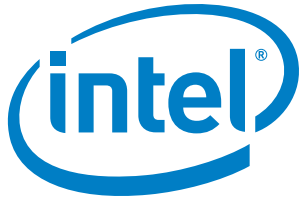


# Intel® C++ Compiler Professional Edition 11.1 for Linux*

In-Depth

## Contents

## Intel® C++ Compiler Professional Edition 11.1 for Linux*

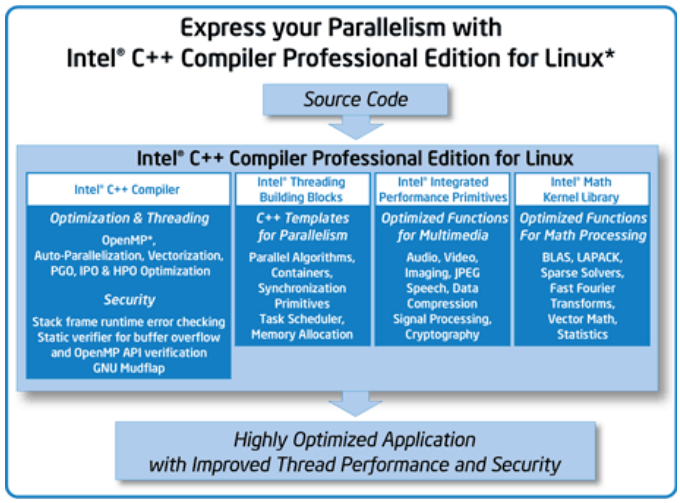
The features you need to create high-performance multithreaded apps for multicore systems. The product includes:

- Intel® C++ Compiler for Linux for IA-32, Intel® 64, and IA-64 architectures
- Intel® Threading Building Blocks
- Intel® Integrated Performance Primitives
- Intel® Math Kernel Library

Compatible with the GNU tool chain to protect your investment in how you develop. The Professional Edition offers a great price compared to the individual components.

Attention Fortran developers—Intel® Compiler Suite Professional Edition 11.1 for Linux* includes everything listed above plus the Intel® Fortran Compiler for Linux. Take advantage of significant price savings over individual components.

## Intel C++ Compiler Professional Edition 11.1— At a Glance




Intel® C++ Compiler Professional Edition Components:

The multithreading and optimization technologies in Intel® compilers and libraries give the Professional Edition its edge in helping you deliver high-performance applications for the latest multicore processors.

## Features

- **Compatibility with the GNU Tool Chain** protects your investment in the way you develop software on and for Linux-based systems. Use Intel C++, gcc, or both!
- **Multithreaded Application Support** including new in 11.0, OpenMP 3.0* (data- and now task-parallelism), and auto-parallelization for simple and efficient software threading.
- **Auto-vectorization** parallelizes code to utilize the Intel® Streaming SIMD Extensions (SSE) instruction set architectures (SSE, SSE2, SSE3, SSSE3, and SSE4) of our latest processors.
- **High-performance Parallel Optimizer (HPO)** restructures and optimizes loops to ensure that auto-vectorization, OpenMP, or auto-parallelization make best use of cache and memory accesses, SIMD instruction sets, and multiple cores. Compiles in a single pass, improving compile time and producing more reliable code.
- **Interprocedural Optimization (IPO)** dramatically improves performance of small- to mid-sized functions, especially in programs containing calls within loops. IPO analysis gives feedback on vulnerabilities and coding errors, such as uninitialized variables or OpenMP API issues, which cannot be detected as well by other compilers.
- **Profile-guided Optimization (PGO)** improves application performance by reducing instruction-cache thrashing, reorganizing code layout, shrinking code size, and reducing branch mispredictions.
- **Intel® Threading Building Blocks** is an award-winning C++ template library that abstracts threads to tasks to create reliable, portable, and scalable parallel applications. Intel® TBB is the most efficient way to implement parallel applications and unleash multicore platform performance.
- **Intel® Math Kernel Library** includes optimized and scalable math routines for maximizing performance, and seamlessly provides forward scaling from current to future manycore platforms.
- **Intel® Integrated Performance Primitives** is an extensive library of multicore-ready, highly optimized software functions for multimedia data processing and communications applications.
- **Optimized-Code Debugging** The Intel® Debugger improves the efficiency of the debugging process on code that has been optimized for Intel® architecture. Includes new threaded code debugging features and a new GUI.
- **New Integrated, Simplified Installation** gets you going with all capabilities quickly and easily. Simplified custom install makes it easy to identify just the components you want.
- **Ongoing Premier Support** now includes online community support forums to speed information flow, in addition to private, password-protected accounts. Technical support, interactive issues management, access to technical and application notes, product updates and more with every commercial and academic license.

## New in This Release

| Feature | Benefit |
|---|---|
| **New Processor Support** | Intel® Atom™ Processor - Create high-performance and battery saving applications for new Mobile Internet Devices! |
| **Compiler and Debugger** | |
| New exception handling | Same user-settable options produce better-optimized code, leading to improved application performance. |
| C++ lambda functions | Already part of the next C++ standard, Intel® C++ lambda functions available now to simplify template libraries' use with loop constructs like STL and Intel TBB. |
| OpenMP 3.0* | OpenMP raises the parallelism abstraction away from the API, simplifying threading and making code more portable. Previously limited to loop-based data parallelism, the new 3.0 standard simplifies both data and task parallelism. |
| Improved valarray option | Templates of array operations that enable low-level hardware features to enhance application performance. No source code change required to use! Implicit valarray loops can either be auto-vectorized or directed to invoke optimized Intel® Integrated Performance Primitives (IPP) library primitives for potential performance gains. |
| SSE2 enabled by default | Take advantage of new Intel® Streaming SIMD Extensions—automatically—through the compiler. No messy low-level coding to get the most from Intel® processors. Resetable for other hosts/targets. |
| Decimal floating point | IEEE 754R Standard implementation overcomes otherwise unavoidable precision issues implied by binary FP formats. Great for banking, accounting, billing, and e-Commerce. |
| Thread-related front-end diagnostics | Provides useful warnings about references and assignments to statically allocated variables and address references of statically allocated variables. |
| Parallel compilation | Supports your build by appropriately allocating files to available processors to take advantage of multicore processors and speed you through your edit/compile/debug cycle. |
| Static Verifier | Find and analyze source file issues. Diagnostics include issues with OpenMP* directives, boundary violations, memory corruptions, memory leak, buffer overflow, and uninitialized memory |
| Parallel Lint for OpenMP* | Performs static analysis to check for OpenMP parallelization correctness. Helps diagnose deadlocks, data races, or potential data dependency—side effects from synchronization issues. |
| Integrated Development Environment | Use in Eclipse CDT 5.0 provides developers with the latest Eclipse support |

| Feature | Benefit |
|---|---|
| **Intel® Threading Building Blocks (Intel® TBB)** | |
| Loops, containers, mutexes, atomic operations, and more | Intel® TBB provides developers with high-level, STL-like library functions to take advantage of parallelism in existing or planned code. Covering loops, containers, mutex controls, atomic operations, complex task scheduling, and more, Intel TBB simplifies threading, saves time, and produces applications that scale as processors are added. |
| Task-oriented threading | Eliminate the tedium and inefficiencies of threading low-level, heavy constructs close to the hardware. Thread the way you think about your application and let the Intel® TBB runtime library worry about the threads. |
| Threaded Performance | Intel® TBB focuses on the particular goal of parallelizing computationally intensive work, delivering higher-level, simpler solutions. |
| Scalable Performance through data-parallelism | Intel® TBB emphasizes programming, which scales well to larger numbers of processors by dividing the collection into smaller pieces. With data-parallel programming, program performance increases as you add processors. |
| Debug and release libraries | Intel® TBB functions come in debug and release forms to support extensive internal checking before building the production version of your software with the release form of the Intel TBB function. This simplifies development and delivers great performance. |
| **Intel® Integrated Performance Primitives (Intel® IPP)** | |
| Intel® Core™ i7 and Atom™ processor support | Continued support for new processors continues to future-proof your investment with assurance of support for each successive generation of processors. |
| Deferred Mode Image Processing (DMIP) Layer | Introduced as a sample on top of Intel® IPP libraries, this provides solutions for pipelined image operations on larger images, utilizes in-memory optimization and improves performance in a multithreading environment. |
| Unified Image Codec (UIC) framework sample library | Standardizes plug- and-play interfaces for various image codecs (JPEG, JPEG2000, etc.) to improve ease of implementation and maintenance. |
| Threaded Static libraries for all functional domains | Provides higher performance threading to static library users. |
| High-level LZO data compression library + improved Zlib, gzip and bzip2 algorithms | High-performance implementations of most popular data compression algorithms using standard interfaces for ease of use. |
| Data Integrity Functional Domain | Reed-Solomon error correcting codes to preserve integrity of data in transmission, storage, and encoding. |
| Variety of new functions and enhancements to Intel IPP samples | Continued added value in response to user requests for added functionality including new signal and image processing transforms and codec enhancements. |

## New in This Release

| Feature | Benefit |
|---|---|
| **Intel® Math Kernel Library (Intel® MKL)** | |
| New "layered" architecture | The new architecture provides maximum support for different development environment configurations and processors in a single package. |
| New threading layer | Link to the version of this layer that matches your development environment and rest assured that Intel® MKL will not have threading incompatibilities with the threading in your application. |
| Discrete Fourier Transform Interface | The DftiCopyDescriptor function has been added for convenience when using the FFTs. The size of statically linked executables calling DFTI has been reduced significantly and complex storage is now available for real-to-real transforms. |
| LAPACK enhancement | The capability to track and/or interrupt the progress of lengthy LAPACK computations has been added. A function called mkl_progress can be defined in a user application, which will be called regularly from a subset of the MKL LAPACK routines. |
| VML extensions | With performance in mind, all VML functions are now threaded. And a new "Enhance Performance" mode is offered for applications where math-function inaccuracies don't dominate parameter inaccuracies (e.g., Monte Carlo simulations and media applications). |
| Sparse BLAS extensions | Improvements include threaded level 3 sparse BLAS triangular solvers and support for all data types (single precision, complex and double complex). |
| **New Linux Support** | Fedora 9*, Ubuntu 8.04*, SLES 11, GNU tool chain 4.2 and 4.3. Eclipse CDT 5.0 support. See the *Release Notes* for a complete list. |
| **Simplified installation** | Streamlined, simplified complete installation for a seamless one-step installation of all components. |
| **New Online Support Community** | Our enhanced online community support forums and knowledge-base search capabilities help you find answers more quickly. This is in addition to private, password-protected accounts available with Premier Support. Go to the support section of the website for more information. |
| **Processor Support** | The addition of support for Intel® Atom™ processors continues to future-proof your investment with assurance of support for each successive generation of processors. That's a key advantage in a world where new hardware platforms come to market with awesome speed. For more details, see the *Release Notes.* |

## Technical Support

With the purchase of the product, you receive one year of technical support and product updates from Intel® Premier Support, our interactive and password-protected issue management and communication website. This premium support service allows you to submit questions, download product updates, and access technical notes, application notes, and other documentation. In addition, we have enhanced our user forums to provide a quick and easy first resource to help with most issues.

## Compatibility

The Intel C++ Compiler for Linux is substantially standards compliant and compatible with the Linux distributions and gcc, and related tools chain. For more specific information, please see the Release Notes. It also supports Intel® Itanium® 2 processors, including Dual-Core Intel® Itanium® 2 processors. Binaries are also substantially compatible with gcc binaries to support a mix and match interoperability capability.

### The Intel C++ Compiler provides the following language conformances:

- ANSI/ISO standard for C language compilation (ISO/IEC 9899:1990)
- ANSI/ISO standard (ISO/IEC 14882:1998) for the C++ language
- OpenMP specification version 3.0

## System Requirements at a Glance

More specific information on installation requirements is available in the Release Notes but, at a glance, Intel C++ Compiler Professional Edition for Linux can be used on, and develop code for, Intel processors since the Intel® Pentium® 4 processor.

We support the use of Intel C++ Compiler Professional Edition on recent releases of Asianux*, Debian*, Red Hat Enterprise Linux*, SUSE LINUX Enterprise Server* and TurboLinux*.

Installation of all components in the product requires 2 GB free disk space.

For more detailed information on system requirements, go to www.intel.com/software/products/systemrequirements/



0609/BLA/CMD/PDF 321500-001