

# **Fujitsu Software Compiler Package V1.0L30**

## **Release Note**

J2UL-2637-03ENZ0(01)  
April 2025

# Preface

---

## Purpose of This Manual

This manual describes functions added to Compiler Package V1.0L30 (hereafter referred to as Compiler Package) or later and provides information on their compatibility.

## Intended Readers

This manual is written for users who use Compiler Package to develop scientific and technical computation programs, and for system administrators who install Compiler Package and set the environment settings.

## Organization of This Manual

This manual is organized as follows.

[Chapter 1 Overview of New Functions](#)

[Chapter 2 Compatibility Information](#)

## Notation Used in This Manual

### Syntax Description Symbols

A syntax description symbol is a symbol that has specific meaning in syntax.

The following syntax symbols are used in this manual.

Symbol name	Symbol	Explanation
Selection symbols	{ }	Only one of the items enclosed in the braces must be selected (items are listed vertically).
		Multiple items are enumerated by this delimiter (items are listed horizontally).
Option symbol	[ ]	An item enclosed in brackets can be omitted. This symbol includes the meaning of the selection symbol "{ }".
Repeat symbol	...	The item immediately preceding the ellipsis can be specified repeatedly in the syntax.

## Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

## Trademarks

- OpenMP is a trademark of OpenMP Architecture Review Board.
- Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries.
- All other trademarks are the property of their respective owners.
- The trademark notice symbol (TM, (R)) is not necessarily added in the system name and the product name, etc. published in this material.

## Date of Publication and Version

Version	Manual code
April 2025, Version 3.1	J2UL-2637-03ENZ0(01)
April 2024, 3rd Version	J2UL-2637-03ENZ0(00)
January 2024, Version 2.5	J2UL-2637-02ENZ0(05)
October 2023, Version 2.4	J2UL-2637-02ENZ0(04)

Version	Manual code
April 2023, Version 2.3	J2UL-2637-02ENZ0(03)
October 2022, Version 2.2	J2UL-2637-02ENZ0(02)
April 2022, Version 2.1	J2UL-2637-02ENZ0(01)
January 2022, 2nd Version	J2UL-2637-02ENZ0(00)
August 2021, Version 1.7	J2UL-2637-01ENZ0(07)
August 2021, Version 1.6	J2UL-2637-01ENZ0(06)
July 2021, Version 1.5	J2UL-2637-01ENZ0(05)
March 2021, Version 1.4	J2UL-2637-01ENZ0(04)
January 2021, Version 1.3	J2UL-2637-01ENZ0(03)
November 2020, Version 1.2	J2UL-2637-01ENZ0(02)
September 2020, Version 1.1	J2UL-2637-01ENZ0(01)
July 2020, 1st Version	J2UL-2637-01ENZ0(00)

## Copyright

Copyright FUJITSU LIMITED 2020-2025

## Update History

Changes	Location	Version
The manual code list was deleted.	Appendix A	Version 3.1
"New Functions in V1.0L30" is added.	1.1	3rd Version
Compatibility information is appended.	Chapter 2 2.7.2	
Manual code for V1.0L30 is added.	Appendix A	
Manual code for V1.0L21 (Patch ID: T01920-05) is added.	Appendix A	Version 2.5
Compatibility information is appended.	Chapter 2 2.1.28	Version 2.4
The section "Other" is added.	2.7	
Manual code for V1.0L21 (Patch ID: T01920-04) is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.2.26 2.2.27 2.2.28 2.3.28 2.3.29 2.3.30	Version 2.3
Manual code for V1.0L21 (Patch ID: T01920-03) is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.1.27 2.2.25 2.3.27 2.6.2	Version 2.2
Manual code for V1.0L21 (Patch ID: T01920-02) is added.	Appendix A	

Changes	Location	Version
Compatibility information is appended.	Chapter 2 2.2.22 2.2.23 2.2.24 2.3.24 2.3.25 2.3.26	Version 2.1
Manual code for V1.0L21 (Patch ID: T01920-01) is added.	Appendix A	
"New Functions in V1.0L21" is added.	1.1	2nd Version
Compatibility information is appended.	Chapter 2 2.4.4	
Manual code for V1.0L21 is added.	Appendix A	
Changed the manual code of Release Note.	A.1.1	Version 1.7
Patch ID of V1.0L20A and patch ID of V1.0L20 are listed together.	-	
Compatibility information is appended.	2.4.3	Version 1.6
Manual code for V1.0L20 (Patch ID: T01887-07) is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.1.25 2.1.26 2.2.21 2.3.23	Version 1.5
Manual code for V1.0L20 (Patch ID: T01887-06) is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.1.21 2.1.22 2.1.23 2.1.24 2.2.20 2.3.22	Version 1.4
Manual code for V1.0L20 (Patch ID: T01887-05) is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.1.19 2.1.20 2.2.19 2.3.21 2.5.8	Version 1.3
Appendix is added.	Appendix A	
Compatibility information is appended.	Chapter 2 2.1.17 2.1.18 2.3.20 2.5.7	Version 1.2
Compatibility information is appended.	Chapter 2 2.1.16	Version 1.1

All rights reserved.

The information in this manual is subject to change without notice.

# Contents

Chapter 1 Overview of New Functions.....	1
1.1 New Functions in V1.0L30.....	1
1.2 New Functions in V1.0L21.....	1
1.3 New Functions in V1.0L20.....	1
1.3.1 Fortran Compiler.....	1
1.3.2 C Compiler.....	4
1.3.3 C++ Compiler.....	8
1.3.4 MPI.....	11
1.3.5 Profiler.....	11
1.3.6 Mathematical Libraries.....	12
Chapter 2 Compatibility Information.....	13
2.1 Fortran Compiler.....	21
2.1.1 Change of the predefined version macro.....	21
2.1.2 Error check for OpenMP clause at compilation time.....	21
2.1.3 Error check for polymorphic variable in OpenMP clause at compilation time.....	22
2.1.4 Change of values of macro and named constant due to OpenMP API version 4.5 support.....	22
2.1.5 Function enhancement of -Knoalias=s option.....	23
2.1.6 Change the target loops of the -Kloop_fission option.....	23
2.1.7 Change the default value of kind of the -Kilfunc[=kind] option.....	24
2.1.8 Abolition of the optimization control specifier LOOP_NOFISSION.....	24
2.1.9 Change of the optimization control specifier LOOP_FISSION.....	25
2.1.10 Incompatibility of the object due to changes in procedure call standard.....	25
2.1.11 Type conversion options in LLVM OpenMP Library.....	26
2.1.12 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change.....	26
2.1.13 Executable program incompatibilities due to changes of header files and modules.....	26
2.1.14 The internal behavior of software pipelining is changed.....	27
2.1.15 The optimization information of software pipelining is changed.....	27
2.1.16 Change of the attribute information of the compilation information for intrinsic function name in constant expression.....	28
2.1.17 Output compiler message for the allocatable assignment of the Fortran 2003 or later standard.....	29
2.1.18 Change the default of the compiler option -Nalloc_assign for the compiler option -X.....	29
2.1.19 Changes of the INQUIRE and OPEN specifications when a directory is specified in the FILE specifier.....	30
2.1.20 Improving time resolution in the SYSTEM_CLOCK built-in subroutine.....	31
2.1.21 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable.....	32
2.1.22 Changing the return value of SIZE in the RANDOM_SEED Built-in Subroutine.....	33
2.1.23 Error check for the SELECT TYPE construct conforming to the Fortran standard at compilation time.....	33
2.1.24 Error check at compilation time for detect execution time errors due to compiler failure.....	34
2.1.25 Changes the runtime message that is printed when an error is detected in the OPEN statement where the NEWUNIT= specifier appears.....	35
2.1.26 Changes the default of the compiler option -Karray_declaration_opt.....	36
2.1.27 Changes of the options to be enabled when the GNU compatible option -fopenmp is specified to be equivalent to the compiler option -Kopenmp.....	36
2.1.28 Value of COUNT_MAX for SYSTEM_CLOCK intrinsic subroutine of Fujitsu Fortran Compiler is improved to be able to count execution time greater than one day.....	36
2.2 C Compiler.....	37
2.2.1 [Trad Mode only] Change of the predefined version macro.....	37
2.2.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed.....	37
2.2.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support.....	38
2.2.4 Obsolescence of the compiler option -Klto.....	38
2.2.5 [Clang Mode only] Remove the compiler option -f{plt no-plt}.....	38
2.2.6 [Clang Mode only] Remove the compiler option -mtune.....	39
2.2.7 [Trad Mode only] Change the target loops of the -Kloop_fission option.....	39
2.2.8 [Trad Mode only] Change the default value of kind of the -Kilfunc[=kind] option.....	40
2.2.9 [Clang Mode only] Change the default value of = {loop procedure} of the -ffj-ilfunc[={loop procedure}] option.....	40
2.2.10 [Trad Mode only] Specification change of the -Krdconv option.....	40

2.2.11 [Clang Mode only] Changes of SIMD optimization information.....	41
2.2.12 [Trad Mode only] Abolition of the optimization control specifier loop_nofission.....	42
2.2.13 [Trad Mode only] Change of the optimization control specifier loop_fission.....	43
2.2.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (# pragma [fj] loop) is invalid.....	43
2.2.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard.....	44
2.2.16 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change.....	44
2.2.17 The internal behavior of software pipelining is changed.....	45
2.2.18 The optimization information of software pipelining is changed.....	45
2.2.19 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option.....	46
2.2.20 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable.....	46
2.2.21 Changes the default of the compiler option -Karray_declaration_opt.....	46
2.2.22 Changing the behavior of the getopt function.....	47
2.2.23 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.....	48
2.2.24 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified.....	49
2.2.25 [Clang Mode only] Change the default behavior due to the addition of the compiler options -f{ debug-info-for-profiling no-debug-info-for-profiling }.....	49
2.2.26 [Clang Mode only] Improvement of the usability for software pipelining and zfill.....	50
2.2.27 [Trad Mode only] The GNU compatible option -fvisibility={default internal hidden protected} deprecated.....	51
2.2.28 [Clang Mode only] The compiler option -f{ fj-loop-interchange fj-no-loop-interchange } deprecated.....	51
2.3 C++ Compiler.....	52
2.3.1 [Trad Mode only] Change of the predefined version macro.....	52
2.3.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed.....	52
2.3.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support.....	53
2.3.4 Obsolescence of the compiler option -Klto.....	53
2.3.5 [Clang Mode only] Remove the compiler option -f{plt no-plt}.....	53
2.3.6 [Clang Mode only] Remove the compiler option -mtune.....	54
2.3.7 [Trad Mode only] Change the target loops of the -Kloop_fission option.....	54
2.3.8 [Trad Mode only] Change the default value of kind of the -Kilfunc[=kind] option.....	55
2.3.9 [Clang Mode only] Change the default value of = {loop procedure} of the -ffj-ilfunc[={loop procedure}] option.....	55
2.3.10 [Trad Mode only] Specification change of the -Krdconv option.....	55
2.3.11 [Clang Mode only] Changes of SIMD optimization information.....	56
2.3.12 [Trad Mode only] Abolition of the optimization control specifier loop_nofission.....	57
2.3.13 [Trad Mode only] Change of the optimization control specifier loop_fission.....	58
2.3.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (# pragma [fj] loop) is invalid.....	58
2.3.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard.....	59
2.3.16 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change.....	59
2.3.17 [Clang Mode only] Incompatibility of executable programs due to the change of the libc++ version.....	60
2.3.18 The internal behavior of software pipelining is changed.....	60
2.3.19 The optimization information of software pipelining is changed.....	60
2.3.20 [Trad Mode only] The initial value of C++ array allocated by new expression without initializer is changed.....	61
2.3.21 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option.....	61
2.3.22 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable.....	62
2.3.23 Changes the default of the compiler option -Karray_declaration_opt.....	62
2.3.24 Changing the behavior of the getopt function.....	62
2.3.25 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.....	64
2.3.26 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified.....	65
2.3.27 [Clang Mode only] Change the default behavior due to the addition of the compiler options -f{ debug-info-for-profiling no-debug-info-for-profiling }.....	65
2.3.28 [Clang Mode only] Improvement of the usability for software pipelining and zfill.....	66
2.3.29 [Trad Mode only] The GNU compatible option -fvisibility={default internal hidden protected} deprecated.....	66
2.3.30 [Clang Mode only] The compiler option -f{ fj-loop-interchange fj-no-loop-interchange } deprecated.....	67

2.4 MPI Library.....	67
2.4.1 The local_options -am for the mpiexec command is changed to -tune.....	67
2.4.2 Executable program incompatibilities due to changes of header files and modules.....	68
2.4.3 Change the behavior when the MCA parameter btl_openib_receive_queues_default_num_bufs is not specified.....	68
2.4.4 Change the default value of the info key romio_ds_write, which specifies the behavior of Data sieving in MPI-IO, and change the processing so that a lock is taken when Data sieving is enabled.....	69
2.5 Profiler.....	69
2.5.1 Change of target for counting up the cost of inlined function in Instant Performance Profiler.....	69
2.5.2 Change of condition for outputting time statistic information of Advanced Performance Profiler.....	69
2.5.3 Change of output of MPI_STARTALL routine in MPI communication cost information of Advanced Performance Profiler.....	70
2.5.4 Change of output information and layout of CPU Performance Analysis Report.....	70
2.5.5 Changes of kinds of CPU Performance Analysis Report.....	71
2.5.6 Changes of output information and layout of CPU Performance Analysis Report (Data Transfer CMGs).....	72
2.5.7 Changes of the Outputting Directory of Profile Data in Profiler.....	72
2.5.8 Change of target for counting up the cost of inlined function in Instant Performance Profiler (cancellation of 2.5.1).....	73
2.6 Mathematical Libraries.....	73
2.6.1 Executable program incompatibilities due to changes of header files and modules.....	73
2.6.2 Changes the mathematical library linked when the compiler option -KNOSVE is specified using the native compiler.....	74
2.7 Other.....	74
2.7.1 Differences in Default Domain of papi by OS Version.....	74
2.7.2 Changing the License Authentication System.....	74

# Chapter 1 Overview of New Functions

## 1.1 New Functions in V1.0L30

This section provides an overview of the features added in Compiler Package V1.0L30.

Function	Description	Reference
Changes to license authentication system	Changed to not build a license server.	Software Description "Notes on Upgrading from Previous Versions" in Chapter "Installation Procedure"

## 1.2 New Functions in V1.0L21

This section provides an overview of the features added in Compiler Package V1.0L21.

Function	Description	Reference
Support for New Hardware	Compatible with PRIMEHPC FX700 server unit (2.6GHz 24C).	None

## 1.3 New Functions in V1.0L20

This section provides an overview of the features added in Compiler Package V1.0L20.

### 1.3.1 Fortran Compiler

Table 1.1 New Functions

Function	Description	Reference
Compiler options	Add/improve the following options: <ul style="list-style-type: none"><li>- {-help --help}</li><li>- -K{array_declaration_opt noarray_declaration_opt}</li><li>- -K{extract_stride_store noextract_stride_store}</li><li>- -K{fp_precision nofp_precision}</li><li>- -K{loop_fission_stripmining[={ M L1 L2 }] loop_nofission_stripmining}</li><li>- -Kloop_fission_threshold=<i>N</i></li><li>- -K{loop_perfect_nest loop_noperfect_nest}</li><li>- -Koptions</li><li>- -K{prefetch_stride[={ soft hard_auto hard_always }] prefetch_nostride}</li><li>- -K{preload nopreload}</li><li>- -K{simd_uncounted_loop simd_nouncounted_loop}</li><li>- -K{simd_use_multiple_structures simd_nouse_multiple_structures}</li><li>- -K{subscript_opt nosubscript_opt}</li><li>- -Kswp_policy={ auto small large }</li><li>- -Ktls_size={ 12 24 32 48 }</li></ul>	Fortran User's Guide, "Compiler Options" in Chapter "Compiling and Linking Fortran Programs"

Function	Description	Reference
	Enhance the function of the -Knoalias option.	Fortran User's Guide, "Compiler Options" in Chapter "Compiling and Linking Fortran Programs"
Environment variable	Add the following environment variable: - FCOMP_UNRECOGNIZED_OPTION	Fortran User's Guide, "Compile Command Environment Variable" in Chapter "Compiling and Linking Fortran Programs"
	Add the following environment variable: - FLIB_TRACEBACK_MEM_SIZE	Fortran User's Guide, "Using Environment Variables for Execution" in Chapter "Executing Fortran Programs"
	Add the following environment variable: - FLIB_BARRIER	Fortran User's Guide, Appendix "Using High-speed Facility"
Execution performance improvement	Improve the following functions: - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) - Register allocation	None
Optimization control specifier	Add the following optimization control specifiers: - [NO]ARRAY_DECLARATION_OPT - [NO]EXTRACT_STRIDE_STORE - [NO]FULLUNROLL_PRE_SIMD[(n)] - LOOP_[NO]FISSION_STRIPMINING[(n L1 L2)] - LOOP_FISSION_TARGET[(CL LS)] - LOOP_FISSION_THRESHOLD(n) - LOOP_[NO]PERFECT_NEST - PREFETCH_[NO]CONDITIONAL - PREFETCH_[NO]INDIRECT - PREFETCH_LINE(n) - PREFETCH_LINE_L2(n) - PREFETCH_NOSEQUENTIAL - PREFETCH_[NO]STRIDE[(SOFT HARD_AUTO HARD_ALWAYS)] - [NO]PRELOAD - SCACHE_ISOLATE_ASSIGN(array1[,array2]...) END_SCACHE_ISOLATE_ASSIGN - SCACHE_ISOLATE_WAY(L2=n1[,L1=n2]) END_SCACHE_ISOLATE_WAY - SIMD_[NO]USE_MULTIPLE_STRUCTURES	Fortran User's Guide, "Using Optimization control line (OCL)" in Chapter "Optimization Functions"

Function	Description	Reference
	- SWP_POLICY({ AUTO SMALL LARGE })	
	Add an array section to the optimization target for the PREFETCH_READ and PREFETCH_WRITE specifiers.	Fortran User's Guide, "Optimization Control Specifiers" in Chapter "Optimization Functions"
Optimization messages	Add the optimization messages for the following functions: <ul style="list-style-type: none"> <li>- Message to suppress inline</li> <li>- Software pipelining</li> </ul>	Fortran Compiler Messages
	Improve the optimization messages for the following functions: <ul style="list-style-type: none"> <li>- Loop fission</li> <li>- Error for usage restriction of stack</li> </ul>	Fortran Compiler Messages
Using high-speed facility	The high-speed facility (the inter-core hardware barrier and the Sector cache) can be used for LLVM OpenMP Library.	Fortran User's Guide, Appendix "Using High-speed Facility"
GNU compatible options	Support the following GNU compatible options: <ul style="list-style-type: none"> <li>- --print-file-name=include</li> <li>- --print-prog-name={ as ld objdump ranlib ar }</li> <li>- --shared</li> <li>- --version</li> <li>- -cpp</li> <li>- -Xlinker <i>option</i></li> <li>- -f{ align-loops[=<i>N</i>] no-align-loops }</li> <li>- -ffp-contract=fast</li> <li>- -f{ inline-functions no-inline-functions }</li> <li>- -floop-parallelize-all</li> <li>- -flto</li> <li>- -f{ omit-frame-pointer no-omit-frame-pointer }</li> <li>- -fopenmp</li> <li>- -f{ openmp-simd no-openmp-simd }</li> <li>- -f{ optimize-sibling-calls no-optimize-sibling-calls }</li> <li>- -f{ pic PIC }</li> <li>- -f{ pie PIE }</li> <li>- -f{ plt no-plt }</li> <li>- -fprofile-dir=<i>path</i></li> <li>- -f{ schedule-insns no-schedule-insns }</li> <li>- -f{ schedule-insns2 no-schedule-insns2 }</li> <li>- -funroll-loops</li> <li>- -f{ unsafe-math-optimizations no-unsafe-math-optimizations }</li> <li>- -g{ dwarf dwarf-4 }</li> </ul>	Fortran User's Guide, Appendix "GNU Compatible Options"

Function	Description	Reference
	<ul style="list-style-type: none"> <li>- <code>-isystem <i>dir</i></code></li> <li>- <code>-march=<i>arch</i>[+<i>features</i>]</code></li> <li>- <code>-mmodel={ <i>small</i> <i>large</i> }</code></li> <li>- <code>-mcpu=<i>cpu</i></code></li> <li>- <code>-m{ <i>pc-relative-literal-loads</i> <i>no-pc-relative-literal-loads</i> }</code></li> <li>- <code>-mtls-size={ 12 24 32 48 }</code></li> <li>- <code>-mtune=<i>cpu</i></code></li> <li>- <code>-pthread</code></li> <li>- <code>-rdynamic</code></li> <li>- <code>-v</code></li> <li>- <code>-w</code></li> </ul>	
Half-precision Type	Enhance the intrinsic procedures compatible with the half-precision type.	Fortran User's Guide, Appendix "Intrinsic Procedures and I/O Statements compatible with Half-precision Type"
Fortran 2008/2018 Standards	Support Fortran 2008 and a subset of Fortran 2018 standards.	<p>Fortran User's Guide, "Assumed type", "Assumed Rank", and "ISO_Fortran_binding.h" in Chapter "Mixed Language Programming"</p> <p>Fortran Language Reference, "Constant expression", "Coarray Reference", "Assignment Statement", "CO_MAX Intrinsic Subroutine", "CO_MIN Intrinsic Subroutine", and "CO_SUM Intrinsic Subroutine"</p>
OpenMP API Version 4.5/5.0 Specifications	Support OpenMP API Version 4.5 and a subset of Version 5.0 specifications.	Fortran User's Guide, "Parallelization by OpenMP Specifications" in "Chapter "Multiprocessing"

### 1.3.2 C Compiler

The C compiler has two user interfaces, Trad Mode and Clang Mode. Note that the new functions listed below may or may not be available depending on the mode. For more information on Trad Mode and Clang Mode, please refer to "C User's Guide".

Table 1.2 New Functions

Function	Trad Mode	Clang Mode	Description	Reference
Compiler options	*	-	<p>Add/improve the following options:</p> <ul style="list-style-type: none"> <li>- <code>{-help --help}</code></li> <li>- <code>-K{array_declaration_opt noarray_declaration_opt}</code></li> <li>- <code>-K{extract_stride_store noextract_stride_store}</code></li> <li>- <code>-K{fp_precision nofp_precision}</code></li> </ul>	C User's Guide, "Compiler Options" in Chapter "From Compilation to Execution"

Function	Trad Mode	Clang Mode	Description	Reference
			<ul style="list-style-type: none"> <li>-K{loop_fission_stripmining[={N L1 L2}]] loop_nofission_stripmining}</li> <li>-Kloop_fission_threshold=N</li> <li>-K{loop_perfect_nest loop_noperfect_nest}</li> <li>-K{prefetch_stride[={soft hard_auto hard_always}]] prefetch_nostride}</li> <li>-K{preload nopreload}</li> <li>-K{simd_uncounted_loop simd_nouncounted_loop}</li> <li>-K{simd_use_multiple_structures  simd_nouse_multiple_structures}</li> <li>-Kswp_policy={auto small large}</li> <li>-Ktls_size={12 24 32 48}</li> </ul>	
	-	*	Add/improve the following options: <ul style="list-style-type: none"> <li>-f{fj-fast-matmul fj-no-fast-matmul}</li> <li>-f{fj-fp-relaxed fj-no-fp-relaxed}</li> <li>-f{fj-hpctag fj-no-hpctag}</li> <li>-f{fj-line fj-no-line}</li> <li>-f{fj-loop-interchange fj-no-loop-interchange} (This option is deprecated in V1.0L21(T01920-03))</li> <li>-f{fj-ocl fj-no-ocl}</li> <li>-f{fj-optlib-string fj-no-optlib-string}</li> <li>-f{fj-prefetch-strong fj-no-prefetch-strong}</li> <li>-f{fj-prefetch-strong-L2 fj-no-prefetch-strong-L2}</li> <li>-fprofile-dir=<i>dir_name</i></li> <li>-mfj-tls-size={12 24 32 48}</li> <li>--coverage</li> </ul>	C User's Guide, "Compiler Options" in Chapter "Clang Mode"
	*	-	The following function is improved so that parameters can be specified. <ul style="list-style-type: none"> <li>-K{rdconv[={1 2}]]nordconv}</li> </ul>	C User's Guide, "Compiler Options" in Chapter "From Compilation to Execution"
Compilation profile file	*	-	Add the following compilation profile files: <ul style="list-style-type: none"> <li>fccpx_trad_PROF</li> <li>fcc_trad_PROF</li> </ul>	C User's Guide, "Compilation Profile File" in Chapter "From Compilation to Execution"
	-	*	Add the following compilation profile files: <ul style="list-style-type: none"> <li>fccpx_clang_PROF</li> <li>fcc_clang_PROF</li> </ul>	C User's Guide, "Compilation Profile File" in Chapter "Clang Mode"
Environment variable	*	-	Add the following environment variables: <ul style="list-style-type: none"> <li>fccpx_trad_ENV</li> <li>fcc_trad_ENV</li> </ul>	C User's Guide, "Environment Variable for Compile Command" in Chapter "From Compilation to Execution"

Function	Trad Mode	Clang Mode	Description	Reference
			- FCOMP_UNRECOGNIZED_OPTION	
	*	-	Add the following environment variable: - FLIB_TRACEBACK_MEM_SIZE	C User's Guide, "Environment Variable for Execution" in Chapter "From Compilation to Execution"
	-	*	Add the following environment variables: - fccpx_clang_ENV - fcc_clang_ENV	C User's Guide, "Environment Variable for Compile Command" in Chapter "Clang Mode"
	-	*	Add the following environment variable: - LD_BIND_NOW	C User's Guide, "Procedure of Execution" in Chapter "Clang Mode"
	*	*	Add the following environment variable: - FLIB_BARRIER	C User's Guide, Appendix "Using High-speed Facility"
Execution performance improvement	*	-	Improve the following functions: - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) - Register allocation	None
Optimization control line	*	-	Enable "#pragma fj" as the format of optimization control lines.	C User's Guide, "Using the Optimization Control Line (OCL)" in Chapter "Optimization"
	-	*	Add the following optimization control line: - #pragma fj <i>Optimization-control-specifier</i>	C User's Guide, "Using the Optimization Control Line (Pragma Directives)" in Chapter "Clang Mode"
Optimization control specifier	*	-	Add the following optimization control specifiers: - [no]array_declaration_opt - [no]extract_stride_store - [no]fullunroll_pre_simd [ <i>n</i> ] - loop_[no]fission_stripmining [ <i>n</i>  "L1" "L2"] - loop_fission_target [cl ls] - loop_fission_threshold <i>n</i> - loop_[no]perfect_nest - prefetch_cache_level <i>c-level</i> - prefetch_[no]conditional - prefetch_[no]indirect - prefetch_[no]infer - prefetch_iteration <i>n</i> - prefetch_iteration_L2 <i>n</i>	C User's Guide, "Using the Optimization Control Line (OCL)" in Chapter "Optimization"

Function	Trad Mode	Clang Mode	Description	Reference
			<ul style="list-style-type: none"> <li>- prefetch_line <i>n</i></li> <li>- prefetch_line_L2 <i>n</i></li> <li>- prefetch_nosequential</li> <li>- prefetch_[no]stride [soft hard_auto hard_always]</li> <li>- [no]preload</li> <li>- scache_isolate_assign <i>array1[,array2]...</i> end_scache_isolate_assign</li> <li>- scache_isolate_way L2=<i>n1</i> [L1=<i>n2</i>] end_scache_isolate_way</li> <li>- simd_[no]use_multiple_structures</li> <li>- swp_policy {auto small large}</li> </ul>	
	-	*	Add the following optimization control specifier: <ul style="list-style-type: none"> <li>- #pragma clang fp contract(fast)</li> </ul>	C User's Guide, "Using the Optimization Control Line (Pragma Directives)" in Chapter "Clang Mode"
Optimization messages	*	-	Add the optimization messages for the following functions: <ul style="list-style-type: none"> <li>- Software pipelining</li> <li>- Error for usage restriction of stack</li> </ul>	C/C++ Compiler Optimization Messages
	*	-	Improve the optimization messages for the following function: <ul style="list-style-type: none"> <li>- Loop fission</li> </ul>	C/C++ Compiler Optimization Messages
Using high-speed facility	*	*	The high-speed facility (the inter-core hardware barrier and the Sector cache) can be used for LLVM OpenMP Library.	C User's Guide, Appendix "Using High-speed Facility"
GNU C compatible option	*	-	Support the following GNU C compatible options: <ul style="list-style-type: none"> <li>- -ffp-contract=fast</li> <li>- -floop-parallelize-all</li> <li>- -fprofile-dir=<i>path</i></li> <li>- -funroll-loops</li> <li>- -f{unsafe-math-optimizations no-unsafe-math-optimizations}</li> <li>- -mcmmodel={small large}</li> </ul>	C User's Guide, Appendix "Compatibility with GNU C specifications"
Add the attribute of GNU Extensions	*	-	Add the following attribute: <ul style="list-style-type: none"> <li>- aarch64_vector_pcs</li> </ul>	C User's Guide, Appendix "Compatibility with GNU C specifications"
Built-in Functions	-	*	Add the SIMD Built-in Functions.	C User's Guide, "SIMD Built-in Functions" in Chapter "Clang Mode"
SVE function	-	*	Output the object files using SVE, when -mcpu=a64fx option is set.	C User's Guide, "Compiler Options" in Chapter "Clang Mode"

Function	Trad Mode	Clang Mode	Description	Reference
OpenMP API Version 4.5/5.0 Specifications	-	*	Support OpenMP API Version 4.5 and a subset of Version 5.0 specifications.	C User's Guide, "Parallelization by OpenMP Specification" in Chapter "Multiprocessing"

### 1.3.3 C++ Compiler

The C++ compiler has two user interfaces, Trad Mode and Clang Mode. Note that the new functions listed below may or may not be available depending on the mode. For more information on Trad Mode and Clang Mode, please refer to "C++ User's Guide".

Table 1.3 New Functions

Function	Trad Mode	Clang Mode	Description	Reference
Compiler options	*	-	Add/improve the following options: <ul style="list-style-type: none"> <li>- {-help --help}</li> <li>-K{array_declaration_opt noarray_declaration_opt}</li> <li>-K{extract_stride_store noextract_stride_store}</li> <li>-K{fp_precision nofp_precision}</li> <li>-K{loop_fission_stripmining[={M L1 L2}]] loop_nofission_stripmining}</li> <li>-Kloop_fission_threshold=<i>N</i></li> <li>-K{loop_perfect_nest loop_noperfect_nest}</li> <li>-K{prefetch_stride[={soft hard_auto hard_always}]] prefetch_nostride}</li> <li>-K{preload nopreload}</li> <li>-K{simd_uncounted_loop simd_nouncounted_loop}</li> <li>-K{simd_use_multiple_structures  simd_nouse_multiple_structures}</li> <li>-Kswp_policy={auto small large}</li> <li>-Ktls_size={ 12 24 32 48}</li> <li>-std={c++17 gnu++17}</li> </ul>	C++ User's Guide, "Compiler Options" in Chapter "From Compilation to Execution"
	-	*	Add/improve the following options: <ul style="list-style-type: none"> <li>-f{fj-fast-matmul fj-no-fast-matmul}</li> <li>-f{fj-fp-relaxed fj-no-fp-relaxed}</li> <li>-f{fj-hpctag fj-no-hpctag}</li> <li>-f{fj-line fj-no-line}</li> <li>-f{fj-loop-interchange fj-no-loop-interchange} (This option is deprecated in V1.0L21(T01920-03))</li> <li>-f{fj-ocl fj-no-ocl}</li> <li>-f{fj-optlib-string fj-no-optlib-string}</li> <li>-f{fj-prefetch-strong fj-no-prefetch-strong}</li> <li>-f{fj-prefetch-strong-L2 fj-no-prefetch-strong-L2}</li> </ul>	C++ User's Guide, "Compiler Options" in Chapter "Clang Mode"

Function	Trad Mode	Clang Mode	Description	Reference
			<ul style="list-style-type: none"> <li>- <code>-fprofile-dir=<i>dir_name</i></code></li> <li>- <code>-mfj-tls-size={ 12 24 32 48 }</code></li> <li>- <code>--coverage</code></li> </ul>	
	*	-	<p>The following function is improved so that parameters can be specified.</p> <ul style="list-style-type: none"> <li>- <code>-K{rdconv[={ 1 2 }]]nordconv }</code></li> </ul>	C++ User's Guide, "Compiler Options" in Chapter "From Compilation to Execution"
Compilation profile file	*	-	<p>Add the following compilation profile files:</p> <ul style="list-style-type: none"> <li>- <code>FCCpx_trad_PROF</code></li> <li>- <code>FCC_trad_PROF</code></li> </ul>	C++ User's Guide, "Compilation Profile File" in Chapter "From Compilation to Execution"
	-	*	<p>Add the following compilation profile files:</p> <ul style="list-style-type: none"> <li>- <code>FCCpx_clang_PROF</code></li> <li>- <code>FCC_clang_PROF</code></li> </ul>	C++ User's Guide, "Compilation Profile File" in Chapter "Clang Mode"
Environment variable	*	-	<p>Add the following environment variables:</p> <ul style="list-style-type: none"> <li>- <code>FCCpx_trad_ENV</code></li> <li>- <code>FCC_trad_ENV</code></li> <li>- <code>FCOMP_UNRECOGNIZED_OPTION</code></li> </ul>	C++ User's Guide, "Environment Variable for Compile Command" in Chapter "From Compilation to Execution"
	*	-	<p>Add the following environment variable:</p> <ul style="list-style-type: none"> <li>- <code>FLIB_TRACEBACK_MEM_SIZE</code></li> </ul>	C++ User's Guide, "Environment Variable for Execution" in Chapter "From Compilation to Execution"
	-	*	<p>Add the following environment variables:</p> <ul style="list-style-type: none"> <li>- <code>FCCpx_clang_ENV</code></li> <li>- <code>FCC_clang_ENV</code></li> </ul>	C++ User's Guide, "Environment Variable for Compile Command" in Chapter "Clang Mode"
	-	*	<p>Add the following environment variable:</p> <ul style="list-style-type: none"> <li>- <code>LD_BIND_NOW</code></li> </ul>	C++ User's Guide, "Procedure of Execution" in Chapter "Clang Mode"
	*	*	<p>Add the following environment variable:</p> <ul style="list-style-type: none"> <li>- <code>FLIB_BARRIER</code></li> </ul>	C++ User's Guide, Appendix "Using High-speed Facility"
Execution performance improvement	*	-	<p>Improve the following functions:</p> <ul style="list-style-type: none"> <li>- SIMD</li> <li>- General Optimization</li> <li>- Instruction Scheduling (Software Pipelining)</li> <li>- Register allocation</li> </ul>	None
Optimization control line	*	-	Enable <code>"#pragma fj"</code> as the format of optimization control lines.	C++ User's Guide, "Using the Optimization Control Line (OCL)" in Chapter "Optimization"
	-	*	<p>Add the following optimization control line:</p> <ul style="list-style-type: none"> <li>- <code>#pragma fj <i>Optimization-control-specifier</i></code></li> </ul>	C++ User's Guide, "Using the Optimization Control Line (Pragma

Function	Trad Mode	Clang Mode	Description	Reference
				Directives)" in "Clang Mode"
Optimization control specifier	*	-	Add the following optimization control specifiers: <ul style="list-style-type: none"> <li>- [no]array_declaration_opt</li> <li>- [no]extract_stride_store</li> <li>- [no]fullunroll_pre_simd [<i>n</i>]</li> <li>- loop_[no]fission_stripmining [<i>n</i> "L1" "L2"]</li> <li>- loop_fission_target [cl ls]</li> <li>- loop_fission_threshold <i>n</i></li> <li>- loop_[no]perfect_nest</li> <li>- prefetch_cache_level <i>c-level</i></li> <li>- prefetch_[no]conditional</li> <li>- prefetch_[no]indirect</li> <li>- prefetch_[no]infer</li> <li>- prefetch_iteration <i>n</i></li> <li>- prefetch_iteration_L2 <i>n</i></li> <li>- prefetch_line <i>n</i></li> <li>- prefetch_line_L2 <i>n</i></li> <li>- prefetch_nosequential</li> <li>- prefetch_[no]stride [soft hard_auto hard_always]</li> <li>- [no]preload</li> <li>- scache_isolate_assign <i>array1[,array2]...</i> end_scache_isolate_assign</li> <li>- scache_isolate_way L2=<i>n1</i> [L1=<i>n2</i>] end_scache_isolate_way</li> <li>- simd_[no]use_multiple_structures</li> <li>- swp_policy {auto small large}</li> </ul>	C++ User's Guide, "Using the Optimization Control Line (OCL)" in Chapter "Optimization"
	-	*	Add the following optimization control specifier: #pragma clang fp contract(fast)	C++ User's Guide, "Using the Optimization Control Line (Pragma Directives)" in Chapter "Clang Mode"
Optimization messages	*	-	Add the optimization messages for the following functions: <ul style="list-style-type: none"> <li>- Software pipelining</li> <li>- Error for usage restriction of stack</li> </ul>	C/C++ Compiler Optimization Messages
	*	-	Improve the optimization messages for the following function: <ul style="list-style-type: none"> <li>- Loop fission</li> </ul>	C/C++ Compiler Optimization Messages
Using high-speed facility	*	*	The high-speed facility (the inter-core hardware barrier and the Sector cache) can be used for LLVM OpenMP Library.	C++ User's Guide, Appendix "Using High-speed Facility"

Function	Trad Mode	Clang Mode	Description	Reference
GNU C++ compatible option	*	-	Support the following GNU C++ compatible options: <ul style="list-style-type: none"> <li>-ffp-contract=fast</li> <li>-floop-parallelize-all</li> <li>-fprofile-dir=<i>path</i></li> <li>-funroll-loops</li> <li>-f{unsafe-math-optimizations no-unsafe-math-optimizations}</li> <li>-mcmodel={small large}</li> </ul>	C++ User's Guide, Appendix "Compatibility with GNU C++ specifications"
Add the attribute of GNU Extensions	*	-	Add the following attribute: <ul style="list-style-type: none"> <li>aarch64_vector_pcs</li> </ul>	C++ User's Guide, Appendix "Compatibility with GNU C++ specifications"
Built-in Functions	-	*	Add the SIMD Built-in Functions.	C++ User's Guide, "SIMD Built-in Functions" in Chapter "Clang Mode"
SVE function	-	*	Output object files using SVE, when -mcpu=a64fx option is set.	C++ User's Guide, "Compiler Options" in Chapter "Clang Mode"
C++17 specifications	*	-	Support a subset of C++17 specifications.	C++ User's Guide, "C++17 Specifications" in Chapter "Language Specifications"
OpenMP API Version 4.5/5.0 Specifications	-	*	Support OpenMP API Version 4.5 and a subset of Version 5.0 specifications.	C++ User's Guide, "Parallelization by OpenMP Specification" in Chapter "Multiprocessing"

### 1.3.4 MPI

Table 1.4 New Functions

Function	Description	Reference
Communication timeout setting	This is a debug function. When this function is set and a hang is detected, debug information is output.	MPI User's Guide, "Communication Timeout Setting" in Chapter "Supplementary Items"

### 1.3.5 Profiler

Table 1.5 New Functions

Function	Description	Reference
Instant Performance Profiler	Corresponds to dynamic process.	Profiler User's Guide, "fjprof_spawn_dir_name Info Key" in Chapter "Instant Performance Profiler"
	Outputs the profile result in the XML format.	Profiler User's Guide, "Details of Profile Result (XML Format)" in Chapter

Function	Description	Reference
		"Instant Performance Profiler"
	Allows the -A option to be specified with the fipp command.	Profiler User's Guide, "Outputting Profile Result" in Chapter "Instant Performance Profiler"
	Corresponds to clang mode.	Profiler User's Guide, "C and C++ Languages" and "Procedure Cost Distribution Information" in Chapter "Instant Performance Profiler"
	Corresponds to libomp.	None
Advanced Performance Profiler	Overhead of region specifying interface is reduced.	Profiler User's Guide, "Measuring Profile Data" in Chapter "Advanced Performance Profiler"
	Allows the -A option to be specified with the fapp command.	Profiler User's Guide, "Outputting Profile Result" in Chapter "Advanced Performance Profiler"
CPU Performance Analysis Report	Add the single report.	Profiler User's Guide, Chapter "CPU Performance Analysis Report"

### 1.3.6 Mathematical Libraries

Table 1.6 New Function

Function	Description	Reference
New routines	Add the following new routines: - Matrix copy and transpose routines	BLAS LAPACK ScaLAPACK User's Guide, "Matrix copy and transpose routines" in Appendix "Routines List"

## Chapter 2 Compatibility Information

Provides information about the incompatibility for functions that have changed from the previous version-level and how to avoid bad side effects if there exists. For "Patch ID", contact the system administrator.

Table 2.1 List of incompatible items by version-level (Fortran Compiler)

Items	Previous version-level (Patch ID)												
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01915-07)	V1.0L20 (T01887-06)	V1.0L20 (T01915-06)	V1.0L20 (T01887-05)	V1.0L20 (T01887-04)	V1.0L20 (T01887-03)	V1.0L20 (T01887-01)
2.1.1 Change of the predefined version macro	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.2 Error check for OpenMP clause at compilation time	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.3 Error check for polymorphic variable in OpenMP clause at compilation time	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.4 Change of values of macro and named constant due to OpenMP API version 4.5 support	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.5 Function enhancement of -Knoalias=s option	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.6 Change the target loops of the -Kloop_fission option	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.7 Change the default value of kind of the -Kilfunc[=kind] option	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.8 Abolition of the optimization control specifier LOOP_NOFISSION	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.9 Change of the optimization control specifier LOOP_FISSION	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.10 Incompatibility of the object due to changes in procedure call standard	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.11 Type conversion options in LLVM OpenMP Library	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.12 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.13 Executable program incompatibilities due to changes of header files and modules	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.14 The internal behavior of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	-	*
2.1.15 The optimization information of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	-	*

Items	Previous version-level (Patch ID)										
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01915-07)	V1.0L20 (T01915-06)	V1.0L20 (T01915-05)	V1.0L20 (T01915-04)	V1.0L20 (T01915-03)
2.1.16 Change of the attribute information of the compilation information for intrinsic function name in constant expression	-	-	-	-	-	-	-	-	-	-	*
2.1.17 Output compiler message for the allocatable assignment of the Fortran 2003 or later standard	-	-	-	-	-	-	-	-	-	*	*
2.1.18 Change the default of the compiler option - Nalloc_assign for the compiler option -X	-	-	-	-	-	-	-	-	-	*	*
2.1.19 Changes of the INQUIRE and OPEN specifications when a directory is specified in the FILE specifier	-	-	-	-	-	-	-	-	*	*	*
2.1.20 Improving time resolution in the SYSTEM_CLOCK built-in subroutine	-	-	-	-	-	-	-	-	*	*	*
2.1.21 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable	-	-	-	-	-	-	-	*	*	*	*
2.1.22 Changing the return value of SIZE in the RANDOM_SEED Built-in Subroutine	-	-	-	-	-	-	-	-	*	*	*
2.1.23 Error check for the SELECT TYPE construct conforming to the Fortran standard at compilation time	-	-	-	-	-	-	-	-	*	*	*
2.1.24 Error check at compilation time for detect execution time errors due to compiler failure	-	-	-	-	-	-	-	-	*	*	*
2.1.25 Changes the runtime message that is printed when an error is detected in the OPEN statement where the NEWUNIT= specifier appears	-	-	-	-	-	-	-	*	*	*	*
2.1.26 Changes the default of the compiler option - Karray_declaration_opt	-	-	-	-	-	-	-	*	*	*	*
2.1.27 Changes of the options to be enabled when the GNU compatible option -fopenmp is specified to be equivalent to the compiler option -Kopenmp	-	-	-	-	*	*	*	*	*	*	*
2.1.28 Value of COUNT_MAX for SYSTEM_CLOCK intrinsic subroutine of Fujitsu Fortran Compiler is improved to be able to count execution time greater than one day	-	-	*	*	*	*	*	*	*	*	*

\*: Incompatible

-: compatible

Table 2.2 List of incompatible items by version-level (C Compiler)

Items	Previous version-level (Patch ID)																			
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01887-07)	V1.0L20A (T01915-07)	V1.0L20 (T01887-06)	V1.0L20A (T01915-06)	V1.0L20 (T01887-05)	V1.0L20A (T01915-05)	V1.0L20 (T01887-04)	V1.0L20A (T01915-04)	V1.0L20 (T01887-03)	V1.0L20A (T01915-03)	V1.0L20 (T01887-01)	V1.0L20A (T01915-01)	V1.0L20 1st Version	V1.0L20A 1st Version
2.2.1 [Trad Mode only] Change of the predefined version macro	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.4 Obsolescence of the compiler option -Klto	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.5 [Clang Mode only] Remove the compiler option -f{plt no-plt}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.6 [Clang Mode only] Remove the compiler option -mtune	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.7 [Trad Mode only] Change the target loops of the -Kloop_fission option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.8 [Trad Mode only] Change the default value of kind of the -Kilfunc[=kind] option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.9 [Clang Mode only] Change the default value of ={loop procedure} of the -ffj-ilfunc[={loop procedure}] option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.10 [Trad Mode only] Specification change of the -Krdconv option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.11 [Clang Mode only] Changes of SIMD optimization information	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.12 [Trad Mode only] Abolition of the optimization control specifier loop_nofission	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.13 [Trad Mode only] Change of the optimization control specifier loop_fission	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (# pragma [fj] loop) is invalid	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.2.16 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*

Items	Previous version-level (Patch ID)											
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01887-07)	V1.0L20 (T01887-06)	V1.0L20 (T01887-05)	V1.0L20 (T01887-04)	V1.0L20 (T01887-03)	V1.0L20 (T01887-01)
2.2.17 The internal behavior of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	*
2.2.18 The optimization information of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	*
2.2.19 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option (*1)					-	-	-	-	-	*	*	*
2.2.20 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable	-	-	-	-	-	-	-	-	*	*	*	*
2.2.21 Changes the default of the compiler option - Karray_declaration_opt	-	-	-	-	-	-	-	-	*	*	*	-
2.2.22 Changing the behavior of the getopt function	-	-	-	-	-	*	*	*	*	*	*	*
2.2.23 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword	-	-	-	-	-	*	*	*	*	*	*	*
2.2.24 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified	-	-	-	-	-	*	*	*	*	*	*	*
2.2.25 [Clang Mode only] Change the default behavior due to the addition of the compiler options - f{debug-info-for-profiling no-debug-info-for-profiling}	-	-	-	-	*	*	*	*	*	*	*	*
2.2.26 [Clang Mode only] Improvement of the usability for software pipelining and zfill	-	-	-	*	*	*	*	*	*	-	-	-
2.2.27 [Trad Mode only] The GNU compatible option -fvisibility={ default internal hidden protected} deprecated	-	-	-	*	*	*	*	*	*	*	*	*
2.2.28 [Clang Mode only] The compiler option -f{fj-loop-interchange fj-no-loop-interchange} deprecated	-	-	-	*	*	*	*	*	*	*	*	*

\*: Incompatible

-: compatible

\*1: Removed from Items due to option deprecation.

Table 2.3 List of incompatible items by version-level (C++ Compiler)

Items	Previous version-level (Patch ID)																			
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01887-07)	V1.0L20 (T01915-07)	V1.0L20 (T01887-06)	V1.0L20 (T01915-06)	V1.0L20 (T01887-05)	V1.0L20 (T01915-05)	V1.0L20 (T01887-04)	V1.0L20 (T01915-04)	V1.0L20 (T01887-03)	V1.0L20 (T01915-03)	V1.0L20 (T01887-01)	V1.0L20A (T01915-01)	V1.0L20 1st Version	V1.0L20A 1st Version
2.3.1 [Trad Mode only] Change of the predefined version macro	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.4 Obsolescence of the compiler option -Klto	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.5 [Clang Mode only] Remove the compiler option -f{plt no-plt}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.6 [Clang Mode only] Remove the compiler option -mtune	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.7 [Trad Mode only] Change the target loops of the -Kloop_fission option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.8 [Trad Mode only] Change the default value of kind of the -Kilfunc[=kind] option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.9 [Clang Mode only] Change the default value of ={loop procedure} of the -ffj-ilfunc[={ loop  procedure}] option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.10 [Trad Mode only] Specification change of the -Krdconv option	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.11 [Clang Mode only] Changes of SIMD optimization information	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.12 [Trad Mode only] Abolition of the optimization control specifier loop_nofission	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.13 [Trad Mode only] Change of the optimization control specifier loop_fission	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (# pragma [fj] loop) is invalid	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.16 The initial value of environment variable OMP_PROC_BIND in The LLVM OpenMP Library is change	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*

Items	Previous version-level (Patch ID)												
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20 (T01887-07)	V1.0L20 (T01887-06)	V1.0L20 (T01887-05)	V1.0L20 (T01887-04)	V1.0L20 (T01887-03)	V1.0L20 (T01887-01)	V1.0L20 1st Version
2.3.17 [Clang Mode only] Incompatibility of executable programs due to the change of the libc++ version	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.18 The internal behavior of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.19 The optimization information of software pipelining is changed	-	-	-	-	-	-	-	-	-	-	-	-	*
2.3.20 [Trad Mode only] The initial value of C++ array allocated by new expression without initializer is changed	-	-	-	-	-	-	-	-	-	-	-	*	*
2.3.21 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option (*1)					-	-	-	-	-	-	*	*	*
2.3.22 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable	-	-	-	-	-	-	-	-	-	*	*	*	*
2.3.23 Changes the default of the compiler option -Karray_declaration_opt	-	-	-	-	-	-	-	-	*	*	*	*	-
2.3.24 Changing the behavior of the getopt function	-	-	-	-	-	*	*	*	*	*	*	*	*
2.3.25 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword	-	-	-	-	-	*	*	*	*	*	*	*	*
2.3.26 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified	-	-	-	-	-	*	*	*	*	*	*	*	*
2.3.27 [Clang Mode only] Change the default behavior due to the addition of the compiler options -f{debug-info-for-profiling no-debug-info-for-profiling}	-	-	-	-	*	*	*	*	*	*	*	*	*
2.3.28 [Clang Mode only] Improvement of the usability for software pipelining and zfill	-	-	-	*	*	*	*	*	*	-	-	-	-
2.3.29 [Trad Mode only] The GNU compatible option -fvisibility={default internal hidden protected} deprecated	-	-	-	*	*	*	*	*	*	*	*	*	*
2.3.30 [Clang Mode only] The compiler option -f{fj-loop-interchange fj-no-loop-interchange} deprecated	-	-	-	*	*	*	*	*	*	*	*	*	*

\*: Incompatible  
 -: compatible  
 \*1: Removed from Items due to option deprecation.

Table 2.4 List of incompatible items by version-level (MPI Library)

Items	Previous version-level (Patch ID)										
	V1.0L10 1st Version	V1.0L20A 1st Version	V1.0L20 1st Version	V1.0L20A (T01915-01)	V1.0L20 (T01887-01)	V1.0L20A (T01915-03)	V1.0L20 (T01887-03)	V1.0L20A (T01915-04)	V1.0L20 (T01887-04)	V1.0L20A (T01915-05)	V1.0L20 (T01887-05)
2.4.1 The local_options -am for the mpiexec command is changed to -tune	*	-	-	-	-	-	-	-	-	-	-
2.4.2 Executable program incompatibilities due to changes of header files and modules	*	-	-	-	-	-	-	-	-	-	-
2.4.3 Change the behavior when the MCA parameter btl_openib_receive_queues_default_num_bufs is not specified	*	*	*	*	*	*	*	*	*	*	*
2.4.4 Change the default value of the info key romio_ds_write, which specifies the behavior of Data sieving in MPI-IO, and change the processing so that a lock is taken when Data sieving is enabled	*	*	*	*	*	*	*	*	*	*	*

\*: Incompatible  
 -: compatible

Table 2.5 List of incompatible items by version-level (Profiler)

Items	Previous version-level (Patch ID)										
	V1.0L10 1st Version	V1.0L20A 1st Version	V1.0L20 1st Version	V1.0L20A (T01915-01)	V1.0L20 (T01887-01)	V1.0L20A (T01915-03)	V1.0L20 (T01887-03)	V1.0L20A (T01915-04)	V1.0L20 (T01887-04)	V1.0L20A (T01915-05)	V1.0L20 (T01887-05)
2.5.1 Change of target for counting up the cost of inlined function in Instant Performance Profiler	*	-	-	-	-	-	-	-	-	-	-
2.5.2 Change of condition for outputting time statistic information of Advanced Performance Profiler	*	-	-	-	-	-	-	-	-	-	-
2.5.3 Change of output of MPI_STARTALL routine in MPI communication cost information of Advanced Performance Profiler	*	-	-	-	-	-	-	-	-	-	-

Items	Previous version-level (Patch ID)									
	V1.0L10 1st Version	V1.0L20A 1st Version	V1.0L20 1st Version	V1.0L20A (T01915-01)	V1.0L20 (T01887-01)	V1.0L20A (T01915-03)	V1.0L20 (T01887-03)	V1.0L20A (T01915-04)	V1.0L20 (T01887-04)	V1.0L20A (T01915-05)
2.5.4 Change of output information and layout of CPU Performance Analysis Report	*	-	-	-	-	-	-	-	-	-
2.5.5 Changes of kinds of CPU Performance Analysis Report	*	-	-	-	-	-	-	-	-	-
2.5.6 Changes of output information and layout of CPU Performance Analysis Report (Data Transfer CMGs)	*	-	-	-	-	-	-	-	-	-
2.5.7 Changes of the Outputting Directory of Profile Data in Profiler	*	*	-	-	-	-	-	-	-	-
2.5.8 Change of target for counting up the cost of inlined function in Instant Performance Profiler (cancellation of 2.5.1)	-	*	*	-	-	-	-	-	-	-

\*: Incompatible

-: compatible

Table 2.6 List of incompatible items by version-level (Mathematical Libraries)

Items	Previous version-level (Patch ID)									
	V1.0L10 1st Version	V1.0L20A 1st Version	V1.0L20 1st Version	V1.0L20A (T01915-01)	V1.0L20 (T01887-01)	V1.0L20A (T01915-03)	V1.0L20 (T01887-03)	V1.0L20A (T01915-04)	V1.0L20 (T01887-04)	V1.0L20A (T01915-05)
2.6.1 Executable program incompatibilities due to changes of header files and modules	*	-	-	-	-	-	-	-	-	-
2.6.2 Changes the mathematical library linked when the compiler option -KNOSVE is specified using the native compiler	*	*	-	-	-	-	-	-	-	-

\*: Incompatible

-: compatible

Table 2.7 List of incompatible items by version-level (Other)

Items	Previous version-level (Patch ID)										
	V1.0L21 (T01920-05)	V1.0L21 (T01920-04)	V1.0L21 (T01920-03)	V1.0L21 (T01920-02)	V1.0L21 (T01920-01)	V1.0L21 1st Version	V1.0L20A (T01915-07)	V1.0L20A (T01915-06)	V1.0L20A (T01915-05)	V1.0L20A (T01915-04)	V1.0L20A (T01915-03)
<a href="#">2.7.2 Changing the License Authentication System</a>	*	*	*	*	*	*	*	*	*	*	*

\*: Incompatible

-: compatible

## 2.1 Fortran Compiler

### 2.1.1 Change of the predefined version macro

#### a. Changes

The predefined version macro is changed.

[Previous version]

The predefined version macro was as follows:

Macro	Value
__fvt_version	800

[Current version]

The predefined version macros are as follows:

Macro	Value
__FRT_major__	The major version number of the compiler
__FRT_minor__	The minor version number of the compiler
__FRT_patchlevel__	The patch level of the compiler
__FRT_version__	The string which represents the version of the compiler

#### b. Potential problems

If the old macro is used in the program, the program might not behave as expected.

#### c. Action

Modify the program to use the new macros.

### 2.1.2 Error check for OpenMP clause at compilation time

#### a. Changes

The diagnostic message is output at compilation time for mistake of OpenMP clause.

The diagnostic message jwd1813i-s is output at compilation time under the following conditions:

1. The compiler option `-Kopenmp` is effective. And,
2. A `PARALLEL` construct that has a `REDUCTION` clause appears. And,
3. A `DO`, `DO SIMD`, or `SECTIONS` construct appears inside the construct of 2. And,
4. The construct of 3 has a `LASTPRIVATE` or `REDUCTION` clause. And,
5. The list item of the `REDUCTION` clause of 2 appears in the clause of 4.

[Previous version]

The diagnostic message `jwd1813i-s` was not output at compilation time and object was created.

[Current version]

The diagnostic message `jwd1813i-s` is output at compilation time.

b. Potential problems

The diagnostic message `jwd1813-s` is output at compilation time and object is not created.

c. Action

Modify program to conform to the OpenMP specification.

## 2.1.3 Error check for polymorphic variable in OpenMP clause at compilation time

---

a. Changes

The diagnostic message is output at compilation time for a polymorphic variable in `REDUCTION` clause of OpenMP.

The diagnostic message `jwd2822i-s` is output at compilation time under the following conditions:

1. The compiler option `-Kopenmp` is effective. And,
2. A `DECLARE REDUCTION` directive that defines derived type's reduction identifier appears. And,
3. The reduction identifier of 2, and a polymorphic variable that has the derived type of 2 as declared type appear in `REDUCTION` clause.

[Previous version]

The diagnostic message `jwd2822i-s` was not output at compilation time and object was created.

[Current version]

The diagnostic message `jwd2822i-s` is output at compilation time.

b. Potential problems

The diagnostic message `jwd2822i-s` is output at compilation time and object is not created.

c. Action

Modify program as follows:

Do not specify the polymorphic variable in `REDUCTION` clause.

## 2.1.4 Change of values of macro and named constant due to OpenMP API version 4.5 support

---

a. Changes

To support OpenMP API version 4.5, the values of the following macro and named constant are changed.

- Macro `_OPENMP`
- Named constant `openmp_version` that is defined in include file `omp_lib.h` and module `omp_lib`

[Previous version]

- Specifying the compiler option -Kopenmp enabled -D\_OPENMP=201307.
- The value of the named constant openmp\_version was 201307.

[Current version]

- Specifying the compiler option -Kopenmp enables -D\_OPENMP=201511.
- The value of the named constant openmp\_version is 201511.

b. Potential problems

Behavior of the program using the macro \_OPENMP or using the named constant openmp\_version is different from before.

c. Action

Modify the program to take into account the change of their values.

## 2.1.5 Function enhancement of -Knoalias=s option

---

a. Changes

The execution result may be incorrect when -Knoalias=s option is specified.

The execution result may be incorrect under the following conditions:

1. The compiler option -Knoalias=s is effective. And,
2. Either of the following structure component appears. And,
  - a. Either of names except rightmost of structure component reference has the POINTER attribute, and the name of rightmost does not have the POINTER attribute.
  - b. Either of names of structure component reference is polymorphic, and either of names of structure component reference has the POINTER attribute.
3. The structure component with the POINTER attribute of 2 and other variable shares storage by either following:
  - a. Pointer assignment statement
  - b. Initialization
  - c. Assignment statement
  - d. ALLOCATE statement with SOURCE= specifier
  - e. Dummy argument

[Previous version]

The execution result was correct when the above condition.

[Current version]

The execution result may be incorrect when the above condition.

b. Potential problems

The execution result may be incorrect when the above condition.

c. Action

Do not specify -Knoalias=s at the compilation time.

## 2.1.6 Change the target loops of the -Kloop\_fission option

---

a. Changes

Change the loop to be analyzed for loop fission when specifying the -Kloop\_fission option.

[Previous version]

When the -Kloop\_fission option was specified, all loops in the source program were analyzed for loop fission.

[Current version]

When the -Kloop\_fission option is specified, loops with the optimization control specifier LOOP\_FFISSION\_TARGET are analyzed for loop fission.

b. Potential problems

Loop fission is not applied only by specifying the -Kloop\_fission option.

c. Action

Specify the optimization control specifier LOOP\_FFISSION\_TARGET[({CL|LS})] for loops to which you want to apply loop fission, and specify the -Kloop\_fission and -Kocl options at the same time. Specify the following compiler options or the optimization control specifiers to change the status of loop fission to improve performance further. Note that the loop fission is not applied when the effect of loop fission cannot be expected.

- The -Kloop\_fission\_threshold=N option or the optimization control specifier LOOP\_FFISSION\_THRESHOLD(N).
- The -Kloop\_fission\_stripmining[={N|L1|L2}] option or the optimization control specifier LOOP\_FFISSION\_STRIPMINING[({N|'L1'|'L2'})].

## 2.1.7 Change the default value of kind of the -Kilfunc[=kind] option

---

a. Changes

The default value of *kind* of the -Kilfunc[=kind] option is changed.

[Previous version]

If a value was not specified for *kind* of the -Kilfunc[=kind] option or the -Kfast option was specified, -Kilfunc=loop was set.

[Current version]

If a value is not specified for *kind* of the -Kilfunc[=kind] option or the -Kfast option is specified, -Kilfunc=procedure is set.

b. Potential problems

The -Kilfunc option may occur calculation errors in the execution results.

Since the default value of the -Kilfunc[=kind] option is changed from "loop" to "procedure", the optimization targets other than loops in addition and the calculation error caused by the execution may change.

c. Action

If you want to revert to the before, specify the -Kilfunc=loop option.

## 2.1.8 Abolition of the optimization control specifier LOOP\_NOFISSION

---

a. Changes

The optimization control specifier LOOP\_NOFISSION is abolished.

[Previous version]

The optimization control specifier LOOP\_NOFISSION was effective.

[Current version]

The optimization control specifier LOOP\_NOFISSION is invalid.

b. Potential problems

The optimization control specifier LOOP\_NOFISSION becomes invalid. If this specifier is specified, the following warning is output.

```
jwdl202i-i "filename", line number: Invalid optimization control specifier in optimization control line.
```

c. Action

You do not need to disable loop fission by the previous optimization control specifier `LOOP_NOFISSION`, because loop fission only works on loops with optimization control specifier `LOOP_FISSION_TARGET`. Therefore, no action is required.

## 2.1.9 Change of the optimization control specifier `LOOP_FISSION`

---

a. Changes

Change the optimization control specifier `LOOP_FISSION` to `LOOP_FISSION_TARGET[({CL|LS})]`. The way of specifying changes as follows:

- The optional parameters of the optimization control specifier can specify the algorithm of the loop fission.
- Only the loops which optimization control specifiers are specified.

[Previous version]

The optimization control specifier `LOOP_FISSION` was effective.

[Current version]

The optimization control specifier `LOOP_FISSION` is invalid.

b. Potential problems

The optimization control specifier `LOOP_FISSION` becomes invalid. If this specifier is specified, the following warning is output.

```
jwdl202i-i "filename", line number: Invalid optimization control specifier in optimization control line.
```

c. Action

Change the optimization control specifier `LOOP_FISSION` to `LOOP_FISSION_TARGET[({CL|LS})]`. You can also specify algorithm for loop fission by the parameters of the optimization control specifier. If the description position of the optimization control line is in a program unit or DO loop unit which is an outer loop of the target, change the description position to the DO loop unit of the target loop to be fissioned.

## 2.1.10 Incompatibility of the object due to changes in procedure call standard

---

a. Changes

Change procedure call standards

[Previous version]

The scalable vector registers to be saved in callee were z8-z31, and SIMD and Floating-Point registers to be saved in callee were v8-v31.

[Current version]

The scalable vector registers to be saved in callee are z8-z23, and SIMD and Floating-Point registers to be saved in callee are v8-v23.

b. Potential problems

When linking objects on the previous standards and the new standards generated from programs which use the `declare simd` construct of OpenMP or objects on the previous standards generated from programs which use functions and operations that are subject to multi-operation functions, the execution result may be incorrect.

c. Action

In the following case, do not link objects on the previous standards, and recompile the program.

- The `declare simd` directive of OpenMP is used.

- The objects are generated with the -Kfast option or -Kmfunc[={1|2|3}] option and the multi-operation function optimization is applied. (\*)

\*) This can be confirmed by the output of the compiler message jwd8300o-i when -Koptmsg=2 is specified.

## 2.1.11 Type conversion options in LLVM OpenMP Library

---

### a. Changes

When runtime option -Wl,-Lb,-Li is used in LLVM OpenMP Library, an argument and a return value of service subroutine become 8byte.

[Previous version]

When runtime option -Wl,-Lb,-Li is used in LLVM OpenMP Library, execution is aborted outputting runtime message jwe1501i-u.

[Current version]

When runtime option -Wl,-Lb,-Li is used, an argument and a return value of service subroutine become 8byte.

### b. Potential problems

Runtime option, environmental variable and error monitor can't control the message jwe1501i-u.

### c. Action

None.

## 2.1.12 The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is change

---

### a. Changes

The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is changed.

[Previous version]

The initial value of environment variable OMP\_PROC\_BIND was 'FALSE'.

[Current version]

The initial value of environment variable OMP\_PROC\_BIND is 'CLOSE'.

### b. Potential problems

Performance improvement can be expected.

### c. Action

There is no necessity for handle.

## 2.1.13 Executable program incompatibilities due to changes of header files and modules

---

### a. Changes

Header files and modules are modified.

MPI header files and modules are not compatible between previous versions and current version.

[Previous version]

Executable programs created in previous version were used.

[Current version]

Executable programs created in previous version cannot be used in current version.

b. Potential problems

If an executable program created in previous version is executed in current version, the program might not behave as expected.

c. Action

The following action is required.

- If the COARRAY feature of Fortran is used in a program, the object program has to be relinked. If no object program remains, the source program has to be recompiled and relinked.

## 2.1.14 The internal behavior of software pipelining is changed

---

a. Changes

The instruction scheduling algorithm used in software pipelining, performed by the options -Kswp, -Kswp\_weak, or -Kswp\_strong, by default is changed.

[Previous version]

A specific algorithm was always used.

[Current version]

A suitable algorithm is selected from multiple ones automatically by default.

b. Potential problems

A selected algorithm may differ from the specific one used by the previous version. It may lead to a performance degradation.

c. Action

You can specify a preferred algorithm by the option -Kswp\_policy. The algorithm used by the previous version can be specified by -Kswp\_policy=small.

## 2.1.15 The optimization information of software pipelining is changed

---

a. Changes

An item is added to the optimization information of software pipelining, which is displayed by -Nlst=t.

[Previous version]

There were three items, IPC, ITR, and MVE, displayed as follows.

```
<<< Loop-information Start >>>
<<< [OPTIMIZATION]
<<<     SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2)
<<< Loop-information End >>>
5      1      do i=1,n
      ...
10     1      enddo
```

[Current version]

An item POL which means the applied instruction scheduling algorithm is added. It shows the actual algorithm selected for the loop by -Kswp\_policy=auto. An example is as follows.

```
<<< Loop-information Start >>>
<<< [OPTIMIZATION]
<<<     SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2, POL: S)
<<< Loop-information End >>>
5      1      do i=1,n
      ...
10     1      enddo
```

b. Potential problems

The displayed format is changed.

c. Action

None.

## 2.1.16 Change of the attribute information of the compilation information for intrinsic function name in constant expression

---

a. Changes

The attribute information for the name of the intrinsic function that is output to the compilation information is changed if all the following conditions are satisfied.

1. The compiler option `-Nlst=a` or `-Nlst=x` is enabled.
2. One of the following is satisfied.
  - 2.1. The intrinsic function of "3." is referenced in the constant expression of the kind type parameter, length type parameter, array bounds, enumerator, initialization of type declaration statement, `PARAMETER` statement, case value, data statement object, `STOP` statement, or `ERROR STOP` statement.
  - 2.2. All the following conditions are satisfied.
    - 2.2.1. The compiler option `-Kopenmp` or `-Kopenmp_simd` is enabled.
    - 2.2.2. The `COLLAPSE`, `SAFELEN`, `SIMDLLEN`, `ORDERED`, `HINT`, or `ALIGNED` clause appears in the OpenMP construct.
    - 2.2.3. The intrinsic function of "3." is referenced in the constant expression of "2.2.2."
  - 2.3. All the following conditions are satisfied.
    - 2.3.1. The compiler option `-O2` or higher is enabled.
    - 2.3.2. All the component specifications of the structure constructor are constant expressions.
    - 2.3.3. The intrinsic function of "3." is referenced in the constant expression of "2.3.2."
    - 2.3.4. The actual arguments of "2.3.3." are all constant expressions.

3. One of the following is satisfied.

3.1. One of the following intrinsic functions is referenced.

ACOS, ACOSD, ACOSH, ACOSQ, AMIN0, ASINH, ATANH, BESSEL_J0, BESSEL_J1, BESSEL_JN, BESSEL_Y0, BESSEL_Y1, BESSEL_YN, BGE, BGT, BLE, BLT, COTAND, COTANQ, DACOS, DACOSD, DACOSQ, DASIN, DCOSQ, DCOTAND, DCOTANQ, DSHIFTL, DSHIFTR, ERFC_SCALED, FINDLOC, HFIX, HYPOT, IALL, IANY, IPARITY, IS_CONTIGUOUS, IS_IOSTAT_END, IS_IOSTAT_EOR, LEADZ, LOG_GAMMA, MASKL, MASKR, MERGE_BITS, MIN1, NORM2, PARITY, POPCNT, POPPAR, RANK, SELECTED_CHAR_KIND, SHIFTA, SHIFTL, SHIFTR, STORAGE_SIZE, TRAILZ
---

3.2. All the following conditions are satisfied.

- 3.2.1. The compiler option `-Nobsfun` is enabled.
- 3.2.2. One of the following intrinsic functions is referenced.

AIMIN0, AJMIN0, I2NINT, IIDINT, IIFIX, IINT, IMIN1, JIDNNT, JMIN1
---

[Previous version]

The attribute information of the intrinsic function name output in the compilation information was "external function name".

[Current version]

The attribute information of the intrinsic function name output in the compilation information is "intrinsic function name".

b. Potential problems

The content of the compilation information is changed.

c. Action

It is not necessary to action.

## 2.1.17 Output compiler message for the allocatable assignment of the Fortran 2003 or later standard

---

a. Changes

Change the compiler message for the allocatable assignment of the Fortran 2003 or later standard.

- When the compiler option `-Nnoalloc_assign` is enabled, the compiler message `jwd2754i-i` is output if the allocatable assignment of the Fortran 2003 or later standard is not operated.
- Change the message text of the compiler message `jwd2881i-i`.

[Previous version]

- The compiler message `jwd2754i-i` was not output.
- The message text was "The allocatable assignment of the Fortran 2003 specification is operated."

[Current version]

- The compiler message `jwd2754i-i` is output.
- The message text is "The allocatable assignment of the Fortran 2003 or later standard is operated."

b. Potential problems

The compiler message is changed.

c. Action

It is not necessary to action.

Specify the compiler option `-f2754` to suppress the output of the compiler message `jwd2754i-i`.

## 2.1.18 Change the default of the compiler option `-Nalloc_assign` for the compiler option `-X`

---

a. Changes

When the compiler option `-X03` or `-X08` is enabled, change the default of the compiler option `-N[No]alloc_assign` from `-Nnoalloc_assign` to `-Nalloc_assign`. The compiler message `jwd2881i-i` may be output because the default was changed to `-Nalloc_assign`.

[Previous version]

When the compiler option `-X03` or `-X08` was enabled, `-Nnoalloc_assign` was set by default.

The compiler message `jwd2881i-i` was not output.

[Current version]

When the compiler option `-X03` or `-X08` is enabled, `-Nalloc_assign` is set by default.

The compiler message `jwd2881i-i` is output.

b. Potential problems

When the compiler option `-X03` or `-X08` is enabled, and `-Nalloc_assign` or `-Nnoalloc_assign` is not specified, then `-Nalloc_assign` is enabled and the allocatable assignment of the Fortran 2003 standard is operated, the execution performance of the assignment statement in which the left side of the assignment statement is allocated and the shapes of the left and right sides match (Fortran 95 language specification) may decrease.

Note that `-X03` is set by default when the file suffix is `.f03` or `.F03`, and `-X08` is set by default when the file suffix is `.f`, `.F`, `.for`, `.FOR`, `.f08` or `.F08`.

Also, because -Nalloc\_assign is enabled and the allocatable assignment of the Fortran 2003 standard is operated, the compiler message jwd2881i-i may be output.

c. Action

For the source program of Fortran 95 language specification, enable the compiler option -Nnoalloc\_assign.

Specify the compiler option -f2881 to suppress the output of the compiler message jwd2881i-i.

## 2.1.19 Changes of the INQUIRE and OPEN specifications when a directory is specified in the FILE specifier

---

a. Changes

The specification is changed when a directory name that does not end in "/" is specified in the INQUIRE and OPEN statements.

[Previous version]

Runtime message (jwe0103i-e) was not output when a directory name that does not end in "/" was specified in the FILE specifier of the INQUIRE statement. The inquiry specifier had a query result value.

Also, if a directory name whose ending is not "/" is specified in the FILE specifier of the OPEN statement, one of the following behaviors occurs.

- A runtime message(jwe0072i-e) was printed for an I/O statement that was executed immediately after the OPEN statement, and the OPEN statement was disabled.
- A runtime message(jwe0022i-s) was printed for the OPEN statement, and the program ended.

[Current version]

If a directory name that does not end in "/" is specified in the FILE specifier of the INQUIRE statement, a runtime message(jwe0103i-e) is output. The INQUIRE statement is disabled and no value is set for the inquiry specifier.

Also, if a directory name whose ending is not "/" is specified in the FILE specifier of the OPEN statement, a runtime message(jwe0102i-e) is output and the OPEN statement becomes invalid.

b. Potential problems

When you query a directory using the INQUIRE statement, a runtime message(jwe0103i-e) is output, the INQUIRE statement is disabled, and no value is set for the query specifier. Therefore, a program that determines the value set in the inquiry specifier may have a different execution behavior.

Also, when a directory is opened, a runtime message(jwe0102i-e) is printed, and the OPEN statement is disabled. If an I/O statement is executed immediately after the OPEN statement, it is input to and output from a named file(fort.nn).

Example:

```
CHARACTER(10),PARAMETER::FILEN='./work'
LOGICAL::CODE/.FALSE./
  INQUIRE(FILE=FILEN,EXIST=CODE)
  IF (CODE.EQV..FALSE.)THEN
    OPEN(10,FILE=FILEN,STATUS='NEW')
    WRITE(10,*)'NEW'
  ELSE
    OPEN(10,FILE=FILEN,STATUS='OLD')
    WRITE(10,*)'OLD'
  ENDIF
END
```

Execution Results:

```
jwe0103i-e line 3 The value of the FILE specifier in an INQUIRE statement is invalid.
error occurs at MAIN__ line 3 loc 0000000000401030 offset 000000000000003c
MAIN__ at loc 0000000000400ff4 called from o.s.
taken to (standard) corrective action, execution continuing.
jwe0102i-e line 5 The value of the FILE specifier in an OPEN statement is invalid (unit=10).
error occurs at MAIN__ line 5 loc 0000000000401068 offset 0000000000000074
```

```

MAIN__      at loc 0000000000400ff4 called from o.s.
taken to (standard) corrective action, execution continuing.
error summary (Fortran)
error number  error level  error count
jwe0102i      e           1
jwe0103i      e           1
total error count = 2

-rw-rw-r-- 1 xxx 5 Dec  9 19:23 fort.10

```

c. Action

If you want to suppress runtime messages(jwe0103i-e, jwe0102i-e), use the ERRSET service subroutine or the runtime option(-Wl,-Re). The use of the value of the query specifier should be changed to match the revised specification.

## 2.1.20 Improving time resolution in the **SYSTEM\_CLOCK** built-in subroutine

a. Changes

In **SYSTEM\_CLOCK**([COUNT, COUNT RATE, COUNT MAX]), the return value is changed when supplied with variables of type eight-byte integer, double, and quadruple precision real.

[Previous version]

The following values were set according to the type of COUNT\_RATE and COUNT\_MAX.

	COUNT_RATE and COUNT_MAX type						
	One-byte integer	Two-byte integer	Four-byte integer	Eight-byte integer	Real	Double precision real	Quadruple precision real
COUNT_RATE value	1	1000	1000	1000	1000	1000	1000
COUNT_MAX value	127	32767	86399999	86399999	Not available		

[Current version]

If COUNT is specified, the following values for COUNT type are set to COUNT\_RATE and COUNT\_MAX.

		COUNT type			
		One-byte integer	Two-byte integer	Four-byte integer	Eight-byte integer
COUNT_RATE value	One-byte integer	1	0 (*1)	0 (*1)	0 (*1)
	Two-byte integer	1	1000	1000	0 (*1)
	Four-byte integer	1	1000	1000	1000000 (*2)
	Eight-byte integer	1	1000	1000	1000000 (*2)
	Real	1	1000	1000	1000000 (*2)
	Double precision real	1	1000	1000	1000000 (*2)
	Quadruple precision real	1	1000	1000	1000000 (*2)
COUNT_MAX value	One-byte integer	127	0 (*1)	0 (*1)	0 (*1)
	Two-byte integer	127	32767	0 (*1)	0 (*1)
	Four-byte integer	127	32767	86399999	0 (*1)
	Eight-byte integer	127	32767	86399999	86399999999 (*2)

If COUNT is not specified, the following values are set according to the type of COUNT\_RATE and COUNT\_MAX.

	COUNT_RATE and COUNT_MAX type						
	One-byte integer	Two-byte integer	Four-byte integer	Eight-byte integer	Real	Double precision real	Quadruple precision real
COUNT_RATE value	1	1000	1000	1000000 (*2)	1000	1000000 (*2)	1000000 (*2)
COUNT_MAX value	127	32767	86399999	86399999999 (*2)	Not available		

\*1) Since the type of COUNT\_RATE or COUNT\_MAX is less than the type of COUNT and the value exceeds the maximum value of the type, COUNT is set to -HUGE (COUNT) and COUNT\_RATE and COUNT\_MAX are set to 0.

\*2) This modification has changed the set value.

b. Potential problems

- When a program outputs the COUNT, COUNT\_RATE, and COUNT\_MAX values, the output values change in the cases of \*1 or \*2 above.
- A program that uses the COUNT, COUNT\_RATE, and COUNT\_MAX values to calculate time may produce a different execution result in the case of \*1 above.

c. Action

If your program uses the COUNT, COUNT\_RATE, and COUNT\_MAX values to calculate time and \*1 above applies, change the argument type so that the value does not exceed the maximum value of the type.

## 2.1.21 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable

a. Changes

The controlling for the Sector cache for the first level cache is changed available when the Sector cache for the second level cache is unavailable.

[Previous version]

The Sector cache for the first level cache was unavailable when the Sector cache for the second level cache was unavailable.

[Current version]

The Sector cache for the first level cache is available when the Sector cache for the second level cache is unavailable.

b. Potential problems

- The runtime message jwe1047i-w is not output.
- The performance may increase or decrease.
  - Fujitsu's mathematical libraries  
The performance may increase.
  - Program describing OCL for controlling the Sector cache  
The performance may increase or decrease.

c. Action

If the environment variable FLIB\_L1\_SCCR\_CNTL is assigned to the value FALSE, the Sector cache for the first level cache is unavailable when the Sector cache for the second level cache is unavailable.

## 2.1.22 Changing the return value of SIZE in the RANDOM\_SEED Built-in Subroutine

---

### a. Changes

In RANDOM\_SEED ([SIZE, PUT, GET]), the return value is changed when SIZE is specified in the argument.

[Previous version]

If SIZE was specified as an argument, 1 was returned.

[Current version]

If SIZE is specified as an argument, 2 is returned.

### b. Potential problems

- When a program outputs the SIZE, the output values is changed.
- The PUT or GET argument is an array of rank one and size greater than or equal to SIZE. Thus, if a program specifies an array of rank one and size less than or equal to 1, the runtime message jwe1304i-e is output.

### c. Action

If a program that is executing PUT or GET as an argument outputs runtime message jwe1304i-e, specify an array of rank one and size more than or equal to 2 as the PUT or GET argument.

## 2.1.23 Error check for the SELECT TYPE construct conforming to the Fortran standard at compilation time

---

### a. Changes

The diagnostic message jwd2489i-s, jwd2330i-s, jwd2104i-s, or jwd2361i-s is output for the program with incorrect SELECT TYPE construct at compilation time.

#### a. The diagnostic message jwd2489i-s is output at compilation time under the following conditions:

1. A SELECT TYPE construct appears. And,
2. A variable appears in the selector of 1. And,
3. An associate name appears in 1. And,
4. The variable of 2. has an ALLOCATABLE attribute. And,
5. The type guard statement of 1. has a "CLASS IS" or "CLASS DEFAULT". And,
6. An assignment statement appears in the block of type guard statement of 5. And,
7. The associate name of 3. appears in the left side of assignment statement of 6.

#### b. The diagnostic message jwd2330i-s is output at compilation time under the following conditions:

1. A SELECT TYPE construct appears. And,
2. A structure component appears in the selector of 1. And,
3. An associate name appears in 1. And,
4. The structure component of 2. has an ALLOCATABLE or POINTER attribute. And,
5. An intrinsic function ALLOCATED or ASSOCIATED appears in the block of type guard statement of 1. And,
6. The associate name of 3. appears in the actual argument of intrinsic function of 5.

#### c. The diagnostic message jwd2104i-s or jwd2361i-s is output at compilation time under the following conditions:

1. A SELECT TYPE construct appears. And,
2. A variable appears in the selector of 1. And,
3. An associate name appears in 1. And,

4. The variable of 2. is of an unlimited polymorphic. And,
5. The type guard statement of 1. has a "TYPE IS". And,
6. An intrinsic type specifier appears in the type guard statement of 1. And,
7. The variable of 4. appears in the block of type guard statement of 5.

[Previous version]

The diagnostic message jwd2489i-s, jwd2330i-s, jwd2104i-s, or jwd2361i-s was not output at compilation time and object was created.

[Current version]

The diagnostic message jwd2489i-s, jwd2330i-s, jwd2104i-s, or jwd2361i-s is output at compilation time and object is not created.

b. Potential problems

The diagnostic message jwd2489i-s, jwd2330i-s, jwd2104i-s, or jwd2361i-s is output at compilation time and object is not created.

c. Action

Modify program as following:

- Specify the variable for selector instead of the variable for associate name in the left side of assignment statement.
- Specify the variable for selector instead of the variable for associate name in the actual argument of intrinsic function.
- Specify the variable for associate name instead of the variable for selector in the block.

## 2.1.24 Error check at compilation time for detect execution time errors due to compiler failure

---

a. Changes

The diagnostic message jwd3999i-w is output when compiler failure can cause a program to end abnormally or incorrect result at execution time.

The diagnostic message jwd3999i-w is output when either a. or b. below is satisfied.

- a.
  1. An ALLOCATE statement appears. And,
  2. The ALLOCATE statement of 1. has a SOURCE specifier. And,
  3. The allocate object of 1. is polymorphic or has a polymorphic component. And,
  4. A variable, function reference, or structure constructor appears in the SOURCE specifier of 2. And,
  5. The object of 4. is polymorphic or has a polymorphic component.
- b.
  1. An assignment statement appears. And,
  2. The left side of 1. is polymorphic or has a polymorphic component. And,
  3. A variable, function reference, or structure constructor appears on the right side of 1. And,
  4. The object of 3. is polymorphic or has a polymorphic component.

[Previous version]

The diagnostic message jwd3999i-w was not output at compilation time.

[Current version]

The diagnostic message jwd3999i-w is output at compilation time.

b. Potential problems

The diagnostic message jwd3999i-w is output at compilation time.

c. Action

If the diagnostic message jwd3999i-w is output, refer to [Programmer Response] in the "Fortran Compiler Messages" that is provided with this product.

## 2.1.25 Changes the runtime message that is printed when an error is detected in the OPEN statement where the NEWUNIT= specifier appears

---

a. Changes

If neither the FILE= specifier nor STATUS= specifier with a value of SCRATCH appears in the OPEN statement where the NEWUNIT= specifier appears, the runtime message jwe1125i-e is output.

[Previous version]

If neither the FILE= specifier nor the STATUS = specifier with the value of SCRATCH appears in the OPEN statement with the NEWUNIT= specifier that is executed after the CLOSE statement, the runtime message jwe0022i-s was output and the program had finished executing.

[Current version]

If neither the FILE= specifier nor the STATUS = specifier with the value of SCRATCH appears in the OPEN statement with the NEWUNIT= specifier that is executed after the CLOSE statement, the runtime message jwe1125i-e is output and the program continues executing.

b. Potential problems

If an OPEN statement containing the IOSTAT= specifier is executed, the value returned to the IOSTAT= specifier will be different. Therefore, a program that determines the value of the IOSTAT= specifier may produce a different execution result.



### Example

---

Example Program

```
CHARACTER(10), DIMENSION(2)::ST/"SCRATCH", "REPLACE"/
DO I=1,2
  OPEN(NEWUNIT=IU, STATUS=ST(I), IOSTAT=IOS)
  SELECT CASE (IOS)
    CASE (0)
      WRITE(IU,*) 'TEST1'
      CLOSE(IU)
    CASE (1)
      PRINT*, IOS, I
      GOTO 11
    CASE (2:)
      PRINT*, IOS, I
      IU=-IU
      WRITE(IU,*) 'TEST2'
      CLOSE(IU)
  END SELECT
ENDDO
END
```

Execution Result

```
1125 2
```

---

c. Action

It is not necessary to action.

## **2.1.26 Changes the default of the compiler option -Karray\_declaration\_opt**

---

### a. Changes

The default of the -Karray\_declaration\_opt option is changed.

[Previous version]

The -Karray\_declaration\_opt option was valid by default.

[Current version]

The -Karray\_declaration\_opt option is invalid by default.

### b. Potential problems

When the -Karray\_declaration\_opt option is not explicitly specified, performance may be degraded.

### c. Action

Specify the -Karray\_declaration\_opt option.

## **2.1.27 Changes of the options to be enabled when the GNU compatible option -fopenmp is specified to be equivalent to the compiler option -Kopenmp**

---

### a. Changes

Changes of the options to be enabled when the GNU compatible option -fopenmp is specified to be equivalent to the compiler option -Kopenmp.

[Previous version]

The compiler options -Kthreadsafe and -Kauto were not enabled when the GNU compatible option -fopenmp was specified.

[Current version]

The compiler options -Kthreadsafe and -Kauto are enabled when the GNU compatible option -fopenmp is specified.

### b. Potential problems

The effective options of the compilation information are changed.

### c. Action

It is not necessary to action.

## **2.1.28 Value of COUNT\_MAX for SYSTEM\_CLOCK intrinsic subroutine of Fujitsu Fortran Compiler is improved to be able to count execution time greater than one day**

---

### a. Changes

The maximum value of COUNT for SYSTEM\_CLOCK intrinsic subroutine of Fujitsu Fortran Compiler is changed.

[Previous version]

The maximum value of COUNT for SYSTEM\_CLOCK intrinsic subroutine was 86399999 for 4-byte integer type, and 86399999999 for 8-byte integer type.

[Current version]

The maximum value of COUNT for SYSTEM\_CLOCK intrinsic subroutine is 2147483647 for 4-byte integer type, and 9223372036854775807 for 8-byte integer type.

### b. Potential problems

The output result changes for a program in which the value of COUNT exceeds the maximum value of COUNT of a previous version or the value of COUNT\_MAX is printed if the type of variable assigned for COUNT or COUNT\_MAX is 4- or 8-byte integer.

c. Action

It is not necessary to action.

## 2.2 C Compiler

---

### 2.2.1 [Trad Mode only] Change of the predefined version macro

---

a. Changes

**[Trad Mode only]**

The predefined version macro is changed.

**[Previous version]**

The predefined version macro was as follows:

Macro	Value
__fcc_version	800

**[Current version]**

The predefined version macros are as follows:

Macro	Value
__FCC_major__	The major version number of the compiler
__FCC_minor__	The minor version number of the compiler
__FCC_patchlevel__	The patch level of the compiler
__FCC_version__	The string which represents the version of the compiler

b. Potential problems

If the old macro is used in the program, the program might not behave as expected.

c. Action

Modify the program to use the new macros.

### 2.2.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed

---

a. Changes

**[Trad Mode only]**

In the compilation, the output of debugging information to the unneeded function not used in the program is suppressed when -O0 option is effective and -g option is specified.

**[Previous version]**

In the compilation, debugging information on the unneeded function not used in the program was output when -O0 option is effective and -g option is specified.

**[Current version]**

In the compilation, debugging information on the unneeded function not used in the program is not output when -O0 option is effective and -g option is specified.

b. Potential problems

The function not used in the program cannot be debugged.

c. Action

It is not necessary to action.

## 2.2.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support

---

a. Changes

**[Clang Mode only]**

To support OpenMP API version 4.5, the value of macro `_OPENMP` is changed for Clang Mode.

**[Previous version]**

When the compiler option `-Nclang` and `-Kopenmp` (or `-fopenmp`) were specified, the value of macro `_OPENMP` was 201107.

**[Current version]**

When the compiler option `-Nclang` and `-Kopenmp` (or `-fopenmp`) are specified, the value of macro `_OPENMP` is 201511.

b. Potential problems

Behavior of the program using the macro `_OPENMP` is different from before.

c. Action

Modify the program to take into account the change of the value.

## 2.2.4 Obsolescence of the compiler option `-Klto`

---

a. Changes

The `-Klto` option is obsolete.

**[Previous version]**

The `-Klto` option was valid.

**[Current version]**

The `-Klto` option is invalid.

b. Potential problems

When the `-Klto` option is specified in Trad Mode, the following warning message is output.

```
warning: lto sub-option specified in -K option is invalid.
```

When the `-Klto` option is specified in Clang Mode, the following error message is output.

```
error: unknown argument '-Klto', did you mean '-flto='?
```

c. Action

To apply link time optimization, specify the `-flto` option in Clang Mode.

## 2.2.5 [Clang Mode only] Remove the compiler option `-f{plt|no-plt}`

---

a. Changes

**[Clang Mode only]**

The `-f{plt|no-plt}` option is not supported in Clang Mode (remove from the manual).

**[Previous version]**

The `-f{plt|no-plt}` option was described in the manual as an option of Clang Mode.

[Current version]

If the source program was compiled using the `-f{plt|no-plt}` option in Clang Mode, this option may not have the effect you expect, so remove it from the manual.

b. Potential problems

No impact for this change.

(If `-fplt` has been specified previously, it has no effect because it does not have code output using PLT.)

c. Action

It is not necessary to action.

## 2.2.6 [Clang Mode only] Remove the compiler option `-mtune`

---

a. Changes

[Clang Mode only]

The `-mtune` option is not supported in Clang Mode (remove from the manual).

[Previous version]

The `-mtune` option description was described in the manual as an option of Clang Mode.

[Current version]

In the case that the source program is translated using the `-mtune` option of Clang Mode, the expected effect of this option may not be obtained, so this option is deleted from the manual.

b. Potential problems

No impact for this change.

(If `-mtune` has been specified previously, it has no effect because it does not have any special optimizations or code output.)

c. Action

For architecture optimization, please specify the `-mcpu={generic|thunderx|thunderx2t99}` option.

## 2.2.7 [Trad Mode only] Change the target loops of the `-Kloop_fission` option

---

a. Changes

[Trad Mode only]

Change the loop to be analyzed for loop fission when specifying the `-Kloop_fission` option.

[Previous version]

When the `-Kloop_fission` option was specified, all loops in the source program were analyzed for loop fission.

[Current version]

When the `-Kloop_fission` option is specified, loops with the optimization control specifier `loop_fission_target` are analyzed for loop fission.

b. Potential problems

Loop fission is not applied only by specifying the `-Kloop_fission` option.

c. Action

Specify the optimization control specifier `loop_fission_target [cl|ls]` for loops to which you want to apply loop fission, and specify the `-Kloop_fission` and `-Kocl` options at the same time. Specify the following compiler options or the optimization control specifiers to change the status of loop fission to improve performance further. Note that the loop fission is not applied when the effect of loop fission cannot be expected.

- The `-Kloop_fission_threshold=N` option or the optimization control specifier `loop_fission_threshold n`.
- The `-Kloop_fission_stripmining[={N|L1|L2}]` option or the optimization control specifier `loop_fission_stripmining [n]"L1"|"L2"]`.

## 2.2.8 [Trad Mode only] Change the default value of kind of the -Kilfunc[=kind] option

---

### a. Changes

#### [Trad Mode only]

The default value of *kind* of the -Kilfunc[=*kind*] option is changed.

#### [Previous version]

If a value was not specified for *kind* of the -Kilfunc[=*kind*] option or the -Kfast option was specified, -Kilfunc=loop was set.

#### [Current version]

If a value is not specified for *kind* of the -Kilfunc[=*kind*] option or the -Kfast option is specified, -Kilfunc=procedure is set.

### b. Potential problems

The -Kilfunc option may occur calculation errors in the execution results.

Since the default value of the -Kilfunc[=*kind*] option is changed from "loop" to "procedure", the optimization targets other than loops in addition and the calculation error caused by the execution may change.

### c. Action

If you want to revert to the before, specify the -Kilfunc=loop option.

## 2.2.9 [Clang Mode only] Change the default value of ={loop|procedure} of the -ffj-ilfunc[={loop|procedure}] option

---

### a. Changes

#### [Clang Mode only]

The default value of ={loop|procedure} of the -ffj-ilfunc option[={loop|procedure}] is changed.

#### [Previous version]

If a value was not specified for ={loop|procedure} of the -ffj-ilfunc option or the -Ofast option was specified, -ffj-ilfunc=loop was set.

#### [Current version]

If a value is not specified for ={loop|procedure} of the -ffj-ilfunc option or the -Ofast option is specified, -ffj-ilfunc=procedure is set.

### b. Potential problems

The -ffj-ilfunc option may occur calculation errors in the execution results.

Since the default value of the -ffj-ilfunc={loop|procedure} option is changed from "loop" to "procedure", the optimization targets other than loops in addition and the calculation error caused by the execution may change.

### c. Action

If you want to revert to the before, specify the -ffj-ilfunc=loop option.

## 2.2.10 [Trad Mode only] Specification change of the -Krdconv option

---

### a. Changes

#### [Trad Mode only]

Change the specification of the -Krdconv option to -Krdconv[=*level*] and control optimization depending on *level*.

#### [Previous version]

- If the -Kfast option was specified, the -Krdconv option was set.

- When the `-Krdconv` option was set:  
Optimization was performed on the assumption that operations of 4-byte integer type do not overflow or wrap around.

[Current version]

- If the `-O2` option or higher is set, the `-Krdconv=1` option is set by default. The `-Krdconv[=level]` option controls the optimization according to *level*. The default value of *level* is 1.
- When the `-Krdconv[=1]` option is set:  
Optimization is performed on the assumption that operations of signed integer type whose size is 4-byte or less do not overflow.
- When the `-Krdconv=2` option is set:  
In addition to the `-Krdconv=1` option, optimization is performed on the assumption that operations of signed integer type whose size is 4-byte or less do not overflow.

#### b. Potential problems

- When the `-Kfast` option or `-Krdconv` option is set:  
Optimization is not performed on the assumption that operations of 4-byte unsigned integer type do not wrap around. This may result in execution performance decrease. It is decided to avoid deriving the `-Krdconv` option from the commonly-used `-Kfast` option and to conform to industry standards.
- When the `-O2` option or higher is set:  
If a program occurs overflows in operations of signed integer type whose size is 4-byte or less, the program execution may be aborted or the execution result may be incorrect. Note that according to the language standard, overflow of signed integer arithmetic results in undefined behavior.

#### c. Action

- When the `-Kfast` option or `-Krdconv[=1]` option is set:  
Specify the `-Krdconv=2` option to perform optimization on the assumption that operations of 4-byte unsigned integer type do not wrap around.
- When the `-O2` option or higher is set:  
It is recommended modify a program if the program occurs overflows in operations of integer type whose size is 4-byte or less. Alternatively, you can avoid the potential problems by specifying the `-Knordconv` option.

## 2.2.11 [Clang Mode only] Changes of SIMD optimization information

#### a. Changes

##### [Clang Mode only]

Changes compilation information and compiler message of SIMD optimization.

##### 1. Compilation information

The compilation information (optimization information) by the compile option `-ffj-lst=t` is changed as follows.

[Previous version]

For SVE vectorized loop, the number of array elements to be processed by one SIMD instruction was shown as "(VL: *length*[,*length*]...)".

```
<<< Loop-information Start >>>
<<<  [OPTIMIZATION]
<<<    SIMD(VL: 4 Interleave: 1)
<<< Loop-information End >>>
v   for (int i = 0; i < N; ++i) {
      a[i] = b[i];
    }
```

[Current version]

The optimization information of the SVE vectorized loop is output as follows. The output format "(VL: AGNOSTIC; VL: *length*[,*length*]... in 128-bit)" means that the loop is vectorized and the SVE vector registers are not fixed to a particular size.

Length shows the number of array elements processing in one SIMD instruction, assuming that the vector register size of the SVE is 128 bits.

```
<<< Loop-information Start >>>
<<< [OPTIMIZATION]
<<< SIMD(VL: AGNOSTIC; VL: 4 in 128-bit Interleave: 1)
<<< Loop-information End >>>
v   for (int i = 0; i < N; ++i) {
      a[i] = b[i];
    }
```

## 2. Compiler message

The compilation message by the compile option `-Rpass=.*` is changed as follows.

### [Previous version]

When loop was SVE vectorized, output the compiler message as follows.

```
test.c:5:3: remark: vectorized loop (vectorization width: 4, interleaved count: 1) [-Rpass=sve-loop-vectorize]
```

### [Current version]

Add SVE vector register size "bit width: 128" to SIMD information as follows.

```
test.c:5:3: remark: vectorized loop (vectorization width: 4, bit width: 128, interleaved count: 1) [-Rpass=sve-loop-vectorize]
```

## b. Potential problems

The output compilation information is changed.

## c. Action

No action is required.

## 2.2.12 [Trad Mode only] Abolition of the optimization control specifier `loop_nofission`

---

### a. Changes

#### [Trad Mode only]

The optimization control specifier `loop_nofission` is abolished.

#### [Previous version]

The optimization control specifier `loop_nofission` was effective.

#### [Current version]

The optimization control specifier `loop_nofission` is invalid.

## b. Potential problems

The optimization control specifier `loop_nofission` becomes invalid. If this specifier is specified, the following warning is output.

```
"filename", line number: warning: identifier "loop_nofission" is undefined.
```

## c. Action

You do not need to disable loop fission by the previous optimization control specifier `loop_nofission`, because loop fission only works on loops with optimization control specifier `loop_fission_target`. Therefore, no action is required.

## 2.2.13 [Trad Mode only] Change of the optimization control specifier `loop_fission`

---

### a. Changes

#### [Trad Mode only]

Change the optimization control specifier `loop_fission` to `loop_fission_target` [cl|ls]. The way of specifying changes as follows:

- The optional parameters of the optimization control specifier can specify the algorithm of the loop fission.
- Only the loops which optimization control specifiers are specified.

#### [Previous version]

The optimization control specifier `loop_fission` was effective.

#### [Current version]

The optimization control specifier `loop_fission` is invalid.

### b. Potential problems

The optimization control specifier `loop_fission` becomes invalid. If this specifier is specified, the following warning is output.

```
"filename", line number: warning: identifier "loop_fission" is undefined.
```

### c. Action

Change the optimization control specifier `loop_fission` to `loop_fission_target` [cl|ls]. You can also specify algorithm for loop fission by the parameters of the optimization control specifier. If the description position of the optimization control line is in a functional or compilation unit or loop unit which is an outer loop of the target, change the description position to the loop unit of the target loop to be fissioned.

## 2.2.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (`# pragma [fj] loop`) is invalid

---

### a. Changes

#### [Clang Mode only]

Change the message when the element following the specifiers of the optimization control line (`# pragma [fj] loop`) is invalid.

#### [Previous version]

- Output "error" when the following element of `prefetch_sequential` specifiers is invalid.
- Output "warning: unsupported" when the following element of `unroll` specifiers is invalid.

```
test.c:6:36: error: invalid argument;expected 'soft' or 'auto'
  #pragma loop prefetch_sequential xxx
                        ^
test.c:7:23: warning: unsupported fj-trad-pragma [-Wfj-unsupported-trad-pragmas]
  #pragma loop unroll "xxx"
                  ^
```

#### [Current version]

Output "warning: invalid argument" if the following elements of `prefetch_sequential` specifiers or `unroll` specifiers are invalid.

```
test.c:6:36: warning: invalid argument 'xxx' - ignoring [-Wignored-pragmas]
  #pragma loop prefetch_sequential xxx
                        ^
test.c:7:23: warning: invalid argument '"xxx"' - ignoring [-Wignored-pragmas]
  #pragma loop unroll "xxx"
                  ^
```

b. Potential problems

There are no bad side effects.

c. Action

It is not necessary to action.

## 2.2.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard

---

a. Changes

**[Trad Mode only]**

Change procedure call standards

**[Previous version]**

The scalable vector registers to be saved in callee were z8-z31, and SIMD and Floating-Point registers to be saved in callee were v8-v31.

**[Current version]**

The scalable vector registers to be saved in callee are z8-z23, and SIMD and Floating-Point registers to be saved in callee are v8-v23.

b. Potential problems

When linking objects on the previous standards and the new standards generated from programs which use the declare simd construct of OpenMP or objects on the previous standards generated from programs which use functions and operations that are subject to multi-operation functions, the execution result may be incorrect.

c. Action

In the following case, do not link objects on the previous standards, and recompile the program.

- The declare simd directive of OpenMP is used.
- The objects are generated with the -Kfast option or -Kmfunc[={ 1|2|3}] option and the multi-operation function optimization is applied. (\*)

\*) This can be confirmed by the output of the compiler message jwd8300o-i when -Koptmsg=2 is specified.

## 2.2.16 The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is change

---

a. Changes

The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is changed.

**[Previous version]**

The initial value of environment variable OMP\_PROC\_BIND was 'FALSE'.

**[Current version]**

The initial value of environment variable OMP\_PROC\_BIND is 'CLOSE'.

b. Potential problems

Performance improvement can be expected.

c. Action

There is no necessity for handle.

## 2.2.17 The internal behavior of software pipelining is changed

---

### a. Changes

The instruction scheduling algorithm used in software pipelining, performed by the options -Kswp, -Kswp\_weak, or -Kswp\_strong, by default is changed.

#### [Previous version]

A specific algorithm was always used.

#### [Current version]

A suitable algorithm is selected from multiple ones automatically by default.

### b. Potential problems

A selected algorithm may differ from the specific one used by the previous version. It may lead to a performance degradation.

### c. Action

You can specify a preferred algorithm by the option -Kswp\_policy. The algorithm used by the previous version can be specified by -Kswp\_policy=small.

## 2.2.18 The optimization information of software pipelining is changed

---

### a. Changes

An item is added to the optimization information of software pipelining, which is displayed by -Nlst=t.

#### [Previous version]

There were three items, IPC, ITR, and MVE, displayed as follows.

```

    <<< Loop-information Start >>>
    <<<  [OPTIMIZATION]
    <<<      SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2)
    <<< Loop-information End >>>
5      for(i=0;i<n;i++){
10         ...
        }
```

#### [Current version]

An item POL which means the applied instruction scheduling algorithm is added. It shows the actual algorithm selected for the loop by -Kswp\_policy=auto. An example is as follows.

```

    <<< Loop-information Start >>>
    <<<  [OPTIMIZATION]
    <<<      SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2, POL: S)
    <<< Loop-information End >>>
5      for(i=0;i<n;i++){
10         ...
        }
```

### b. Potential problems

The displayed format is changed.

### c. Action

None.

## 2.2.19 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option

---

### a. Changes

#### [Clang Mode only]

The condition that induces the -ffj-loop-interchange option is changed.

#### [Previous version]

The -ffj-loop-interchange option was not induced from other options.

#### [Current version]

The -ffj-loop-interchange option is induced from -O2 option or higher.

### b. Potential problems

When -O2 option or higher is set, elements of cache access may change. It may lead to a performance improvement or degradation.

### c. Action

If performance degrades, use the -ffj-no-loop-interchange option.

## 2.2.20 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable

---

### a. Changes

The controlling for the Sector cache for the first level cache is changed available when the Sector cache for the second level cache is unavailable.

#### [Previous version]

The Sector cache for the first level cache was unavailable when the Sector cache for the second level cache was unavailable.

#### [Current version]

The Sector cache for the first level cache is available when the Sector cache for the second level cache is unavailable.

### b. Potential problems

- The runtime message jwe1047i-w is not output.
- The performance may increase or decrease.
  - Fujitsu's mathematical libraries  
The performance may increase.
  - Program describing OCL for controlling the Sector cache  
The performance may increase or decrease.

### c. Action

If the environment variable FLIB\_L1\_SCCR\_CNTL is assigned to the value FALSE, the Sector cache for the first level cache is unavailable when the Sector cache for the second level cache is unavailable.

## 2.2.21 Changes the default of the compiler option -Karray\_declaration\_opt

---

### a. Changes

The -Karray\_declaration\_opt option was valid by default, when -O0 or higher option is valid.

#### [Previous version]

The -Karray\_declaration\_opt option was valid by default, when -O0 or higher option was valid.

#### [Current version]

The -Karray\_declaration\_opt option is invalid by default.

b. Potential problems

When the `-Karray_declaration_opt` option is not explicitly specified, performance may be degraded.

c. Action

Specify the `-Karray_declaration_opt` option.

## 2.2.22 Changing the behavior of the `getopt` function

---

a. Changes

Change the behavior of the `getopt`, `getopt_long`, and `getopt_long_only` functions.

[Previous version]

When the `getopt`, `getopt_long`, or `getopt_long_only` function was executed under any of the following conditions, if a non-option argument (an argument that does not begin with a '-') was followed by an argument that begins with a '-', it was not recognized as an option string.

- Using the MPI library
- Interlanguage linking with Fortran
- One or more of the following compiler options are specified:

[General Options for Compiler]

`-SSL2`  
`-SSL2BLAMP`

[Options for Optimization]

`-O2`  
`-O3`

[Options for Language Specifications]

`--linkfortran`  
`--linkcoarray`

[-K option]

`-Kfast`  
`-Kocl`  
`-Kopenmp`  
`-Kparallel` (Trad Mode only)  
`-Kparallel_strong` (Trad Mode only)  
`-Kvisimpact` (Trad Mode only)

[-N option]

`-NRtrap` (Trad Mode only)  
`-Nfmtl=fmt_mode` (Trad Mode only)  
`-Nhook_func` (Trad Mode only)  
`-Nhook_time`  
`-Nquickdbg={subchk|heapchk}` (Trad Mode only)  
`-Nrt_tune` (Trad Mode only)  
`-Nrt_tune_func` (Trad Mode only)  
`-Nrt_tune_loop={all|innermost}` (Trad Mode only)

[Current version]

When the `getopt`, `getopt_long`, or `getopt_long_only` function is executed, arguments that begin with '-' after a non-optional arguments (arguments that do not begin with '-') are recognized as option strings.

b. Potential problems

Arguments that were not recognized as option strings when parsed with the `getopt`, `getopt_long`, or `getopt_long_only` function may be recognized as option strings.



## Example

### - Example Program

```

$ cat getopttest.c
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char **argv)
{
    int aflag = 0, bflag = 0, cflag = 0, c, index;

    opterr = 0;
    while ((c = getopt(argc, argv, "abc")) != -1) {
        switch (c) {
            case 'a':
                aflag = 1;
                break;
            case 'b':
                bflag = 1;
                break;
            case 'c':
                cflag = 1;
                break;
            default:
                printf("user:Unknown option character `\\x%x'.\\n", optopt);
                return 1;
        }
    }
    printf("aflag = %d, bflag = %d, cflag = %d\\n", aflag, bflag, cflag);
    for (index = optind; index < argc; index++)
        printf("user:Non-option argument %s\\n", argv[index]);
    return 0;
}

```

### - Execution Result

The getopt function recognizes all arguments '-a', '-b', and '-c' that begin with '-' as optional arguments.

```

$ gcc -Kfast getopttest.c
$ ./a.out -a arg1 -b arg2 -c arg3
aflag = 1, bflag = 1, cflag = 1
user:Non-option argument arg1
user:Non-option argument arg2
user:Non-option argument arg3

```

### c. Action

Make the first character of the string specified in the third argument of the getopt, getopt\_long, and getopt\_long\_only functions '+', or set the environment variable POSIXLY\_CORRECT.

## 2.2.23 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword

### a. Changes

[Trad Mode only]

The specification is changed so that a message is output and an error is generated during the compilation if any of compiler reserved registers listed below is specified in the inline asm clobber list or the register name asm keyword.

Compiler reserved registers: x19, w19, r19, x29, w29, r29, sp, xzr, xsp, wzr, wsp

[Previous version]

No error messages were output when a compiler reserved register was specified in the inline asm clobber list or the register name asm keyword.

[Current version]

An error message is output and compilation error is generated if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.

b. Potential problems

An error message is output and compilation error is generated if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.

c. Action

Please do not specify any compiler reserved registers in the inline asm clobber list or the register name asm keyword.

## 2.2.24 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified

---

a. Changes

[Clang Mode only]

Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified.

[Previous version]

When the -fno-omit-frame-pointer option was specified, the optimization that did not keep the frame pointer register of leaf functions was worked.

[Current version]

When the -fno-omit-frame-pointer option is specified, the frame pointer register of leaf functions is kept.

b. Potential problems

When the -fno-omit-frame-pointer is specified, performance may be degraded.

c. Action

If performance degrades, specify the -momit-leaf-frame-pointer option.

Note that you get the call graph information of the Instant Performance Profiler, specify the -fno-omit-frame-pointer option.

## 2.2.25 [Clang Mode only] Change the default behavior due to the addition of the compiler options -f{debug-info-for-profiling|no-debug-info-for-profiling}

---

a. Changes

[Clang Mode only]

Change the default behavior due to the addition of the compiler options -f{debug-info-for-profiling|no-debug-info-for-profiling}.

The compiler option -fdebug-info-for-profiling generates signatures for additional information available to the Profiler. The signature is the information used by overload resolution.

If the compiler option -fdebug-info-for-profiling is omitted, the signature is generated as additional information.

[Previous version]

The signature was not generated as additional information.

Therefore, the Instant Performance Profiler displayed the function cost per function name for inlined functions when the -Minlined option was effective.

[Current version]

If these options are omitted, the signature is generated as additional information.

Therefore, the Instant Performance Profiler displays the function cost per signature for inlined functions when the -Minlined option is effective.

b. Potential problems

The size of the object program generated by omitting these options may become larger than that of the previous version.

c. Action

If you want to revert to the before, specify the compiler option -fno-debug-info-for-profiling.

## 2.2.26 [Clang Mode only] Improvement of the usability for software pipelining and zfill

---

a. Changes

[Clang Mode only]

Make the following changes to improve the usability of software pipelining and zfill.

1. To enable software pipelining for a loop which is vectorized by SVE instructions or enable zfill, not only the compiler option of these features -ffj-swp and -ffj-zfill[=*N*] but also the compiler option -msve-vector-bits=512 is needed to be enabled. Therefore, the behavior is changed to induce the compiler option -msve-vector-bits=512 when the compiler option -ffj-swp or -ffj-zfill[=*N*] is specified.
2. The compiler message is changed when the loop analysis in software pipelining fails.

[Previous version]

1. No compiler options were induced when the compiler option -ffj-swp or -ffj-zfill[=*N*] was specified.
2. The compiler message "loop analyzation failed" was output when the loop analysis in software pipelining failed.

[Current version]

1. The compiler option -msve-vector-bits=512 is induced when the compiler option -ffj-swp or -ffj-zfill[=*N*] is specified.
2. The compiler message "loop analysis failed" is output when the loop analysis in software pipelining fails.

b. Potential problems

1. The following problems may occur due to the induction of the compiler option -msve-vector-bits=512.
  - If the compiler option -ffj-swp or -ffj-zfill[=*N*] is specified and -msve-vector-bits=512 is not specified, the compiler terminates with error in compiling programs using SIMD built-in functions in **[Current version]** due to the restriction of the compiler option -msve-vector-bits=512, though such programs could be compiled in **[Previous version]**.
  - The behavior of optimization will change and the execution performance may be reduced due to the induction. If the words "bit width: 128" are not included in the compiler message of SIMD extension which is output when the compiler option -Rpass=.\* is specified, this change may affect.
2. The compiler message is changed.

c. Action

1. Specify the compiler option -msve-vector-bits=scalable after the compiler options -ffj-swp and -ffj-zfill[=*N*].
2. It is not necessary to action.

## 2.2.27 [Trad Mode only] The GNU compatible option `-fvisibility={default|internal|hidden|protected}` deprecated

---

### a. Changes

#### [Trad Mode only]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` is deprecated and removed from the manual.

#### [Previous version]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` was supported.

#### [Current version]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` is deprecated.

### b. Potential problems

When the GNU compatible option `-fvisibility={default|internal|hidden|protected}` is specified, a warning message is output at the compile time.

### c. Action

- Use Clang Mode if you want to specify the GNU compatible option `-fvisibility={default|internal|hidden|protected}`.
- The same effect can be achieved by modifying the program to use the `__attribute__` feature in Trad Mode.

Example:

```
int __attribute__((visibility("hidden"))) g;
```

## 2.2.28 [Clang Mode only] The compiler option `-f{fj-loop-interchange|fj-no-loop-interchange}` deprecated

---

### a. Changes

#### [Clang Mode only]

The compiler option `-f{fj-loop-interchange|fj-no-loop-interchange}`, which is for loop interchange optimization, is not supported and removed from the manual.

#### [Previous version]

The compiler option `-ffj-loop-interchange` was supported.

#### [Current version]

The compiler option `-ffj-loop-interchange` is unsupported.

### b. Potential problems

Since loop interchange optimization does not work, the execution performance may be reduced.

### c. Action

If you want to apply loop interchange optimization, do one of the following.

- Specify the compiler option `-Kloop_interchange` in Trad Mode.
- When using Clang Mode, modify the source program as follows.

Before

```
for(int i = 0; i < N; i++){
    for(int j = 0; j < N; j++){
        a[j][i] = a[j][i] + b[j][i];
    }
}
```

After (Interchange inner and outer loops)

```
for(int j = 0; j < N; j++){
    for(int i = 0; i < N; i++){
        a[j][i] = a[j][i] + b[j][i];
    }
}
```

## 2.3 C++ Compiler

---

### 2.3.1 [Trad Mode only] Change of the predefined version macro

---

a. Changes

**[Trad Mode only]**

The predefined version macro is changed.

**[Previous version]**

The predefined version macro was as follows:

Macro	Value
__FCC_VERSION	800

**[Current version]**

The predefined version macros are as follows:

Macro	Value
__FCC_major__	The major version number of the compiler
__FCC_minor__	The minor version number of the compiler
__FCC_patchlevel__	The patch level of the compiler
__FCC_version__	The string which represents the version of the compiler

b. Potential problems

If the old macro is used in the program, the program might not behave as expected.

c. Action

Modify the program to use the new macros.

### 2.3.2 [Trad Mode only] For information on the unneeded function, output of debugging information is suppressed

---

a. Changes

**[Trad Mode only]**

In the compilation, the output of debugging information to the unneeded function not used in the program is suppressed when -O0 option is effective and -g option is specified.

**[Previous version]**

In the compilation, debugging information on the unneeded function not used in the program was output when -O0 option is effective and -g option is specified.

**[Current version]**

In the compilation, debugging information on the unneeded function not used in the program is not output when -O0 option is effective and -g option is specified.

b. Potential problems

The function not used in the program cannot be debugged.

c. Action

It is not necessary to action.

### 2.3.3 [Clang Mode only] Change of value of macro due to OpenMP API version 4.5 support

---

a. Changes

**[Clang Mode only]**

To support OpenMP API version 4.5, the value of macro `_OPENMP` is changed for Clang Mode.

**[Previous version]**

When the compiler option `-Nclang` and `-Kopenmp` (or `-fopenmp`) were specified, the value of macro `_OPENMP` was 201107.

**[Current version]**

When the compiler option `-Nclang` and `-Kopenmp` (or `-fopenmp`) are specified, the value of macro `_OPENMP` is 201511.

b. Potential problems

Behavior of the program using the macro `_OPENMP` is different from before.

c. Action

Modify the program to take into account the change of the value.

### 2.3.4 Obsolescence of the compiler option `-Klto`

---

a. Changes

The `-Klto` option is obsolete.

**[Previous version]**

The `-Klto` option was valid.

**[Current version]**

The `-Klto` option is invalid.

b. Potential problems

When the `-Klto` option is specified in Trad Mode, the following warning message is output.

```
warning: lto sub-option specified in -K option is invalid.
```

When the `-Klto` option is specified in Clang Mode, the following error message is output.

```
error: unknown argument '-Klto', did you mean '-flto='?
```

c. Action

To apply link time optimization, specify the `-flto` option in Clang Mode.

### 2.3.5 [Clang Mode only] Remove the compiler option `-f{plt|no-plt}`

---

a. Changes

**[Clang Mode only]**

The `-f{plt|no-plt}` option is not supported in Clang Mode (remove from the manual).

[Previous version]

The `-f{plt|no-plt}` option was described in the manual as an option of Clang Mode.

[Current version]

If the source program was compiled using the `-f{plt|no-plt}` option in Clang Mode, this option may not have the effect you expect, so remove it from the manual.

b. Potential problems

No impact for this change.

(If `-fplt` has been specified previously, it has no effect because it does not have code output using PLT.)

c. Action

It is not necessary to action.

## 2.3.6 [Clang Mode only] Remove the compiler option `-mtune`

---

a. Changes

**[Clang Mode only]**

The `-mtune` option is not supported in Clang Mode (remove from the manual).

[Previous version]

The `-mtune` option description was described in the manual as an option of Clang Mode.

[Current version]

In the case that the source program is translated using the `-mtune` option of Clang Mode, the expected effect of this option may not be obtained, so this option is deleted from the manual.

b. Potential problems

No impact for this change.

(If `-mtune` has been specified previously, it has no effect because it does not have any special optimizations or code output.)

c. Action

For architecture optimization, please specify the `-mcpu={generic|thunderx|thunderx2t99}` option.

## 2.3.7 [Trad Mode only] Change the target loops of the `-Kloop_fission` option

---

a. Changes

**[Trad Mode only]**

Change the loop to be analyzed for loop fission when specifying the `-Kloop_fission` option.

[Previous version]

When the `-Kloop_fission` option was specified, all loops in the source program were analyzed for loop fission.

[Current version]

When the `-Kloop_fission` option is specified, loops with the optimization control specifier `loop_fission_target` are analyzed for loop fission.

b. Potential problems

Loop fission is not applied only by specifying the `-Kloop_fission` option.

c. Action

Specify the optimization control specifier `loop_fission_target [cl|ls]` for loops to which you want to apply loop fission, and specify the `-Kloop_fission` and `-Kocl` options at the same time. Specify the following compiler options or the optimization control specifiers to change the status of loop fission to improve performance further. Note that the loop fission is not applied when the effect of loop fission cannot be expected.

- The `-Kloop_fission_threshold=N` option or the optimization control specifier `loop_fission_threshold n`.
- The `-Kloop_fission_stripmining[={N|L1|L2}]` option or the optimization control specifier `loop_fission_stripmining [n]"L1"|"L2"]`.

### 2.3.8 [Trad Mode only] Change the default value of kind of the `-Kilfunc[=kind]` option

---

#### a. Changes

##### [Trad Mode only]

The default value of *kind* of the `-Kilfunc[=kind]` option is changed.

##### [Previous version]

If a value was not specified for *kind* of the `-Kilfunc[=kind]` option or the `-Kfast` option was specified, `-Kilfunc=loop` was set.

##### [Current version]

If a value is not specified for *kind* of the `-Kilfunc[=kind]` option or the `-Kfast` option is specified, `-Kilfunc=procedure` is set.

#### b. Potential problems

The `-Kilfunc` option may occur calculation errors in the execution results.

Since the default value of the `-Kilfunc[=kind]` option is changed from "loop" to "procedure", the optimization targets other than loops in addition and the calculation error caused by the execution may change.

#### c. Action

If you want to revert to the before, specify the `-Kilfunc=loop` option.

### 2.3.9 [Clang Mode only] Change the default value of `={loop|procedure}` of the `-ffj-ilfunc[={loop|procedure}]` option

---

#### a. Changes

##### [Clang Mode only]

The default value of `={loop|procedure}` of the `-ffj-ilfunc option[={loop|procedure}]` is changed.

##### [Previous version]

If a value was not specified for `={loop|procedure}` of the `-ffj-ilfunc` option or the `-Ofast` option was specified, `-ffj-ilfunc=loop` was set.

##### [Current version]

If a value is not specified for `={loop|procedure}` of the `-ffj-ilfunc` option or the `-Ofast` option is specified, `-ffj-ilfunc=procedure` is set.

#### b. Potential problems

The `-ffj-ilfunc` option may occur calculation errors in the execution results.

Since the default value of the `-ffj-ilfunc={loop|procedure}` option is changed from "loop" to "procedure", the optimization targets other than loops in addition and the calculation error caused by the execution may change.

#### c. Action

If you want to revert to the before, specify the `-ffj-ilfunc=loop` option.

### 2.3.10 [Trad Mode only] Specification change of the `-Krdconv` option

---

#### a. Changes

##### [Trad Mode only]

Change the specification of the `-Krdconv` option to `-Krdconv[=level]` and control optimization depending on *level*.

[Previous version]

- If the -Kfast option was specified, the -Krdconv option was set.
- When the -Krdconv option was set:  
Optimization was performed on the assumption that operations of 4-byte integer type do not overflow or wrap around.

[Current version]

- If the -O2 option or higher is set, the -Krdconv=1 option is set by default. The -Krdconv[=*level*] option controls the optimization according to *level*. The default value of *level* is 1.
- When the -Krdconv[=1] option is set:  
Optimization is performed on the assumption that operations of signed integer type whose size is 4-byte or less do not overflow.
- When the -Krdconv=2 option is set:  
In addition to the -Krdconv=1 option, optimization is performed on the assumption that operations of signed integer type whose size is 4-byte or less do not overflow.

b. Potential problems

- When the -Kfast option or -Krdconv option is set:  
Optimization is not performed on the assumption that operations of 4-byte unsigned integer type do not wrap around. This may result in execution performance decrease. It is decided to avoid deriving the -Krdconv option from the commonly-used -Kfast option and to conform to industry standards.
- When the -O2 option or higher is set:  
If a program occurs overflows in operations of signed integer type whose size is 4-byte or less, the program execution may be aborted or the execution result may be incorrect. Note that according to the language standard, overflow of signed integer arithmetic results in undefined behavior.

c. Action

- When the -Kfast option or -Krdconv[=1] option is set:  
Specify the -Krdconv=2 option to perform optimization on the assumption that operations of 4-byte unsigned integer type do not wrap around.
- When the -O2 option or higher is set:  
It is recommended modify a program if the program occurs overflows in operations of integer type whose size is 4-byte or less. Alternatively, you can avoid the potential problems by specifying the -Knordconv option.

## 2.3.11 [Clang Mode only] Changes of SIMD optimization information

---

a. Changes

[Clang Mode only]

Changes compilation information and compiler message of SIMD optimization.

1. Compilation information

The compilation information (optimization information) by the compile option -ffj-lst=t is changed as follows.

[Previous version]

For SVE vectorized loop, the number of array elements to be processed by one SIMD instruction was shown as "(VL: *length*[,*length*]...)".

```
<<< Loop-information Start >>>
<<<  [OPTIMIZATION]
<<<    SIMD(VL: 4 Interleave: 1)
<<< Loop-information End >>>
v   for (int i = 0; i < N; ++i) {
      a[i] = b[i];
    }
```

#### [Current version]

The optimization information of the SVE vectorized loop is output as follows. The output format "(VL: AGNOSTIC; VL: *length*[,*length*]... in 128-bit)" means that the loop is vectorized and the SVE vector registers are not fixed to a particular size.

Length shows the number of array elements processing in one SIMD instruction, assuming that the vector register size of the SVE is 128 bits.

```
<<< Loop-information Start >>>
<<<  [OPTIMIZATION]
<<<   SIMD(VL: AGNOSTIC; VL: 4 in 128-bit Interleave: 1)
<<< Loop-information End >>>
v   for (int i = 0; i < N; ++i) {
      a[i] = b[i];
    }
```

#### 2. Compiler message

The compilation message by the compile option `-Rpass=.` is changed as follows.

#### [Previous version]

When loop was SVE vectorized, output the compiler message as follows.

```
test.c:5:3: remark: vectorized loop (vectorization width: 4, interleaved count: 1) [-Rpass=sve-loop-vectorize]
```

#### [Current version]

Add SVE vector register size "bit width: 128" to SIMD information as follows.

```
test.c:5:3: remark: vectorized loop (vectorization width: 4, bit width: 128, interleaved count: 1) [-Rpass=sve-loop-vectorize]
```

#### b. Potential problems

The output compilation information is changed.

#### c. Action

No action is required.

## 2.3.12 [Trad Mode only] Abolition of the optimization control specifier `loop_nofission`

---

#### a. Changes

##### [Trad Mode only]

The optimization control specifier `loop_nofission` is abolished.

#### [Previous version]

The optimization control specifier `loop_nofission` was effective.

#### [Current version]

The optimization control specifier `loop_nofission` is invalid.

#### b. Potential problems

The optimization control specifier `loop_nofission` becomes invalid. If this specifier is specified, the following warning is output.

```
"filename", line number: warning: identifier "loop_nofission" is undefined.
```

#### c. Action

You do not need to disable loop fission by the previous optimization control specifier `loop_nofission`, because loop fission only works on loops with optimization control specifier `loop_fission_target`. Therefore, no action is required.

## 2.3.13 [Trad Mode only] Change of the optimization control specifier `loop_fission`

---

### a. Changes

#### [Trad Mode only]

Change the optimization control specifier `loop_fission` to `loop_fission_target` [cl|ls]. The way of specifying changes as follows:

- The optional parameters of the optimization control specifier can specify the algorithm of the loop fission.
- Only the loops which optimization control specifiers are specified.

#### [Previous version]

The optimization control specifier `loop_fission` was effective.

#### [Current version]

The optimization control specifier `loop_fission` is invalid.

### b. Potential problems

The optimization control specifier `loop_fission` becomes invalid. If this specifier is specified, the following warning is output.

```
"filename", line number: warning: identifier "loop_fission" is undefined.
```

### c. Action

Change the optimization control specifier `loop_fission` to `loop_fission_target` [cl|ls]. You can also specify algorithm for loop fission by the parameters of the optimization control specifier. If the description position of the optimization control line is in a functional or compilation unit or loop unit which is an outer loop of the target, change the description position to the loop unit of the target loop to be fissioned.

## 2.3.14 [Clang Mode only] Change the message if the Fujitsu compiler's proprietary optimization control line (`# pragma [fj] loop`) is invalid

---

### a. Changes

#### [Clang Mode only]

Change the message when the element following the specifiers of the optimization control line (`# pragma [fj] loop`) is invalid.

#### [Previous version]

- Output "error" when the following element of `prefetch_sequential` specifiers is invalid.
- Output "warning: unsupported" when the following element of `unroll` specifiers is invalid.

```
test.c:6:36: error: invalid argument;expected 'soft' or 'auto'
  #pragma loop prefetch_sequential xxx
                        ^
test.c:7:23: warning: unsupported fj-trad-pragma [-Wfj-unsupported-trad-pragmas]
  #pragma loop unroll "xxx"
                        ^
```

#### [Current version]

Output "warning: invalid argument" if the following elements of `prefetch_sequential` specifiers or `unroll` specifiers are invalid.

```
test.c:6:36: warning: invalid argument 'xxx' - ignoring [-Wignored-pragmas]
  #pragma loop prefetch_sequential xxx
                        ^
test.c:7:23: warning: invalid argument '"xxx"' - ignoring [-Wignored-pragmas]
  #pragma loop unroll "xxx"
                        ^
```

b. Potential problems

There are no bad side effects.

c. Action

It is not necessary to action.

## 2.3.15 [Trad Mode only] Incompatibility of the object due to changes in procedure call standard

---

a. Changes

**[Trad Mode only]**

Change procedure call standards

**[Previous version]**

The scalable vector registers to be saved in callee were z8-z31, and SIMD and Floating-Point registers to be saved in callee were v8-v31.

**[Current version]**

The scalable vector registers to be saved in callee are z8-z23, and SIMD and Floating-Point registers to be saved in callee are v8-v23.

b. Potential problems

When linking objects on the previous standards and the new standards generated from programs which use the declare simd construct of OpenMP or objects on the previous standards generated from programs which use functions and operations that are subject to multi-operation functions, the execution result may be incorrect.

c. Action

In the following case, do not link objects on the previous standards, and recompile the program.

- The declare simd directive of OpenMP is used.
- The objects are generated with the -Kfast option or -Kmfunc[={ 1|2|3}] option and the multi-operation function optimization is applied. (\*)

\*) This can be confirmed by the output of the compiler message jwd8300o-i when -Koptmsg=2 is specified.

## 2.3.16 The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is change

---

a. Changes

The initial value of environment variable OMP\_PROC\_BIND in The LLVM OpenMP Library is changed.

**[Previous version]**

The initial value of environment variable OMP\_PROC\_BIND was 'FALSE'.

**[Current version]**

The initial value of environment variable OMP\_PROC\_BIND is 'CLOSE'.

b. Potential problems

Performance improvement can be expected.

c. Action

There is no necessity for handle.

## 2.3.17 [Clang Mode only] Incompatibility of executable programs due to the change of the libc++ version

---

### a. Changes

#### [Clang Mode only]

The version of libc++ for the Clang Mode, which is used as the C++ standard template library, is changed.

#### [Previous version]

The version 6.0 of libc++ was used.

#### [Current version]

The version 7.0 of libc++ is used.

### b. Potential problems

If an executable program which is created by Clang Mode of the previous version with the `-stdlib=libc++` option is executed, the program might not behave as expected due to the incompatibility of libc++.

### c. Action

The object program has to be relinked.

If no object program remains, the source program has to be recompiled and relinked.

## 2.3.18 The internal behavior of software pipelining is changed

---

### a. Changes

The instruction scheduling algorithm used in software pipelining, performed by the options `-Kswp`, `-Kswp_weak`, or `-Kswp_strong`, by default is changed.

#### [Previous version]

A specific algorithm was always used.

#### [Current version]

A suitable algorithm is selected from multiple ones automatically by default.

### b. Potential problems

A selected algorithm may differ from the specific one used by the previous version. It may lead to a performance degradation.

### c. Action

You can specify a preferred algorithm by the option `-Kswp_policy`. The algorithm used by the previous version can be specified by `-Kswp_policy=small`.

## 2.3.19 The optimization information of software pipelining is changed

---

### a. Changes

An item is added to the optimization information of software pipelining, which is displayed by `-Nlst=t`.

#### [Previous version]

There were three items, IPC, ITR, and MVE, displayed as follows.

```

5          <<< Loop-information Start >>>
          <<< [OPTIMIZATION]
          <<<      SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2)
          <<< Loop-information End >>>
10         for(i=0;i<n;i++){
          ...
          }
```

[Current version]

An item POL which means the applied instruction scheduling algorithm is added. It shows the actual algorithm selected for the loop by -Kswp\_policy=auto. An example is as follows.

```

5          <<< Loop-information Start >>>
          <<< [OPTIMIZATION]
          <<<      SOFTWARE PIPELINING(IPC: 1.50, ITR: 11, MVE: 2, POL: S)
          <<< Loop-information End >>>
10      for(i=0;i<n;i++){
          ...
          }
```

b. Potential problems

The displayed format is changed.

c. Action

None.

## 2.3.20 [Trad Mode only] The initial value of C++ array allocated by new expression without initializer is changed

---

a. Changes

**[Trad Mode only]**

The initial value is changed when the type of the variable allocated by the new expression in C++ is a class type and the initial value of the new expression is not specified.

[Previous version]

Initial value was not set (indefinite value).

[Current version]

Set the initial value to 0 value.

b. Potential problems

When you refer to the value of a variable allocated by the array new expression without an initial value, it is always 0 value.

c. Action

It is not necessary to action.

## 2.3.21 [Clang Mode only] Change of condition that induces the -ffj-loop-interchange option

---

a. Changes

**[Clang Mode only]**

The condition that induces the -ffj-loop-interchange option is changed.

[Previous version]

The -ffj-loop-interchange option was not induced from other options.

[Current version]

The -ffj-loop-interchange option is induced from -O2 option or higher.

b. Potential problems

When -O2 option or higher is set, elements of cache access may change. It may lead to a performance improvement or degradation.

c. Action

If performance degrades, use the -ffj-no-loop-interchange option.

## 2.3.22 Controlling the Sector cache for the first level cache when the Sector cache for the second level cache is unavailable

---

### a. Changes

The controlling for the Sector cache for the first level cache is changed available when the Sector cache for the second level cache is unavailable.

#### [Previous version]

The Sector cache for the first level cache was unavailable when the Sector cache for the second level cache was unavailable.

#### [Current version]

The Sector cache for the first level cache is available when the Sector cache for the second level cache is unavailable.

### b. Potential problems

- The runtime message jwe1047i-w is not output.
- The performance may increase or decrease.
  - Fujitsu's mathematical libraries  
The performance may increase.
  - Program describing OCL for controlling the Sector cache  
The performance may increase or decrease.

### c. Action

If the environment variable FLIB\_L1\_SCCR\_CNTL is assigned to the value FALSE, the Sector cache for the first level cache is unavailable when the Sector cache for the second level cache is unavailable.

## 2.3.23 Changes the default of the compiler option -Karray\_declaration\_opt

---

### a. Changes

The -Karray\_declaration\_opt option was valid by default, when -O0 or higher option is valid.

#### [Previous version]

The -Karray\_declaration\_opt option was valid by default, when -O0 or higher option was valid.

#### [Current version]

The -Karray\_declaration\_opt option is invalid by default.

### b. Potential problems

When the -Karray\_declaration\_opt option is not explicitly specified, performance may be degraded.

### c. Action

Specify the -Karray\_declaration\_opt option.

## 2.3.24 Changing the behavior of the getopt function

---

### a. Changes

Change the behavior of the getopt, getopt\_long, and getopt\_long\_only functions.

#### [Previous version]

When the getopt, getopt\_long, or getopt\_long\_only function was executed under any of the following conditions, if a non-option argument (an argument that does not begin with a '-') was followed by an argument that begins with a '-', it was not recognized as an option string.

- Using the MPI library
- Interlanguage linking with Fortran

- One or more of the following compiler options are specified:

[General Options for Compiler]

-SSL2  
-SSL2BLAMP

[Options for Optimization]

-O2  
-O3

[Options for Language Specifications]

--linkfortran  
--linkcoarray

[-K option]

-Kfast  
-Kocl  
-Kopenmp  
-Kparallel (Trad Mode only)  
-Kparallel\_strong (Trad Mode only)  
-Kvisimpact (Trad Mode only)

[-N option]

-NRtrap (Trad Mode only)  
-Nfntl=*fnt\_mode* (Trad Mode only)  
-Nhook\_func (Trad Mode only)  
-Nhook\_time  
-Nquickdbg={subchk|heapchk} (Trad Mode only)  
-Nrt\_tune (Trad Mode only)  
-Nrt\_tune\_func (Trad Mode only)  
-Nrt\_tune\_loop={all|innermost} (Trad Mode only)

[Current version]

When the getopt, getopt\_long, or getopt\_long\_only function is executed, arguments that begin with '-' after a non-optional arguments (arguments that do not begin with '-') are recognized as option strings.

b. Potential problems

Arguments that were not recognized as option strings when parsed with the getopt, getopt\_long, or getopt\_long\_only function may be recognized as option strings.



## Example

- Example Program

```
$ cat getopttest.c
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char **argv)
{
    int aflag = 0, bflag = 0, cflag = 0, c, index;

    opterr = 0;
    while ((c = getopt(argc, argv, "abc")) != -1) {
        switch (c) {
            case 'a':
                aflag = 1;
                break;
            case 'b':
```

```

        bflag = 1;
        break;
    case 'c':
        cflag = 1;
        break;
    default:
        printf("user:Unknown option character `\\x%x'.\\n", optopt);
        return 1;
    }
}
printf("aflag = %d, bflag = %d, cflag = %d\\n", aflag, bflag, cflag);
for (index = optind; index < argc; index++)
    printf("user:Non-option argument %s\\n", argv[index]);
return 0;
}

```

#### - Execution Result

The getopt function recognizes all arguments '-a', '-b', and '-c' that begin with '-' as optional arguments.

```

$ gcc -Kfast getopttest.c
$ ./a.out -a arg1 -b arg2 -c arg3
aflag = 1, bflag = 1, cflag = 1
user:Non-option argument arg1
user:Non-option argument arg2
user:Non-option argument arg3

```

#### c. Action

Make the first character of the string specified in the third argument of the getopt, getopt\_long, and getopt\_long\_only functions '+', or set the environment variable POSIXLY\_CORRECT.

## 2.3.25 [Trad Mode only] Specification change so that a message is output and an error is generated during the compilation if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword

#### a. Changes

##### [Trad Mode only]

The specification is changed so that a message is output and an error is generated during the compilation if any of compiler reserved registers listed below is specified in the inline asm clobber list or the register name asm keyword.

Compiler reserved registers: x19, w19, r19, x29, w29, r29, sp, xzr, xsp, wzr, wsp

##### [Previous version]

No error messages were output when a compiler reserved register was specified in the inline asm clobber list or the register name asm keyword.

##### [Current version]

An error message is output and compilation error is generated if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.

#### b. Potential problems

An error message is output and compilation error is generated if a compiler reserved register is specified in the inline asm clobber list or the register name asm keyword.

#### c. Action

Please do not specify any compiler reserved registers in the inline asm clobber list or the register name asm keyword.

## 2.3.26 [Clang Mode only] Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified

---

### a. Changes

#### [Clang Mode only]

Change of keeping the frame pointer register in leaf functions when the -fno-omit-frame-pointer option is specified.

#### [Previous version]

When the -fno-omit-frame-pointer option was specified, the optimization that did not keep the frame pointer register of leaf functions was worked.

#### [Current version]

When the -fno-omit-frame-pointer option is specified, the frame pointer register of leaf functions is kept.

### b. Potential problems

When the -fno-omit-frame-pointer is specified, performance may be degraded.

### c. Action

If performance degrades, specify the -fomit-leaf-frame-pointer option.

Note that you get the call graph information of the Instant Performance Profiler, specify the -fno-omit-frame-pointer option.

## 2.3.27 [Clang Mode only] Change the default behavior due to the addition of the compiler options -f{debug-info-for-profiling|no-debug-info-for-profiling}

---

### a. Changes

#### [Clang Mode only]

Change the default behavior due to the addition of the compiler options -f{debug-info-for-profiling|no-debug-info-for-profiling}.

The compiler option -fdebug-info-for-profiling generates signatures for additional information available to the Profiler. The signature is the information used by overload resolution.

If the compiler option -fdebug-info-for-profiling is omitted, the signature is generated as additional information.

#### [Previous version]

The signature was not generated as additional information.

Therefore, the Instant Performance Profiler displayed the function cost per function name for inlined functions when the -Minlined option was effective.

#### [Current version]

If these options are omitted, the signature is generated as additional information.

Therefore, the Instant Performance Profiler displays the function cost per signature for inlined functions when the -Minlined option is effective.

### b. Potential problems

The size of the object program generated by omitting these options may become larger than that of the previous version.

### c. Action

If you want to revert to the before, specify the compiler option -fno-debug-info-for-profiling.

## 2.3.28 [Clang Mode only] Improvement of the usability for software pipelining and zfill

---

### a. Changes

#### [Clang Mode only]

Make the following changes to improve the usability of software pipelining and zfill.

1. To enable software pipelining for a loop which is vectorized by SVE instructions or enable zfill, not only the compiler option of these features `-ffj-swp` and `-ffj-zfill[=N]` but also the compiler option `-msve-vector-bits=512` is needed to be enabled. Therefore, the behavior is changed to induce the compiler option `-msve-vector-bits=512` when the compiler option `-ffj-swp` or `-ffj-zfill[=N]` is specified.
2. The compiler message is changed when the loop analysis in software pipelining fails.

#### [Previous version]

1. No compiler options were induced when the compiler option `-ffj-swp` or `-ffj-zfill[=N]` was specified.
2. The compiler message “loop analyzation failed” was output when the loop analysis in software pipelining failed.

#### [Current version]

1. The compiler option `-msve-vector-bits=512` is induced when the compiler option `-ffj-swp` or `-ffj-zfill[=N]` is specified.
2. The compiler message “loop analysis failed” is output when the loop analysis in software pipelining fails.

### b. Potential problems

1. The following problems may occur due to the induction of the compiler option `-msve-vector-bits=512`.
  - If the compiler option `-ffj-swp` or `-ffj-zfill[=N]` is specified and `-msve-vector-bits=512` is not specified, the compiler terminates with error in compiling programs using SIMD built-in functions in **[Current version]** due to the restriction of the compiler option `-msve-vector-bits=512`, though such programs could be compiled in **[Previous version]**.
  - The behavior of optimization will change and the execution performance may be reduced due to the induction. If the words “bit width: 128” are not included in the compiler message of SIMD extension which is output when the compiler option `-Rpass=.` is specified, this change may affect.
2. The compiler message is changed.

### c. Action

1. Specify the compiler option `-msve-vector-bits=scalable` after the compiler options `-ffj-swp` and `-ffj-zfill[=N]`.
2. It is not necessary to action.

## 2.3.29 [Trad Mode only] The GNU compatible option `-fvisibility={default|internal|hidden|protected}` deprecated

---

### a. Changes

#### [Trad Mode only]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` is deprecated and removed from the manual.

#### [Previous version]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` was supported.

#### [Current version]

The GNU compatible option `-fvisibility={default|internal|hidden|protected}` is deprecated.

### b. Potential problems

When the GNU compatible option `-fvisibility={default|internal|hidden|protected}` is specified, a warning message is output at the compile time.

c. Action

- Use Clang Mode if you want to specify the GNU compatible option `-fvisibility={default|internal|hidden|protected}`.
- The same effect can be achieved by modifying the program to use the `__attribute__` feature in Trad Mode.

Example:

```
int __attribute__((visibility("hidden"))) g;
```

## 2.3.30 [Clang Mode only] The compiler option `-f{fj-loop-interchange|fj-no-loop-interchange}` deprecated

---

a. Changes

**[Clang Mode only]**

The compiler option `-f{fj-loop-interchange|fj-no-loop-interchange}`, which is for loop interchange optimization, is not supported and removed from the manual.

**[Previous version]**

The compiler option `-ffj-loop-interchange` was supported.

**[Current version]**

The compiler option `-ffj-loop-interchange` is unsupported.

b. Potential problems

Since loop interchange optimization does not work, the execution performance may be reduced.

c. Action

If you want to apply loop interchange optimization, do one of the following.

- Specify the compiler option `-Kloop_interchange` in Trad Mode.
- When using Clang Mode, modify the source program as follows.

Before

```
for(int i = 0; i < N; i++){
    for(int j = 0; j < N; j++){
        a[j][i] = a[j][i] + b[j][i];
    }
}
```

After (Interchange inner and outer loops)

```
for(int j = 0; j < N; j++){
    for(int i = 0; i < N; i++){
        a[j][i] = a[j][i] + b[j][i];
    }
}
```

## 2.4 MPI Library

---

### 2.4.1 The local\_options -am for the mpiexec command is changed to -tune

---

a. Changes

The local\_options -am for the mpiexec command is changed to -tune.

**[Previous version]**

The option for the mpiexec command which specifies configuration file of MCA parameter to MPI program was -am.

[Current version]

The option for the mpiexec command which specifies configuration file of MCA parameter to MPI program is -tune.

b. Potential problems

The option -am for the mpiexec command which specifies configuration file of MCA parameter cannot be used.

c. Action

Change the option for the mpiexec command which specifies configuration file of MCA parameter to -tune.

## 2.4.2 Executable program incompatibilities due to changes of header files and modules

---

a. Changes

Header files and modules are modified.

MPI header files and modules are not compatible between previous versions and current version.

[Previous version]

Executable programs created in previous version were used.

[Current version]

Executable programs created in previous version cannot be used in current version.

b. Potential problems

If an executable program created in previous version is executed in current version, the program might not behave as expected.

c. Action

The following actions are required.

- If MPI is used in a program, the source program has to be recompiled and relinked.
- If the COARRAY feature of Fortran is used in a program, the object program has to be relinked. If no object program remains, the source program has to be recompiled and relinked.
- If the SSL II/MPI is used in a program, the source program has to be recompiled and relinked.
- If the ScaLAPACK is used in a program, the object program has to be relinked. If no object program remains, the source program has to be recompiled and relinked.

## 2.4.3 Change the behavior when the MCA parameter btl\_openib\_receive\_queues\_default\_num\_bufs is not specified

---

a. Changes

The behavior when the MCA parameter btl\_openib\_receive\_queues\_default\_num\_bufs is not specified is changed.

[Previous version]

The number of SRQ buffers was set to 1024.

[Current version]

The MPI library automatically sets the number of SRQ buffers according to the number of processes.

b. Potential problems

Since reducing memory usage is also considered, the execution performance of a program may decrease.

c. Action

Specify the following option for the mpiexec command when executing the MPI program. However, note that memory usage may increase.

```
--mca btl_openib_receive_queues_default_num_bufs 1024
```

## 2.4.4 Change the default value of the info key `romio_ds_write`, which specifies the behavior of Data sieving in MPI-IO, and change the processing so that a lock is taken when Data sieving is enabled

---

### a. Changes

The default value of the info key `romio_ds_write`, which specifies the behavior of Data sieving in MPI-IO is changed.

In addition, the mode used when Data sieving is enabled is changed to a mode that takes a lock.

#### [Previous version]

The default value of the info key `romio_ds_write` was automatic.

In addition, when Data sieving was enabled, the processing was performed in a mode that did not take a lock.

#### [Current version]

The default value of the info key `romio_ds_write` is disable.

In addition, when Data sieving is enabled, the processing is performed in a mode that takes a lock.

### b. Potential problems

Data sieving may be able to improve the performance even when writing to a file. Therefore, the execution performance of some programs may decrease due to changing the default value of the info key `romio_ds_write` to disable.

In addition, even if Data sieving is explicitly enabled, the execution performance may decrease due to the additional processing of taking a lock.

### c. Action

None.

## 2.5 Profiler

---

### 2.5.1 Change of target for counting up the cost of inlined function in Instant Performance Profiler

---

#### a. Changes

When cost information is output by Instant Performance Profiler, target for counting up the cost of inlined function is changed.

##### [Previous version]

When cost information is output by Instant Performance Profiler, cost of the function was counted up to the caller function.

##### [Current version]

When cost information is output by Instant Performance Profiler, cost of the function is counted up to the callee function.

#### b. Potential problems

When cost information is output by Instant Performance Profiler, cost of the function is counted up to the callee function.

#### c. Action

None.

### 2.5.2 Change of condition for outputting time statistic information of Advanced Performance Profiler

---

#### a. Changes

Condition for outputting time statistic information of Advanced Performance Profiler is changed.

[Previous version]

Advanced Performance Profiler always output time statistic information.

[Current version]

If the -Icpupa option is effective, the -Inocputime option is effective and does not output User CPU time and System CPU time.

If the -Icpupa option is effective and output User CPU time and System CPU time, you must specify the -Icputime option. If the -Icpupa option is not effective, User CPU time and System CPU time are output.

b. Potential problems

When the -Icpupa option of fapp command is effective, User CPU time and System CPU time are not output. When the -Icpupa option of fapp command is not effective, User CPU time and System CPU time are output.

c. Action

Specify the -Icputime option of fapp command if the -Icpupa option of fapp command is effective and outputting User CPU time and System CPU time is necessary.

## 2.5.3 Change of output of MPI\_STARTALL routine in MPI communication cost information of Advanced Performance Profiler

---

a. Changes

When MPI communication cost information for MPI\_STARTALL routine is output by Advanced Performance Profiler, the content is changed.

[Previous version]

In the MPI communication cost information for MPI\_STARTALL routine, only information of MPI\_STARTALL routine is output. Output line was fixed as 1.

[Current version]

In the MPI communication cost information for MPI\_STARTALL routine, information of MPI\_STARTALL routine and persistent collective communication request routine starting MPI\_STARTALL is output. Output line may be plural.

b. Potential problems

When MPI communication cost information for MPI\_STARTALL routine is output, output line may be plural.

c. Action

None.

## 2.5.4 Change of output information and layout of CPU Performance Analysis Report

---

a. Changes

Output information and layout of CPU Performance Analysis Report are changed.

[Previous version]

1. "Graph. Busy rate" was output.
2. "Non-contiguous gather load instruction rate(%) ( /Load-store instruction)", "Non-contiguous scatter store instruction rate(%) ( /Load-store instruction)", and "Prefetch rate(%) ( /Load-store instruction)" in table "Cache" were output.
3. "CMG0", "CMG1", "CMG2", and "CMG3" in table "Data Transfer CMGs" were output.
4. "Fill instruction" and "Spill instruction" in table "Instruction" of detailed report were output.
5. "Graph. Instruction" was output.
6. "L1 pipeline 0 Active element rate(%)", "L1 pipeline 1 Active element rate(%)", "Floating-point register fill instruction", "Predicate register fill instruction", "Floating-point register spill instruction", and "Predicate register spill instruction" in table "EXTRA" were output.

[Current version]

1. "Graph. Busy rate" are not output.
2. "Non-contiguous gather load instruction rate(%) ( /Load-store instruction)", "Non-contiguous scatter store instruction rate(%) ( /Load-store instruction)", "Prefetch rate(%) ( /Load-store instruction)" in table "Cache" are not output.
3. "CMG0", "CMG1", "CMG2", "CMG3" in table "Data Transfer CMGs" are not output.
4. "Floating-point register fill instruction" and "Predicate register fill instruction" are output instead of "Fill instruction" in table "Instruction" of detailed report. "Floating-point register spill instruction" and "Predicate register spill instruction" are output instead of "Spill instruction" in table "Instruction" of detailed report.
5. "Graph. Instruction" is not output.
6. "Floating-point pipeline A Active element rate(%)" and "Floating-point pipeline B Active element rate(%)" move to table "Busy". "Floating-point register fill instruction", "Predicate register fill instruction", "Floating-point register spill instruction", and "Predicate register spill instruction" move to table "Instruction". "L1 pipeline 0 Active element rate(%)" and "L1 pipeline 1 Active element rate(%)" move to table "Busy".
7. Whole layout is changed.

b. Potential problems

"Graph. Busy rate", "Graph. Instruction", "Non-contiguous gather load instruction rate(%) ( /Load-store instruction)", "Non-contiguous scatter store instruction rate(%) ( /Load-store instruction)", "Prefetch rate(%) ( /Load-store instruction)", "CMG0", "CMG1", "CMG2", and "CMG3" cannot be referred to.

"Fill instruction" and "Spill instruction" in detailed report cannot be referred to.

c. Action

- When content in "Graph. Busy rate" needs to be referred to, see graph in "Graph. Cycle accounting".
- When value of "Fill instruction" or "Spill instruction" needs to be referred to, add two values that are divided in detailed report.
- When equivalent contents with "Non-contiguous gather load instruction rate(%) ( /Load-store instruction)" and "Non-contiguous scatter store instruction rate(%) ( /Load-store instruction)" needs to be referred to, see "Non-contiguous gather load instruction" or "Non-contiguous scatter store instruction" respectively. When equivalent contents with "Prefetch rate(%) ( /Load-store instruction)" add values of "Contiguous prefetch instruction", "Gathering prefetch instruction", and "Scalar prefetch instruction".
- None for others.

## 2.5.5 Changes of kinds of CPU Performance Analysis Report

---

a. Changes

Count of measurement and file output for creating CPU Performance Analysis Report is changed.

[Previous version]

Measurement and file output by Advanced Performance Profiler should be done 5, 11, or 17 times.

[Current version]

Measurement and file output by Advanced Performance Profiler should be done 1, 5, 11, or 17 times.

b. Potential problems

Measurement and file output by Advanced Performance Profiler should be done 1, 5, 11, or 17 times.

c. Action

None.

## 2.5.6 Changes of output information and layout of CPU Performance Analysis Report (Data Transfer CMGs)

---

### a. Changes

Output information and layout of CPU Performance Analysis Report (Data Transfer CMGs) are changed.

[Previous version]

1. The write information items "CMG0", "CMG1", "CMG2", and "CMG3" were output.
2. The write and read information items "Memory" were output.

[Current version]

1. The write information items "CMG0", "CMG1", "CMG2", and "CMG3" are not output.
2. The write and read information items "Memory" are not output.
3. The write and read entries "Own memory" and "Other Memory" are output.
4. The line of the write and read information is output in order of read, write.

### b. Potential problems

The values of items "CMG0", "CMG1", "CMG2", and "CMG3" in the write information cannot be referred to.

### c. Action

- See "Own memory" instead of the read and write information items "Memory".
- There is no solution for the write information items "CMG0", "CMG1", "CMG2", and "CMG3".

## 2.5.7 Changes of the Outputting Directory of Profile Data in Profiler

---

### a. Changes

The outputting directory of profile data which collecting commands (fipp -C or fapp -C) generate in Instant Performance Profiler and Advanced Performance Profiler is changed.

[Previous version]

Profile data is output to a directory specified by -d option like below.

```
<Directory specified by -d option>/<Profile data name>
```

[Current version]

Profile data is output to a directory subordinate to a directory specified by -d option. Directory layout is like below.

```
<Directory specified by -d option>/<Sub-directory>/<Profile data name>
```

### b. Potential problems

Profiler does not output profile data to the directory specified by -d option, and fails to execute analyzing commands and an error message is output like below when either collecting commands (fipp -C, fapp -C) or analyzing commands (fipp -A, fapp -A) use older version.

- Instant Performance Profiler

```
fipp -A: profiling data is invalid. : [Directory specified by -d option]
```

- Advanced Performance Profiler

```
fapp -A: profiling data is invalid. : [Directory specified by -d option]
```

### c. Action

None.

## 2.5.8 Change of target for counting up the cost of inlined function in Instant Performance Profiler (cancellation of 2.5.1)

---



### Information

Cancel "2.5.1 Change of target for counting up the cost of inlined function in Instant Performance Profiler" and return to the original specifications.

#### a. Changes

In the cost information output by Instant Performance Profiler, target for counting up the cost of inlined function is changed.

##### [Previous version]

In the cost information output by Instant Performance Profiler, the cost of inlined function was counted up in the cost of itself, not the cost of the caller function.

##### [Current version]

In the cost information output by Instant Performance Profiler, the cost of inlined function is counted up in the cost of the caller function, not an inlined function.

#### b. Potential problems

In the cost information output by Instant Performance Profiler, target for counting up the cost of inlined function is changed.

#### c. Action

It is not necessary to action.

If you want the cost information which is output by Instant Performance Profiler of previous version, contact a System Engineer or the Fujitsu Support Desk.

## 2.6 Mathematical Libraries

---

### 2.6.1 Executable program incompatibilities due to changes of header files and modules

---

#### a. Changes

Header files and modules are modified.

MPI header files and modules are not compatible between previous versions and current version.

##### [Previous version]

Executable programs created in previous version were used.

##### [Current version]

Executable programs created in previous version cannot be used in current version.

#### b. Potential problems

If an executable program created in previous version is executed in current version, the program might not behave as expected.

#### c. Action

The following actions are required.

- If the SSL II/MPI is used in a program, the source program has to be recompiled and relinked.
- If the ScaLAPACK is used in a program, the object program has to be relinked. If no object program remains, the source program has to be recompiled and relinked.

## 2.6.2 Changes the mathematical library linked when the compiler option -KNOSVE is specified using the native compiler

---

### a. Changes

Changes the mathematical libraries to be linked when the compiler option -KNOSVE is specified, in addition to the compiler option to link the Fujitsu mathematical libraries using the native compiler.

The mathematical library of general-purpose was installed only on the login node to reduce the disk space on the compute node, but it should be installed on the compute node as well. This allows the mathematical libraries of general-purpose that could only be linked by a cross compiler to be linked by a native compiler, so that the libraries linked by both are common.

[Previous version]

Even though the compiler option -KNOSVE was specified to native compiler, the library using SVE was linked.

[Current version]

When the compiler option -KNOSVE is specified to native compiler, the library of general-purpose is linked.

### b. Potential problems

When the mathematical library is linked using the native compiler with the compiler option -KNOSVE, the performance may be degraded.

### c. Action

It is not necessary to action.

## 2.7 Other

---

### 2.7.1 Differences in Default Domain of papi by OS Version

---

The default value of the domain that defines the event collection range of papi varies depending on the OS version. This may cause differences in measurement.

Hereinafter, Red Hat Enterprise Linux is referred to as RHEL.

[RHEL 8.7 or earlier]

```
Default domain=USER/KERNEL/SUPERVISOR
```

Collects events that occur in user space, kernel space, or supervisor space.

[RHEL 8.8 or later]

```
Default domain=USER
```

Collects only events that occur in user space.



#### Information

.....

This incompatibility does not affect measurement with the Profiler of Compiler Package.

The domain can be changed using the papi\_set\_domain(3) function. See the Linux man pages for details.

.....

### 2.7.2 Changing the License Authentication System

---

#### a. Changes

Change the license authentication system.

[Previous version]

The license server was set up on a login node or a compute node.

[Current version]

Does not set up the license server.

b. Potential problems

- Compiler versions prior to V1.0L21 will not be able to compile.
- Changes the result of the applicability judgment of the Compiler Package.

c. Action

- When compiling with compilers prior to version V1.0L21, setting the value of the environment variable FCOMP\_CHECK\_LICENSE to 'nocheck' at compile time allows compilation with version V1.0L30 without a license server.

example:

```
export FCOMP_CHECK_LICENSE=nocheck
```

- The result of the applicability judgment is different between V1.0L21 and V1.0L30 of the Compiler Package.  
When requesting issuance of the applicability certificate of Compiler Package V1.0L30, request it as "Compiler Package V1.0L30."