

Transition to the Intel® Fortran Compiler

Barbara Perz

Intel Technical Consulting Engineer

October 19, 2023

The Intel logo is located in the bottom left corner of the slide. It consists of a stylized graphic of four overlapping squares in shades of blue, arranged in a descending staircase pattern. To the right of this graphic is the word "intel" in a lowercase, white, sans-serif font, followed by a registered trademark symbol (®).

intel®

Our Fortran Compilers 2023 and Beyond



Agenda

- Intel Fortran Compiler 2023 Overview
- Getting Started and Porting From IFORT to IFX
- Q&A

Celebrating IFX Feature Completion!



Our Goal:
IFX feature parity with IFORT
with 2023.0.0

ACCOMPLISHED!

Full Fortran 2018

IFORT directives, options,
& behaviors

Legacy DEC extensions

Microsoft* Visual Studio support

AND MORE!

Acceleration with Intel GPUs

Our Fortran Solution 2023

Intel® Fortran Compiler (ifx)

Our Fortran compiler tuned for

4th Gen Intel® Xeon® Scalable processors (code-named Sapphire Rapids), Intel® Xeon® CPU Max Series (code-named Sapphire Rapids HBM) and the Intel® Data Center GPU Max Series (code-named Ponte Vecchio)

Fortran Language Feature parity with IFORT
With Comparable Performance

Intel® Fortran Compiler Classic (ifort)

Dependable, proven features
and performance for pre-2023 Intel CPU products

Because you need advanced Fortran language features and the absolute best performance for your applications on Intel solutions

CHOICE! Continuity! Features! Performance!

IFX: Driving a New Era in Accelerated Computing

IFX: ALL that you you like in IFORT *PLUS*

- OpenMP* 5.x Standards, offload to Intel GPUs from Fortran
An open, portable Standard maintains your investment
Best in class OpenMP features and support
- Fortran 2018 DO CONCURRENT supports automatic offload to Intel GPUs

Protecting your Fortran Investment

Same Fortran parser/analyzer you know and love from IFORT

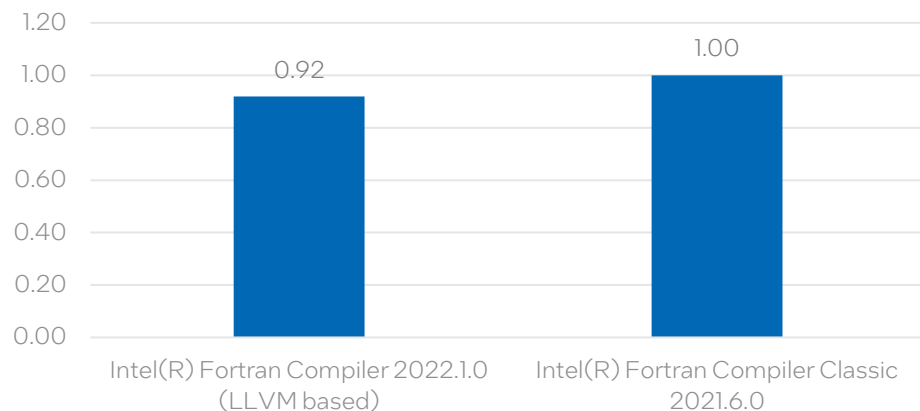
- Supports legacy DEC extensions, *all F2018*, ifort directives and features
- The majority of IFORT compiler directives and options you have used for years. And Microsoft Visual Studio* integration for Windows*

- *Binary compatible, mix and match ifx and ifort*

Intel® Fortran Compilers Build Time Performance on Linux*

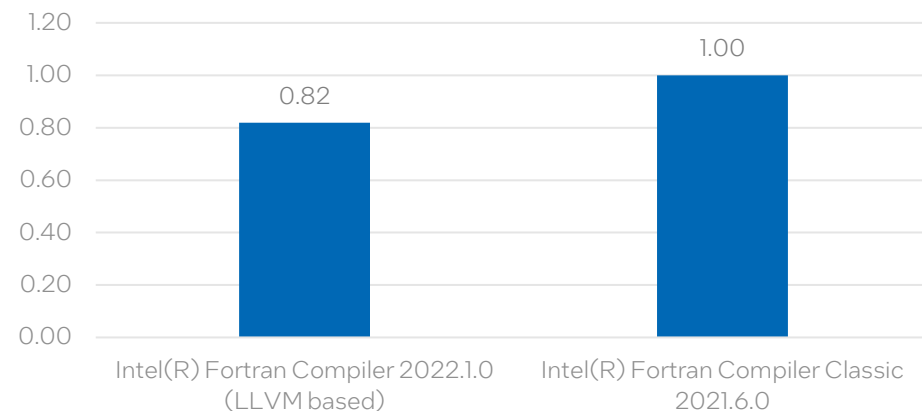
Build time performance advantage relative between Intel compilers on Intel® Core™ i7-8700K Processor

Build Time Performance for SPECrate 2017 Integer Suite @ 64 Bit (est.)
(Lower is Better)



Estimated: Built time measurement of the geometric mean of the Fortran workloads from the SPECrate* 2017 Integer suite

Build Time Performance for SPECrate 2017 Floating Point Suite @ 64 Bit (est.)
(Lower is Better)



Estimated: Built time measurement of the geometric mean of the Fortran workloads from the SPECrate* 2017 Floating Point suite

Performance varies by use, configuration, and other factors. Learn more at www.intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

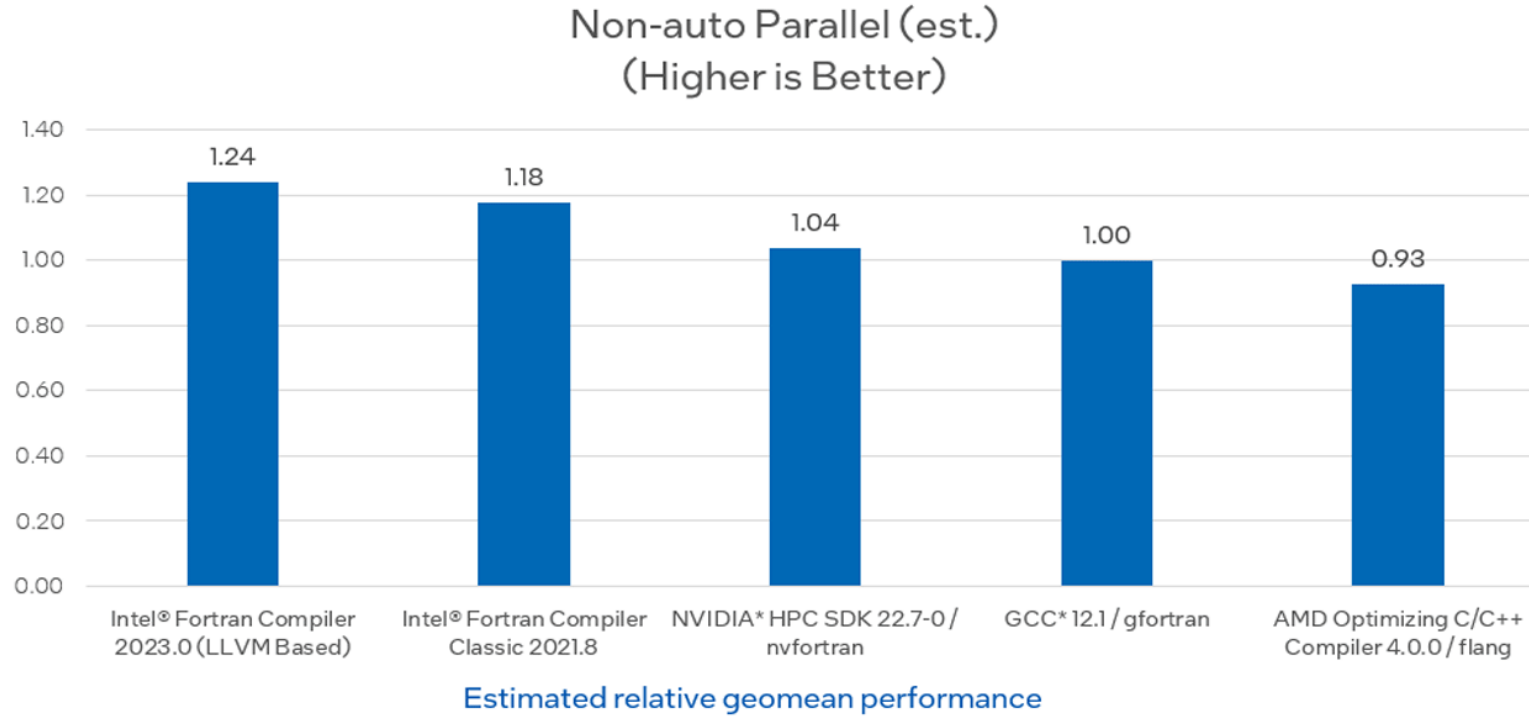
Your costs and results may vary. Intel technologies may require enabled hardware, software, or service activation.

More information on the SPEC benchmarks can be found at: <http://www.spec.org>

Configuration: Testing by Intel as of Mar 16, 2022. Configuration: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 16G x2 DDR4 2666. Red Hat Enterprise Linux release 8.0 (Ootpa), 4.18.0-80.el8.x86_64. Software: Intel(R) Fortran Compiler for applications running on Intel(R) 64, Version 2022.1.0 Build 20220316. Intel(R) Fortran Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version 2021.6.0 Build 20220226_000000. Compiler switches: Intel(R) 64 Compiler Classic: ifort -O2 -xCORE-AVX512, Intel(R) Fortran Compiler: ifx -O2 -xCORE-AVX512.

Intel® Fortran Compiler Boosts Application Performance on Linux*

Performance Advantage Measured by Polyhedron* Fortran Benchmark on Intel® Core™ i9-12900K Processor



Testing Date: Performance results are based on testing by Intel as of December 2, 2022 and may not reflect all publicly available security updates.

Configuration Details and Workload Setup: Intel® Core™ i9-12900K CPU @ 5.2GHz, i9-12900K, 16G x2 DDR5 4800. Software: Intel® Fortran Compiler for applications running on Intel® 64, Version 2023.0.0 Build 20221201, Intel® Fortran Compiler Classic for applications running on Intel 64, Version 2021.8.0 Build 20221119_000000, GCC 12.1.0 / gfortran, AMD* Optimizing C/C++ Compiler 4.0.0 / flang - AMD clang version 14.0.6 (CLANG: AOCC_4.0.0-Build#434 2022_10_28) (based on LLVM Mirror.Version.14.0.6), NVIDIA* HPC SDK 22.7.0 / nvfortran. Red Hat Enterprise* Linux release 8.4 (Ootpa), 4.18.0-372.9.1.el8.x86_64. Non-auto Parallel compiler switches: Intel® Fortran Compiler: -Ofast -xalderlake -fito -nostandard-realloc-lhs. Intel® Fortran Compiler Classic: -fast -xCORE-AVX2 -nostandard-realloc-lhs. GCC / gfortran: -Ofast -mfpmath=sse -fito -march=alderlake -funroll-loops. AMD* Optimizing C/C++ Compiler / flang: compile: -O3 -ffast-math -march=znver3 -fveclib=AMDLIBM -fito -mllvm -unroll-aggressive -mllvm -unroll-threshold=500; link: -WI, -mllvm -WI, -inline-recursion=4 -WI, -mllvm -WI, -lsrc-in-nested-loop -WI, -mllvm -WI, -enable-iv-split -fito -WI, -mllvm -WI, -region-vectorize -WI, -mllvm -WI, -function-specialize -WI, -mllvm -WI, -align-all-nofallthru-blocks=6 -WI, -mllvm -WI, -reduce-array-computations=3 -O3 -ffast-math -march=znver3 -fveclib=AMDLIBM -fito -lamdlibm -lflang -lamdlibm -lm. NVIDIA HPC SDK / nvfortran: -fast -Mipa=fast,inline -Mallocatable=O3 -Mfprelaxed -Mstack_arrays.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more at www.intel.com/PerformanceIndex. Your costs and results may vary.

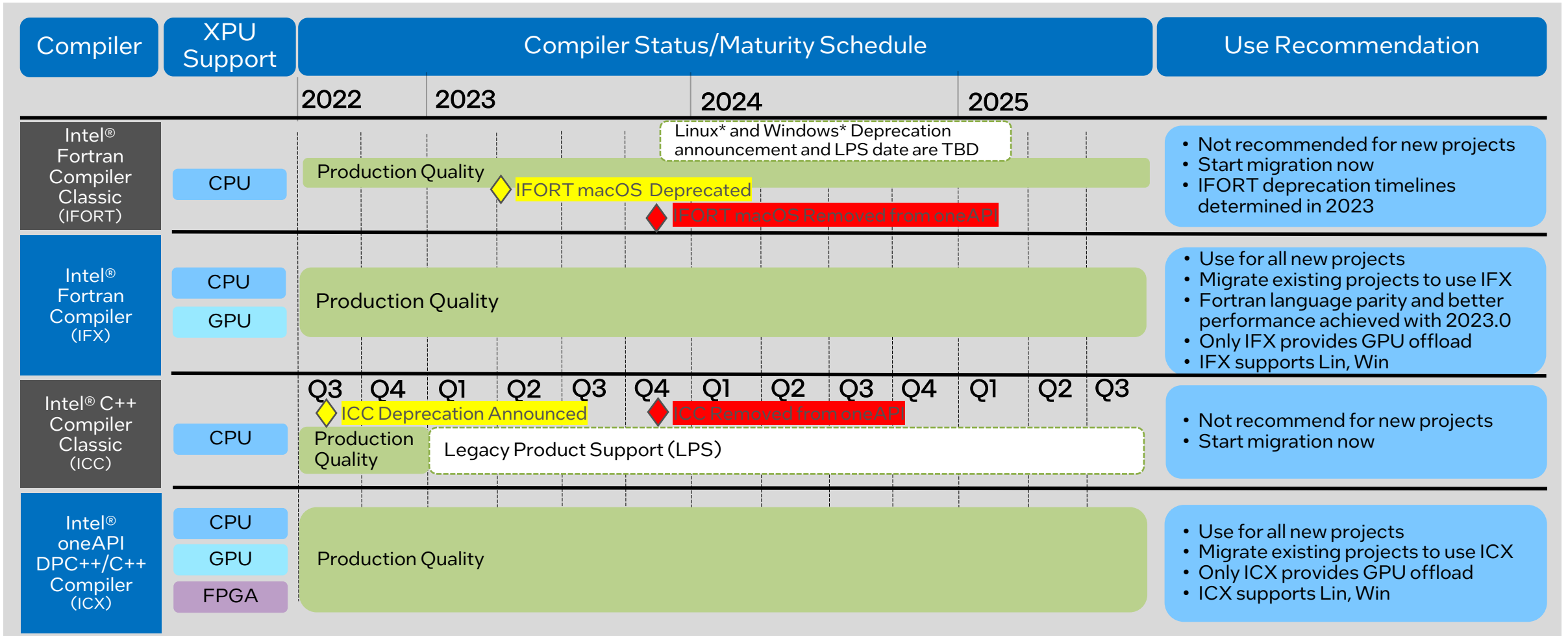
Compilers Roadmap and Timeline Update 2023

Intel® oneAPI Toolkits Release Schedule 2023 (Tentative)

- 2023.0.0 released 12/11/2022
- 2023.1.0 released 04/06/2023
- 2023.2.0 released 07/19/2023
- 2024.0.0 *scheduled** Q4 2023

* All product updates and releases and scheduled dates are subject to change

Intel Compilers Transition Roadmap



CPU = Intel Xeon, Core, Atom processors
 GPU = Intel Integrated and discrete GPU's
 FPGA = Intel FPGA's (Agilex, Stratix and Arria)

Legacy Product Support (LPS): Maintenance mode. Fix only critical bugs from priority customers and CVE issues.

Intel® Compiler Classic (ifort) macOS Deprecation

- macOS oneAPI HPC and Base Toolkits deprecation notice published with 2023.1.0 release. **NO DATES SET FOR Linux* or Windows***
 - Intel oneAPI Base and HPC Toolkits **for macOS** on x86 are now deprecated and will be discontinued in the 2024.0 release. Several Intel-led open source developer tool projects will continue supporting macOS on Apple Silicon including oneAPI Threading Building Blocks (oneTBB) and Intel® Implicit SPMD Program Compiler and we welcome the opportunity to work with contributors to expand support to additional tools in the future.
- For more info, [blog post](#)

Support Definitions

“Deprecation” similar to Language Standards definition:

- The act or process of marking the feature or product as obsolete, to discourage its use and warn users that it **may** be phased out in the future, but not removing the capability immediately, so as to allow for continued compatibility for a period of time.

“Legacy Product Support Support”

- Follows our support for older product versions. Older compilers provided on Intel® Registration Center downloads:

<https://www.intel.com/content/www/us/en/developer/articles/release-notes/oneapi-supported-product-versions.html>