IBM XL C/C++ for Linux, V13.1.4

Improve application performance and developer productivity using the latest IBM XL C/C++ compiler for Linux

Highlights

IBM® XL C/C++ for Linux, V13.1.4:

- Delivers an XL C/C++ compiler to support little endian Linux distributions.
- XL C/C++ for Linux on little endian distributions offers an alternate no-charge community edition. The full XL C/C++ for Linux compiler continues to be available and orderable.
- Leverages the capabilities of the latest POWER8® architecture.
- Maximizes application performance through industry leading optimization technology.
- Eases application migration to Power Systems™ through conformance to the C99 and C++03 standards. XL C/C++ for Linux on little endian distributions is also compliant with the C11 and C++11 standards.
- XL C/C++ for Linux on little endian distributions improves developer productivity with full support for OpenMP 3.1 and partial support for OpenMP 4.5.

Delivers an XL C/C++ compiler to support little endian Linux distributions

Starting with V13.1.1, IBM XL C/C++ for Linux contains two C/C++ compilers that support:

- Big endian Linux distributions on IBM Power® servers configured for big endian mode, supporting RHEL 6, RHEL 7, and SLES 11
- Little endian Linux distributions on IBM Power servers configured for little endian mode supporting Ubuntu Server 14.04, 14.10, and 16.04, RHEL 7.1, RHEL 7.2, CentOS 7, SUSE Linux Enterprise Server 12, and SUSE Linux Enterprise Server 12 SP1

The XL C/C++ for Linux compiler for the little endian architecture leverages the Clang infrastructure from the open source community for a portion of its compiler frontend. Clang is a component of the LLVM open source compiler and toolchain project and provides the C and C++ language family front end for LLVM. The XL C/C++ compiler combines the Clang front-end infrastructure with IBM advanced optimization technology and code generation.

The XL C/C++ for Linux compiler for little endian Linux distributions also provides a greater level of GNU compatibility.

Offers a community edition for developer and partner use

Starting from V15.1.4, XL C/C++ for Linux Community Edition on little endian distributions is available for download and deployment. It is a no-charge, entry-level C/C++ compiler for the developers and partners who have lightweight production requirements. The full XL C/C++ for Linux compiler continues to be available for order.

For a comparison of the full version, trial version, and Community Edition of the XL C/C++ for Linux compiler, see the following table.

Table 1. The trial version, the community edition, and the full version

Feature	Trial	Community Edition	Full version
License term	Limited: 60 days	Unlimited	Unlimited
Cost	No charge	No charge	Priced per user
Licensed for production environment		✓	✓
O2	✓	✓	✓
O3	✓	✓	✓
Inter-procedural analysis (IPA)	✓	✓*	✓

Table 1. The trial version, the community edition, and the full version (continued)

Feature	Trial	Community Edition	Full version
Profile-directed feedback (PDF)	✓	✓	✓
Loop optimization (-qhot)	✓	✓	✓
OpenMP	✓		✓
O5	✓		✓
Subscription and support			✓

[*] Advanced IPA is available only at O5 and is not enabled in the community edition.

While the community edition embodies most of the core features of the fully warranted XL C/C++ for Linux compiler, it does not support optimization level 5 for generation of the highest level of optimization or OpenMP constructs for parallel programming. In addition XL C/C++ for Linux Community Edition is not warranted and does not provide for any subscription, service, or support. The release of the community edition allows for convenient availability of the XL C/C++ compiler so that developers can experience the advantages of IBM compiler technology on the POWER8 platform. To obtain full-warranty, full-function, and world-class support from IBM, you should consider licensing the full XL C/C++ for Linux compiler.

Leverages the capabilities of the latest POWER8 architecture

XL C/C++ for Linux, V13.1.4 generates code that leverages the capabilities of the latest POWER8 architecture. Compiler suboptions for architecture and tuning specify code generation on the POWER8 processor architecture. `-qarch=pwr8` instructs the compiler to produce code that can fully exploit the POWER8 architecture. `-qtune=pwr8` enables optimizations, such as instruction scheduling, that maximize performance on POWER8 systems, while allowing for binary compatibility with previous POWER® processors.

XL C/C++ provides built-in functions for direct programmer access to the POWER architecture. While most programmers will rely on the compiler to exploit processor features automatically, built-in functions give you an easy way to access specific instructions or processor features using C or C++ function call syntax and C or C++ variables. XL C/C++ for Linux, V13.1.4 provides built-in functions supporting POWER8 features such as vector processing, cryptography, cache management and transactional memory.

The Mathematical Acceleration Subsystem (MASS) libraries contain frequently used elementary and special mathematical functions that enable improved performance over the corresponding standard system library functions. These highly tuned MASS libraries are enhanced to support the POWER8 architecture. In addition to the scalar library, MASS for POWER8 includes both a vector version (for arbitrary length arrays) and a SIMD or Single-Instruction Multiple Data version (for the vector datatypes). All the vector and SIMD POWER8 MASS libraries exploit the POWER8 vector instruction set, as does the scalar library for little endian Linux distributions.

Maximizes application performance through industry-leading compiler optimization technology

The optimization and hardware exploitation features in IBM XL C/C++ help improve programming productivity. The XL C/C++ compiler generates code that delivers leading-edge performance from existing and new hardware, often with no source code changes.

XL C/C++ for Linux implements the GCC symbol visibility attributes, pragmas, and option which describe whether and how an entity that is defined in one module can be referenced or used in other modules. Using the visibility attributes for entities, you can get the following benefits:

- Decreasing the size of shared libraries
- Reducing the chance of symbol collision
- Allowing more optimization for the compile and link phases
- Improving the efficiency of dynamic linking

XL C/C++ provides faster compile time for large applications. The compiler now uses less memory for some advanced optimizations. Header files that are repeatedly used in your source code are cached within the compiler to speed up compilation. The compiler front end and optimizer use the greater addressability of the 64-bit process space, thereby allowing significantly larger programs to be optimized.

Profile directed feedback (PDF) optimization collects information about an application run with typical input data and then applies transformations to the program based on that information. PDF can ensure that the performance of the application is optimized for its important inputs. Application profile monitoring and profile directed feedback capabilities minimize the need for manual tuning to achieve desirable performance on large, complex applications.

Eases application migration to IBM Power Systems

Make your applications portable with the XL compilers, which offer industry compliant programming languages and extensions. XL compilers help programmers easily maintain and run their applications on IBM systems.

IBM XL C/C++ conforms to the following programming language specifications for C/C++: C89, C99, C++ 98, and C++03. XL C/C++ for Linux, V13.1.4, on little endian distributions is also compliant with C11 and C++11 standards.

Augmenting the standardized language levels, the XL C/C++ compiler has implemented C and C++ language extensions to support vector programming and a subset of GNU C and C++ language extensions. In addition, the XL C++ compiler maintains close support of Boost C++ library releases.

C11 and C++11 features

XL C/C++ for Linux, V13.1.4, for little endian distributions is C11 and C++11 compliant. In V13.1.3, the following C11 and C++11 features are added for little endian distributions:

- Atomic types and operations (partial support) to enable synchronization and communication between threads.
- Thread-Local Storage to maintain data that is local to a thread. Note that the compiler fully supports the thread-local feature only when GCC runtime library V4.8 or later is used.

Full support of OpenMP 3.1 and support for a small subset of OpenMP 4.5

XL C/C++ for Linux provides full support for OpenMP 3.1 so programmers can automate parallel programming and take advantage of multiprocessor systems. Some of the features include finer control of the number of threads used in nested parallelism, full control of where a thread can switch from one task to another task, and more types of atomic operation to better synchronize parallel code.

XL C/C++ for Linux, V13.1.4, for little endian distributions also supports the following OpenMP 4.5 features:

- omp_get_num_places function
- omp_get_partition_num_places function
- omp_get_partition_place_nums function
- omp_get_place_num_procs function
- omp_get_place_proc_ids function
- omp_get_place_num function
- The omp_get_proc_bind function
- The OMP_PLACES environment variable
- The proc_bind clause

The following environment variables are extended to control the thread affinity policy:

- OMP_DYNAMIC
- OMP_DISPLAY_ENV
- OMP_PROC_BIND
- OMP_THREAD_LIMIT

The ATOMIC directive is extended to support sequentially consistent atomic operations by specifying a new optional clause seq_cst.

Summary

IBM compilers allow applications to take advantage of virtually all the hardware exploitation features provided by IBM processors including POWER8. By utilizing leading-edge optimization technologies in IBM compilers, organizations can improve their return on investment in hardware assets, while increasing programmer productivity.

Organizations often wait until they upgrade their hardware to upgrade their compilers. However, given that the compilers can deliver significant improvements in application performance and programmer productivity, compilers offer a cost-effective way to get more out of existing technology. By periodically upgrading compilers, programmers can take advantage of new language, usability and optimization features, and stay ahead of competitors on the technology curve.

For more information

To learn more about the IBM XL C/C++ for Linux compiler, contact your IBM representative, IBM Business Partner, or visit: XL C/C++ for Linux at www.ibm.com/software/products/en/xlcpp-linux/.

Get started today by downloading a trial version or community edition of XL C/C++ for Linux at: www.ibm.com/developerworks/downloads/r/xlcpluslinux/.

April 2016

References in this document to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.