



## A Brief Overview of the IBM XL Compiler Family

June 2004

### Overview

This paper contains an overview of IBM XL C/C++/Fortran compilers, outlining their history, unique capabilities and cross-platform support. You will also find a performance overview and a summary of their current strategic direction.

Three additional papers discuss the optimizations available in the IBM XL compiler family, the specific features supported in the IBM XL C/C++ compilers for Mac OS X, and an overview of the IBM XL Fortran compiler for Mac OS X:

- Code Optimization with IBM XL Compilers
- XL C/C++ Advanced Edition for Mac OS X
- XL Fortran Advanced Edition for Mac OS X

You can find more information about the IBM compiler family, including downloadable trial versions, on the web at:

<http://www.ibm.com/software/awdtools/fortran/>  
<http://www.ibm.com/software/awdtools/ccompilers/>

Those web sites also contain information on contacting IBM if you have questions about the IBM compiler family or this document. You may also contact the compiler group by sending e-mail to [compinfo@ca.ibm.com](mailto:compinfo@ca.ibm.com).

### XL Compiler History

The IBM XL compilers are the result of a joint effort of an Advanced Technology project between the IBM Toronto Lab, the Yorktown Heights Research Lab, and other IBM facilities. This collaboration began in the 1980s with the first POWER chips and continued with the introduction of the XL Fortran for AIX compiler in 1989, and subsequently, the C and C++ for AIX compilers in 1991. Always available on the AIX operating system, the IBM XL compiler family was released for Linux operating systems running on iSeries and pSeries hardware in 2003. In 2004, the XL compiler family formally welcomed Mac OS X into the fold, after a popular and successful beta program in 2003.

Today IBM continues a more than twenty year investment in the development of POWER series and the PowerPC series chips. The compiler team's quality assurance

infrastructure is among the finest in the industry. Each compiler executes over a million C, C++, and Fortran test scenarios. IBM's premier customer service ensures fast turn around time and mandates timely solutions to customer problems.. Both the compiler and processor design teams are strongly committed to maximizing performance on all POWER and PowerPC processors.

## Multiple-Platform C/C++ and Fortran

IBM has developed extensive and mature industry-leading optimization technology that covers multiple platforms including AIX, Mac OS X, OS/400, z/OS, z/VM, Linux for iSeries and pSeries, and PASE (for AS400). The modular structure of these compilers ensures that optimizations and functionality are delivered on all platforms to all languages. Further, each product derives from a common code base, so features and optimizations are tested by multiple languages on multiple platforms.

The IBM compilers strive to maximize the performance of scientific, technical, and commercial applications on server platforms. Multiple operating system availability ensures cross-platform portability, augmented by standards compliance. The IBM compilers also focus on maximizing performance of key UNIX and SPEC benchmarks. The compilers are compliant with ISO C 1989, ISO C 1999, ISO C++ 1998, and Fortran 77/90/95 international programming language standards, as well as the OpenMP industry standard (OpenMP V1.0 for C/C++ and OpenMP V2.0 for XL Fortran). A high degree of source and option compatibility is maintained across the PowerPC platforms. Standards compliance ensures ease of code portability between multiple operating systems and hardware platforms.

The entire family of compilers is widely accepted among the scientific and technical communities. IBM compilers build key IBM products such as the AIX operating system, DB2, and Lotus Domino. Important ISVs and system houses around the world use IBM compilers to build their applications. These applications cover financial, government, petroleum, telecommunications, travel, retail, health care, and automotive industries. IBM compilers are also active in supercomputing efforts that use massive workstation server farms that demand the utmost in performance. These business interests help motivate the drive for performance of the XL compiler family.

### ***IBM Compiler Availability***

Table 1: IBM C/C++ and Fortran compilers available by platform as of June 2004

<b>Platform</b>	<b>C/C++</b>	<b>Fortran</b>
Apple (Mac OS X)	XL C/C++ V6.0 Advanced Edition for Mac OS X	XL Fortran Advanced Edition V8.1 for Mac OS X
pSeries (Linux) iSeries (Linux)	VisualAge C/C++ V6.0 for Linux	XL Fortran V8.1 for Linux

pSeries (AIX)	C for AIX V6.0, VisualAge C++ V6.0 for AIX	XL Fortran V8.1 for AIX
zSeries (z/OS, z/VM)	C/C++ for z/OS V1.5	IBM VS Fortran V2
iSeries (OS/400)	ILE C/C++ (feature of WebSphere Development Studio for iSeries)	N/A

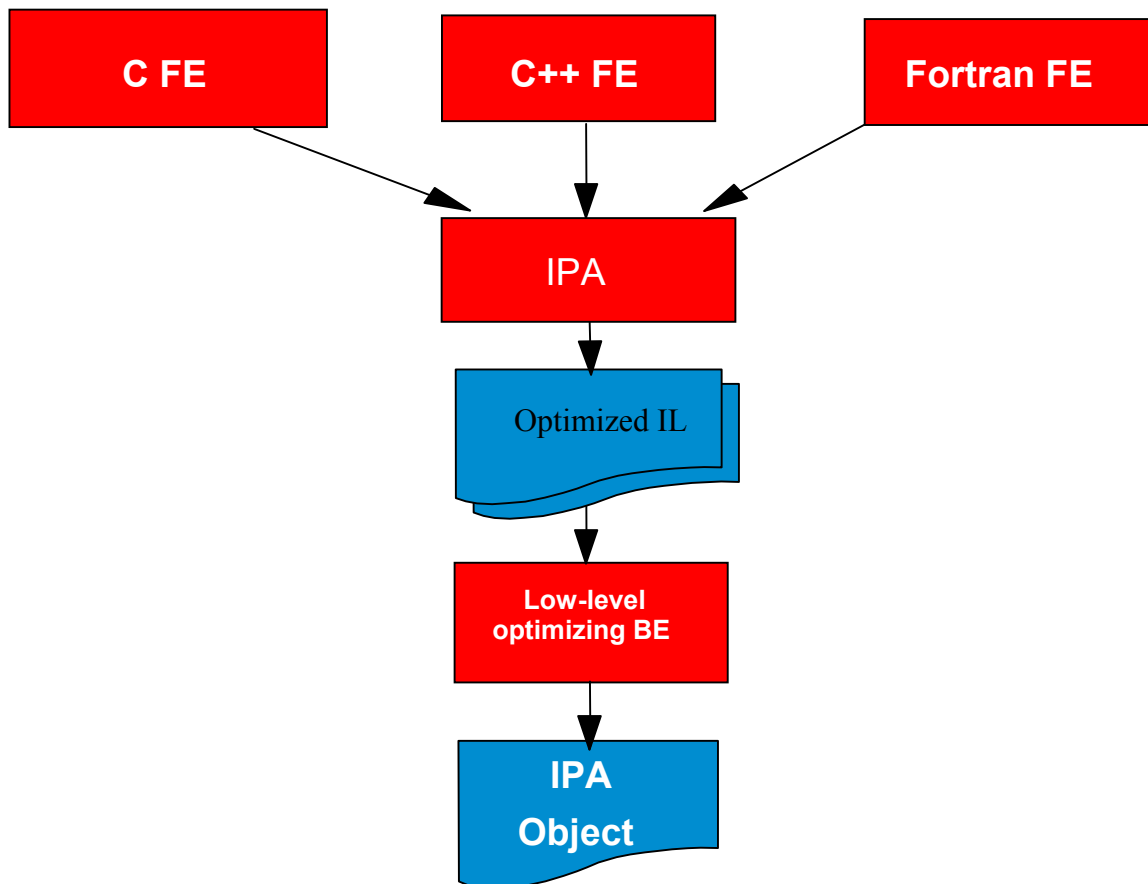
## IBM XL Compiler Architecture

The IBM compilers use a modular design, beginning with multiple language-parsing front-ends (FE) that generate a unified intermediate language (IL). This IL is processed by a low-level optimizing back-end (BE), thus enabling common optimizations that apply to all languages.

## ***We have O5!***

The highest optimization level currently available to the IBM compiler family is -O5. This level initiates aggressive optimization and the highest level of Interprocedural Analysis (IPA). The -O5 optimization process begins with the front-end emitting an intermediate language which is analyzed and transformed by IPA. The optimized intermediate language is then processed by the low-level optimizing back end for further optimization and object code creation. The following diagram illustrates the -O5 compile-step process:

Figure 1: Compile path with IPA used on the compile step

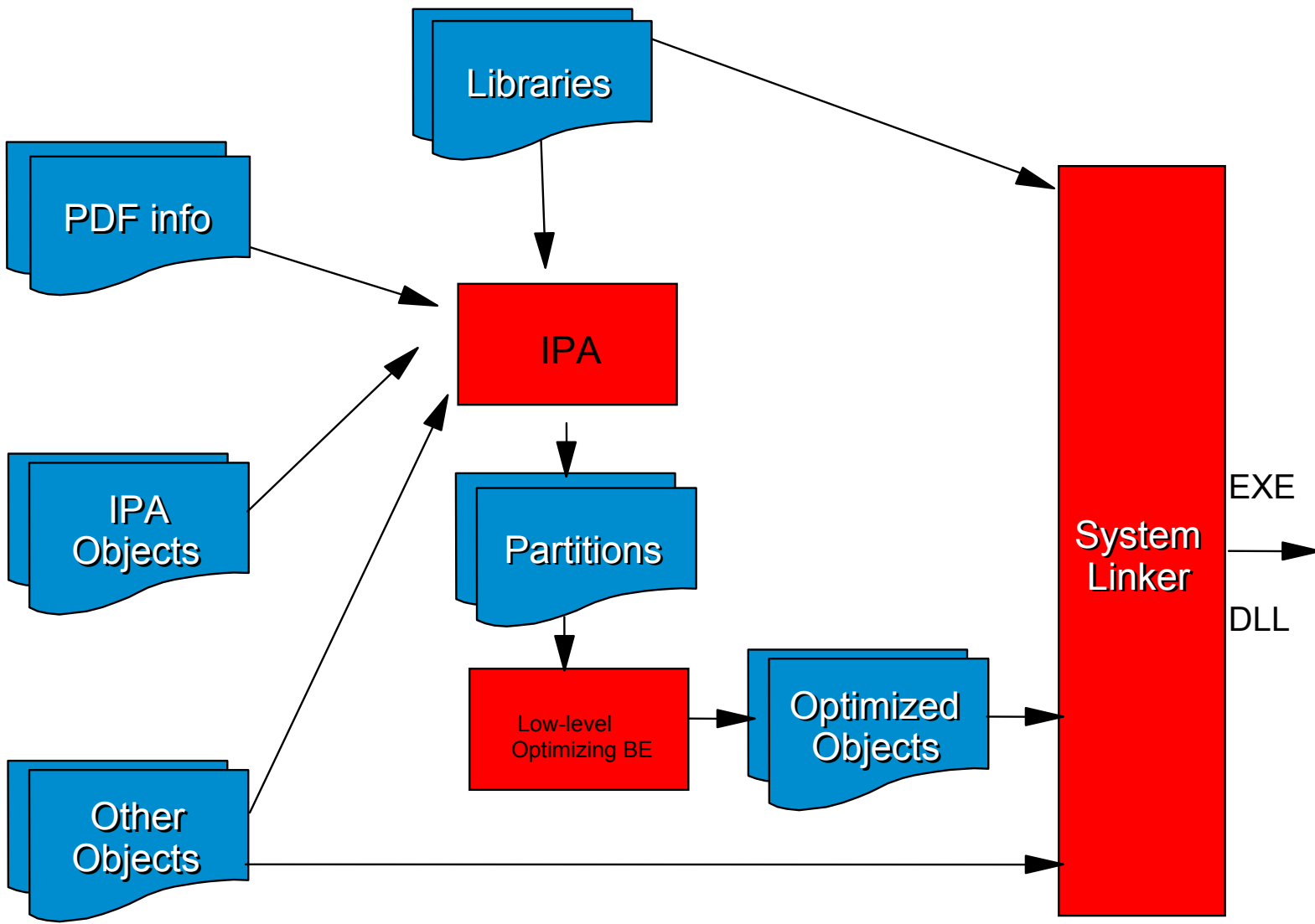


The IPA link-step optimization process combines many sources of information to achieve a view of the entire application. IPA performs the following tasks on the application link step:

- Examines libraries for all external symbols,
- Considers any profile-directed feedback (PDF) optimization information
- Processes IPA objects from the IPA compile step and normal non-IPA objects

The IPA optimizer separates the result into logically organized partitions. These partitions are subsequently processed by the low-level optimizing back end. The output of this process is an optimized set of objects which the system linker can use to create the final executable or shared library.

Figure 2: Link path when IPA is invoked



## Performance and IBM Chip Design

The key strength of the IBM XL compiler family is performance. The XL compilers are unmatched in their ability to optimize and tune code for execution on POWER-series and PowerPC platforms. The performance gain from years of compiler optimization experience can be seen in the release-to-release compiler improvements for the POWER4 processor.

The POWER4 chip is a 64-bit ISA PowerPC with a nominal clock speed of 1.3 GHz. It has 2 FXUs, 2 FPUs, 2 LSUs, 1 BRU, and 1 CRLU per core. Each core contains one L1 cache. The instruction L1 cache is 64K using direct write. The data L1 cache is 32K 2-way FIFO. There are two cores per chip, sharing with 3x480K 8-way pseudo-LRU L2 cache per chip. There are four chips and 4x32MB L3 cache per module. This leads to 1 to 4 modules or 8 to 32-way configurations.

The PowerPC 970 (G5 chip) was based on an early POWER5 design with reduced power consumption. This was achieved by reducing one processor, reducing the L2 cache from 1.5 MB to 0.5 MB and eliminating the L3 cache. However, it supports the AltiVec/VMX instruction set and runs at a core speed of 2GHz.

The AltiVec support in POWERPC 970 allowed for 32 128-bit registers, and 162 SIMD instructions. There are 4-way single precision vector FP instructions, and 4, 8, and 16 way vector fixed point instructions. The PowerPC 970 has two execution units: Permute and ALU. Permute handles vector permutation, and splat instructions. ALU handles vector add, subtract, floating-point math. The ALU has three subunits: simple fixed, complex fixed, and float.

These chips offer key architectural features for optimizations that the IBM XL compiler family exploits. There are very long floating point pipelines with no bypass. They can also enable aggressive out-of-order executions. The deep instruction fetch and decode pipelines permit microcoding and cracking. There are also 8 prefetch buffers, 16M page support, stream touch instructions, excellent dynamic branch prediction, new branch hints, and dedicated execution unit for CR logic.

Using informal measurements of SPEC CPU2000 benchmarks, the following table summarizes the improvements on AIX measuring the difference between the previous releases of IBM XL C/C++/Fortran and the current releases.

Table 2: Estimated compiler improvement on POWER4 chip on AIX

	<b>POWER4 with XL C/C++ v5.0 and XL Fortran V7.1</b>	<b>POWER4 with XL C/C++ v6.0 and XL Fortran V8.1</b>	<b>Change</b>
<b>SPECint2000</b>	<b>650</b>	<b>790</b>	<b>21.5%</b>
<b>SPECfp2000</b>	<b>967</b>	<b>1098</b>	<b>13.5%</b>

Specific POWER4 optimizations were exploited to make these numbers possible:

- Architecture-neutral and architecture-specific code paths
- Precise machine model for scheduling
- New loop transformations for deep pipelines
- New aggressive branch optimizations
- Optimized usage of hardware-expanded instructions
- Modulo scheduled loops

There were also enhanced profile-directed optimizations and improved whole program optimizations for locality and parallelism. These improved the handling of loop nests with calls. Inlining and cloning were tuned to loop optimization. Array data flow handling was also improved.

The same compiler code base is also built on SuSE, and Red Hat Linux distributions for pSeries. Comparing the latest IBM XL C/C++/Fortran compilers (VisualAge C++ V6.0 and XL FORTRAN V8.1 for Linux) versus comparable V3.2 GNU compilers, SPECint2000 is estimated to be 23% faster while SPECfp2000 is estimated to be 60% faster. These results are not derived using the formal SPEC benchmark process.

Similar informal measurement of the SPEC benchmark not done using the specified SPEC processes was performed on Mac OS X. The IBM C/C++/Fortran compilers (VisualAge C++ V6.0 and XL Fortran V8.1 for Mac OS X) are about 12% faster for SPECint2000 compared to GCC 3.3 (the native GCC on MAC OS X) and about 35.8% faster for SPECfp2000.

## Language Standards

The XL compiler team maintains membership in the ISO C and C++ committee representing Canada and IBM. This allows IBM to help drive the strategic direction of the languages as standards evolve, and deliver the changes as they become ratified. The

compiler team is also a member of J3, the primary Fortran 2003 development body. Both C/C++ and Fortran teams have membership in the OpenMP consortium.

The main changes to the C++ standard are the *Technical Corrigendum 1* which details ongoing work in defect reports collected worldwide. These TC1 items combine to form the 2003 C++ standard. Additional updates come in the form of the *C++ Library Extensions Technical Report* in 2001. These reports proposed extensions to the C++ library. Although not official as of this paper (June 2004), it is the intention of most vendors to implement these library extensions which will add useful libraries to the Standard library. These additions include the hash library, smart pointers, template metaprogramming utilities, and special math functions that are useful for research purposes. Finally, a further report named the *Technical Report on C++ Performance* was published in 2003 that is used internally by the development team to improve C++ compiler performance.

The evolution of the C and C++ languages has consistently driven the standards bodies to formalize the extensions proposed by the software community. The C language has extensions for embedded programming. There is extension to the C and C++ language to support UTF-16, and native decimal arithmetic support. There is an effort by Microsoft to define a new dialect of C++ to standardize C# in the marketplace. The C++ Evolution Working group endeavors to meet customer demand for third-party, open-source libraries. This working group is in the process of standardizing shared library behavior, thread and garbage collection, and changes to templates.

IBM compiler team members also participate in the GNU Committee which meets annually to examine the direction of the GNU compilers and tool chain. IBM is also currently active on the SPEC2004 CPU search committee which selects new benchmarks for the next generation of hardware and compiler performance test. When the new SPEC2004 benchmarks are released, they will define the direction that all vendor hardware and compilers will take for many years.

Ratification of the working draft of Fortran 2003 is expected by the end of 2004. XL Fortran 8.1 already contains a subset of important Fortran 2003 draft standard features.

The IBM family of compilers is also involved in parallel computing. The compilers implement the OpenMP standard. The latest OpenMP standard Version 2.0 for C and C++ aligns with the Fortran OpenMP Version 2.0 API and clarifies ambiguities in the original specification. The major new features in OpenMP 2.0 are:

- COPYPRIVATE for broadcast of sequential reads
- Portable timing routines
- Control over the number of threads for multilevel parallelism
- Simplified re-privatization rules

## ***IBM and GNU***

While ensuring that the IBM compiler family has always offered diverse functionality and excellent performance to their users, IBM also has a long standing interest in the development of the GNU compiler. IBM contributes to GCC development both financially and technically, including large parts of the original GCC code for RS/6000, and the 64-bit retarget for Linux on PowerPC. Further technical contributions include an instruction scheduler, as well as the rights to the graph coloring register allocation patent for use in GCC. Today IBM still seeks to make contributions to the GNU effort in the areas of Single Static Assignment (SSA) optimizations, auto-vectorization, and loop optimization. With these additions, IBM and many others help the GNU compiler grow, while GNU still maintains autonomy, and forges ahead in its own direction for language development and optimization. Consequently, users have the unique opportunity to choose between the GNU rapid adoption of experimental new language features, and the IBM compiler family's drive for optimization to make the PowerPC processor the fastest possible, coupled with premium customer support. However, this picture is now changing as IBM compilers evolve, adopting most of the GNU language features as well as achieving near complete binary compatibility with GNU compilers.

A major enhancement in the latest release of IBM XL C/C++ is the staged implementation of GNU V3.2 language extensions. This enhancement was delivered as source and binary compatibility to enable high degree of interoperability with specific GNU compilers. The intent of this GNU compatibility project is to phase in the support in stages:

1. Enable compilation of POSIX compliant source on Linux using our compiler and the GNU headers
2. Add in the most common feature and language extensions
3. Implement full support for enough of the extensions to compile the Linux kernel and most popular Linux applications

The second phase was achieved in Version 6 of the IBM C/C++ compilers, so that the IBM compilers have most of the common GNU language extensions.

Even more innovative is the guarantee for GNU interoperability with IBM C/C++ compilers achieved in the IBM C/C++ V6.0 compilers. The binary compatibility promise is based on the need to reuse the large repository of GNU-built libraries without the need to rebuild. The common-vendor C++ ABI permits a limited form of cross-vendor C and C++ compatibility by specifying the name mangling, object model, and exception handling mechanisms. However, it does not guarantee complete cross-vendor binary compatibility because of differences acknowledged by the C++ ABI. These undefined areas will affect the degree of interoperability of the final product.

The IBM compilers also have a close affinity with third party tool chains, and seek to adapt to GNU tools such as gdb and asm, Apple tools such as Project Builder, XCode, and Apple Performance tools like the Shark. There are also specific AIX and Linux performance tools which can identify performance bottlenecks.

Together, these compatibility capabilities permit porting of most Linux code built with the IBM compilers as well as building the Linux and the AIX kernel.

## **Summary**

The IBM compiler family offers premier innovation and capability on the AIX, Linux, and Mac OS X PowerPC platforms as well as zSeries and iSeries platforms. The support on multiple PowerPC chips and platforms enables easy porting of applications and optimizations that are tailored to PowerPC hardware. IBM's reputation for providing stable compilers with premium customer support allows large critical applications to rely on the XL compiler family. The flexibility of the compilers allows you to control the type and the depth of optimization analysis through compiler options that specify the level and kinds of optimization best suited for your application. The source and binary compatibility with the GNU compilers give you access to the vast array of Open Source software and libraries. Support of the leading standards ensures compliance for the future.

IBM's long history of compiler development gives you control of mature industry-leading optimization technology such as Interprocedural Analysis (IPA), and a unique set of PowerPC optimizations that fully exploit the hardware architecture's capabilities. This optimization strength combines with the IBM compiler family's robustness, capability, and standards conformance on multiple hardware and operating system platforms to produce a product set unmatched in the industry.

## ***Trial Versions and Purchasing***

You can download trial versions of the IBM compilers for Mac OS X, Linux, and AIX at these IBM web sites:

<http://www.ibm.com/software/awdtools/fortran/xlfortran>

<http://www.ibm.com/software/awdtools/ccompilers>

Purchasing information for IBM compilers is also available on those web sites. You can also purchase the Mac OS X versions of the XL compilers from Absoft© Corporation, on IBM OEM partner, at <http://www.absoft.com>.

## ***Contacting IBM***

IBM welcomes your comments. You can send them to [compinfo@ca.ibm.com](mailto:compinfo@ca.ibm.com) or mail them to this address:

XL Compiler Development  
Department SQT  
Application Development Technology Centre  
Software Division Toronto Laboratory IBM Canada Ltd.  
8200 Warden Avenue  
Markham, Ontario  
Canada – L6G 1C7

***Copyright Notice***

© Copyright IBM Corp. 2004. All Rights Reserved.

IBM is trademark or registered trademark of International Business Machines Corporation in the U.S., other countries or both.