
Intel(R) Threading Building Blocks - Release Notes
Version 2.0

Contents

Overview and Product Contents

A description of the contents and directory structure of Intel(R) Threading Building Blocks can be found in the file README.txt. You can view this file either before installation, or after installation in the product "doc" sub-directory.

What's New

Some of the major features of Intel(R) Threading Building Blocks include:

- Generic, concurrent, thread-safe containers: hash table, vector, queue.

- Generic parallel algorithms: parallel for, reduce, scan (prefix), while, pipeline, sort.

- Atomic operations: read, write, fetch-and-store, fetch-and-add, compare-and-swap.

- Locks: spin, reader-writer, queuing, OS-wrapper.

- Dynamic libraries. Support for multiple architectures, platforms, operating systems and compilers. Debug library versions.

- A brief Getting Started Guide that walks through an example to help you quickly learn the basics of using Intel(R) Threading Building Blocks.

- Extensive tutorial and reference documentation including several hands-on examples that illustrate how to use the major features.

Some of the new features in the 2.0 release include:

- New operating system version support.

- Parallel algorithms can now be used without specifying a fixed grain size.

- Xcode* IDE projects are now provided for the examples for use on Mac OS* systems when using Xcode* tool suite 2.4 or higher. For graphical examples, Xcode* IDE projects also include native Mac OS* graphics support.

- New convex hull example for parallel_reduce().

- Improved Intel(R) Thread Profiler support.

- Other functional, performance and documentation improvements.

See the documents in the product "doc" sub-directory, after installation, for more detailed information.

System Requirements

Hardware - Minimum Requirements

Microsoft* Windows* Systems
Intel(R) Pentium(R) 4 processor

Linux* Systems
Intel(R) Pentium(R) 4 processor or
Intel(R) Itanium(R) 2 processor
Mac OS* Systems
Intel(R) Core(TM) Solo processor
512 MB of RAM
300 MB of disk space

Hardware - Recommended

Microsoft* Windows* Systems
Intel(R) Pentium(R) 4 processor supporting Hyper-Threading Technology or Intel(R) Xeon(R) processor or higher
Linux* Systems
Intel(R) Pentium(R) 4 processor supporting Hyper-Threading Technology or Intel(R) Xeon(R) processor or higher, or Intel(R) Itanium(R) 2 processor
Mac OS* Systems
Intel(R) Core(TM) Duo processor
1 GB of RAM

Hardware - Supported

Intel(R) Pentium(R) 4 processor
Intel(R) Xeon(R) processor
Intel(R) Pentium(R) D processor
64-bit Intel(R) Xeon(R) processor
Intel(R) Core(TM) Solo processor
Intel(R) Core(TM) Duo processor
Intel(R) Core(TM) 2 Duo processor
Intel(R) Itanium(R) 2 processor (Linux* systems only)
Non Intel(R) processors compatible with the above processors

Software - Minimum Requirements

Supported operating system (see below)
Supported compiler (see below)
Xcode* tool suite 2.2.1 or higher (Mac OS* systems only)
Microsoft* Internet Explorer* 6.0 or higher (Windows* systems)
Adobe(R) Reader(R)* 6.0 or higher

Software - Recommended

Intel(R) C++ Compiler 9.0 or higher (Windows* and Linux* systems)
Intel(R) C++ Compiler 9.1 or higher (Mac OS* systems)
Intel(R) Thread Checker 3.0 or higher
Intel(R) Thread Profiler 3.0 or higher
Web browser (Linux* and Mac OS* systems)

Software - Supported Operating Systems

Microsoft* Windows* Systems
Microsoft* Windows* XP Professional
Microsoft* Windows* Server 2003
Microsoft* Windows* Vista
Linux* Systems
Red Hat* Enterprise Linux* 3, 4 and 5
(when using Red Hat* Enterprise Linux* 4 with Intel(R) Itanium(R) processors, operating system Update 2 or higher is recommended)
Red Hat* Fedora* Core 4, 5 and 6
(not with Intel(R) Itanium(R) processors)
Asianux* 2.0
Red Flag* DC Server 5.0
Haansoft* Linux* Server 2006
Miracle Linux* v4.0
SuSE* Linux* Enterprise Server (SLES) 9 and 10
SGI* Propack* 4.0 (with Intel(R) Itanium(R) processors only)
SGI* Propack* 5.0 (not with IA-32 architecture processors)
Mandriva/Mandrake* Linux* 10.1.06
(not with Intel(R) Itanium(R) processors)
Turbolinux* GreatTurbo Enterprise Server 10 SP1
(not with Intel(R) Itanium(R) processors)
Mac OS* Systems
Mac OS* 10.4.4 or higher

Software - Supported Compilers

Microsoft* Visual C++* 7.1 (Microsoft* Visual Studio* .NET 2003, Windows* systems only)
Microsoft* Visual C++ 8.0 (Microsoft* Visual Studio* 2005, Windows* systems only)
Intel(R) C++ Compiler 9.0 or higher (Windows* and Linux* systems)
Intel(R) C++ Compiler 9.1 or higher (Mac OS* systems)
For each supported Linux* operating system, the standard gcc version provided with that operating system is supported,

including: 3.2, 3.3, 3.4, 4.0, 4.1
For each supported Mac OS* operating system, the standard gcc
version provided with that operating system is supported,
including: 4.0.1 (Xcode* tool suite 2.2.1 or higher)

Installation

Detailed installation instructions for Intel(R) Threading Building Blocks can be found in the file INSTALL.txt. You can view this file either before installation, or after installation in the product "doc" sub-directory.

Known Issues

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

Library Issues

- The cache_aligned_allocator template is incompatible with the Microsoft* deque and list containers.

- The atomic<long long> and atomic<unsigned long long> templates are not supported when using the Microsoft* Visual C++* 7.1 (Microsoft* Visual Studio* .NET 2003) compiler.

- When using exceptions, note that an exception must be caught within the same task that throws the exception.

- To allow more accurate results to be obtained with Intel(R) Thread Checker or Intel(R) Thread Profiler, download the latest update releases of these products before using them with Intel(R) Threading Building Blocks.

- If you are using Intel(R) Threading Building Blocks and OpenMP* constructs mixed together in rapid succession in the same program, and you are using Intel(R) 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(R) 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(R) 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(R) 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 "doc" sub-directory for more details on debug vs. release libraries.

Library Example Issues

- When building an example on Windows* systems, either via a Makefile or via a Microsoft* Visual Studio* project, a spurious warning may be encountered of the form:

  cl : Command line warning D4002 : ignoring unknown option '/G-'

  This warning can be ignored; the compilation should complete and the compiled example should execute correctly.

- The Xcode* IDE projects for the examples, on Mac OS* systems, currently only support Xcode* tool suite 2.4 or higher. Use 'make' commands, as described in each example's index.html page, to build and run the examples on Mac OS* systems when using Xcode* tool suite versions prior to 2.4.

Windows* Installation Issues

- If the version of Windows* Installer on your system is older than version 2.0, the installer will automatically upgrade it and you will need to reboot the computer after the installation is complete.

- Scripts are used in the normal install process and may trip false positives in certain antivirus software. You may need to disable script blocking or you may be able to temporarily allow scripts for the installation process.

Linux* Installation Issues

- Some versions of RPM do not allow installation to a non-default installation directory, including RPM 4.0.2 (resolved in RPM 4.0.3) and RPM 4.1 (resolved in RPM 4.11 to 4.2).

- When installing on Linux*, there is a known issue where the registration step at the end of the installation program may freeze after your e-mail address is entered. If this occurs, you may interrupt the installation program (e.g., via pressing <ctrl-C>); the installation should already have completed. In this case, you may register via the Intel(R) Software Development Products Registration Center by visiting:

    https://registrationcenter.intel.com/

- When installing on Mandriva/Mandrake* Linux* 10.1.06, there is a known issue where the installation program may freeze after printing "installing RPM package...". If this occurs, try re-installing and choose to install without using RPM.

Mac OS* Installation Issues

- Xcode* tool suite 2.2.1 or higher must be installed prior to installing Intel(R) Threading Building Blocks on Mac OS* systems.

Technical Support

A rich repository of self-help product information such as tutorials, getting started tips, known product issues, product errata, compatibility information and answers to frequently asked questions can be found at the Intel(R) Software Development Products Technical Support site by visiting:

    http://www.intel.com/software/products/support/

To receive technical support and product updates for the tools provided in this product, please register at the Intel(R) Software Development Products Registration Center by visiting:

https://registrationcenter.intel.com/

If you have questions or problems getting started with Intel(R) Threading Building Blocks, please visit:

https://registrationcenter.intel.com/support/

For problems or technical questions, contact Intel technical support through your Intel(R) Premier Support account by visiting:

https://premier.intel.com/

To submit an issue via the Intel(R) Premier Support website, perform the following steps:

1. Ensure that Java* and JavaScript* are enabled in your web browser.
2. Go to https://premier.intel.com/
3. Type in your Login and Password. Both are case-sensitive.
4. Click the "Submit Issues" button.
5. Read the Confidentiality Statement and click the "I Accept" button.
6. Click the "Go" button next to the "Product" drop-down list.
7. Click the "Submit Issue" link in the left navigation bar.
8. Choose "Development Environment (tools,SDV,EAP)" from the "Product Type" drop-down list.
9. If this is a software or license-related issue, choose the appropriate "Intel(R) Threading Building Blocks for <platform>" item from the "Product Name" drop-down list.
10. Enter your question and complete the fields, in the windows that follow, to submit your issue.

Note the following guidelines when forming your problem report or product suggestion:

1. The specific release and package identification information for Intel(R) Threading Building Blocks can be found in the file tbbsupport.txt in the directory where the product was installed. Please note this information when submitting your problem or question.
2. Describe your difficulty or suggestion. For problem reports, please be as specific as possible. Include specific compiler and/or linker command-line options. Include as small a testcase

as is adequate to illustrate the problem.

3. Describe your system configuration. Be sure to include specific information that may be applicable, including: operating system, name and version of installed applications, and anything else that may be relevant to help address your concern.

Related Products and Services

You can find out about other Intel software development products through the Intel web site at:

http://www.intel.com/software/products/

These particular products and services may be useful in conjunction with Intel(R) Threading Building Blocks.

The Intel(R) C++ Compilers are an important part of making software run at top speed and fully support the latest Intel(R) IA-32 and Intel(R) Itanium(R) processors. For more information, visit:

http://www.intel.com/software/products/compilers/

Intel(R) Thread Checker can locate source locations that cause deadlocks, data races and other thread safety issues in threaded programs or programs that use threaded runtimes. For more information, visit:

http://www.intel.com/software/products/threading/

Intel(R) Thread Profiler is a performance tuning tool for parallel programs that use Win32*, POSIX*, OpenMP* or custom synchronization. For more information, visit:

http://www.intel.com/software/products/threading/

The VTune(TM) Performance Analyzer allows you to evaluate how your application is utilizing the CPU and helps you determine if there are modifications you can make to improve your application's performance. For more information, visit:

http://www.intel.com/software/products/vtune/

The Intel(R) Performance Library Suite provides a set of routines optimized for various Intel processors. For more information, visit:

http://www.intel.com/software/products/perflib/

The Intel(R) Software College provides interactive tutorials, documentation, and code samples that teach Intel(R) architecture and software optimization techniques. For more information, visit:

http://www.intel.com/software/college/

Disclaimer and Legal Information

Current characterized software defects are available on request.

Developers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Improper use of reserved or undefined features or instructions may cause unpredictable behavior or failure in developer's software code when running on an Intel processor. Intel reserves these features or instructions for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from their unauthorized use.