



# IBM XL Fortran compiler overview

## **December 2014**

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, AIX, Blue Gene, Blue Gene/L, Blue Gene/P, Blue Gene/Q, DB2, Domino, Lotus, POWER, POWER6, POWER7, POWER7+, POWER8, Power, PowerPC, Power Architecture, Power Systems, Power PC, z/OS, and z/VM 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](http://www.ibm.com/legal/copytrade.shtml).

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

© **Copyright IBM Corporation 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

## Overview

The IBM® XL Fortran compilers are full-featured Fortran language compilers available for AIX® and Linux on Power®, and Blue Gene®/Q. IBM XL Fortran supports the latest international programming language standards and industry specifications. The XL Fortran compilers also contain core optimization technologies available on other IBM hardware-based systems, such as z/OS® and z/VM®. More information about IBM XL Fortran is available at:

- <http://www.ibm.com/software/products/en/fortcompfami>
- <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=b10932b4-0edd-4e61-89f2-6e478ccba9aa>

---

## Current IBM XL Fortran releases

As of December 2014, the following IBM XL Fortran compiler products are available:

- IBM XL Fortran for Linux, V15.1.1
- IBM XL Fortran for AIX, V15.1
- IBM XL Fortran for Linux, V15.1
- IBM XL Fortran for AIX, V14.1
- IBM XL Fortran for Linux, V14.1
- IBM XL Fortran for Blue Gene/Q, V14.1
- IBM XL Fortran for AIX, V13.1
- IBM XL Fortran for Linux, V13.1



---

## IBM XL Fortran history

IBM XL Fortran was first released in 1989. From 1990, XL Fortran has been available on IBM Power Systems™ since the hardware's inception. Always available on the AIX operating system, XL Fortran was released for Linux operating systems in 2003. IBM announced the Blue Gene® program in December 1999, leading to the release of XL Fortran versions for Blue Gene®/L in 2006, Blue Gene®/P in 2007, and Blue Gene/Q in 2012.

The XL Fortran compiler contains industry-leading optimization technology that has continually evolved since the mid-1980s. The optimization technology in XL Fortran maximizes the performance of code executing on PowerPC® and POWER-series processors. The close relationship between IBM's compiler and chip design groups ensures that XL Fortran can take maximum advantage of IBM processor technology as it becomes available. Additionally, the compiler group has a unique opportunity to influence design decisions in the production of chips, like the POWER8™ chip, so that XL Fortran can exploit its fullest potential. IBM XL Fortran for AIX, V15.1 and IBM XL Fortran for Linux, V15.1 fully support POWER8 processors.

IBM XL Fortran has been and continues to be the Fortran compiler of choice on AIX. XL Fortran is used to measure system performance and announce benchmark results, such as the SPEC benchmark. IBM XL Fortran shares common compilation technology with the IBM XL C and XL C++ compilers, specifically key components for optimization and targeting specific architectures. The IBM XL Fortran compiler is used to build parts of other key IBM products such as AIX. XL Fortran is not only a compiler for large, broad-spectrum applications. Over the years, customers have come to rely on IBM XL Fortran's stability, versatility, and performance to build critical applications both large and small.



---

## Standards support

IBM XL Fortran for Linux, V15.1.1, IBM XL Fortran for AIX, V15.1, IBM XL Fortran for Linux, V15.1, and IBM XL Fortran for Blue Gene/Q, V14.1 fully implement the FORTRAN 77, Fortran 90, Fortran 95, and Fortran 2003 language standards, and partially implement the Fortran 2008 language standard. Further, XL Fortran for AIX, V15.1 and XL Fortran for Linux, V15.1 support a partial implementation of the OpenMP Fortran API Version 4.0 and a complete implementation of the OpenMP Fortran API Version 3.1, which gives users the power and versatility to develop portable SMP applications and to use task level parallelization and loop parallelization.

Different invocation commands, such as `xl`, `xl90`, `xl95`, `xl2003`, and `xl2008`, allow you to automatically supply default options for a particular Fortran standard. The compiler also implements a language-level compiler option, `-qlanglvl`, which can report non-conformant source constructs. Additionally, the IBM XL Fortran runtime library supports environment variables that control behavior for a particular standard, or report non-conformant usage while an application is executing.

IBM is a member of the ISO/IEC JTC1/SC22/WG5 Committee, INCITS PL22.3 (formerly J3) Fortran Technical Committee, and the OpenMP Language Architecture Review Board. These bodies are responsible for the Fortran language and the OpenMP API extension. IBM is thus in a position to understand and participate in the latest updates, clarifications, and recommendations to the Fortran standard and industry specifications.

---

## Available TS 29113 features

IBM XL Fortran for Linux, V15.1.1 supports the following selected language interoperability features from TS 29113:

- Assumed-rank objects
- Assumed-type objects
- Interoperable procedures with dummy arguments that have `ALLOCATABLE`, `OPTIONAL`, or `POINTER` attributes
- Interoperable variables in asynchronous communication
- Relaxed restrictions on the `C_F_POINTER`, `C_F_PROCPOINTER`, `C_FUNLOC`, and `C_LOC` module procedures from the intrinsic module `ISO_C_BINDING`
- The `ISO_Fortran_binding.h` header file
- The `RANK` intrinsic procedure

---

## Available Fortran 2008 standard features

IBM XL Fortran for Linux, V15.1.1 supports the following selected features of the Fortran 2008 standard:

- 64-bit integers
- `BACK=` arguments in `MAXLOC` and `MINLOC`
- Complex part designators
- Double colon separators (`::`) in `PROCEDURE` statements

- Enhancements to the **ISO\_FORTRAN\_ENV** intrinsic module including new constants and the **COMPILER\_VERSION** and **COMPILER\_OPTIONS** inquiry functions
- Extensions to the generic resolution rules
- **FINDLOC** intrinsic procedure
- Implied-shape arrays
- Improvements to the **ALLOCATE** statement, including adding the **MOLD=** specifier and copying the bounds of the source array
- Improvements to the **STOP** and **EXIT** statements
- Impure elemental procedures
- Internal procedures as actual arguments or procedure pointer targets
- The **STORAGE\_SIZE** inquiry intrinsic procedure
- Intrinsic procedures for manipulating bits through combined shifting, merging, masking, or shifting
- Language-level checking for new and obsolescent Fortran 2008 features
- Maximum standard array rank has been increased to 15
- New and improved intrinsic procedures
  - Error, gamma, and bit-counting intrinsics
  - The **EXECUTE\_COMMAND\_LINE** intrinsic
  - The **RADIX=** argument on the **SELECTED\_REAL\_KIND** intrinsic
  - Trigonometric and hyperbolic intrinsics
- New optional argument presence rules with respect to allocatable and pointer dummy arguments
- Pointer dummy argument enhancement
- Separate module subprograms
- Submodules
- The **C\_SIZE\_OF** inquiry function in the **ISO\_C\_BINDING** intrinsic module
- The declaration of multiple type-bound procedures in a single procedure statement
- The **BLOCK** construct
- The **CONTIGUOUS** attribute and **IS\_CONTIGUOUS** intrinsic function
- The **END** statement for internal and module subprograms
- The **ERROR STOP** statement
- The **MODULE** prefix specifier
- The **NEWUNIT=** specifier in the **OPEN** statement
- Type specification in the **FORALL** statement and construct
- Using the **TYPE** statement for declaring variables of intrinsic type

---

## Available Fortran 2003 standard features

IBM XL Fortran began delivering support for the Fortran 2003 standard as early as the V8.1 release. As of the V13.1 release, the XL Fortran compiler fully supports the Fortran 2003 standard.



---

## Language extensions

In addition to standards conformance, IBM XL Fortran features many common language extensions, as well as code-porting features. IBM XL Fortran implements IBM Fortran language extensions such as SAA Fortran, and a number of extensions added in VS Fortran. These extensions simplify porting existing Fortran code from other platforms to IBM XL Fortran compilation targets. Featured industry language extensions include:

- 128-bit floating-point data type
- Cray (integer) pointers
- Structure records
- Union maps
- **BYTE, STATIC, AUTOMATIC**
- **SIZEOF** intrinsic

IBM XL Fortran provides *xlftutility*, a Fortran 90 module that you can use in your applications to access operating system features such as system timers and process query functions without having to write the access code yourself. The Pthreads Library Module provides you with Fortran language interfaces to the AIX and Linux operating systems pthread libraries. You can use the *f\_pthread* module to parallelize and thread-safe your code.

A rich set of Power-based functionality further enhances the compiler. IBM XL Fortran includes a large number of intrinsic procedures and directives to give you source-level access to hardware-level operations such as cache control, data prefetching, and hardware data-manipulation instructions. The *xlfp\_util* module adds additional intrinsic functions that allow you to both control and query the hardware floating-point status and control register. The **VECTOR** intrinsic data type and dozens of intrinsic functions give you direct access to the powerful VMX, VSX (for AIX and Linux) and QPX (for Blue Gene/Q) vector instructions available in Power chips like the PowerPC 970, POWER6<sup>®</sup>, POWER7<sup>®</sup>, and POWER8.

IBM XL Fortran also includes many directives that assist in the optimization of your applications in a number of different ways, including:

- Directives that transform source constructs, such as **SUBSCRIPTORDER** and **COLLAPSE**, that you can apply to more efficiently use memory at execution time.
- Directives to guide the compiler in optimization transformations such as loop unrolling.
- Directives that inform the compiler that certain constructs, such as **DO** loops, have particular behaviors or attributes. This allows the optimizer to identify additional opportunities for optimization that may not be possible otherwise.

## OpenMP API Version 4.0 specification

Starting from IBM XL Fortran for AIX, V15.1, and IBM XL Fortran for Linux, V15.1, the XL Fortran compiler supports the following OpenMP 4.0 features:

- capture clause enhancements
- `OMP_DISPLAY_ENV` environment variable

## OpenMP API Version 3.1 specification

Starting from IBM XL Fortran for AIX, V14.1, IBM XL Fortran for Linux, V14.1, and IBM XL Fortran for Blue Gene/Q, V14.1, the XL compilers support the following OpenMP 3.1 updates:

- Adds FINAL and MERGEABLE clauses to the TASK construct to support optimization.
- Adds the TASKYIELD construct to allow users to specify where in the program can perform task switching.
- Adds the `omp_in_final` runtime library routine to support specialization of final task regions.
- Extends the ATOMIC construct to include READ, WRITE, and CAPTURE forms; adds the UPDATE clause to apply the existing form of the ATOMIC construct.
- Allows dummy arguments with the INTENT(IN) attribute to be specified on the FIRSTPRIVATE clause.
- Allows unallocated allocatable arrays to be specified on the COPYIN clause.
- Allows Fortran 90 pointers to be specified on the FIRSTPRIVATE clause.
- Adds the OMP\_PROC\_BIND environment variable to control whether OpenMP threads are allowed to move between processors.
- Extends the OMP\_NUM\_THREADS environment variable to specify the number of threads to use for nested parallel regions.

---

## Support for POWER8 processors

### Mathematical Acceleration Subsystem (MASS) libraries for POWER8 processors

- **Scalar libraries.** The MASS library interfaces include the following features:
  - The scalar functions have generic interfaces that can be called with **REAL(4)** or **REAL(8)** arguments.
  - The scalar functions are marked pure. You can call them from pure procedures.
  - The scalar functions are marked elemental. You can call them with an array argument and apply them to all the array elements.
  - The intent of the argument is specified to assist in compiler error checking.
- **Vector libraries.** The vector MASS library **libmassvp8.a** contains vector procedures that have been tuned for the POWER8 architecture. The MASS vector library interfaces include the following features:
  - The vector functions have generic interfaces that can be called with **REAL(4)** or **REAL(8)** arguments.
  - The vector functions are marked pure. You can call them from pure procedures.
  - The intent of the argument is specified to assist in compiler error checking.
- **SIMD libraries.** The MASS SIMD library **libmass\_simdp8.a** contains an accelerated set of frequently used math intrinsic procedures that provide improved performance over the corresponding standard system library procedures. The MASS SIMD library interfaces include the following features:
  - The SIMD functions are marked pure. You can call them from pure procedures.
  - The intent of the argument is specified to assist in compiler error checking.

### Compiler options for POWER8 processors

- The **-qarch** compiler option specifies the processor architecture for which code is generated. The **-qtune** compiler option tunes instruction selection, scheduling, and other architecture-dependent performance enhancements to run best on a specific hardware architecture.
- The **-qarch=pwr8** suboption produces object code containing instructions that will run on the POWER8 hardware platforms. With the **-qtune=pwr8** suboption, optimizations are tuned for the POWER8 hardware platforms.

**POWER8 hardware directives and intrinsics.** New hardware directives and intrinsics are added to support the following POWER8 processor features:

- POWER8 intrinsics for vector processing
- POWER8 cryptography intrinsics
- POWER8 transactional memory intrinsics
- POWER8 prefetch directives
- POWER8 prefetch intrinsic procedures



---

## Support for Blue Gene/Q architecture

### Quad Processing eXtension data types and intrinsic procedures

The XL Fortran compiler for Blue Gene/Q supports the Quad Processing eXtension (QPX) instruction set in the PowerPC processor. New data types and intrinsic procedures are introduced to support the QPX instructions. With the QPX intrinsic procedures, you can efficiently manipulate vector operations in your application.

### Mathematical Acceleration Subsystem (MASS) libraries

- **Vector libraries.** The vector MASS library **libmassv.a** contains vector procedures that have been tuned for the Blue Gene/Q architecture. These procedures are used in 64-bit mode.
- **SIMD libraries.** The MASS SIMD library **libmass\_simd.a** contains an accelerated set of frequently used math intrinsic procedures that provide improved performance over the corresponding standard system library procedures.

### -qarch=qp and -qtune=qp options

The **-qarch=qp** compiler option produces object code that runs on Blue Gene/Q platforms. The **-qtune=qp** compiler option specifies that optimizations are tuned for Blue Gene/Q platforms. They are enabled by default.



---

## What's new in IBM XL Fortran for Linux, V15.1.1

IBM XL Fortran for Linux, V15.1.1 includes two Fortran compilers: IBM XL Fortran for Linux, V15.1 for big endian distributions and IBM XL Fortran for Linux, V15.1.1 for little endian distributions. The following operating systems and new features are supported.

### Operating system support

IBM XL Fortran for Linux, V15.1 for big endian distributions supports the following operating systems:

- SUSE Linux Enterprise Server 11 Service Pack 2 (SLES 11 SP2)
- Red Hat Enterprise Linux 6.4 (RHEL 6.4) or later
- Red Hat Enterprise Linux 7.0 (RHEL 7.0) or later

IBM XL Fortran for Linux, V15.1.1 for little endian distributions supports the following operating systems:

**Note:** XL Fortran packages two Linux compilers in a single product. For XL Fortran for Linux V15.1.1 (little endian distributions), the following operating systems are only on IBM POWER8.

- Ubuntu 14.04 or Ubuntu 14.10 or later
- SUSE Linux Enterprise Server 12 (SLES 12) or later

### Migrating from big endian Linux to little endian Linux

IBM XL Fortran for Linux, V15.1.1 for little endian distributions is compatible with versions of the compiler running on the POWER8 big endian systems. There are, however, some differences to consider.

- To help migrate programs from big-endian systems, you can use the **-qaltivec** option to toggle the vector element sequence in registers to big-endian or little-endian element order.

For more information, see “**Program migration from big-endian systems**” in the *XL Fortran for Linux, V15.1.1 Optimization and Programming Guide*.

### Relaxed restrictions on the C\_F\_POINTER, C\_F\_PROCPOINTER, C\_FUNLOC, and C\_LOC module procedures from the intrinsic module ISO\_C\_BINDING

Some restrictions on module procedure C\_F\_POINTER, C\_F\_PROCPOINTER, C\_FUNLOC, and C\_LOC of the intrinsic module ISO\_C\_BINDING that were put forward by earlier Fortran language standards are removed according to TS29113.

For more information, see “C\_F\_POINTER”, “C\_F\_PROCPOINTER”, “C\_FUNLOC”, and “C\_LOC” in the *XL Fortran for Linux, V15.1.1 (Little Endian Distributions) Language Reference*.

### Passing scalar actual arguments that correspond to assumed-size dummy arguments of assumed-type

You can pass a scalar as the actual argument that corresponds to an assumed-size dummy argument of assumed-type. An example is provided in “Assumed-size arrays” in the *XL Fortran for Linux, V15.1.1 (Little Endian Distributions) Language Reference*.

**Inquiry intrinsic procedure: STORAGE\_SIZE (A, KIND) (Fortran 2008)**

The **STORAGE\_SIZE** procedure returns the storage size in bits for an element of an array that has the dynamic type and type parameters of A. For more information, see “STORAGE\_SIZE (A, KIND) (Fortran 2008)”, in the *XL Fortran for Linux, V15.1.1 (Little Endian Distributions) Language Reference*.



---

## Porting code to IBM XL Fortran

IBM XL Fortran has several options to assist you in porting Fortran code from other systems. The **-qport** option includes suboptions that allow you to toggle compiler behaviors that can assist in porting non-standard code. There are many other compiler options that can assist with code porting or reproducing results from other platforms, including such functionality as:

- Altering default data type sizes and data type interaction rules
- Controlling certain program flow behaviors
- Influencing what types of comment-form and conditional compilation directives are accepted
- Altering the naming scheme for global names seen by the linker
- Specifying floating-point manipulation rules such as rounding mode
- Altering the behavior of the IBM XL Fortran runtime through environment variables



---

## Utilization tracking and reporting

**Note:** The utilization tracking and reporting feature is only for big endian systems.

On AIX and Linux, you can use the utilization reporting tool to help determine whether the use of the compiler by your organization matches your compiler license entitlements. When enabled, each invocation of the compiler is recorded in a compiler utilization file. The utilization reporting tool can then be used to generate a report of the overall usage of the compiler within your organization. In particular, the report indicates whether the compiler usage complies with the number of Concurrent User licenses that you have acquired.



---

## Programming with IBM XL Fortran

IBM XL Fortran supports a traditional command-line development toolset common on many platforms and works well with standard build tools such as *make*.

The binary code IBM XL Fortran produces is compatible with IBM XL C/C++ for AIX, IBM XL C/C++ for Linux, and IBM XL C/C++ for Blue Gene/Q. This gives you the flexibility of coding your application with interlanguage calls where appropriate. IBM XL Fortran provides several compilation options to assist application development in a mixed-language environment. The **BIND(C)** Fortran 2003 standard feature supported by IBM XL Fortran allows you to program interlanguage calls in a standards-conforming portable manner.

---

## Debugging IBM XL Fortran applications

You can debug applications built with IBM XL Fortran using debuggers such as the standard AIX dbx debugger and Linux gdb debugger. Various third parties have marketed software products for debugging IBM XL Fortran applications, including the TotalView debugger available from Rogue Wave Software and the DDT debugger available from Allinea.

**Note:** References to these third party products are provided for your convenience only. Check with the vendor for more details.

IBM XL Fortran for AIX, V15.1, IBM XL Fortran for Linux, V15.1, IBM XL Fortran for Linux, V15.1.1, and IBM XL Fortran for Blue Gene/Q, V14.1 support debugging applications that were optimized using the **-O2** option. XL Fortran lets you balance performance and debugging capability by using a multilevel **-g** option. The IBM XL Fortran runtime library supplies traceback capabilities through procedures you can call from your application. The runtime library contains default signal and exception-handling support, but is flexible enough to allow you to override these with your own handlers. The compiler also supplies options to assist in debugging applications.

---

## Program analysis

IBM XL Fortran produces binary code that can be used with system standard performance analysis tools such as *gprof* or, on AIX, *tprof*. Using the **-p** compiler option allows your application to emit the information that some profilers require.

The compiler can assist you in analyzing your program in several ways. A listing facility can show you your source code with embedded diagnostic messages in addition to seeing them on the screen. Other options allow you to select the severity of the messages reported, or filter out specific messages completely.

---

## IBM XL Fortran runtime library

The IBM XL Fortran runtime library is an integral part of the product. The runtime is available in both threadsafe and non-threadsafe variations, allowing you to choose the library best suited to your application's performance requirements. Many environment variables are recognized by the runtime and allow you to direct the runtime behavior. These include variables for language conformance and porting issues, input-output characteristics, error reporting, and multithreading

behaviors. The XL Fortran run time library can be redistributed with your XL Fortran applications to your customers. Alternatively, the XL Fortran runtime library can be downloaded by users of your applications.

The IBM XL Fortran runtime library is available for download at:

<http://www.ibm.com/support/docview.wss?rs=43&uid=swg21156900>

---

## Optimization capabilities

One of the key strengths of IBM XL Fortran is optimization. The compiler offers the benefits of optimization technology that has been evolving at IBM since the late 1980s, combining extensive hardware knowledge with a comprehensive understanding of compiler technology and what users look for in a compiler when building end-user applications. The optimization can decrease execution time and make your applications run faster, producing code that is highly tuned for execution on PowerPC and Power platforms. Improving optimization is a key goal of the IBM compiler team, and one that will continue to be a major focus with each iteration of the IBM XL Fortran compiler.

The optimizer includes five base optimization levels; **-O0**, **-O2**, **-O3**, **-O4**, and **-O5**. These levels allow you to choose from minimal optimization to intense program analysis that provides benefits even across programming languages. Optimization analyses range from local basic block to subprogram to file-level to whole-program analysis. The higher the optimization level, the more intense the program analysis becomes as increasingly sophisticated optimization techniques are applied to your code.

At any optimization level, the compiler performs transformations that result in performance improvements, while still executing your code the way it was written. At higher levels, the compiler can trade numeric precision for execution speed. If this effect is not desired, you can specify compiler options such as **-qstrict** to prevent such trade-offs. Other options such as **-qsmallstack** or **-qcompact** allow you to bias optimization decisions in favor of smaller stack space or program size.

The IBM XL Fortran compiler does not limit your optimization choices unnecessarily. All of the optimization capabilities, including those discussed above, can be combined. You choose the levels and types of optimizations best suited to your application and build constraints, putting ultimate control of how your application builds and runs firmly in your hands.

For more information on optimization, see the *Code optimization with the IBM XL Compilers* whitepaper and the IBM XL Fortran *Optimization and Programming Guide*.





---

## Documentation

IBM XL Fortran supplies comprehensive documentation describing the functionality and capabilities of the compiler.

- The *Compiler Reference* describes all aspects of the compilation process including: compiling, linking, and executing your application. This reference also details the behavior of all command line options available for compilation, and environment variables that you can use to control program execution.
- The *Language Reference* documents the entire Fortran language, including all supported standards. This reference also details the many useful extensions IBM has added to Fortran. It also identifies certain language constructs as being from a particular language revision, such as Fortran 2003, or as an IBM extension.
- The *Optimization and Programming Guide* discusses common programming tasks with IBM XL Fortran, such as OpenMP SMP programming. This guide also details the optimization process to ensure you take advantage of all the optimization capabilities the compiler has to offer, whether you are a beginner, or already have some experience with IBM XL compiler optimization capabilities.
- Other documentation shipped with IBM XL Fortran includes the *Installation Guide* and *Getting Started With XL Fortran*.
- Man pages for all utilities shipped with the compiler, as well as compiler invocation commands are also included.

IBM XL Fortran compilers product documentation is available online. You can access it from the following library page links:

- IBM XL Fortran for AIX: <http://www.ibm.com/support/docview.wss?uid=swg27036673>
- IBM XL Fortran for Linux: <http://www.ibm.com/support/docview.wss?uid=swg27036672>
- IBM XL Fortran for Blue Gene/Q: <http://www.ibm.com/support/docview.wss?uid=swg27036671>

The IBM XL Fortran compilers also include man pages for all utilities and compiler invocation commands.

An extensive collection of technical materials, trials and demos, support information, and features and benefits of IBM XL Fortran can be found at the following URL:

<http://www.ibm.com/software/products/en/fortcompfami>



---

## Premier customer service

IBM XL Fortran comes with IBM's premier service and support. The IBM Service and Support team is dedicated to providing you with responsive platform and cross-platform software support. For complex or code-related problems, IBM employs specialized service teams with access to compiler development experts. The vision of IBM Service and Support is to achieve a level of support excellence that exceeds customer expectations and differentiates IBM in the marketplace. You will always have access to the right level of IBM expertise when you need it. For more information about the latest updates about supported IBM XL Fortran compilers, see <http://www.ibm.com/support/docview.wss?uid=swg21156900> and [http://www.ibm.com/support/entry/portal/ctt/Software/Rational/Fortran\\_Compilers](http://www.ibm.com/support/entry/portal/ctt/Software/Rational/Fortran_Compilers).



---

## Summary

The IBM XL Fortran compilers conform to Fortran standards and include many common industry language extensions and features. They give you the flexibility to easily port your code to the IBM XL Fortran family of compilers on supported platforms. The compilation and optimization technology in IBM XL Fortran is designed to deliver the best performing applications on the Power Architecture®. IBM XL Fortran is IBM's premier Fortran compiler, and has become synonymous with performance and quality for small and large customers around the world.



---

## Trial versions and purchasing

The following link contains the XL Fortran compiler product web pages. To download the trial versions of the compilers, navigate to the download page.

<http://www.ibm.com/software/products/en/fortcompfami>

Information on how to buy IBM XL Fortran is also available at the above web site.





---

## Contacting IBM

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







Printed in USA