COBOL-IT
OpenSource Business Oriented Language

The COBOL-IT® Compiler's Performance Advantage

COBOL-IT – 231, rue Saint Honore – 75001 Paris – France

COBOL-IT
OpenSource Business Oriented Language

The COBOL-IT® Compiler's Performance Advantage

Delivering COBOL-IT's performance advantage.

The COBOL-IT compiler generates intermediate "C" code, which must then be compiled with a "C" compiler for the target platform at hand to produce a working executable file. It is a fairly common technique, applied to compilers for different languages, including "C++", Eiffel, Prolog, Fortran, Pascal and Lisp. Beyond the obvious gain in portability, applying this technique to COBOL delivers significant performance advantages over alternative strategies.

Bringing top performance to your application.

It is of course intuitively clear that executable objects produced from intermediate "C" code are faster than objects produced by a byte code compiler and run with a virtual machine interpreter. It has taken Sun Microsystems several years of continuous investment to get their Java Just-In-Time virtual machine to deliver performance of the same order of magnitude as the performance delivered by native, and a significant part of the effort had to do with issues that do not apply to COBOL at all such as garbage collection.

Leveraging the best optimization technologies.

Generating intermediate "C" code is also faster than native code generation in most cases, as it can capitalize on the huge efforts in optimization that have been invested in "C" compilers for the various platforms at hand.

Making the most effective use of technical resources.

Generating intermediate "C" code is more sensible than generating native code directly also, in that it places the responsibility for optimal handling of complex architectures, such as pipelining, on the provider of the "C" compiler, where it properly belongs. Allowing the creation of native code to be handled by the native "C" compilers eliminates a resource drain, and eliminates the inevitable need to compromise for needed flexibility, while providing access to all the available optimization techniques.

Shortening time-to-market.

Reusing and optimizing existing "C" source code significantly shortens time-to-market whenever a new platform or a new version of an existing platform must be supported, as it only relies on the existence of "C" compilers which are usually developed by the hardware vendors together with the processor architectures.

Continuing to improve COBOL-IT code generation.

Earlier versions of the COBOL-IT compiler generated a single "C" function for a COBOL program. This approach produces very large "C" functions, up to a point where some optimizers can no longer be applied due to performance or resource consumption constraints. A recent change to the code generation scheme addresses this issue by dividing the generated code into a number of smaller functions. This allows the most aggressive optimizations to be applied.

Out-performing the competition.

Benchmarks consistently show that COBOL-IT 's generated code outperforms the object code of COBOL compilers producing native code.

For more information, please contact sales@cobol-it.com.

For more information about COBOL-IT and our products and services, please visit www.cobol-it.com.

CONTACT INFORMATION

COBOL-IT
231, rue Saint Honoré - 75001 Paris - France
Phone: +33 1 75 43 05 50 Email: info@cobol-it.com
Fax: +33 1 75 43 05 16 Web: www.cobol-it.com
Follow us on Twitter: https://twitter.com/cobol_it

We have Distributors and Technical Hubs worldwide.
For a complete listing, check our website:
http://www.cobol-it.com/contact.php