

IBM XL C/C++ for Linux, V16.1.1



Optimize your HPC and Cognitive applications and fully exploit POWER9 architecture using the latest IBM XL C/C++ for Linux compiler

IBM XL C/C++ for Linux, V16.1.1



Optimize your HPC and Cognitive applications and fully exploit POWER9 architecture using the latest IBM XL C/C++ for Linux compiler

November 2018

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.

IBM, the IBM logo, ibm.com[®], POWER, POWER9, Power Architecture[®], and Power Systems are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

IBM XL C/C++ for Linux, V16.1.1

Optimize your HPC and Cognitive applications and fully exploit POWER9 architecture using the latest IBM XL C/C++ for Linux compiler

Highlights

IBM® XL C/C++ for Linux, V16.1.1:

- Leverages the capabilities of the latest POWER9™ architecture.
- Fully supports the OpenMP 4.5 specification for productive programming.
- Maximizes application performance through industry leading optimization technology.
- Eases application migration to Power Systems™.
- Offers an alternative no-charge, fully functional Community Edition.

Leverages the capabilities of the latest POWER9 architecture

XL C/C++ for Linux, V16.1.1 generates code that leverages the capabilities of the latest POWER9 architecture. Compiler suboptions for architecture and tuning specify code generation on the POWER9 architecture. Use `-qarch=pwr9` to instruct the compiler to produce code that can fully exploit the POWER9 architecture. Use `-qtune=pwr9` to enable optimizations, such as instruction scheduling, that maximize performance on the POWER9 architecture, while allowing for binary compatibility with previous POWER® processors.

XL C/C++ for Linux 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, V16.1.1 provides built-in functions targeting the POWER9 architecture such as fixed-point, binary floating-point, binary-coded decimal, and vector built-in functions.

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. The vector library and SIMD library in XL C/C++ for Linux, V16.1.1 contain functions that are tuned for the POWER9 architecture.

Fully supports the OpenMP 4.5 specification for productive programming

OpenMP is a portable and scalable programming model that gives programmers a simple, flexible, and standard interface for developing parallel applications for platforms ranging from the desktop to the supercomputer. The combination of the IBM POWER processors and the NVIDIA GPUs provides a platform for heterogeneous high-performance computing that can run several technical computing workloads efficiently. The computational capability is built on top of massively parallel and multithreaded cores within the NVIDIA GPUs and the IBM POWER processors.

XL C/C++ for Linux fully supports the OpenMP 4.5 parallel programming model to accelerate your applications.

In XL C/C++ for Linux, V16.1.1, you can use the `-qtgtarch` option to specify the real or virtual GPU architectures where the code can run, overriding the default GPU architecture. This allows the compiler to take maximum advantage of the capabilities and machine instructions which are specific to a GPU architecture, or common to a virtual architecture.

XL C/C++ for Linux, V16.1.1 also provides high performance computing with support for Power® System servers such as the POWER9 servers.

Maximizes application performance through industry-leading compiler optimization technology

The optimization and hardware exploitation features in IBM XL C/C++ for Linux help improve programming productivity. XL C/C++ for Linux 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 options that 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

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 and 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++ for Linux conforms to the following programming language specifications for C/C++: C89, C99, C++ 98, C++03, C11, and C++11 standards. It also supports a subset of the C++14 standard.

XL C/C++ for Linux, V16.1.1 leverages the Clang V4.0 front-end technology which provides a large degree of compatibility with GCC. 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.

XL C/C++ for Linux also provides a great level of GNU compatibility.

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.

Offers a no-charge, fully functional Community Edition

XL C/C++ for Linux Community Edition is a no-charge, fully functional C/C++ compiler. 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 POWER9 architecture. This product is not warranted and does not provide for any subscription, service, or support.

The full XL C/C++ for Linux compiler priced per user continues to be available for order. To obtain full warranty and world-class IBM support, consider licensing the full XL C/C++ for Linux compiler.

Summary

IBM compilers allow applications to take advantage of virtually all the hardware exploitation features provided by IBM processors. 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 or download the Community Edition, visit XL C/C++ for Linux at <https://www.ibm.com/us-en/marketplace/xl-cpp-linux-compiler-power>.



Printed in USA