encouraged to read *Getting Started with the Intel® Parallel Composer* which can be found in the Intel® Parallel Composer Documentation folder.

## 1.1 Change History

This section highlights important changes in product updates. For a list of corrections to reported problems, please read Intel® Parallel Composer Fixes List.

Update 3

- Intel® Threading Building Blocks updated to 2.2 Update 1
- Corrections to reported problems

Update 2 Revised

- The Intel® 64 architecture compiler was rebuilt to correct an issue where this compiler would not run on certain Windows 7* and Windows Server 2008* systems. Correctness of generated code is not an issue.
- Intel® Threading Building Blocks (Intel® TBB) changed to version 2.2 in Update 2 – this was not noted earlier. The installation folder for Intel® TBB on the Intel® 64 architecture is now named “intel64” instead of “em64t”.
- Microsoft Windows 7* was added as a supported operating system

Update 2

- The Parallel Debugger Extension can now display taskwaits
- Corrections to reported problems

Update 1

- The Parallel Debugger Extension no longer fails due to a dependency on MSVCR71.DLL
- Added note about requiring use of the dynamic-linked OpenMP library (default) to use the Parallel Debugger Extension

- Corrections to reported problems

## 1.2 Product Contents

*Intel Parallel Composer* includes the following components:

- Intel® C++ Compilers for building applications that run on IA-32 or Intel® 64 architecture systems running the Windows* operating system
- Intel® Integrated Performance Primitives 6.1 Update 2
- Intel® Threading Building Blocks 2.2 Update 1
- Intel® Parallel Debugger Extension
- Integration into Microsoft* development environments
- Sample programs
- On-disk documentation

## 1.3 System Requirements

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

- A PC 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)
    - For the best experience, a multi-core or multi-processor system is recommended
- 1GB RAM
- 3GB free disk space for all product features and all architectures
- Microsoft Windows XP*, Microsoft Windows Vista*, Microsoft Windows 7*, Microsoft Windows Server 2003* or Microsoft Windows Server 2008* (embedded editions not supported)
- When installed on Microsoft Windows Server:
    - Microsoft Visual Studio 2008* Standard Edition (or higher edition) SP1 with C++ and “x64 Compiler and Tools” components installed [1]
- When installed on Microsoft Windows XP, Windows Vista or Windows Server 2003, one of:
    - Microsoft Visual Studio 2008* Standard Edition (or higher edition) with C++ and “x64 Compiler and Tools” components installed [1]
    - Microsoft Visual Studio 2005* Standard Edition (or higher edition) with C++ and “x64 Compiler and Tools” components installed [1]
- To read the on-disk documentation, Adobe Reader* 7.0 or later

Notes:

1. Microsoft Visual Studio 2005 and 2008 Standard Edition installs the “x64 Compiler and Tools” component by default – the Professional and higher editions require a “Custom” install to select this.
2. Applications can be run on the same Windows versions as specified above for development. Applications may also run on non-embedded 32-bit versions of Microsoft

Windows earlier than Windows XP, though Intel does not test these for compatibility. Your application may depend on a Win32 API routine not present in older versions of Windows. You are responsible for testing application compatibility. You may need to copy certain run-time DLLs onto the target system to run your application.

## 1.4 Documentation

Product documentation can be found in the `Documentation` folder as shown under Installation Folders. If you are new to Intel® Parallel Composer, please begin with *Getting Started with the Intel® Parallel Composer*.

## 1.5 Samples

Sample applications can be found in the `Samples` folder as shown under Installation Folders.

## 1.6 Technical Support

If you did not register your compiler during installation, please do so at the Intel® Software Development Products Registration Center. Registration entitles you to free technical support, product updates and upgrades for the duration of the support term.

For information about how to find 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 Installation

## 2.1 Pre-Installation Steps

### 2.1.1 Configure Visual Studio for 64-bit Applications

If you are using Microsoft Visual Studio 2005 or 2008 and will be developing 64-bit applications you may need to change the configuration of Visual Studio to add 64-bit support.

If you are using Visual Studio 2005/2008 Standard Edition, no configuration is needed to build 64-bit applications. For other editions:

1. From Control Panel > Add or Remove Programs, select “Microsoft Visual Studio 2005 (or 2008) > Change/Remove. The Visual Studio Maintenance Mode window will appear. Click Next.
2. Click Add or Remove Features
3. Under “Select features to install”, expand Language Tools > Visual C++
4. If the box “X64 Compiler and Tools” is not checked, check it, then click Update. If the box is already checked, click Cancel.

### 2.1.2 Installation on Microsoft Windows Vista*

On Microsoft Windows Vista, Microsoft Visual Studio 2005 users should install Visual Studio 2005 Service Pack 1 (VS 2005 SP1) as well as the Visual Studio 2005 Service Pack 1 Update for Windows Vista, which is linked to from the VS 2005 SP1 page. After installing these updates, you must ensure that Visual Studio runs with Administrator permissions, otherwise you will be unable to use the Intel compiler. For more information, please see Microsoft's Visual Studio on Windows Vista page (http://msdn2.microsoft.com/en-us/vstudio/aa948853.aspx) and related documents.

## 2.2 Installation

If you are installing the product for the first time, please be sure to have the product serial number available as you will be asked for it during installation. A valid license is required for installation and use.

To begin installation, insert the first product DVD in your computer's DVD-ROM drive; the installation should start automatically. If it does not, open the top-level folder of the DVD-ROM drive in Windows Explorer and double-click on `setup.exe`.

If you received your product as a downloadable file, double-click on the executable file (`.EXE`) to begin installation. Note that there are several different downloadable files available, each providing different combinations of components. Please read the download web page carefully to determine which file is appropriate for you.

You do not need to uninstall previous versions or updates before installing a newer version – the new version will replace the older version. If you have Intel C++ Compiler Professional Edition installed, installing Intel Parallel Composer will replace Intel C++ Compiler's integration into Microsoft Visual Studio, but the other compiler version will remain available for use from the command line.

## 2.3 Installation Folders

The product installs into a folder arrangement as shown below. Not all folders will be present in a given installation. If other Intel® Parallel Studio tools are installed, they will share the top-level installation folder.

- `C:\Program Files\Intel\Parallel Studio\`
  - `Composer`
    - `bin`
      - `ia32`
      - `ia32_intel64`
    - `Documentation`
    - `include`
    - `ipp`
      - `ia32`
      - `intel64`
    - `help`

- lib
  - ia32
  - intel64
- perf_headers
- Samples
- tbb
  - ia32
  - inte64
  - include
  - examples
- setup_c
- VSDebugExtension

Where the folders under `bin, include, lib, ipp` and `tbb` are used as follows:

- `ia32`: Files used in building applications that run on IA-32 architecture systems
- `intel64`: Files used in building applications that run on Intel® 64 architecture systems
- `ia32_intel64`: Files used in building applications that run on Intel® 64 architecture systems. The compiler executable in this folder runs on either IA-32 or Intel® 64 architecture systems.

If you are installing on a system with a non-English language version of Windows, 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.

## 2.4 Installation Known Issues

During installation, Intel® software cannot handle Unicode characters in license paths and the names of the licenses. Intel® software tries to find licenses in the standard location (`%CommonProgramFiles%\Intel\Licenses`, most commonly `C:\Program Files\Common Files\Intel\Licenses` on 32-bit systems and `C:\Program Files (x86)\Common Files\Intel\Licenses` on 64-bit systems). Do not place licenses in folders or paths containing localized characters. For example: C:¥インテル¥ライセンス. Do not rename licenses obtained from Intel using localized characters. For example マイライセンス.lic. Do not set the INTEL_LICENSE_FILE environment variable to contain directory paths and license names containing localized characters. Keep licenses either in the standard location (see above), or use ASCII characters in directory names and license names. For example: `C:\Intel\Licenses and License.lic`.

# 3 Intel® C++ Compiler

This section summarizes changes, new features and late-breaking news about the Intel C++ Compiler as part of Intel Parallel Composer.

## 3.1 Known Issues

### 3.1.1 Using a Source Control System

If your project is managed under a source control system, for example, Microsoft Visual Source Safe* or Microsoft Visual Studio Team Foundation Server*, there are additional steps you must follow in order to use the Intel C++ project system with your project. A detailed article on this topic is available at http://software.intel.com/en-us/articles/tips-on-using-the-intel-c-compiler-with-source-code-control-software/

### 3.1.2 Option to Clean Project after Conversion to use Intel C++

When the option is selected to use Intel C++ for a project, a dialog box appears offering to "clean" the project (remove results of previous builds). This step is recommended and is selected by default; you may choose to disable the "clean" operation so that you may perform the "clean" manually.

However, if you are using a non-English version of Microsoft Visual Studio, this dialog does not appear and you must "clean" the project manually.

# 4 Intel® Integrated Performance Primitives

This section summarizes changes, new features and late-breaking news about the Intel Integrated Performance Primitives as part of Intel Parallel Composer.

## 4.1 Known Issues

Please note the following with respect to this particular release of Intel® Integrated Performance Primitives

### 4.1.1 Intel® IPP Static libraries are provided as a separate download

The Intel® IPP static libraries are provided as a separate download. See here for download and installation instructions if you would like to use them.

### 4.1.2 Intel® IPP Cryptography libraries are available as a separate download

The Intel® IPP cryptography libraries are available as a separate download. For download and installation instructions, login to the Intel(R) Software Development Products Registration Center and review the product description.

# 5 Intel® Threading Building Blocks

This section summarizes changes, new features and late-breaking news about the Intel Threading Building Blocks as part of Intel Parallel Composer.

## 5.1 Known Issues

Please note the following with respect to this particular release of Intel Threading Building Blocks.

### 5.1.1 Issue When Multiple Visual Studio Versions Are Installed

Intel Threading Building Blocks (Intel® TBB) provides separate sets of dynamic-linked libraries (DLLs) for the various supported versions of Microsoft Visual Studio. If you have more than one supported Visual Studio version installed, for example, 2005 and 2008, installing Intel Threading Building Blocks will add the DLL folders for all supported Visual Studio versions to the system `PATH` environment variable; typically with the newest version listed first. Since the DLLs have the same names for the different Visual Studio versions, this means that only the first set as found on `PATH` will be used. This may cause problems for applications built with one version of Visual Studio but run using the DLLs for a different version.

If you have more than one Visual Studio version installed, Intel recommends that you choose one version for your Intel® TBB application development and remove the folder for the other version from `PATH`. The folder names are of the form:

- VS2005: C:\Program Files\Intel\Parallel Studio\Composer\tbb\ia32\vc8\bin
- VS2008: C:\Program Files\Intel\Parallel Studio\Composer\tbb\ia32\vc9\bin

The path may vary depending on architecture and system language. Intel Threading Building Blocks does not provide static libraries.

### 5.1.2 Library Issues

- To allow more accurate results to be obtained with Intel® Thread Checker or Intel® Thread Profiler, download the latest update releases of these products before using them with Intel Threading Building Blocks.
- If you are using Intel Threading Building Blocks and OpenMP* constructs mixed together in rapid succession in the same program, and you are using Intel compilers for your OpenMP* code, set KMP_BLOCKTIME to a small value (e.g., 20 milliseconds) to improve performance. This setting can also be made within your OpenMP* code via the kmp_set_blocktime() library call. See the Intel compiler OpenMP* documentation for more details on KMP_BLOCKTIME and kmp_set_blocktime().
- In general, non-debug ("release") builds of applications or examples should link against the non-debug versions of the Intel Threading Building Blocks libraries, and debug builds should link against the debug versions of these libraries. On Windows systems, compile with /MD and use Intel Threading Building Blocks release libraries, or compile with /MDd and use debug libraries; not doing so may cause run-time failures. See the Tutorial in the product "Documentation" sub-directory for more details on debug vs. release libraries.

# 6 Intel® Parallel Debugger Extension

This section summarizes changes, new features and late-breaking news about the Intel Parallel Debugger Extension as part of Intel Parallel Composer.

## 6.1 Known Issues

- If you are using Microsoft Visual Studio 2005, there are three Intel-specific exceptions that must be enabled manually. Select `Debug` > `Exceptions`, expand the `Win32 Exceptions` tree, and enable items:

```
a1a01db0 Intel Parallel Debugger Extension Exception 0
a1a01db1 Intel Parallel Debugger Extension Exception 1
a1a01db2 Intel Parallel Debugger Extension Exception 2
```

This needs to be done once per project.

- Use of the Intel Parallel Debugger Extension requires that the OpenMP library be linked dynamically, which is the default. If you wish to use the Parallel Debugger Extension, do not use `/Qopenmp-link:static` to specify static linking of the OpenMP Library.

- Disabling the Intel Debugging exceptions during a debug session may cause Visual Studio (up to Visual Studio 2008, SP1) to hang.

- Be sure to enable the parallel debug instrumentation before you start parallel debugging: `Properties` > `Configuration Properties` > `C/C++` > `Debug` > `Enable Parallel Debug Checks`. Otherwise, the debugger will not detect data sharing events nor break on re-entrant calls.

- If you are using Microsoft Visual Studio 2008 and debugging 64-bit applications, you must have Visual Studio 2008 Service Pack 1 installed.

    - You can debug 64-bit applications under Visual Studio 2005 and 2008 without Service Packs only if they are linked to the low memory area. If not linked to the low memory area, you will not see any events until the debuggee terminates. After termination, all events are displayed in the event window. In order to debug 64-bit applications properly, set the base address to `0x10000` in `Project` > `Properties` > `Linker` > `Advanced`.

- If you also have Intel® Visual Fortran Compiler installed, you may see an error message from the Intel Parallel Debugger Extension when running Fortran programs under the Visual Studio debugger. The error message can be ignored.

- Local variables are displayed as "`???`" in the Data Sharing Events window.

- The SSE Registers window does not work for 64-bit applications - the window shows "`???`"

- Filters on static local variables are not set correctly via context menu.

- Reentrancy detection stops in Disassembly view.

- The debugger extension windows remain empty when their placement is changed from "docked" to "floating". The workaround is to either keep them docked or to restart the debug session after the placement was changed.

- The debugger extension requires the application to be started from Visual Studio. It does not work when attaching to an existing process.

- Windows settings are restored to default (Hexadecimal) when the window is hidden or closed and reopened again.

### 6.2 Documentation

Intel Parallel Debugger Extension Documentation can be accessed via the Help menu of Microsoft Visual Studio or by clicking the Help button of specific dialog boxes. Context Sensitive Help is also available by clicking the function key F1 after activation of a Debugger Extension window.

## 7 Disclaimer and Legal Information

and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

Celeron, Centrino, Intel, Intel logo, Intel386, Intel486, Intel Atom, Intel Core, Itanium, MMX, Pentium, VTune, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.

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