



FUJITSU Software

Technical Computing Suite V2.0

A decorative horizontal band with a red-to-dark-red gradient, featuring abstract, glowing white and red lines that swirl and intersect, creating a sense of motion and technology.

Technical Computing Language

User's Guide

J2UL-1923-03ENZ0(00)
June 2016

Preface

Purpose of This Manual

This manual provides an overview of the functions of Technical Computing Language (hereafter referred to as this system). It also contains points to be noted when migrating from existing products.

This system is corresponds as follows:

- Supercomputer PRIMEHPC FX10 system
- Supercomputer PRIMEHPC FX100 system
- PC cluster comprised of PC server PRIMERGY

Intended Readers

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

Notation Used in This Manual

In this manual,

- Supercomputer PRIMEHPC FX10 system is referred to as FX10 system.
- Supercomputer PRIMEHPC FX100 system is referred to as FX100 system.
- PC cluster comprised of PC server PRIMERGY is referred to as PC Cluster.
- FX10 system and FX100 system are generically referred to as PRIMEHPC system.

Organization of This Manual

This manual consists of the following chapters:

[Chapter 1 Technical Computing Language Overview](#)

Provides an overview of the system.

[Chapter 2 Functions of Components](#)

Provides an overview of the function of each component.

[Chapter 3 Additional Functions Overview](#)

Provides an overview of additional functions to the previous release of this product.

[Chapter 4 Compatibility Information](#)

Provides compatibility information with earlier releases of this product.

[Chapter 5 Migrating from former Language Package V3](#)

Explains points to be noted when migrating from a former Language Package V3 system to this system.

[Chapter 6 Notes on migration from FX10 system to FX100 system](#)

Explains notes when migrating from FX10 system to FX100 system.

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviation
Microsoft(R) Office Excel 2003	Excel
Microsoft(R) Office Excel 2007	
Microsoft(R) Excel 2010	

Full Name	Abbreviation
Microsoft(R) Excel 2013	
Microsoft(R) Excel 2011 for Mac	

Export Controls

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

Trademarks

- Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Red Hat, and Red Hat Enterprise Linux are trademarks of Red Hat, Inc., registered in the U.S. and other countries.
- OpenMP is a trademark of the OpenMP Architecture Review Board.
- Mac and Mac OS are registered trademarks of Apple Inc.
- Microsoft, Windows, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- All other trademarks are the property of their respective owners.

Date of Publication and Version

Version	Manual code
June 2016, 3rd Version	J2UL-1923-03ENZ0(00)
November 2015, 2nd Version	J2UL-1923-02ENZ0(00)
February 2015, Version 1.1	J2UL-1923-01ENZ0(01)
October 2014, 1st Version	J2UL-1923-01ENZ0(00)

Copyright

Copyright FUJITSU LIMITED 2014-2016

Update History

Changes	Location	Version
Standard supported by the C++ compiler is updated.	2.1.3	3rd Version
Additional functions overview is appended. FX10 System Generation Number:13	3.1.1.1	
	3.1.2.1	
	3.1.3.1	
Additional functions overview is appended. FX100 System Generation Number:04	3.1.1.2	
	3.1.2.2	
	3.1.3.2	
	3.2.1.2	
	3.2.2.2	
Additional functions overview is appended. PC Cluster Generation Number:15	3.2.4.2	
	3.1.1.3	
	3.1.2.3	
	3.1.3.3	
	3.2.1.3	
	3.2.2.3	
	3.4.2.3	

Changes	Location	Version
Compatibility information is appended. FX10 System Generation Number:13	4.1.1.1 4.1.2.1 4.1.3.1	
Compatibility information is appended. FX100 System Generation Number:04	4.1.1.2 4.1.2.2 4.1.3.2	
Compatibility information is appended. PC Cluster Generation Number:15	4.1.1.3 4.1.2.3 4.1.3.3	
Compatibility information is appended. FX100 System Generation Number:03	4.2.4.1	
Compatibility information is appended. PC Cluster Generation Number:14	4.2.4.2	
The table of each component is renewed.	6.2	
The article is corrected for the MPI-3.0 support.	1.1 2.2	2nd Version
Additional functions overview is appended. FX10 System Generation Number:12	3.1.1.1 3.1.2.1 3.1.3.1	
Additional functions overview is appended. FX100 System Generation Number:03	3.1.1.2 3.1.2.2 3.1.3.2 3.2.1.2 3.2.2.2 3.2.4.2 3.2.5.2 3.3.2 3.4.2.2	
Additional functions overview is appended. PC Cluster Generation Number:14	3.1.1.3 3.1.2.3 3.1.3.3 3.2.1.3 3.2.2.3 3.2.4.3 3.2.5.3 3.3.3 3.4.2.3	
Compatibility information is appended. FX10 System Generation Number:12	4.1.1.1	
Compatibility information is appended. FX100 System Generation Number:03	4.1.1.2 4.1.2.1 4.1.3.1 4.1.4.1	
Compatibility information is appended. PC Cluster Generation Number:14	4.1.1.3 4.1.4.2	
The article on "Change in source code" is changed.	6.1	
Generation Number to consider is appended.	6.2	
The article "Abbreviations" is appended.	Preface	

Changes	Location	Version
Additional functions overview is appended. FX10 System Generation Number:11	3.1.1.1 3.1.2.1 3.1.3.1 3.2.4.1 3.2.5.1 3.2.6.1 3.3.1	
Additional functions overview is appended. FX100 System Generation Number:02	3.1.1.2 3.1.2.2 3.1.3.2 3.2.1.2 3.2.4.2 3.2.5.2 3.2.6.2 3.3.2	
Additional functions overview is appended. PC Cluster Generation Number:13	3.1.1.3 3.1.2.3 3.1.3.3 3.3.3	
Compatibility information is appended. FX10 System Generation Number:11	4.1.1.1 4.1.2.1 4.1.3.1	
Compatibility information is appended. FX100 System Generation Number:02	4.1.1.2 4.1.2.2 4.1.3.2	
Compatibility information is appended. PC Cluster Generation Number:13	4.1.1.3 4.1.2.3 4.1.3.3	
Compatibility information for MPI Library is appended.	4.2.4.1 4.2.4.2 6.2.3	
Generation Number to consider is appended.	6.2	

Refer to the following for the update history of Technical Computing Suite V1.0.

Edition	Date	Revised Location	Description
1	2012-11-05	Chapter 3	Additional functions overview is appended.
		Chapter 4	Compatibility information is appended.
2	2013-08-12	Preface	An explanation of the trademark Mac and Windows is appended.
		1.3.2	An explanation of component packages is appended.
		Chapter 3	Additional functions overview is appended. - FX10 System Generation Number:07, PC Cluster Generation Number:10 - FX10 System Generation Number:06, PC Cluster Generation Number:09
		Chapter 4	Compatibility information is appended. - FX10 System Generation Number:07, PC Cluster Generation Number:10 - FX10 System Generation Number:06, PC Cluster Generation Number:09

Edition	Date	Revised Location	Description
3	2013-10-31	Chapter 3	Additional functions overview is appended. - FX10 System Generation Number:08
4	2013-11-11	1.1 2.1.1 2.1.2 2.1.3	The version of OpenMP specification is changed.
		1.1 2.3	The version of MPI standards is changed.
		2.1.1	A part of Fortran 2008 standards is appended to language specifications.
		Chapter 3	Additional functions overview is appended. - FX10 System Generation Number:09, PC Cluster Generation Number:11
		Chapter 4	Compatibility information is appended. - FX10 System Generation Number:09, PC Cluster Generation Number:11
5	2014-10-01	-	Reworked the format all over with the version upgrade of Technical Computing Suite V2.0.
		-	FX100 system is added.
		1.1 2.3	The version of MPI standards is changed.
		2.1.1 2.1.2 2.1.3	CPU name is changed.
		2.1.3	Language specifications supported by the C++ compiler is appended.
		2.4.2	Largepage performance information is added.
		Chapter 3	Additional functions overview is appended. - FX10 System Generation Number:10, PC Cluster Generation Number:12
		Chapter 4	Compatibility information is appended. - FX10 System Generation Number:10, PC Cluster Generation Number:12
		Chapter 6	New article.

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

Contents

Chapter 1 Technical Computing Language Overview.....	1
1.1 What is Technical Computing Language?.....	1
1.2 Applicable Fields.....	1
1.3 System Configuration.....	2
1.3.1 Manual Organization.....	2
1.3.2 Component Packages.....	3
1.3.3 System Configuration from Hardware Perspective.....	4
1.3.3.1 Client (Windows Operating System/ Mac Operating System).....	4
1.3.3.2 Login Node (Linux Operating System).....	5
1.3.3.3 PRIMEHPC System (Computing Nodes).....	5
1.3.3.4 PC Cluster (Computing Nodes).....	6
1.3.4 Program Development Flow.....	6
1.3.4.1 Creating and Executing a Program from a Client.....	6
1.3.4.2 Compiling and Executing an Existing Program from a Login Node.....	7
1.3.4.3 Debugging an Existing Program from a Client.....	8
1.3.4.4 Tuning an Existing Program from a Client.....	9
Chapter 2 Functions of Components.....	11
2.1 Compiler.....	11
2.1.1 Fortran Compiler.....	11
2.1.2 C Compiler.....	12
2.1.3 C++ Compiler.....	12
2.1.4 XPFortran Translator.....	13
2.2 Mathematical Libraries.....	13
2.3 MPI Library.....	14
2.4 Programming Workbench.....	14
2.4.1 Debugger.....	15
2.4.2 Profiler.....	15
Chapter 3 Additional Functions Overview.....	17
3.1 Compiler.....	17
3.1.1 Fortran Compiler.....	17
3.1.1.1 FX10 System.....	17
3.1.1.2 FX100 System.....	21
3.1.1.3 PC Cluster.....	23
3.1.2 C Compiler.....	28
3.1.2.1 FX10 System.....	28
3.1.2.2 FX100 System.....	32
3.1.2.3 PC Cluster.....	34
3.1.3 C++ Compiler.....	38
3.1.3.1 FX10 System.....	38
3.1.3.2 FX100 System.....	42
3.1.3.3 PC Cluster.....	45
3.1.4 XPFortran Translator.....	49
3.1.4.1 FX10 System.....	49
3.1.4.2 PC Cluster.....	49
3.2 Mathematical Libraries.....	50
3.2.1 SSL II.....	50
3.2.1.1 FX10 System.....	50
3.2.1.2 FX100 System.....	50
3.2.1.3 PC Cluster.....	51
3.2.2 C-SSL II.....	52
3.2.2.1 FX10 System.....	52
3.2.2.2 FX100 System.....	52
3.2.2.3 PC Cluster.....	54

3.2.3 SSL II/MPI.....	55
3.2.3.1 FX10 System.....	55
3.2.3.2 PC Cluster.....	55
3.2.4 BLAS.....	56
3.2.4.1 FX10 System.....	56
3.2.4.2 FX100 System.....	56
3.2.4.3 PC Cluster.....	56
3.2.5 LAPACK.....	56
3.2.5.1 FX10 System.....	56
3.2.5.2 FX100 System.....	57
3.2.5.3 PC Cluster.....	57
3.2.6 ScaLAPACK.....	58
3.2.6.1 FX10 System.....	58
3.2.6.2 FX100 System.....	58
3.2.6.3 PC Cluster.....	58
3.3 MPI Library.....	58
3.3.1 FX10 System.....	58
3.3.2 FX100 System.....	60
3.3.3 PC Cluster.....	61
3.4 Programming Workbench.....	61
3.4.1 Debugger.....	61
3.4.1.1 FX10 System.....	61
3.4.2 Profiler.....	62
3.4.2.1 FX10 System.....	62
3.4.2.2 FX100 System.....	62
3.4.2.3 PC Cluster.....	63
3.4.3 User GUI.....	63
3.4.3.1 PC Cluster.....	63
Chapter 4 Compatibility Information.....	64
4.1 Migrating to V2.0L30 (FX10 System Generation Number:13, FX100 System Generation Number:04, PC Cluster Generation Number:15).....	64
4.1.1 Fortran Compiler.....	64
4.1.1.1 FX10 System.....	64
4.1.1.2 FX100 System.....	64
4.1.1.3 PC Cluster.....	64
4.1.2 C++ Compiler.....	64
4.1.2.1 FX10 System.....	64
4.1.2.2 FX100 System.....	65
4.1.2.3 PC Cluster.....	65
4.1.3 MPI Library.....	65
4.1.3.1 FX10 System.....	65
4.1.3.2 FX100 System.....	65
4.1.3.3 PC Cluster.....	65
4.2 Migrating to V2.0L20 (FX10 System Generation Number:12, FX100 System Generation Number:03, PC Cluster Generation Number:14).....	66
4.2.1 Fortran Compiler.....	66
4.2.1.1 FX10 System.....	66
4.2.1.2 FX100 System.....	66
4.2.1.3 PC Cluster.....	66
4.2.2 C Compiler.....	66
4.2.2.1 FX100 System.....	66
4.2.3 C++ Compiler.....	66
4.2.3.1 FX100 System.....	66
4.2.4 MPI Library.....	67
4.2.4.1 FX100 System.....	67
4.2.4.2 PC Cluster.....	67

4.3 Migrating to V2.0L10 (FX10 System Generation Number:11, FX100 System Generation Number:02, PC Cluster Generation Number:13).....	67
4.3.1 Fortran Compiler.....	67
4.3.1.1 FX10 System.....	67
4.3.1.2 FX100 System.....	67
4.3.1.3 PC Cluster.....	68
4.3.2 C Compiler.....	68
4.3.2.1 FX10 System.....	68
4.3.2.2 FX100 System.....	68
4.3.2.3 PC Cluster.....	68
4.3.3 C++ Compiler.....	68
4.3.3.1 FX10 System.....	68
4.3.3.2 FX100 System.....	68
4.3.3.3 PC Cluster.....	68
4.4 Migrating to V2.0L10 (FX10 System Generation Number:10, PC Cluster Generation Number:12).....	69
4.4.1 Fortran Compiler.....	69
4.4.1.1 FX10 System.....	69
4.4.1.2 PC Cluster.....	69
4.4.2 C Compiler.....	69
4.4.2.1 FX10 System.....	69
4.4.2.2 PC Cluster.....	69
4.4.3 C++ Compiler.....	70
4.4.3.1 FX10 System.....	70
4.4.3.2 PC Cluster.....	70
4.4.4 MPI Library.....	70
4.4.4.1 FX10 System.....	70
4.4.4.2 PC Cluster.....	70
4.5 Migrating to V1.0L30 (FX10 System Generation Number:09, PC Cluster Generation Number:11).....	70
4.5.1 Fortran Compiler.....	70
4.5.1.1 FX10 System.....	70
4.5.1.2 PC Cluster.....	71
4.5.2 C Compiler.....	71
4.5.2.1 FX10 System.....	71
4.5.2.2 PC Cluster.....	71
4.5.3 C++ Compiler.....	71
4.5.3.1 FX10 System.....	71
4.5.3.2 PC Cluster.....	71
4.5.4 MPI Library.....	72
4.5.4.1 FX10 System.....	72
4.5.4.2 PC Cluster.....	72
4.5.5 Programming Workbench.....	72
4.5.5.1 FX10 System.....	72
4.5.5.2 PC Cluster.....	72
4.6 Migrating to V1.0L20 (FX10 System Generation Number:07, PC Cluster Generation Number:10).....	72
4.6.1 Fortran Compiler.....	72
4.6.1.1 FX10 System.....	72
4.6.1.2 PC Cluster.....	72
4.6.2 C Compiler.....	73
4.6.2.1 FX10 System.....	73
4.6.2.2 PC Cluster.....	73
4.6.3 C++ Compiler.....	73
4.6.3.1 FX10 System.....	73
4.6.3.2 PC Cluster.....	73
4.7 Migrating to V1.0L20 (FX10 System Generation Number:06, PC Cluster Generation Number:09).....	73
4.7.1 Fortran Compiler.....	73
4.7.1.1 FX10 System.....	73
4.7.1.2 PC Cluster.....	73

4.7.2 C Compiler.....	73
4.7.2.1 FX10 System.....	73
4.7.2.2 PC Cluster.....	73
4.7.3 C++ Compiler.....	74
4.7.3.1 FX10 System.....	74
4.7.3.2 PC Cluster.....	74
4.8 Migrating to V1.0L20 (FX10 System Generation Number:05, PC Cluster Generation Number:08).....	74
4.8.1 Fortran Compiler.....	74
4.8.1.1 FX10 System.....	74
4.8.1.2 PC Cluster.....	74
4.8.2 C Compiler.....	74
4.8.2.1 FX10 System.....	74
4.8.2.2 PC Cluster.....	75
4.8.3 C++ Compiler.....	75
4.8.3.1 FX10 System.....	75
4.8.3.2 PC Cluster.....	75
4.8.4 MPI Library.....	75
4.8.4.1 FX10 System.....	75
4.8.4.2 PC Cluster.....	75
 Chapter 5 Migrating from former Language Package V3.....	 76
 Chapter 6 Notes on migration from FX10 system to FX100 system.....	 77
6.1 Common guidelines.....	77
6.2 Modification of each component.....	77
6.2.1 Compiler.....	77
6.2.1.1 Fortran Compiler.....	77
6.2.1.2 C Compiler.....	80
6.2.1.3 C++ Compiler.....	82
6.2.1.4 XPFortran Translator.....	85
6.2.2 Mathematical Libraries.....	85
6.2.2.1 SSL II.....	85
6.2.2.2 C-SSL II.....	85
6.2.2.3 BLAS.....	85
6.2.2.4 LAPACK.....	86
6.2.3 MPI Library.....	86
6.2.4 Programming Workbench.....	86
6.2.4.1 Debugger.....	86
6.2.4.2 Profiler.....	87

Chapter 1 Technical Computing Language Overview

This chapter presents an overview of Technical Computing Language under the following headings:

- What is Technical Computing Language?
- Applicable Fields
- System Configuration

1.1 What is Technical Computing Language?

Technical Computing Language is intended for supercomputer PRIMEHPC system and PC cluster. It is software that assists with all stages, from development through to execution of high-functionality parallel application programs written in Fortran, C, C++, and the parallel program language XPFortran.

Technical Computing Language has the following features:

Assists with development of "high-functionality" parallel programs

- Compiler optimization technology that elicits high CPU execution performance
- Various functions that assist with highly parallelized programming using hybrid parallels (*1)
 - Compiler automatic parallelization function that simplifies thread parallelization
 - Parallel program language XPFortran that simplifies process parallels
 - High-functionality MPI communication library that underpins process parallelization
 - Optimized mathematics library and parallel edition mathematics library (thread parallels, MPI parallels)

*1: Hybrid parallel is a parallel programming model containing both thread parallels and process parallels.

Assists with efficient development of large-scale parallel application programs

- Compilers equipped with high-level optimization functions and automatic parallelization functions
- Translator for translating from parallel program language XPFortran to MPI programs
- Parallel debugger
- Profiler that underpins program performance tuning
- Optimized mathematics library and parallel edition mathematics library

Assists with development of "highly portable" programs

- Compilers that conform to international program language standards (Fortran, C, and C++)
- Support for various industry standard specifications
 - C and C++ compilers that support GNU compiler extended specifications
 - Compilers that conform to OpenMP 3.1 standards
 - MPI library that conforms to MPI Standard
 - FX100 System / PC Cluster: MPI-3.0 Standard
 - FX10 System: MPI-2.2 Standard and a part of MPI-3.0 Standard
- Provision of BLAS, LAPACK, and ScaLAPACK

1.2 Applicable Fields

Technical Computing Language will be used in various fields that require high calculation processing capability and the development and execution of scientific and technical computing application programs on PRIMEHPC system or PC cluster.

Technical Computing Language enables high parallelization and is particularly effective for the development and execution of application programs for large-scale scientific and technical computing.

In addition, Technical Computing Language is suitable for the following processing:

- Development and execution of large-scale scientific and technical computing application programs written in Fortran, C, and C++
- Development and execution of large-scale and highly parallelized application programs for process parallels and thread parallels

1.3 System Configuration

This section explains the following aspects of Technical Computing Language system configuration:

- Manual organization
- Component packages
- System configuration from hardware perspective
- Program development flow

1.3.1 Manual Organization

The following manuals are provided with this system:

Reader Task	Relevant Manual	Format
- Learn about the Fortran syntax supported by the Fortran compiler	Fortran Language Reference	PDF
- Learn how to use the Fortran compiler - Learn Fortran programming	Fortran User's Guide	PDF
- Learn how to use the COARRAY Feature - Learn COARRAY programming	Fortran User's Guide Additional Volume COARRAY	PDF
- Learn about error messages during compilation using the Fortran compiler	Fortran Compiler Messages	PDF
- Learn how to use the C compiler - Learn about the C compiler extended language specifications based on C and OpenMP specifications	C User's Guide	PDF
- Learn how to use the C++ compiler - Learn about the C++ compiler extended language specifications based on C++ and OpenMP specifications	C++ User's Guide	PDF
- Learn about optimization messages from the C and C++ compiler	C/C++ Compiler Optimization Messages	PDF
- Learn how to use the XPFortran translator - Learn about XPFortran language specifications	XPFortran User's Guide	PDF
- Learn about messages displayed during execution of Fortran, C, and C++ programs	Fortran/C/C++ Runtime Messages	PDF
- Learn about runtime information output functions, and how to use them	Runtime Information Output Function User's Guide	PDF
- Learn how to develop programs using programming workbench tools	Programming Workbench User's Guide	PDF

Reader Task	Relevant Manual	Format
- Learn how to use profiler statistical information and time series information	Profiler User's Guide	PDF
- Learn about the rank allocation optimization tool functions, and how to use them	Rank Map Automatic Tuning Tool User's Guide	PDF
- Learn about debugger GUI functions, and how to use them	Debugger User's Guide	PDF
- Learn how to use the Mathematical Libraries	Programmer's Guide for Usage of Mathematical Libraries	PDF
- Learn about the functions and use of the Scientific Subroutine Library II (SSL II)	SSL II User's Guide FUJITSU SSL II Extended Capabilities User's Guide FUJITSU SSL II Extended Capabilities User's Guide II	PDF
- Learn about the functions and usage of the Scientific Subroutine Library II Thread-Parallel Capabilities	Fujitsu SSL II Thread-Parallel Capabilities User's Guide	PDF
- Learn about the functions and use of the C Scientific Subroutine Library II (C-SSL II)	Fujitsu C-SSL II User's Guide	PDF
- Learn about the functions and usage of the C Scientific Subroutine Library II Thread-Parallel Capabilities	Fujitsu C-SSL II Thread-Parallel Capabilities User's Guide	PDF
- Learn about the functions and usage of the Scientific Subroutine Library II/MPI (SSL II/MPI)	Fujitsu SSL II/MPI User's Guide	PDF
- Obtain an overview of the functions provided by BLAS, LAPACK and ScaLAPACK, and learn how they relate to the BLAS, LAPACK, and ScaLAPACK published by Netlib	BLAS LAPACK ScaLAPACK User's Guide	PDF
- Learn about the function of Fortran and C++ versions of Fast Basic Operations Library for Quadruple Precision and how to use it	Fast Basic Operations Library for Quadruple Precision User's Guide	PDF
- Learn how to use the MPI library	MPI User's Guide	PDF

View PDF manuals in the latest Adobe(R) Reader(R) version.

1.3.2 Component Packages

This system consists of the following packages.

- Package for the master environment

This is the latest environment of this system.

- Package for the generation environment

This system can assign the first edition and each patch release the specific generation (hereafter called "Generation Number").

It means that this generation management allows you to have more than one patch release of this system at the same time.



Example

Example of master environment and generation environment to which are installed packages of first edition and some patches for FX100 system

Binaries installed for both the master environment and latest generation environment (/opt/FJSVmxlang/GM-2.0.0-*nn*) are identical.

```

/opt/FJSVmxlang/          ..... V2.0 product (master environment)
/GM-2.0.0-04             ..... V2.0 product (generation environment)

```

```

/GM-2.0.0-05      . . . . . V2.0 patch  (generation environment)
:
/GM-2.0.0-nn(*1) . . . . . V2.0 patch  (generation environment)

```

*1: "*nn*" means a generation number.

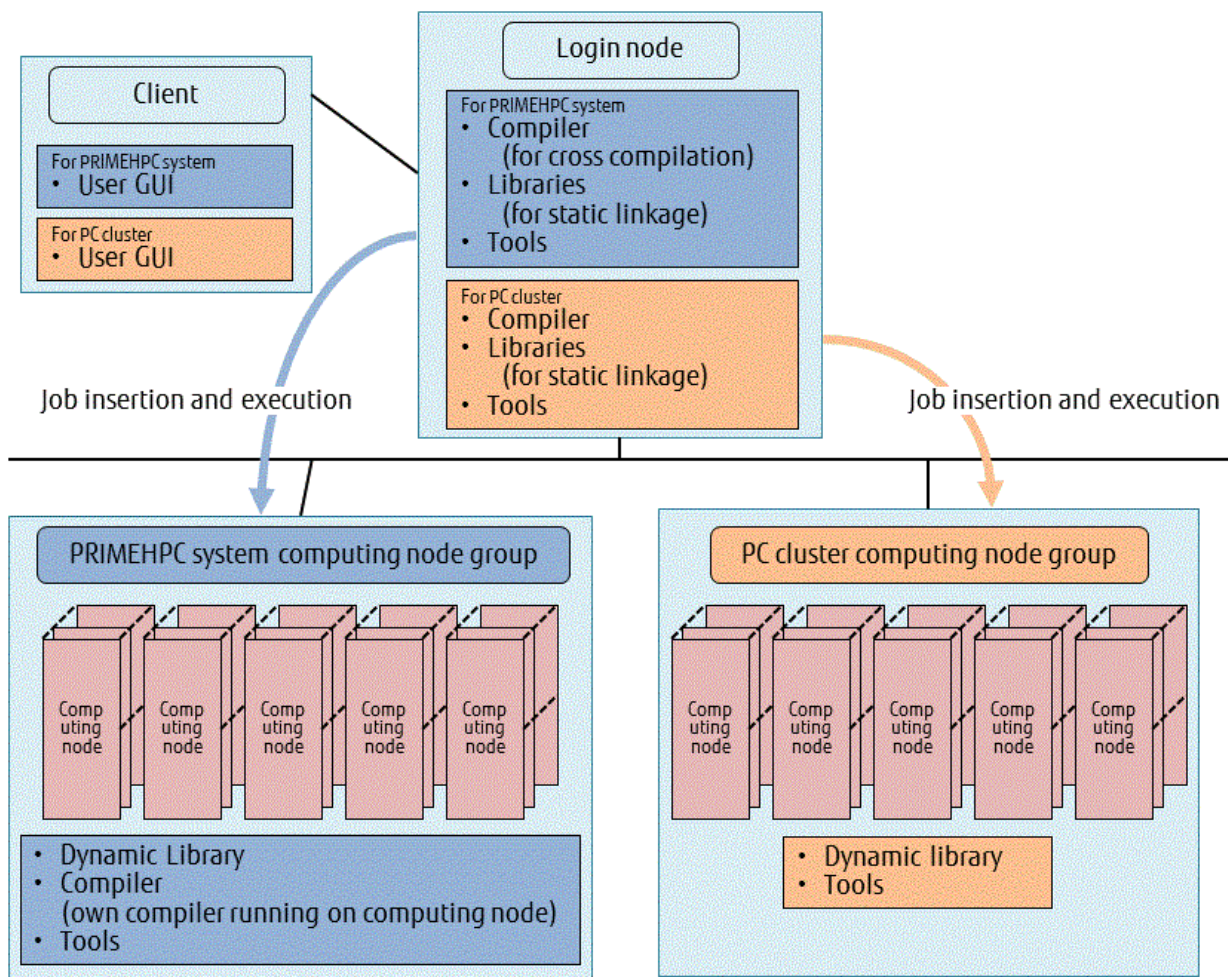
1.3.3 System Configuration from Hardware Perspective

The types of computer running Technical Computing Language can be broadly classified as follows:

1. Client
2. Login node
3. PRIMEHPC system (computing node group)
4. PC cluster (computing node group)

The example of system configuration is shown below. Refer to the Job Operation Software manual for details about system configuration.

Figure 1.1 Example of System configuration



1.3.3.1 Client (Windows Operating System/ Mac Operating System)

A client computer uses the programming workbench and GUI-based tasks to perform programming operations such as program creation, editing, compilation, and execution, and also debugger and profiler operations for programs for PRIMEHPC system or PC cluster.

To perform the various programming operations for PRIMEHPC system, install the following PRIMEHPC system components:

- Tools
 - The programming workbench user GUI for PRIMEHPC system

To perform the various programming operations for PC clusters, install the following PC cluster components:

- Tools
 - The programming workbench user GUI for PC clusters

1.3.3.2 Login Node (Linux Operating System)

The login node runs the compilers for PRIMEHPC system or PC cluster and creates executable programs that use command-based operations to perform the actual program compilation and link-editing. This node uses job operation software to execute the created programs as ordered jobs on the PRIMEHPC system or PC cluster computation nodes.

To develop programs for PRIMEHPC system, install the following PRIMEHPC system components:

- PRIMEHPC system compiler (for cross compilation)
 - Fortran compiler
 - C compiler
 - C++ compiler
 - XPFortran translator
 - MPI program compilation commands (MPI library)
- PRIMEHPC system libraries (for static linkage)
 - Runtime libraries of each compiler
 - Mathematics library
- PRIMEHPC system tools
 - Processing part on the login node side of the programming workbench (Web service program)
 - Profiler information output commands

To develop programs for PC clusters, install the following PC cluster components:

- PC cluster compilers
 - Fortran compiler
 - C compiler
 - C++ compiler
 - XPFortran translator
 - MPI program compilation commands (MPI library)
- PC cluster libraries (for static linkage)
 - Runtime libraries of each compiler
 - Mathematics library
- PC cluster tools
 - Processing part on the login node side of the programming workbench (Web service program)
 - Profiler information output commands

1.3.3.3 PRIMEHPC System (Computing Nodes)

The executable programs created on the PRIMEHPC system login node are executed as jobs by the computing nodes.

In addition, compilers that run on the PRIMEHPC system computing nodes allow program compilation and link editing to be executed as jobs on the computing nodes.

Install the following PRIMEHPC system components on the PRIMEHPC system computing nodes:

- Runtime dynamic libraries
 - Runtime libraries of each compiler
 - MPI library
- Tools
 - Profiling data collection commands
 - Profiler information output commands

If required, also install the following components:

- Compilers (own compilers running on the computing node)
 - Fortran compiler
 - C compiler
 - C++ compiler
 - XPFortran translator
 - MPI program compilation commands (MPI library)

1.3.3.4 PC Cluster (Computing Nodes)

The executable programs created on the PC cluster login node are executed as jobs by the computing nodes.

Install the following PC cluster components on the PC cluster computing nodes:

- Runtime dynamic libraries
 - Runtime libraries of each compiler
 - MPI library
- Tools
 - Profiling data collection commands
 - Profiler information output commands

1.3.4 Program Development Flow

The following sections provide typical examples of basic program development work flows (for example, program creation, compilation, execution, debugging, tuning) with attention given to the components explained in the sections above.

1.3.4.1 Creating and Executing a Program from a Client

The following workflow applies when performing operations from program creation to execution from a client.

1. Creating the program

Use the programming workbench "file operation functions" to create and edit the program intended for each of the target computing nodes. The created program is actually deployed on the shared file system at the login node.

If required, write a program that uses the various functions provided by the mathematics library or the MPI library.

2. Compiling the program

Use the "application construction functions" of the programming workbench at the client side to compile the program created in Step 1 using the appropriate compiler and to create an executable program. The executable program is created on the shared file system at the login node.

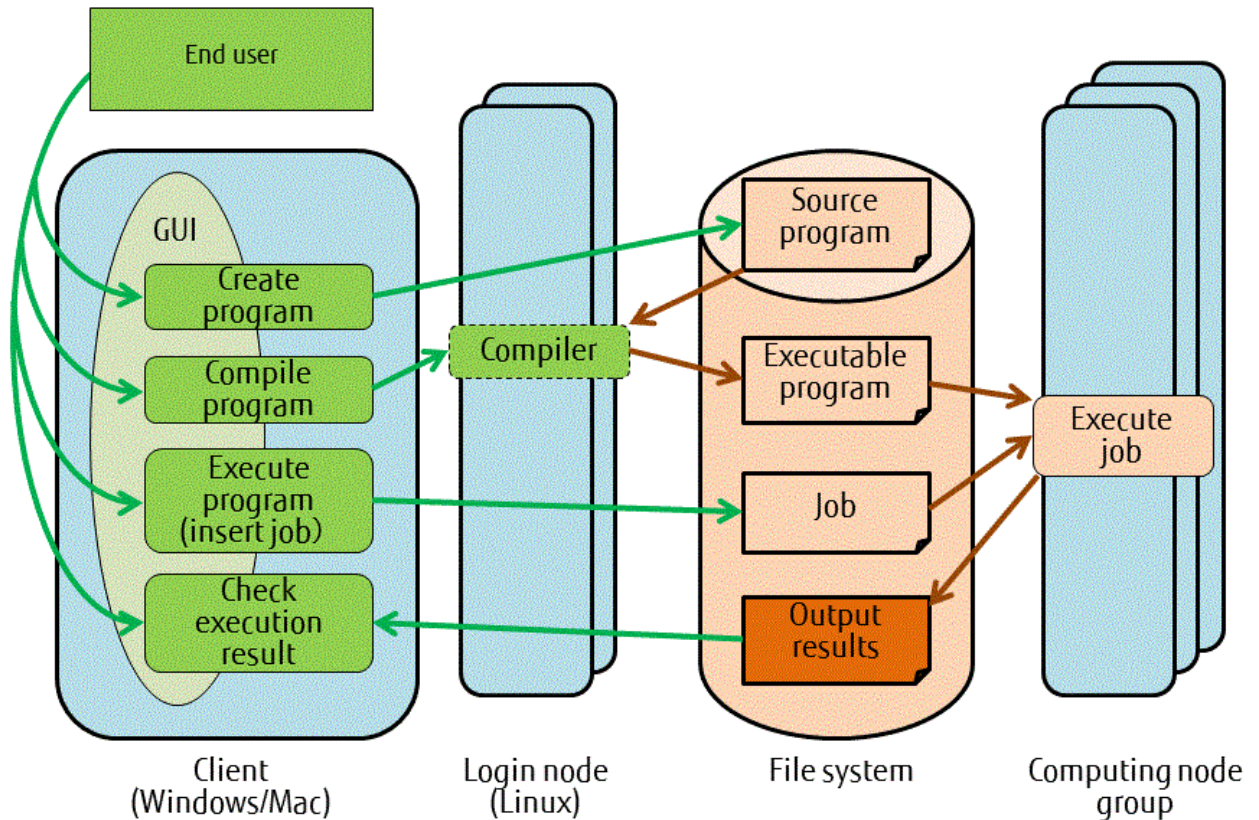
3. Executing the program

Use the "application execution functions" of the programming workbench at the client side to execute the program created in Step 2 at the relevant computing node. Actual program execution is passed to the job operation software as a job.

4. Checking the program execution results

If program execution results are output as a file, use the programming workbench "file operation functions" from the client side to check the contents of these output results.

Figure 1.2 Program creation and execution from a client



1.3.4.2 Compiling and Executing an Existing Program from a Login Node

The following workflow applies when performing operations from compilation to execution of an existing program from a login node.

1. Compiling the program

At the login node, start the appropriate compiler and create the executable program.

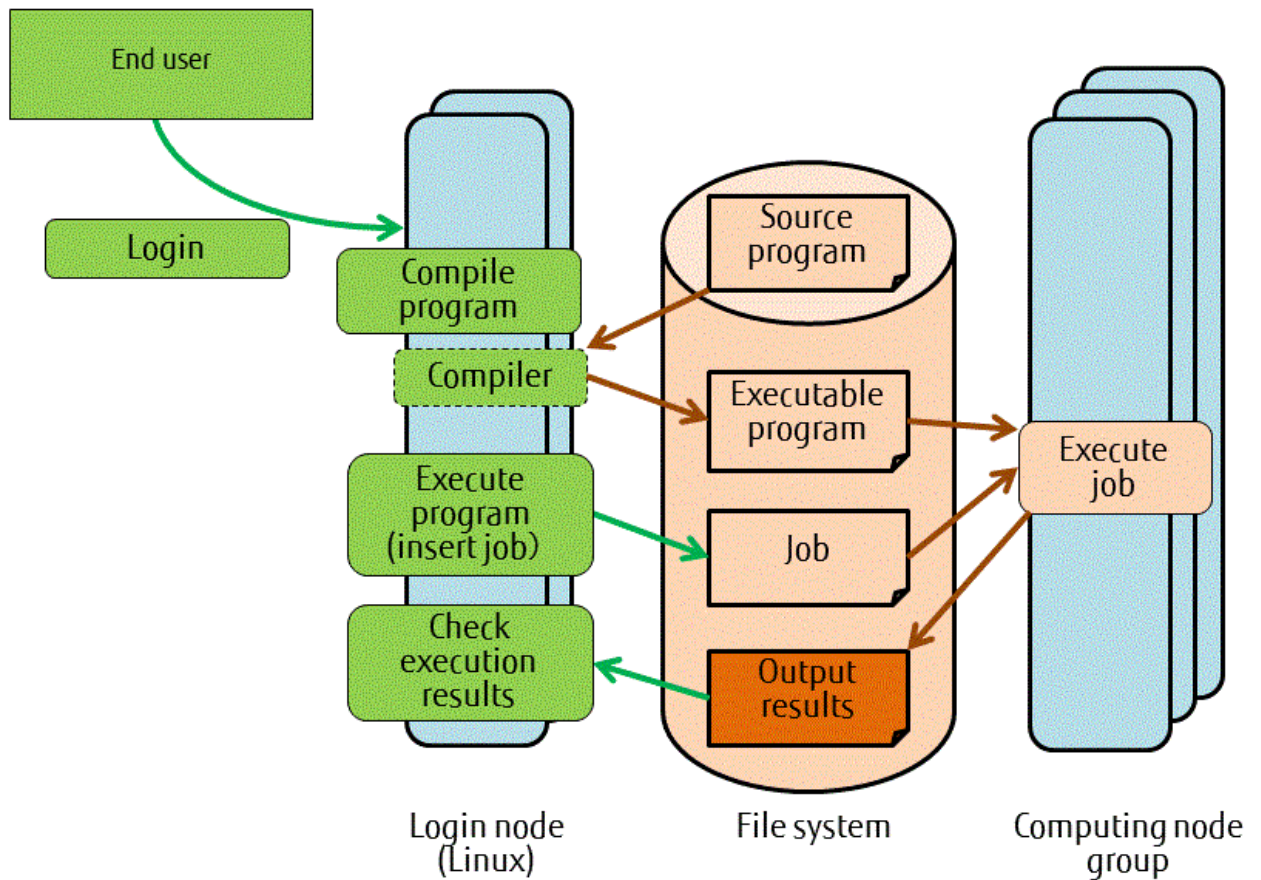
2. Executing the program

Prepare a job script for executing the program created in Step 1 as a job, and pass program execution to the job operation software on the login node (throw the `pjsub` command).

3. Checking the program execution results

If program execution results are output as a file, check the contents of these output results from the login node.

Figure 1.3 Compilation and execution of an existing program from a login node



1.3.4.3 Debugging an Existing Program from a Client

The following workflow applies when debugging an existing program from a client.

1. Compiling the program

Use the "application construction functions" of the programming workbench at the client side to compile the program using the appropriate compiler and to create an executable program for debugging. When compiling the program, specify compile option `-g`. The executable program for debugging is created on the shared file system at the login node.

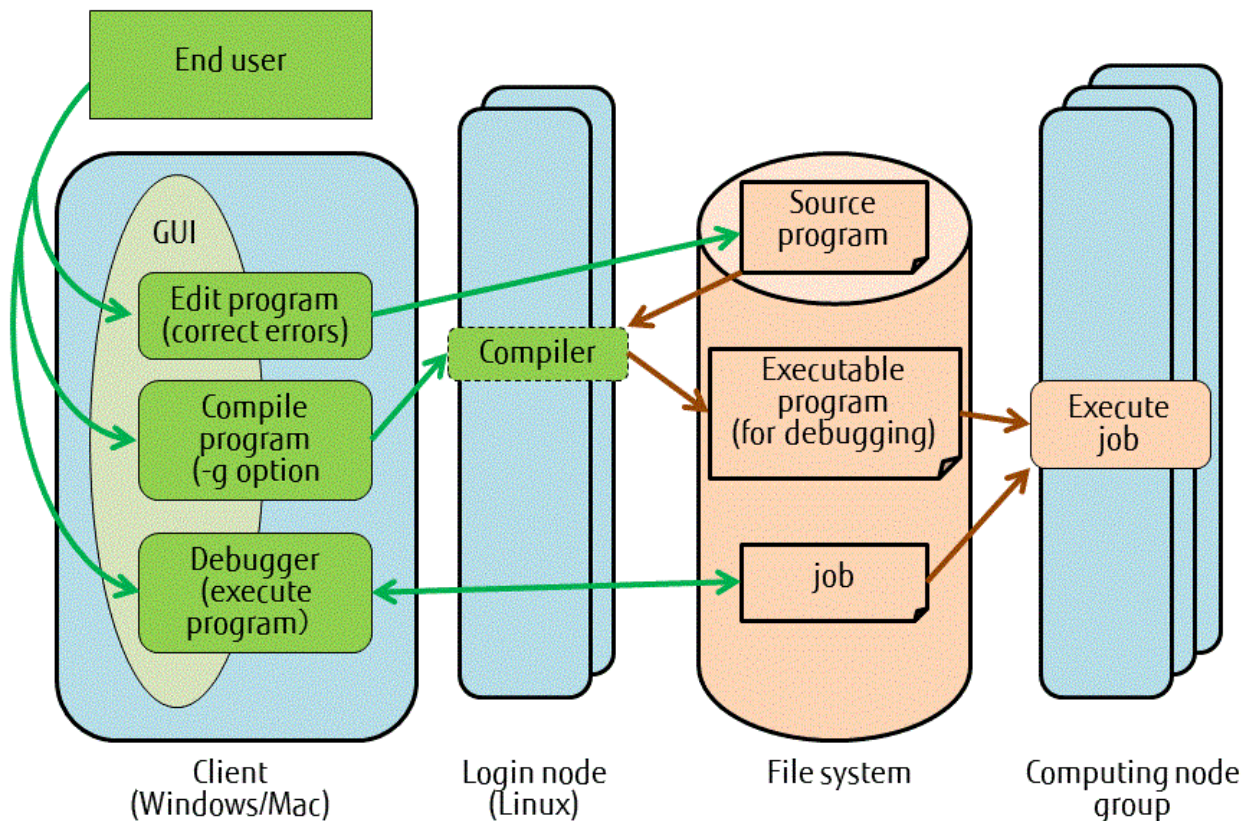
2. Starting the debugger

Use the "launcher functions" of the programming workbench at the client side to start the "debugger". Under the debugger, start execution of the program created in Step 1, and start debugging.

3. Editing the program (correcting program errors)

Use the "file operation functions" of the programming workbench at the client side to correct the program.

Figure 1.4 Debugging an existing program from a client



1.3.4.4 Tuning an Existing Program from a Client

The following workflow applies when tuning an existing program from a client.

1. Compiling the program

Use the "application construction functions" of the programming workbench at the client side to compile the program using the appropriate compiler and to create an executable program. The executable program is created on the shared file system at the login node.

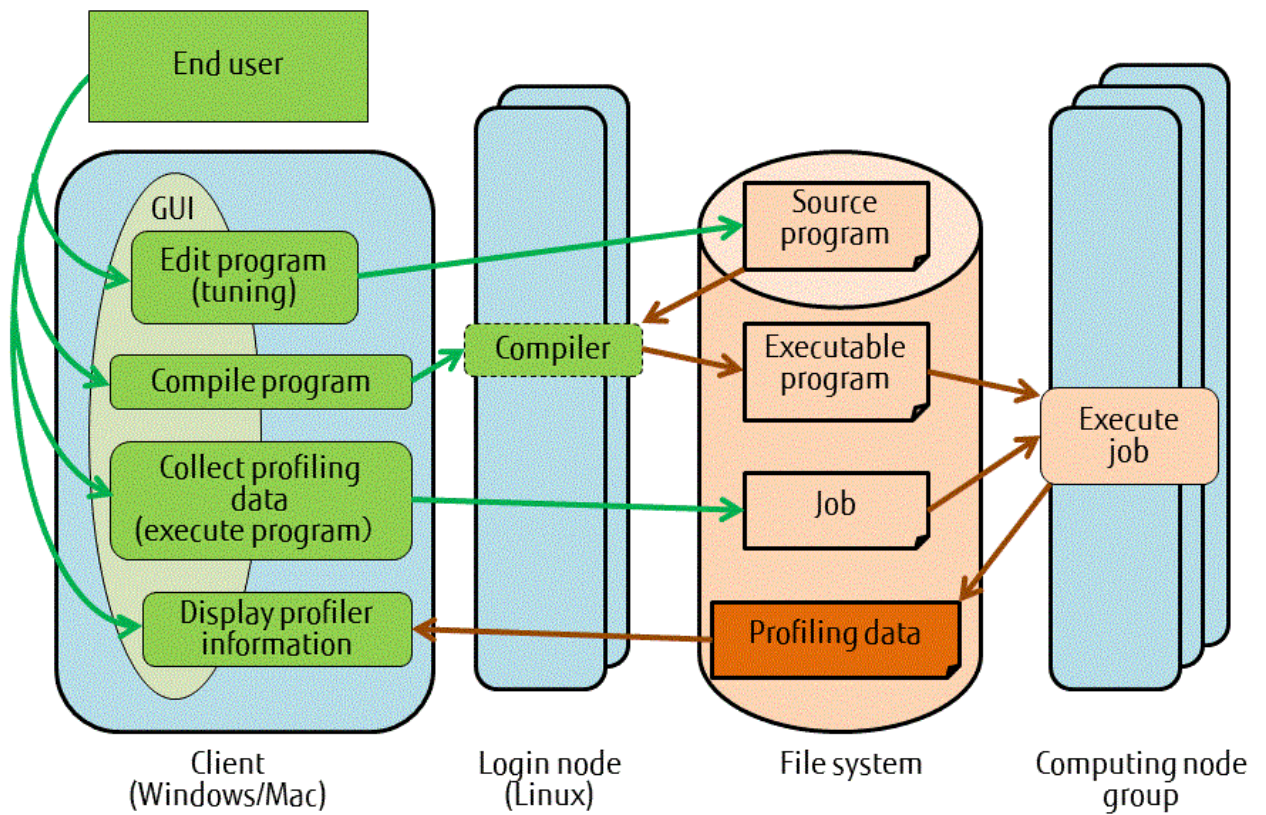
2. Starting the profiler

Use the "launcher functions" of the programming workbench at the client side to start the "profiler". Under the profiler, execute the program created in Step 1, and collect profiling data. When program execution finishes, use the profiler to display the various types of profiler information, and from this information obtain the hints required for tuning the program performance.

3. Editing the program (modifying the program for tuning purposes)

Use the "file operation functions" of the programming workbench at the client side to modify and improve the program.

Figure 1.5 Tuning an existing program from a client



Chapter 2 Functions of Components

2.1 Compiler

2.1.1 Fortran Compiler

The Fortran compiler compiles programs written in Fortran and can create executable programs that are highly optimized, in particular to achieve high CPU execution performance for targeted computing nodes. If required, the Fortran compiler can also create executable programs capable of automatic parallelization or thread-level parallel execution using OpenMP specifications. An execution environment for the fast and secure execution of these executable programs is also prepared.

An overview of Fortran compiler functions is presented below. Refer to the *Fortran User's Guide* for Fortran compiler details.

1. Language specifications

The Fortran compiler supports the following standards specifications:

- ISO/IEC 1539-1:2010 (Fortran 2008 standards) a part of specification
- ISO/IEC 1539-1:2004, JIS X 3001-1:2009 (Fortran 2003 standards)
- ISO/IEC 1539-1:1997, JIS X 3001-1:1998 (Fortran 95 standards)
- Fortran 90 standards and FORTRAN77 standards

The Fortran compiler also supports the main industry standard Fortran language specifications. Refer to the *Fortran Language Reference* for information on the language specifications supported by the Fortran compiler.

2. Main functions

The Fortran compiler is equipped with an "Automatic parallelization function" that enables programs that automatically perform thread-level parallel processing to be created just by specifying the compile options. The Fortran compiler also supports the "OpenMP API Version 3.1 specifications" that perform thread-level parallel processing by means of directive lines specified within the program. A shared memory system is a prerequisite for the automatic parallelization and parallel processing that use OpenMP specifications, and these functions are enabled within the computing nodes. Used in conjunction with the MPI library, these functions support the highly efficient hybrid parallel programming model (thread parallels + MPI process parallels).

The Fortran compiler also supports a wide variety of other functions, such as program debugging (procedure reference verification, undefined data reference, array size validity), precision sensitivity diagnosis through application of precision reduction, trace-back map, error monitoring, and output of compilation information containing parallelization and optimization information.

3. Optimization functions

The Fortran compiler is equipped with optimization functions like those listed below, and can create object programs capable of fast execution at computing nodes. In addition, various optimization control lines are provided in order to facilitate these optimizations from within programs.

- Optimization that reconfigures multiple loops
- Function for scheduling instructions to suit the CPU processor characteristics
- Efficient use of cache by means of pre-fetch instructions
- Optimization functions that effectively use the SPARC64 XIfx (FX100 system) and SPARC64 IXfx (FX10 system) HPC extended mechanisms (if a compiler for a PRIMEHPC system):
 - Increasing degree of parallelism (register extension)
 - Reducing save and restore count for register contents (register extension)
 - Change to SIMD (support also for loops containing IF statements)
 - Software control of sector cache by optimization control lines

2.1.2 C Compiler

The C compiler compiles programs written in C and, like the Fortran compiler, can create executable programs highly optimized to achieve high CPU execution performance for targeted computing nodes. If required, the C compiler can also create executable programs capable of automatic parallelization or thread-level parallel execution using OpenMP specifications.

An overview of C compiler functions is presented below. Refer to the *C User's Guide* for C compiler details.

1. Language specifications

The C compiler supports the following standards specifications:

- ISO/IEC 9899:1990 (C89 standards)
- ISO/IEC 9899:1999 (C99 standards)

The C compiler also supports GNU compiler extended specifications.

2. Main functions

The C compiler is equipped with an "Automatic parallelization function" that enables programs that automatically perform thread-level parallel processing to be created just by specifying the compile options. The C compiler also supports the "OpenMP API Version 3.1 specifications" that perform thread-level parallel processing by means of directive lines specified within the program. A shared memory system is a prerequisite for the automatic parallelization and parallel processing that use OpenMP specifications, and these functions are enabled within the computing nodes. Used in conjunction with the MPI library, these functions support the highly efficient hybrid parallel programming model (thread parallels + MPI process parallels).

The C compiler also supports GNU compiler extended specifications and has superior portability from a variety of systems.

3. Optimization functions

The C compiler is equipped with optimization functions like those listed below, and can create object programs capable of fast execution at computing nodes. In addition, various optimization control lines are provided in order to facilitate these optimizations from within programs.

- Optimization that reconfigures multiple loops
- Function for scheduling instructions to suit the CPU processor characteristics
- Efficient use of cache by means of pre-fetch instructions
- Optimization functions that effectively use the SPARC64 XIfx (FX100 system) and SPARC64 IXfx (FX10 system) HPC extended mechanisms (if a compiler for a PRIMEHPC system):
 - Increasing degree of parallelism (register extension)
 - Reducing save and restore count for register contents (register extension)
 - Change to SIMD (support also for loops containing IF statements)
 - Software control of sector cache by optimization control lines

2.1.3 C++ Compiler

The C++ compiler compiles programs written in C++ language and, like the Fortran compiler, can create executable programs highly optimized to achieve high CPU execution performance for targeted computing nodes. If required, the C++ compiler can also create executable programs capable of automatic parallelization or thread-level parallel execution using OpenMP specifications.

An overview of C++ compiler functions is presented below. Refer to the *C++ User's Guide* for C++ compiler details.

1. Language specifications

The C++ compiler supports the following standards specifications:

- ISO/IEC 14882:2003 (C++03 standards)
- ISO/IEC 14882:2011 (C++11 standards)

The C++ compiler also supports GNU compiler extended specifications.

2. Main functions

The C++ compiler is equipped with an "Automatic parallelization function" that enables programs that automatically perform thread-level parallel processing to be created just by specifying the compile options. The C++ compiler also supports the "OpenMP API Version 3.1 specifications" that perform thread-level parallel processing by means of directive lines specified within the program. A shared memory system is a prerequisite for the automatic parallelization and parallel processing that use OpenMP specifications, and these functions are enabled within the computing nodes. Used in conjunction with the MPI library, these functions support the highly efficient hybrid parallel programming model (thread parallels + MPI process parallels).

The C++ compiler also supports GNU compiler extended specifications and has superior portability from a variety of systems.

3. Optimization functions

The C++ compiler is equipped with optimization functions like those listed below, and can create object programs capable of fast execution at computing nodes. In addition, various optimization control lines are provided in order to facilitate these optimizations from within programs.

- Optimization that reconfigures multiple loops
- Function for scheduling instructions to suit the CPU processor characteristics
- Efficient use of cache by means of pre-fetch instructions
- Optimization functions that effectively use the SPARC64 XIfx (FX100 system) and SPARC64 IXfx (FX10 system) HPC extended mechanisms (if a compiler for a PRIMEHPC system):
 - Increasing degree of parallelism (register extension)
 - Reducing save and restore count for register contents (register extension)
 - Change to SIMD (support also for loops containing IF statements)
 - Software control of sector cache by optimization control lines

2.1.4 XPFortran Translator

The XPFortran translator takes programs written in XPFortran that are intended for distributed memory type parallel computer and converts them to MPI programs coded in Fortran.

Use of the parallel program language XPFortran makes it easy to develop highly parallel programs based on programs in the range that uses Fortran 90 standards. Program coding extended from the Fortran 90 standards specifications as XPFortran is provided as directives (XPFortran extended optimization control lines), and therefore can also be operated as ordinary Fortran programs.

In principle, XPFortran program execution is execution using process parallels. However, unlike ordinary distributed memory type parallel computing, XPFortran can define global memory space shared between multiple processes. By using this global memory space, programming can be performed as if it were a shared memory type parallel processing system even though execution takes place via parallel processes. Synchronization, exclusive control between processes, and functions for data transfer between processes are also provided.

In addition, XPFortran can be used in conjunction with thread parallels and supports the highly efficient hybrid parallel programming model (thread parallels + process parallels by means of XPFortran).

Refer to the *XPFortran User's Guide* for information on XPFortran.

2.2 Mathematical Libraries

In addition to Fujitsu's own mathematics libraries (SSL II and C-SSL II) which are widely used within Japan by R&D users, the linear algebra field libraries (BLAS, LAPACK, and ScaLAPACK) developed in the US are also provided.

These mathematics libraries (except for ScaLAPACK) make it possible for the same routine to be called from multiple threads simultaneously (thread safe).

Important SSL II functions also provide the thread parallel routines that describe the parallel mathematics computing algorithms intended for shared memory type scalar parallel computing in OpenMP Fortran. In addition, these parallel mathematics computing algorithms are also provided as C-SSL II thread parallel routines.

Three-dimensional Fourier transforms parallelized by MPI are provided by SSL II/MPI.

Thread parallel routines are also provided for all BLAS Level 3 routines, for the major Level 2 routines, and for the major LAPACK routines.

All of these mathematical libraries are tuned so that optimized execution performance is obtained at each computing node.

Refer to the *Programmer's Guide for Usage of Mathematical Libraries* for information on Usage of Mathematical Libraries.

Refer to the *SSL II User's Guide*, *FUJITSU SSL II Extended Capabilities User's Guide*, *FUJITSU SSL II Extended Capabilities User's Guide II* and *Fujitsu SSL II Thread-Parallel Capabilities User's Guide* for information on SSL II.

Refer to the *Fujitsu C-SSL II User's Guide* and the *Fujitsu C-SSL II Thread-Parallel Capabilities User's Guide* for information on C-SSL II.

Refer to the *Fujitsu SSL II/MPI User's Guide* for information on SSL II/MPI.

Refer to the *BLAS LAPACK ScaLAPACK User's Guide* for information on BLAS, LAPACK and ScaLAPACK.

Refer to the *Fast Basic Operations Library for Quadruple Precision User's Guide* for information on Fast Basic Operations Library for Quadruple Precision.

2.3 MPI Library

The MPI library (message passing library) conforms to the MPI-2.2 Standard and MPI-3.0 Standard (*1) prescribed by the MPI Forum. The MPI library supports the interconnect of each of the target system, PRIMEHPC system and PC cluster, and achieves high performance and conserves memory. The MPI library for PRIMEHPC system in particular provides optimization suited to network topologies having from one- to three-dimensional torus structures.

The MPI library switches to the transfer method that best suits the transmitted data length in order to make point-to-point communication faster. For PRIMEHPC system, in addition to transmitted data length, process allocation (rank allocation) and the number of hops are also taken into account when switching to the optimum transfer method.

The MPI library for PRIMEHPC system does not use point-to-point communication for collective communication functions that are used frequently. Instead, it recognizes the interconnect topology structure and adopts special-purpose algorithms that reduce congestion in order to make collective communications faster. High-speed barrier and reduction using the interconnect hardware functions are also possible.

The MPI library can be used from Fortran, C, and C++ compilers.

Refer to the *MPI User's Guide* for information on the MPI library.

*1: MPI library that conforms to MPI Standard:

- FX100 System: MPI-3.0 Standard
- PC Cluster: MPI-3.0 Standard
- FX10 System: MPI-2.2 Standard and a part of MPI-3.0 Standard

2.4 Programming Workbench

The programming workbench is a GUI development environment that enables all operations to be performed from the client side using integrated views in all phases of application development for the target system.

Programming workbench has its own file explorer and editor, as well as high-functionality development tools such as a debugger and profiler.

The following functions can be used from the programming workbench:

- Login: The login window for logging in to the login node and launching the programming workbench main window
- Launcher: The main window that launches the various development support functions
- File operations: The file explorer that assists with file creation and file operations
- File editing: The editor that assists with displaying and editing file contents
- Application construction: The builder that assists with creating applications

- Application execution: The executor that assists with execution of the created applications
- Debugging: The debugger that assists with application debugging
- Profiling: The profiler that assists with application tuning

Refer to the *Programming Workbench User's Guide* for information on the programming workbench.

2.4.1 Debugger

The debugger enables the operations below to be performed for application programs written in Fortran, C, or C++. The debugger also enables debugging of programs that support thread parallels and MPI process parallels.

- Controlling program execution
- Setting program execution stop positions
- Displaying expressions and variable values
- Displaying and selecting call stacks

The debugger can also use the command line interface to display debugging information.

The following debugging methods are available for the debugger:

- Ordinary debugging: The programming workbench debugging function at the client (user terminal) throws a job, executes the program from its start, and performs debugging. During debugging, program expressions and variables can be displayed, execution can be controlled, and execution stop positions can be set.
- Core file debugging: Core file debugging uses the core files output when a job ends abnormally to investigate statically the status at the time of abnormal termination.
- Job ID attach debugging: Job attach debugging specifies a job ID and captures all processes of the job. This helps with investigation of the program status if, for example, the job specified by the job ID does not end during execution of that job.

Refer to the *Debugger User's Guide* for debugger details.

2.4.2 Profiler

The profiler is a performance analysis tool that can collect various types of information required for analyzing the performance of application programs written in Fortran, C, and C++. The profiler can also collect profiler information for programs that support thread parallels and MPI process parallels.

The profiler is comprised of an instant profiler, an advanced profiler, and a tracer.

- Instant profiler (a profiler capable of big-picture and comprehensive performance analysis)

The instant profiler can perform big-picture and comprehensive performance analysis with low overheads, even for large-scale parallel applications. It is suitable for any sort of application because it can collect information in a fixed-size file even with long jobs. The following information can be obtained:

- Time statistics information (elapsed time, user CPU time, system CPU time, etc.)
 - Cost information: Sampling is performed during application execution and those counts are output as costs in procedure, loop, and line units.
 - Hardware monitor information: Processor operation circumstances during application execution are output.
 - Call graph information: Procedure call routes are output along with their costs.
 - Source code information: Each line of source code is output with cost information appended.
- Advanced profiler (a profiler capable of analyzing MPI information as well as local and detailed performance analysis)

In cases requiring local and detailed analysis, such as analysis of MPI information and CPU arithmetic unit operation, the advanced profiler can obtain cost measurements that make use of hardware monitor information and time-series information for various events. The following information can be obtained:

- MPI information: The average values, maximum values, and minimum values are output for the MPI function execution count, message length, execution time, and wait time.
- Hardware monitor information: Processor operation circumstances during application execution are output.
- Largepage performance information: Largepage performance information is output.
- Tracer (MPI library and user function time-series information)

The tracer collects in a time series the execution information for user functions called within source code by the MPI library and users.

Refer to the *Profiler User's Guide* for profiler details.

Chapter 3 Additional Functions Overview

This chapter explains functions that have been added with this release.

The "VL" and "Generation Number" in the following tables mean the version level of products and the Generation Number of patches to which functions are added.

For detail of "Generation Number", refer to "1.3.2 Component Packages".

3.1 Compiler

3.1.1 Fortran Compiler

3.1.1.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L30	13	Compiler option	"[= <i>kind</i>]" is added to the -Kilfunc option.	Fortran User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variable is added. - LIBRARY_PATH	Fortran User's Guide "2.3 Compile Command Environment Variable"
		Execution performance improvement	The following functions are improved. - General optimization - SIMD - Automatic parallelization	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
V2.0L20	12	Compiler option	The following compiler options are added. - -K{openmp_tls openmp_notls} - -K{intentopt nointentopt}	Fortran User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. - Automatic Parallelization - General Optimization	None
V2.0L10	11	Compiler option	The following compiler options are added. - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation}	Fortran User's Guide "2.2 Compiler Options"
		Execution performance improvement	The following functions are improved. - Automatic Parallelization	None

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) 	
		Optimization control specifier	<p>The following optimization control specifier is improved.</p> <ul style="list-style-type: none"> - ARRAY_MERGE 	Fortran User's Guide "9.10 Using Optimization control line (OCL)"
	10	Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> -K{loop_versioning loop_noversioning} -K{parallel_fp_precision parallel_nofp_precision} -K{omitfp noomitfp} -g0 	Fortran User's Guide "2.2 Compiler Options"
		Optimization control specifier	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - CLONE - {LOOP_VERSIONING LOOP_NOVERSIONING} 	Fortran User's Guide "9.10 Using Optimization control line (OCL)"
		Execution performance improvement	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization - Instruction Scheduling (Software Pipelining) 	None
			The execution performance of an array assignment statement is improved in the FORALL construct.	None
			The inline expansion function is improved when compiler option -Knoalias=s is effective.	None
			The execution performance of the threadprivate directive of OpenMP is improved.	None
		Fortran 2008 Specifications	Data object has the implicit SAVE attribute in declaration part of the module.	Fortran Language Reference
V1.0L30	09	Fortran 2008 Specifications	<p>The following specifications are added.</p> <ul style="list-style-type: none"> - OPTIONAL attribute of dummy argument and, actual argument of non-associated pointer, non-allocated allocatable or reference of intrinsic function NULL(). - Bound remapping list specification is added in pointer assignment statement - Specifications are added in module ISO_FORTRAN_ENV and ISO_C_BINDING. - CONCURRENT in DO construct. - CONTIGUOUS attribute and intrinsic function IS_CONTIGUOUS. 	Fortran Language Reference

VL	Generation Number	Function	Description	Reference
			- MOLD specifier in ALLOCATE statement.	
		Fortran 2003 Specifications	The following checking is added in compilation time. - Reference non pure final procedure in pure procedure.	Fortran Language Reference
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	Fortran User's Guide "12.3 Parallelization by OpenMP Specifications"
		Compiler option -AU for Fujitsu provided module	Specification of compiler option -AU is improved for Fujitsu provided module.	None
		Optimization messages improvement	Optimization messages of software pipelining are improved.	Fortran Compiler Messages
		Execution performance improvement	Execution performance is improved when user's procedure reference has a pointer array or assumed shape array in actual argument and compiler option -H is specified.	None
			The following execution performances are improved. - Automatic parallelization - SIMD - General optimization (unrolling, etc.) - Inline expansion of intrinsic functions (tan, exp, and matmul)	None
V1.0L20	07	Execution performance improvement	Execution performance is improved when user's function reference has a pointer array or assumed shape array in actual argument.	None
			Compiler option -Ksimd_region_constant is added.	Fortran User's Guide "2.2 Compiler Options"
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	Fortran User's Guide "F.5 Migrating to V1.0L20(Generation Number:07)"
	06	Control of the initial value zero in an undefined variable	Compiler option -N{setvalue nosetvalue} is added.	Fortran User's Guide "2.2 Compiler Options"
		Compilation performance improvement	Compilation performance is improved when POINTER components of derived type exist in derived type.	None
		Improvement the variable value of the undefined specification	Improves to fix the variable value of the specification that is the undefined value in Fortran standard or OpenMP standard.	None
		Execution performance improvement	The following execution performances are improved. - Automatic parallelization	None

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - SIMD - General optimization (full unrolling, etc.) - Instruction scheduling (software pipelining) 	
		Optimization messages and optimization information improvement	<p>The following optimization messages and optimization information are improved.</p> <ul style="list-style-type: none"> - Automatic parallelization - SIMD - Software pipelining 	Fortran Compiler Messages
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - <code>-Nlst[={a d i m p t x}]</code> - <code>-Nlst_out=file</code> 	Fortran User's Guide "2.2 Compiler Options"
			<p>The message for unrecognized compiler option is output.</p>	Fortran User's Guide "F.6 Migrating to V1.0L20(Generation Number:06)"
		Performance improvement of Runtime information output function	<p>The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.</p>	None
		Performance improvement of Fortran input/output statements	<p>The memory access performance is improved by decreasing the use of thread local storage.</p>	None
	05	Fortran 2008 Specifications	<p>Support intrinsic procedure ACOSH, ASINH, ATANH, BGE, BGT, BLE, and BLT, etc.</p>	Fortran Language Reference
		Release restrictions for Fortran 2003 specification	<p>Release the following restrictions for Fortran 2003 specification:</p> <ul style="list-style-type: none"> - FINAL subroutine - ASSOCIATE construct / SELECT TYPE construct - Type bound procedure of user defined operation and assignment - Intrinsic procedure in initialization expression - Structure constructor of component has CLASS(*) and ALLOCATABLE attribute 	Fortran Language Reference
		Addition of optimization control specifiers	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - TEMP_PRIVATE - FIRST_PRIVATE - LAST_PRIVATE - PARALLEL_CYCLIC - {LOOP_PART_PARALLEL LOOP_NOPART_PARALLEL} 	Fortran User's Guide "9.10 Using Optimization control line (OCL)", and "12.2.3.2.4 Optimization Control Specifiers for Automatic Parallelization"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - {LOOP_PART_SIMD LOOP_NOPART_SIMD} - UNSWITCHING - {SHORTLOOP NOSHORTLOOP} - SIMD_LISTV 	
		Execution performance improvement	<p>The following execution performances are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) - Intrinsic functions (aint and dint, etc.) - Promotion of optimization for large loop 	Fortran User's Guide "2.2 Compiler Options"
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.1.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	Compiler option	"[= <i>kind</i>]" is added to the -Kilfunc option.	Fortran User's Guide "2.2 Compiler Options"
			The following compiler option is added. <ul style="list-style-type: none"> - -K{ simd_reduction_product simd_noreduction_product } 	Fortran User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variable is added. <ul style="list-style-type: none"> - LIBRARY_PATH 	Fortran User's Guide "2.3 Compile Command Environment Variable"
		Fortran 2003 Specifications	Add the user-defined derived type input/output specification.	Fortran User's Guide Fortran Language Reference
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None

VL	Generation Number	Function	Description	Reference
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
V2.0L20	03	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_tls openmp_notls} - -K{FLTLTD NOFLTLTD} - -K{HPC_ACE HPC_ACE2} - -K{intentopt nointentopt} - -K{lto nolto} - -K{simd_separate_stride simd_noseparate_stride} - -N{ coarray nocoarray } 	Fortran User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization 	None
		Optimization control specifier	The following optimization control specifiers are added. <ul style="list-style-type: none"> - {FLTLTD NOFLTLTD} - SIMD[(ALIGNED UNALIGNED)] - {SIMD_REDUNDANT_VL(<i>n</i>) SIMD_NOREDUNDANT_VL} - UXSIMD[(ALIGNED UNALIGNED)] 	Fortran User's Guide "9.10 Using Optimization control line (OCL)"
		Fortran 2003 Specifications	Add the following specifications: <ul style="list-style-type: none"> - Extend PROCEDURE statement in interface block - Parameterized derived types 	Fortran Language Reference
		Fortran 2008 Specifications	Add the following specifications: <ul style="list-style-type: none"> - BLOCK construct - Implied shape array - ERROR STOP statement - Extend STOP statement - Double colon after PROCEDURE in interface block 	Fortran Language Reference
		COARRAY feature	COARRAY feature is supported.	Fortran User's Guide Additional Volume COARRAY

VL	Generation Number	Function	Description	Reference
				Fortran Language Reference
V2.0L10	02	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation} 	Fortran User's Guide "2.2 Compiler Options"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) 	None
		Optimization control specifier	The following optimization control specifier is improved. <ul style="list-style-type: none"> - ARRAY_MERGE 	Fortran User's Guide "9.10 Using Optimization control line (OCL)"
		Optimization information improvement	SIMD optimization information is improved. ("VL: <i>length</i> " is adding displayed.)	Fortran User's Guide "4.1.2.1 Compilation Information Options"

3.1.1.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -KCORE_AVX2 - -K{fp_contract nofp_contract} 	Fortran User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variable is added. <ul style="list-style-type: none"> - LIBRARY_PATH 	Fortran User's Guide "2.3 Compile Command Environment Variable"
		Fortran 2003 Specifications	Add the user-defined derived type input/output specification.	Fortran User's Guide Fortran Language Reference
		COARRAY feature	COARRAY feature is supported.	Fortran User's Guide Additional Volume COARRAY Fortran Language Reference
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None

VL	Generation Number	Function	Description	Reference
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
V2.0L20	14	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_tls openmp_notls} - -K{intentopt nointentopt} - -K{lto nolto} - -N{ coarray nocoarray } 	Fortran User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
		Fortran 2003 Specifications	Add the following specifications: <ul style="list-style-type: none"> - Extend PROCEDURE statement in interface block - Parameterized derived types 	Fortran Language Reference
		Fortran 2008 Specifications	Add the following specifications: <ul style="list-style-type: none"> - BLOCK construct - Implied shape array - ERROR STOP statement - Extend STOP statement - Double colon after PROCEDURE in interface block 	Fortran Language Reference
V2.0L10	13	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation} - -K{simd_extension simd_noextension} - -K{simd_maskload simd_nomaskload} 	Fortran User's Guide "2.2 Compiler Options"
		Optimization control specifier	The following optimization control specifiers are added. <ul style="list-style-type: none"> - {SIMD_EXTENSION SIMD_NOEXTENSION} - {SIMD_MASKLOAD SIMD_NOMASKLOAD} 	Fortran User's Guide "9.11 Using Optimization control line (OCL)"
			The following optimization control specifier is improved. <ul style="list-style-type: none"> - ARRAY_MERGE 	Fortran User's Guide "9.11 Using Optimization control line (OCL)"

VL	Generation Number	Function	Description	Reference
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization 	None
		Optimization information improvement	SIMD optimization information is improved. ("VL: <i>length</i> " is adding displayed.)	Fortran User's Guide "4.1.2.1 Compilation Information Options"
	12	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - <code>-K{loop_versioning loop_noversioning}</code> - <code>-K{parallel_fp_precision parallel_nofp_precision}</code> - <code>-g0</code> 	Fortran User's Guide "2.2 Compiler Options"
		Optimization control specifier	The following optimization control specifiers are added. <ul style="list-style-type: none"> - <code>CLONE</code> - <code>{LOOP_VERSIONING LOOP_NOVERSIONING}</code> 	Fortran User's Guide "9.10 Using Optimization control line (OCL)"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
			The execution performance of an array assignment statement is improved in the FORALL construct.	None
			The inline expansion function is improved when compiler option <code>-Knoalias=s</code> is effective.	None
			The execution performance of the threadprivate directive of OpenMP is improved.	None
		Fortran 2008 Specifications	Data object has the implicit SAVE attribute in declaration part of the module.	Fortran Language Reference
V1.0L30	11	Fortran 2008 Specifications	The following specifications are added. <ul style="list-style-type: none"> - OPTIONAL attribute of dummy argument and, actual argument of non-associated pointer, non-allocated allocatable or reference of intrinsic function NULL(). - Bound remapping list specification is added in pointer assignment statement - Specifications are added in module ISO_FORTRAN_ENV and ISO_C_BINDING. - CONCURRENT in DO construct. - CONTIGUOUS attribute and intrinsic function IS_CONTIGUOUS. MOLD specifier in ALLOCATE statement.	Fortran Language Reference

VL	Generation Number	Function	Description	Reference
		Fortran 2003 Specifications	The following checking is added in compilation time. Reference non pure final procedure in pure procedure.	Fortran Language Reference
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	Fortran User's Guide "12.3 Parallelization by OpenMP Specifications"
		Compiler option -AU for Fujitsu provided module	Specification of compiler option -AU is improved for Fujitsu provided module.	None
		Execution performance improvement	Execution performance is improved when user's procedure reference has a pointer array or assumed shape array in actual argument and compiler option -H is specified.	None
			The following execution performances are improved. - Automatic parallelization - SIMD - General optimization (unrolling, etc.) - Inline expansion of intrinsic functions (matmul)	None
V1.0L20	10	Execution performance improvement	Execution performance is improved when user's function reference has a pointer array or assumed shape array in actual argument.	None
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	Fortran User's Guide "G.5 Migrating to V1.0L20(Generation Number:10)"
	09	Control of the initial value zero in an undefined variable	Compiler option -N{setvalue nosetvalue} is added.	Fortran User's Guide "2.2 Compiler Options"
		Compilation performance improvement	Compilation performance is improved when POINTER components of derived type exist in derived type.	None
		Improvement the variable value of the undefined specification	Improves to fix the variable value of the specification that is the undefined value in Fortran standard or OpenMP standard.	None
		Execution performance improvement	The following execution performances are improved. - Automatic parallelization - SIMD - General optimization (full unrolling, etc.) - Instruction scheduling	None
		Optimization messages and optimization	The following optimization messages and optimization information are improved.	Fortran Compiler Messages

VL	Generation Number	Function	Description	Reference
		information improvement	<ul style="list-style-type: none"> - Automatic parallelization - SIMD 	
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - <code>-Nlst[={a d i m p t x}]</code> - <code>-Nlst_out=<i>file</i></code> 	Fortran User's Guide "2.2 Compiler Options"
			The message for unrecognized compiler option is output.	Fortran User's Guide "G.6 Migrating to V1.0L20(Generation Number:09)"
		Performance improvement of Runtime information output function	The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.	None
	08	Fortran 2008 Specifications	Support intrinsic procedure ACOSH, ASINH, ATANH, BGE, BGT, BLE, and BLT, etc.	Fortran Language Reference
		Release restrictions for Fortran 2003 specification	<p>Release the following restrictions for Fortran 2003 specification:</p> <ul style="list-style-type: none"> - FINAL subroutine - ASSOCIATE construct / SELECT TYPE construct - Type bound procedure of user defined operation and assignment - Intrinsic procedure in initialization expression - Structure constructor of component has CLASS(*) and ALLOCATABLE attribute 	Fortran Language Reference
		Addition of optimization control specifiers	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - TEMP_PRIVATE - FIRST_PRIVATE - LAST_PRIVATE - PARALLEL_CYCLIC - {LOOP_PART_PARALLEL LOOP_NOPART_PARALLEL} - {LOOP_PART_SIMD LOOP_NOPART_SIMD} - UNSWITCHING 	Fortran User's Guide "9.10 Using Optimization control line (OCL)", and "12.2.3.2.4 Optimization Control Specifiers for Automatic Parallelization"
		Execution performance improvement	<p>The following execution performances are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) 	Fortran User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - Intrinsic functions (aint and dint, etc.) - Promotion of optimization for large loop 	
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.2 C Compiler

3.1.2.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L30	13	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	"[={loop procedure}]" is added to the -Kilfunc option.	C User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variables are added. <ul style="list-style-type: none"> - CPATH - C_INCLUDE_PATH - LIBRARY_PATH 	C User's Guide "2.3 Environment Variable for Compile Command"
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - Built-in functions with names that start with '__atomic' - __builtin_index - __builtin_isinf - __builtin_isinff - __builtin_isinfl - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex 	C User's Guide "E.1.2 Built-in Functions"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None

VL	Generation Number	Function	Description	Reference
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
V2.0L20	12	GNU C Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C User's Guide "E.1 GNU C Extensions"
		Compiler option	The following compiler option is added. - -K{openmp_tls openmp_notls}	C User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. - Automatic Parallelization - General Optimization	None
V2.0L10	11	Compiler option	The following compiler options are added. - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation}	C User's Guide "2.2 Compiler Options"
		Execution performance improvement	The following functions are improved. - Automatic Parallelization - SIMD - General Optimization - Instruction Scheduling (Software Pipelining)	None
		Built-in Functions	The following Built-in Functions are added. - __sync_synchronize - __sync_lock_test_and_set - __sync_lock_release	C User's Guide "E.1.2 Built-in Functions"
	10	Compiler option	The following compiler options are added. - -K{loop_versioning loop_noversioning} - -K{parallel_fp_precision parallel_nofp_precision} - -K{omitfp noomitfp} - -g0	C User's Guide "2.2 Compiler Options"
		Optimization control specifier	The following optimization control specifiers are added. - clone - {loop_versioning loop_noversioning}	C User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance improvement	The following functions are improved. - Automatic Parallelization - General Optimization	None

VL	Generation Number	Function	Description	Reference
			- Instruction Scheduling (Software Pipelining)	
			The execution performance of schedule(static) of OpenMP loop construct is improved.	None
V1.0L30	09	Control of the initial value zero in an undefined variable	Compiler option -N{setvalue nosetvalue} is added.	C User's Guide "2.2.2.6 -N Option"
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - __sync_fetch_and_add - __sync_fetch_and_sub - __sync_fetch_and_or - __sync_fetch_and_and - __sync_fetch_and_xor - __sync_fetch_and_nand - __sync_add_and_fetch - __sync_sub_and_fetch - __sync_or_and_fetch - __sync_and_and_fetch - __sync_xor_and_fetch - __sync_nand_and_fetch - __sync_bool_compare_and_swap - __sync_val_compare_and_swap 	C User's Guide "E.1.2 Built-in Functions"
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	C User's Guide "4.3 Parallelization by OpenMP Specifications"
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Optimization messages improvement	Optimization messages of software pipelining are improved.	C/C++ Compiler Optimization Messages
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (unrolling, etc.) - Inline expansion of intrinsic functions (tan, exp) 	None
V1.0L20	07	Execution performance improvement	Compiler option -Ksimd_region_constant is added.	C User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	C User's Guide "H.4 Migrating to V1.0L20(Generation Number:07)"
	06	Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (inline expansion, full unrolling, etc.) - Instruction scheduling (software pipelining) 	None
		Optimization messages and optimization information improvement	The following optimization messages and optimization information are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - Software pipelining 	C/C++ Compiler Optimization Messages
		Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -Nlst[={p t}] - -Nlst_out=<i>file</i> 	C User's Guide "2.2.2.6 -N Option"
			The message for unrecognized compiler option is output.	C User's Guide "H.5 Migrating to V1.0L20(Generation Number:06)"
		Performance improvement of Runtime information output function	The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.	None
	05	GNU C compatible options	The following options became available regardless of the -Xg option. <ul style="list-style-type: none"> - -M, -MD, -MF, -MM, -MMD, -MP, -MT The following GNU C compatible option is supported. <ul style="list-style-type: none"> - --version 	C User's Guide "2.2 Compiler Options" and "E.2 GNU C Compatible Options"
		Addition of optimization control specifiers	The following optimization control specifiers are added. <ul style="list-style-type: none"> - temp_private - first_private - last_private - parallel_cyclic - {loop_part_parallel loop_nopart_parallel} 	C User's Guide "3.4.1 Using the Optimization Control Line (OCL) ", and "4.2.3.2 Optimization Control Line"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - {loop_part_simd loop_nopart_simd} - unswitching - {shortloop noshortloop} - simd_listv 	
		Execution performance improvement	<p>The following execution performances are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) - Promotion of optimization for large loop 	C User's Guide "2.2 Compiler Options"
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.2.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	<p>"[={loop procedure}]" is added to the -Kilfunc option.</p> <p>The following compiler option is added.</p> <ul style="list-style-type: none"> - -K{ simd_reduction_product simd_noreduction_product } 	C User's Guide "2.2 Compiler Options"
		Environment variable	<p>The following environment variables are added.</p> <ul style="list-style-type: none"> - CPATH - C_INCLUDE_PATH - LIBRARY_PATH 	C User's Guide "2.3 Environment Variable for Compile Command"
		Built-in Functions	<p>The following Built-in Functions are added.</p> <ul style="list-style-type: none"> - Built-in functions with names that start with '__atomic' - __builtin_index - __builtin_isinf - __builtin_isinff - __builtin_isinfl 	C User's Guide "E.1.2 Built-in Functions"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex 	
		Execution performance improvement	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
		Optimization message	Optimization messages of link time optimization are improved.	C/C++ Compiler Optimization Messages
V2.0L20	03	GNU C Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C User's Guide "E.1 GNU C Extensions"
		SIMD built-in functions	SIMD built-in functions for HPC-ACE and HPC-ACE2 are supported.	C User's Guide "Appendix F SIMD built-in functions"
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - -K{openmp_tls openmp_notls} - -K{FLTLTD NOFLTLTD} - -K{HPC_ACE HPC_ACE2} - -K{lto nocto} - -K{simd_separate_stride simd_noseparate_stride} 	C User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			<p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization 	None
		Optimization control specifier	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - {ftld nofltd} - simd[aligned unaligned] - {simd_redundant_v l simd_noredundant_vl} - uxsimd[aligned unaligned] 	C User's Guide "3.4.1 Using the Optimization Control Line (OCL)"

VL	Generation Number	Function	Description	Reference
V2.0L10	02	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation} 	C User's Guide "2.2 Compiler Options"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) 	None
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - __sync_synchronize - __sync_lock_test_and_set - __sync_lock_release 	C User's Guide "E.1.2 Built-in Functions"
		Optimization information improvement	SIMD optimization information is improved. ("VL: <i>length</i> " is adding displayed.)	C User's Guide "5.1.2.2.7 Details Optimization Information"

3.1.2.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -KCORE_AVX2 - -K{fp_contract nofp_contract} 	C User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variables are added. <ul style="list-style-type: none"> - CPATH - C_INCLUDE_PATH - LIBRARY_PATH 	C User's Guide "2.3 Environment Variable for Compile Command"
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - Built-in functions with names that start with '__atomic' - __builtin_index - __builtin_isinf - __builtin_isinff - __builtin_isinfl 	C User's Guide "E.1.2 Built-in Functions"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex - Built-in functions with names that start with '_sync' 	
		Execution performance improvement	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
		Optimization message	Optimization messages of link time optimization are improved.	C/C++ Compiler Optimization Messages
V2.0L20	14	GNU C Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C User's Guide "E.1 GNU C Extensions"
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - K{openmp_tls openmp_notls} - K{lto nocto} 	C User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			<p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
V2.0L10	13	Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - K{openmp_assume_norecurrence openmp_noassume_norecurrence} - N{cancel_overtime_compilation nocancel_overtime_compilation} - K{simd_extension simd_noextension} - K{simd_maskload simd_nomaskload} 	C User's Guide "2.2 Compiler Options"
		Optimization control specifier	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - {simd_extension simd_noextension} - {simd_maskload simd_nomaskload} 	C User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization 	None

VL	Generation Number	Function	Description	Reference
		improvement	<ul style="list-style-type: none"> - SIMD - General Optimization 	
		Optimization information improvement	SIMD optimization information is improved. (" VL: <i>length</i> " is adding displayed.)	C User's Guide " 5.1.2.2.6 Details Optimization Information"
	12	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - K{loop_versioning loop_noversioning} - K{parallel_fp_precision parallel_nofp_precision} - g0 	C User's Guide "2.2 Compiler Options"
		Optimization control specifier	The following optimization control specifiers are added. <ul style="list-style-type: none"> - clone - {loop_versioning loop_noversioning} 	C User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
			The execution performance of schedule(static) of OpenMP loop construct is improved.	None
V1.0L30	11	Control of the initial value zero in an undefined variable	Compiler option -N{setvalue nosetvalue} is added.	C User's Guide "2.2.2.6 -N Option"
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	C User's Guide "4.3 Parallelization by OpenMP Specifications"
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (unrolling, etc.) 	None
V1.0L20	10	Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	C User's Guide "H.4 Migrating to V1.0L20(Generation Number: 10)"
	09	Execution performance improvement	The following execution performances are improved.	None

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (inline expansion, full unrolling, etc.) - Instruction scheduling 	
		Optimization messages and optimization information improvement	<p>The following optimization messages and optimization information are improved.</p> <ul style="list-style-type: none"> - Automatic parallelization - SIMD 	C/C++ Compiler Optimization Messages
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - <code>-Nlst[={p t}]</code> - <code>-Nlst_out=file</code> 	C User's Guide "2.2.2.6 -N Option"
			The message for unrecognized compiler option is output.	C User's Guide "H.5 Migrating to V1.0L20(Generation Number:09)"
		Performance improvement of Runtime information output function	The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.	None
	08	GNU C compatible options	<p>The following options became available regardless of the <code>-Xg</code> option.</p> <ul style="list-style-type: none"> - <code>-M, -MD, -MF, -MM, -MMD, -MP, -MT</code> <p>The following GNU C compatible option is supported.</p> <ul style="list-style-type: none"> - <code>--version</code> 	C User's Guide "2.2 Compiler Options" and "E.2 GNU C Compatible Options"
		Addition of optimization control specifiers	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - <code>temp_private</code> - <code>first_private</code> - <code>last_private</code> - <code>parallel_cyclic</code> - <code>{loop_part_parallel loop_nopart_parallel}</code> - <code>{loop_part_simd loop_nopart_simd}</code> - <code>unswitching</code> 	C User's Guide "3.4.1 Using the Optimization Control Line (OCL) ", and "4.2.3.2 Optimization Control Line"
		Execution performance improvement	<p>The following execution performances are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) 	C User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
			- Promotion of optimization for large loop	
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.3 C++ Compiler

3.1.3.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L30	13	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C++ User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	"[={loop procedure}]" is added to the -Kilfunc option.	C++ User's Guide "2.2 Compiler Options"
			The following compiler options are added. - -Nstl=libc++ - -stdlib= <i>stl_kind</i>	C++ User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variables are added. - CPATH - CPLUS_INCLUDE_PATH - LIBRARY_PATH	C++ User's Guide "2.3 Environment Variable for Compile Command"
		C++11 specifications	The C++11 specifications are supported.	C++ User's Guide "6.3 C++11 Specifications"
		Built-in Functions	The following Built-in Functions are added. - Built-in functions with names that start with '_atomic' - __builtin_index - __builtin_isinf - __builtin_isinff - __builtin_isinfl - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex	C++ User's Guide "D.1.2 Built-in Functions"

VL	Generation Number	Function	Description	Reference
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization 	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
V2.0L20	12	C++11 specifications(part)	The C++11 specifications (part) are supported further.	C++ User's Guide "6.3 C++11 Specifications"
		GNU C++ Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C++ User's Guide "D.1 GNU C++ Extensions"
		GNU C++ Compatible Options	The following GNU C++ compatible options are supported. <ul style="list-style-type: none"> - -std={c++03 c++11 gnu++03 gnu++11} 	C++ User's Guide "D.2 GNU C++ Compatible Options"
		Compiler option	The following compiler option is added. <ul style="list-style-type: none"> - -K{openmp_tls openmp_notls} 	C++ User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
V2.0L10	11	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation} 	C++ User's Guide "2.2 Compiler Options"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) 	None
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - __sync_synchronize - __sync_lock_test_and_set - __sync_lock_release 	C++ User's Guide "D.1.2 Built-in Functions"
	10	Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -K{loop_versioning loop_noversioning} 	C++ User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - -K{parallel_fp_precision parallel_nofp_precision} - -K{remove_inlinefunction noremove_inlinefunction} - -K{omitfp noomitfp} - -g0 	
		Optimization control specifier	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - clone - {loop_versioning loop_noversioning} 	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance improvement	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization - Instruction Scheduling (Software Pipelining) 	None
			The execution performance of schedule(static) of OpenMP loop construct is improved.	None
		C++11 specifications (part)	<p>The following compiler option is added.</p> <ul style="list-style-type: none"> - -std={c++03 c++11} 	C++ User's Guide "2.2.2.4 Options for Language Specifications"
V1.0L30	09	Control of the initial value zero in an undefined variable	Compiler option -N{setvalue nosetvalue} is added.	C++ User's Guide "2.2.2.7 -N Option"
		Built-in Functions	<p>The following Built-in Functions are added.</p> <ul style="list-style-type: none"> - __sync_fetch_and_add - __sync_fetch_and_sub - __sync_fetch_and_or - __sync_fetch_and_and - __sync_fetch_and_xor - __sync_fetch_and_nand - __sync_add_and_fetch - __sync_sub_and_fetch - __sync_or_and_fetch - __sync_and_and_fetch - __sync_xor_and_fetch - __sync_nand_and_fetch - __sync_bool_compare_and_swap - __sync_val_compare_and_swap 	C++ User's Guide "D.1.2 Built-in Functions"
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	C++ User's Guide "4.3 Parallelization by

VL	Generation Number	Function	Description	Reference
				OpenMP Specifications"
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Optimization messages improvement	Optimization messages of software pipelining are improved.	C/C++ Compiler Optimization Messages
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (unrolling, etc.) - Inline expansion of intrinsic functions (tan, exp) 	None
V1.0L20	07	Execution performance improvement	Compiler option -Ksimd_region_constant is added.	C++ User's Guide "2.2 Compiler Options"
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	C++ User's Guide "G.4 Migrating to V1.0L20(Generation Number:07)"
	06	Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (inline expansion, full unrolling, etc.) - Instruction scheduling (software pipelining) 	None
		Optimization messages and optimization information improvement	The following optimization messages and optimization information are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - Software pipelining 	C/C++ Compiler Optimization Messages
		Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - -Nlst[={p/t}] - -Nlst_out=<i>file</i> - -Nstl=500fast 	C++ User's Guide "2.2.2.7 -N Option"
			The message for unrecognized compiler option is output.	C++ User's Guide "G.5 Migrating to

VL	Generation Number	Function	Description	Reference
				V1.0L20(Generation Number:06)"
		Performance improvement of Runtime information output function	The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.	None
	05	GNU C++ compatible options	The following options became available regardless of the -Xg option. <ul style="list-style-type: none"> - -M, -MD, -MF, -MM, -MMD, -MP, -MT The following GNU C++ compatible option is supported. <ul style="list-style-type: none"> - --version 	C++ User's Guide "2.2 Compiler Options" and "D.2 GNU C++ Compatible Options"
		Standard template library	Upgrades the standard template library.	C++ User's Guide "2.2.2.7 -N Option" - Nstl={500 521} option
		Addition of optimization control specifiers	The following optimization control specifiers are added. <ul style="list-style-type: none"> - temp_private - first_private - last_private - parallel_cyclic - {loop_part_parallel loop_nopart_parallel} - {loop_part_simd loop_nopart_simd} - unswitching - {shortloop noshortloop} - simd_listv 	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL) ", and"4.2.3.2 Optimization Control Line"
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) - Promotion of optimization for large loop 	C++ User's Guide "2.2 Compiler Options"
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.3.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C++ User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	"[={loop procedure}]" is added to the -Kilfunc option.	C++ User's Guide "2.2 Compiler Options"
			The following compiler options are added. <ul style="list-style-type: none"> - -K{simd_reduction_product simd_noreduction_product} - -Nstl=libc++ - -stdlib=<i>stl_kind</i> 	C++ User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variables are added. <ul style="list-style-type: none"> - CPATH - CPLUS_INCLUDE_PATH - LIBRARY_PATH 	C++ User's Guide "2.3 Environment Variable for Compile Command"
		C++11 specifications	The C++11 specifications are supported.	C++ User's Guide "6.3 C++11 Specifications"
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - Built-in functions with names that start with '__atomic' - __builtin_index - __builtin_isinf - __builtin_isinff - __builtin_isinfl - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex 	C++ User's Guide "D.1.2 Built-in Functions"
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization - Link time optimization 	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None

VL	Generation Number	Function	Description	Reference
		Optimization message	Optimization messages of link time optimization are improved.	C/C++ Compiler Optimization Messages
V2.0L20	03	C++11 specifications(part)	The C++11 specifications (part) are supported further.	C++ User's Guide "6.3 C++11 Specifications"
		GNU C++ Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C++ User's Guide "D.1 GNU C++ Extensions"
		GNU C++ Compatible Options	The following GNU C++ compatible options are supported. - -std={c++03 c++11 gnu++03 gnu++11}	C++ User's Guide "D.2 GNU C++ Compatible Options"
		SIMD built-in functions	SIMD built-in functions for HPC-ACE and HPC-ACE2 are supported.	C++ User's Guide "Appendix E SIMD built-in functions"
		Compiler option	The following compiler options are added. - -K{openmp_tls openmp_notls} - -K{FLTLD NOFLTLD} - -K{HPC_ACE HPC_ACE2} - -K{lto nocto} - -K{simd_separate_stride simd_noseparate_stride} - --linkcoarray	C++ User's Guide "2.2 Compiler Options"
		Execution performance improvement	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
			The following functions are improved. - Automatic Parallelization - SIMD - General Optimization	None
Optimization control specifier	The following optimization control specifiers are added. - {ftld nofltd} - simd[aligned unaligned] - {simd_redundant_vl simd_noredundant_vl} - uxsimd[aligned unaligned]	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL)"		
V2.0L10	02	Compiler option	The following compiler options are added. - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation}	C++ User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
		Execution performance improvement	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - SIMD - General Optimization - Instruction Scheduling (Software Pipelining) 	None
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - <code>__sync_synchronize</code> - <code>__sync_lock_test_and_set</code> - <code>__sync_lock_release</code> 	C++ User's Guide "D.1.2 Built-in Functions"
		Optimization information improvement	SIMD optimization information is improved. ("VL: <i>length</i> " is adding displayed.)	C++ User's Guide "5.1.2.2.7 Details Optimization Information"

3.1.3.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	Suffix of Input Files	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C++ User's Guide "2.1.2 Input Files for the Compile Command"
		Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - <code>-KCORE_AVX2</code> - <code>-K{fp_contract nofp_contract}</code> - <code>-Nstl=libc++</code> - <code>-stdlib=stl_kind</code> 	C++ User's Guide "2.2 Compiler Options"
		Environment variable	The following environment variables are added. <ul style="list-style-type: none"> - <code>C_PATH</code> - <code>CPLUS_INCLUDE_PATH</code> - <code>LIBRARY_PATH</code> 	C++ User's Guide "2.3 Environment Variable for Compile Command"
		C++11 specifications	The C++11 specifications are supported.	C++ User's Guide "6.3 C++11 Specifications"
		Built-in Functions	The following Built-in Functions are added. <ul style="list-style-type: none"> - Built-in functions with names that start with <code>'__atomic'</code> - <code>__builtin_index</code> - <code>__builtin_isinf</code> - <code>__builtin_isinff</code> - <code>__builtin_isinfl</code> 	C++ User's Guide "D.1.2 Built-in Functions"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex - Built-in functions with names that start with '_sync' 	
		Execution performance improvement	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - General optimization - SIMD - Automatic parallelization - Link time optimization 	None
		Compilation performance improvement	The size of the object program when -Nline option is specified is reduced.	None
		Optimization message	Optimization messages of link time optimization are improved.	C/C++ Compiler Optimization Messages
V2.0L20	14	C++11 specifications(part)	The C++11 specifications (part) are supported further.	C++ User's Guide "6.3 C++11 Specifications"
		GNU C++ Extensions	Assembler instructions with C expression operands are supported when the -Xa option is effective.	C++ User's Guide "D.1 GNU C++ Extensions"
		GNU C++ Compatible Options	<p>The following GNU C++ compatible options are supported.</p> <ul style="list-style-type: none"> - -std={c++03 c++11 gnu++03 gnu++11} 	C++ User's Guide "D.2 GNU C++ Compatible Options"
		Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - -K{openmp_tls openmp_notls} - -K{lto no_lto} - --linkcoarray 	C++ User's Guide "2.2 Compiler Options"
		Execution performance improvement	<p>The execution performance of the collapse clause of the loop construct of OpenMP is improved.</p> <p>The following functions are improved.</p> <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
V2.0L10	13	Compiler option	<p>The following compiler options are added.</p> <ul style="list-style-type: none"> - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation} - -K{simd_extension simd_noextension} 	C++ User's Guide "2.2 Compiler Options"

VL	Generation Number	Function	Description	Reference
			- <code>-K{simd_maskload simd_nomaskload}</code>	
		Optimization control specifier	The following optimization control specifiers are added. - <code>{simd_extension simd_noextension}</code> - <code>{simd_maskload simd_nomaskload}</code>	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance improvement	The following functions are improved. - Automatic Parallelization - SIMD - General Optimization	None
		Optimization information improvement	SIMD optimization information is improved. ("VL: <i>length</i> " is adding displayed.)	C++ User's Guide "5.1.2.2.6 Details Optimization Information"
	12	Compiler option	The following compiler options are added. - <code>-K{loop_versioning loop_noversioning}</code> - <code>-K{parallel_fp_precision parallel_nofp_precision}</code> - <code>-K{remove_inlinefunction noremove_inlinefunction}</code> - <code>-g0</code>	C++ User's Guide "2.2 Compiler Options"
		Optimization control specifier	The following optimization control specifiers are added. - <code>clone</code> - <code>{loop_versioning loop_noversioning}</code>	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL)"
		Execution performance improvement	The following functions are improved. - Automatic Parallelization - General Optimization	None
			The execution performance of <code>schedule(static)</code> of OpenMP loop construct is improved.	None
		C++11 specifications (part)	The following compiler option is added. - <code>-std={c++03 c++11}</code>	C++ User's Guide "2.2.2.4 Options for Language Specifications"
V1.0L30	11	Control of the initial value zero in an undefined variable	Compiler option <code>-N{setvalue nosetvalue}</code> is added.	C++ User's Guide "2.2.2.7 -N Option"
		OpenMP API Version 3.1 specifications	OpenMP API Version 3.1 specifications are supported.	C++ User's Guide "4.3 Parallelization by OpenMP Specifications"
		Compilation performance improvement	The optimization processing time of inline expansion is improved.	None

VL	Generation Number	Function	Description	Reference
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (unrolling, etc.) 	None
V1.0L20	10	Compilation performance improvement	The optimization processing time of inline expansion is improved.	None
		Linking option	The error message is output when specifying parallelization options are omitted at linking.	C++ User's Guide "G.4 Migrating to V1.0L20(Generation Number:10)"
	09	Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD - General optimization (inline expansion, full unrolling, etc.) - Instruction scheduling 	None
		Optimization messages and optimization information improvement	The following optimization messages and optimization information are improved. <ul style="list-style-type: none"> - Automatic parallelization - SIMD 	C/C++ Compiler Optimization Messages
		Compiler option	The following compiler options are added. <ul style="list-style-type: none"> - <code>-Nlst[={p t}]</code> - <code>-Nlst_out=file</code> - <code>-Nstl=500fast</code> 	C++ User's Guide "2.2.2.7 -N Option"
			The message for unrecognized compiler option is output.	C++ User's Guide "G.5 Migrating to V1.0L20(Generation Number:09)"
		Performance improvement of Runtime information output function	The Runtime Information Output Function is improved to decrease the overhead when the source program size is large.	None
	08	GNU C++ compatible options	The following options became available regardless of the <code>-Xg</code> option. <ul style="list-style-type: none"> - <code>-M</code>, <code>-MD</code>, <code>-MF</code>, <code>-MM</code>, <code>-MMD</code>, <code>-MP</code>, <code>-MT</code> The following GNU C++ compatible option is supported. <ul style="list-style-type: none"> - <code>--version</code> 	C++ User's Guide "2.2 Compiler Options" and "D.2 GNU C++ Compatible Options"
		Standard template library	Upgrades the standard template library.	C++ User's Guide "2.2.2.7 -N Option" -

VL	Generation Number	Function	Description	Reference
				Nstl={ 500 521 } option
		Addition of optimization control specifiers	The following optimization control specifiers are added. <ul style="list-style-type: none"> - temp_private - first_private - last_private - parallel_cyclic - {loop_part_parallel loop_nopart_parallel} - {loop_part_simd loop_nopart_simd} - unswitching 	C++ User's Guide "3.4.1 Using the Optimization Control Line (OCL) ", and "4.2.3.2 Optimization Control Line"
		Execution performance improvement	The following execution performances are improved. <ul style="list-style-type: none"> - Automatic Parallelization (-Kloop_part_parallel, etc.) - SIMD (-Kloop_part_simd, etc.) - General Optimization (Loop Unswitching enhance, etc.) - Promotion of optimization for large loop 	C++ User's Guide "2.2 Compiler Options"
		Compilation performance improvement	When compiling, used memory is reduced.	None
		Messages about optimization status	Output of the obstruction cause messages of optimization is improved.	None

3.1.4 XPFortran Translator

3.1.4.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L20	05	Addition of environment variable	The following environment variable is supported. <ul style="list-style-type: none"> - TMPDIR <p>The temporary directory that the xpfirtpx/xpfir command uses can be changed. When environment variable TMPDIR is not set, it is used /tmp.</p>	XPFortran User's Guide "2.1.2 How to Use the XPFortran System"

3.1.4.2 PC Cluster

VL	Generation Number	Function	Description	Reference
V1.0L20	08	Addition of environment variable	The following environment variable is supported. <ul style="list-style-type: none"> - TMPDIR 	XPFortran User's Guide "2.1.2 How to Use the XPFortran System"

VL	Generation Number	Function	Description	Reference
			The temporary directory that the xpfmt command uses can be changed. When environment variable TMPDIR is not set, it is used /tmp.	

3.2 Mathematical Libraries

3.2.1 SSL II

3.2.1.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L30	09	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VSRLU LU decomposition of real sparse matrices - DM_VSRLUX A system of linear equations with LU decomposed real sparse matrices - DM_VSRS A system of linear equations with real sparse matrices (LU decomposition method) 	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"

3.2.1.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VSSSLU LU decomposition of a structurally symmetric real sparse matrix - DM_VSSSLUX A system of linear equations with LU-decomposed structurally symmetric real sparse matrices - DM_VSSSS A system of linear equations with structurally symmetric real sparse matrices (LU decomposition method) 	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"
V2.0L20	03	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VMVSCCC Multiplication of a complex sparse matrix and a complex vector (compressed column storage method) - DM_VRANU5 	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"

VL	Generation Number	Function	Description	Reference
			<p>Generation of uniform random numbers [0,1) (MRG8)</p> <ul style="list-style-type: none"> - DM_VSCLU <p>LU decomposition of an unsymmetric complex sparse matrix</p> <ul style="list-style-type: none"> - DM_VSCLUX <p>A system of linear equations with LU-decomposed unsymmetric complex sparse matrices</p> <ul style="list-style-type: none"> - DM_VSCS <p>A system of linear equations with unsymmetric complex sparse matrices (LU decomposition method)</p>	
V2.0L10	02	Performance improvement of FFT	FFT routine is tuned for FX100.	None

3.2.1.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VSSSLU <p>LU decomposition of a structurally symmetric real sparse matrix</p> <ul style="list-style-type: none"> - DM_VSSSLUX <p>A system of linear equations with LU-decomposed structurally symmetric real sparse matrices</p> <ul style="list-style-type: none"> - DM_VSSSS <p>A system of linear equations with structurally symmetric real sparse matrices (LU decomposition method)</p>	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"
V2.0L20	14	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VMVSCCC <p>Multiplication of a complex sparse matrix and a complex vector (compressed column storage method)</p> <ul style="list-style-type: none"> - DM_VRANU5 <p>Generation of uniform random numbers [0,1) (MRG8)</p> <ul style="list-style-type: none"> - DM_VSCLU <p>LU decomposition of an unsymmetric complex sparse matrix</p> <ul style="list-style-type: none"> - DM_VSCLUX 	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"

VL	Generation Number	Function	Description	Reference
			<p>A system of linear equations with LU-decomposed unsymmetric complex sparse matrices</p> <ul style="list-style-type: none"> - DM_VSCS <p>A system of linear equations with unsymmetric complex sparse matrices (LU decomposition method)</p>	
V1.0L30	11	New routines	<p>New routine of SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - DM_VSRLU <p>LU decomposition of real sparse matrices</p> <ul style="list-style-type: none"> - DM_VSRLUX <p>A system of linear equations with LU decomposed real sparse matrices</p> <ul style="list-style-type: none"> - DM_VSRS <p>A system of linear equations with real sparse matrices (LU decomposition method)</p>	FUJITSU SSL II Thread-Parallel Capabilities User's Guide "Part II Usage of Subroutines"

3.2.2 C-SSL II

3.2.2.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L20	05	New routines	<p>New routine of C-SSL II</p> <ul style="list-style-type: none"> - c_dvcft3 <p>One-dimensional discrete complex Fourier transforms (Radix 2, for data sequence with a constant stride)</p>	FUJITSU C-SSL II User's Guide "Description of the C-SSL II Routines"

3.2.2.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	New routines	<p>New routine of C-SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - c_dm_vmvscce <p>Multiplication of a complex sparse matrix and a complex vector (compressed column storage method)</p> <ul style="list-style-type: none"> - c_dm_vranu5 <p>Generation of uniform random numbers [0,1) (MRG8).</p> <ul style="list-style-type: none"> - c_dm_vsclu <p>LU decomposition of an unsymmetric complex sparse matrix</p>	FUJITSU C-SSL II Thread-Parallel Capabilities User's Guide "Description of the C-SSL II Routines"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - c_dm_vsclux A system of linear equations with LU-decomposed unsymmetric complex sparse matrices - c_dm_vscs A system of linear equations with unsymmetric complex sparse matrices (LU decomposition method) 	
V2.0L20	03	New routines	<p>New routine of C-SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - c_dm_vjdhecr Eigenvalues and eigenvectors of an Hermitian sparse matrix (Jacobi-Davidson method, compressed row storage method) - c_dm_vjdnhcr Eigenvalues and eigenvectors of a complex sparse matrix (Jacobi-Davidson method, compressed row storage method) - c_dm_vlcspaxcr1 System of linear equations with non-Hermitian symmetric complex sparse matrices (Conjugate A-Orthogonal Conjugate Residual method with preconditioning by incomplete LDL^T decomposition, symmetric compressed row storage method) - c_dm_vlspaxcr2 System of linear equations with unsymmetric real sparse matrices (Induced Dimension Reduction method with preconditioning by sparse approximate inverse, compressed row storage method) - c_dm_vradau5 System of stiff ordinary differential equations or differential-algebraic equations (Implicit Runge-Kutta method) - c_dm_vsrlu LU decomposition of an unsymmetric real sparse matrix - c_dm_vsrlux A system of linear equations with LU-decomposed unsymmetric real sparse matrices - c_dm_vsrsl A system of linear equations with unsymmetric real sparse matrices (LU decomposition method) 	<p>FUJITSU C-SSL II Thread-Parallel Capabilities User's Guide "Description of the C-SSL II Routines"</p>

3.2.2.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	New routines	<p>New routine of C-SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - c_dm_vmvsccc Multiplication of a complex sparse matrix and a complex vector (compressed column storage method) - c_dm_vranu5 Generation of uniform random numbers [0,1) (MRG8). - c_dm_vsclu LU decomposition of an unsymmetric complex sparse matrix - c_dm_vsclux A system of linear equations with LU-decomposed unsymmetric complex sparse matrices - c_dm_vscs A system of linear equations with unsymmetric complex sparse matrices (LU decomposition method) 	<p>FUJITSU C-SSL II Thread-Parallel Capabilities User's Guide "Description of the C-SSL II Routines"</p>
V2.0L20	14	New routines	<p>New routine of C-SSL II Thread-Parallel Capabilities</p> <ul style="list-style-type: none"> - c_dm_vjdhecr Eigenvalues and eigenvectors of an Hermitian sparse matrix (Jacobi-Davidson method, compressed row storage method) - c_dm_vjdnhcr Eigenvalues and eigenvectors of a complex sparse matrix (Jacobi-Davidson method, compressed row storage method) - c_dm_vlcspsxcr1 System of linear equations with non-Hermitian symmetric complex sparse matrices (Conjugate A-Orthogonal Conjugate Residual method with preconditioning by incomplete LDL^T decomposition, symmetric compressed row storage method) - c_dm_vlspaxcr2 System of linear equations with unsymmetric real sparse matrices (Induced Dimension Reduction method with preconditioning by sparse approximate inverse, compressed row storage method) 	<p>FUJITSU C-SSL II Thread-Parallel Capabilities User's Guide "Description of the C-SSL II Routines"</p>

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - c_dm_vradau5 System of stiff ordinary differential equations or differential-algebraic equations (Implicit Runge-Kutta method) - c_dm_vsrlu LU decomposition of an unsymmetric real sparse matrix - c_dm_vsrlux A system of linear equations with LU-decomposed unsymmetric real sparse matrices - c_dm_vsrs A system of linear equations with unsymmetric real sparse matrices (LU decomposition method) 	
V1.0L20	08	New routines	<p>New routine of C-SSL II</p> <ul style="list-style-type: none"> - c_dvcft3 One-dimensional discrete complex Fourier transforms (Radix 2, for data sequence with a constant stride) 	FUJITSU C-SSL II User's Guide "Description of the C-SSL II Routines"

3.2.3 SSL II/MPI

3.2.3.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L30	09	New routines	<p>New routine of SSL II/MPI</p> <ul style="list-style-type: none"> - DS_V3DRCF3 Three-dimensional discrete real Fourier transforms (mixed radix of 2, 3, 5 and 7, volumetric decomposition) 	FUJITSU SSL II/MPI User's Guide "Part II Usage of Subroutines"

3.2.3.2 PC Cluster

VL	Generation Number	Function	Description	Reference
V1.0L30	11	New routines	<p>New routine of SSL II/MPI</p> <ul style="list-style-type: none"> - DS_V3DRCF3 Three-dimensional discrete real Fourier transforms (mixed radix of 2, 3, 5 and 7, volumetric decomposition) 	FUJITSU SSL II/MPI User's Guide "Part II Usage of Subroutines"

3.2.4 BLAS

3.2.4.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L10	11	Shared library	Shared library of BLAS is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"

3.2.4.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L30	04	Performance improvement	The performance of DGEMM is improved when following conditions are met. 1. TRANSA='T'. And, 2. The number of threads is 16 or 32.	None
V2.0L20	03	CBLAS	CBLAS, which is C interface to the BLAS, is supported.	BLAS LAPACK ScaLAPACK User's Guide "A.1 BLAS"
V2.0L10	02	Shared library	Shared library of BLAS is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"

3.2.4.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L20	14	CBLAS	CBLAS, which is C interface to the BLAS, is supported.	BLAS LAPACK ScaLAPACK User's Guide "A.1 BLAS"

3.2.5 LAPACK

3.2.5.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L10	11	Shared library	Shared library of LAPACK is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"

VL	Generation Number	Function	Description	Reference
	10	LAPACK version 3.5.0	New source code which is improved in version 3.5.0 of Netlib LAPACK is used for DBDSDC and SBDSDC.	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"
V1.0L20	05	LAPACK version 3.4.1	LAPACK version 3.4.1 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Computing the complete CS decomposition - Level-3 BLAS symmetric indefinite solve and symmetric indefinite inversion - Improved QR factorization - LAPACK C interface 	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"

3.2.5.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L20	03	LAPACK version 3.5.0	LAPACK version 3.5.0 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Symmetric/Hermitian linear equation routines with rook pivoting algorithm 	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"
		PLASMA	PLASMA, which is parallelized LAPACK with pthread, is supported.	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"
V2.0L10	02	Shared library	Shared library of LAPACK is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"

3.2.5.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L20	14	LAPACK version 3.5.0	LAPACK version 3.5.0 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Symmetric/Hermitian linear equation routines with rook pivoting algorithm 	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"
		PLASMA	PLASMA, which is parallelized LAPACK with pthread, is supported.	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"
V1.0L20	08	LAPACK version 3.4.1	LAPACK version 3.4.1 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Computing the complete CS decomposition 	BLAS LAPACK ScaLAPACK User's Guide "A.2 LAPACK"

VL	Generation Number	Function	Description	Reference
			<ul style="list-style-type: none"> - Level-3 BLAS symmetric indefinite solve and symmetric indefinite inversion - Improved QR factorization - LAPACK C interface 	

3.2.6 ScaLAPACK

3.2.6.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L10	11	Shared library	Shared library of ScaLAPACK is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"
V1.0L30	09	ScaLAPACK version 2.0.2	ScaLAPACK version 2.0.2 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Nonsymmetric Eigenvalue Problem - MRRR (Multiple Relatively Robust Representations) algorithm 	BLAS LAPACK ScaLAPACK User's Guide "A.3 ScaLAPACK"

3.2.6.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L10	02	Shared library	Shared library of ScaLAPACK is supported.	Programmer's Guide for Usage of Mathematical Libraries "2.2.7 Using shared libraries"

3.2.6.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V1.0L30	11	ScaLAPACK version 2.0.2	ScaLAPACK version 2.0.2 of Netlib is supported. The following new routines are added. <ul style="list-style-type: none"> - Nonsymmetric Eigenvalue Problem MRRR (Multiple Relatively Robust Representations) algorithm 	BLAS LAPACK ScaLAPACK User's Guide "A.3 ScaLAPACK"

3.3 MPI Library

3.3.1 FX10 System

VL	Generation Number	Function	Description	Reference
V2.0L10	11	Support of the sequential program execution	Add --nompi option of mpiexec to specify a sequential program.	MPI User's Guide "4.1 Execution Command Formats"
		Support of the execution definition file specification	Add --app option of mpiexec to specify an execution definition file.	MPI User's Guide "4.1 Execution Command Formats"
		Performance improvement of MPI_Scatterv	MCA parameter coll_tuned_scatterv_use_linear_sync is added to use MPI_Scatterv whose performance was enhanced by improving algorithms.	MPI User's Guide "4.2 MCA Parameters"
		Support of the non-contiguous mode	Add support for a non-contiguous mode.	MPI User's Guide
	10	Support of MPI-3.0 Standard	Add support for a part of the MPI-3.0 Standard.	MPI User's Guide
		Performance improvement of Persistent requests	To improve the performance of persistent communication, the extended interface is added.	MPI User's Guide "5.4 Extended Persistent Communication Requests Interface"
		Improvement of node utilization rates	Add support for a mesh mode of a node-exclusive job and a node-sharing job.	MPI User's Guide
V1.0L30	09	Support of MPI-2.2 Standard	Add support for the MPI-2.2 Standard.	MPI User's Guide
		Performance improvement of MPI_Allreduce and MPI_Bcast	The performance of MPI_Allreduce and MPI_Bcast is enhanced when the shape of job is one-dimensional or two-dimensional.	None
		Support of hybrid parallel job	Add support for a new format of VCOORD_FILE corresponding to a parallel hybrid job.	MPI User's Guide "4.1 Execution Command Formats" and "4.5 VCOORD_FILE file format"
		Improvement of Tofu Barrier Communication	MCA parameter coll_tbi_use_on_max_min is added that uses Tofu barrier communication for floating point datatypes MPI_MAX and MPI_MIN.	MPI User's Guide "4.2 MCA Parameters"
		Outputs stack trace information	MCA parameter orte_abort_print_stack is added that outputs stack trace information when the MPI program ends abnormally.	MPI User's Guide "4.2 MCA Parameters"
V1.0L20	08	Performance improvement of MPI_Allgatherv and MPI_Alltoallv on multiple processes in a node	The performance of MPI_Allgatherv and MPI_Alltoallv is enhanced by improving algorithms.	None
	07	Performance improvement of MPI_Allgather and MPI_Alltoall on multiple processes in a node	The performance of MPI_Allgather and MPI_Alltoall is enhanced by improving algorithms.	None

VL	Generation Number	Function	Description	Reference
	06	Output improvement of MPI Statistical Information Section Specifying	The MPI library is improved to accumulate the execution time when the FJMPI_COLLECTION_START/FJMPI_COLLECTION_STOP function is looped two or more times.	MPI User's Guide "5.3 MPI Statistical Information Section Specifying Interface"
		Performance improvement in making communicator	MCA parameter coll_tbi_use_on_comm_dup is added that the execution time becomes faster when the MPI program repeats frequently duplication of communicators or creation of windows.	MPI User's Guide "4.2 MCA Parameters"
	05	Tofu barrier communication increases speeds within node.	When multiple processes are allocated within one node, faster communication speeds using Tofu barrier communication can be achieved by algorithm tuned for communication within node.	None
		Improve performance of MPI_Bcast and MPI_Allreduce when use multi-procs in nodes.	Improve intra-node communication of MPI_Bcast and MPI_Allreduce.	None
		Rank query interface for the six dimensional coordinate	Add the extended functions for the six dimensional coordinate.	MPI User's Guide "5.1 Rank Query Interface"
		Addition of algorithm of Alltoall	The algorithm of Alltoall optimized for six dimension hexahedron is added.	None
		Hasty Rendezvous communication supports put/get operations.	The MCA parameter below is added, and Hasty Rendezvous communication encourages overlap of calculations and communication. - -mca pml_ob1_use_hasty_rendezvous 2 Hasty Rendezvous communication can be used put/get operations.	MPI User's Guide "4.2 MCA Parameters"
		Improve performance of MPI_Gather and MPI_Allgather.	Add new algorithm of MPI_Gather.	None

3.3.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L20	03	Support of MPI-3.0 Standard	Add support for the MPI-3.0 Standard.	MPI User's Guide
V2.0L10	02	Support of the sequential program execution	Add --nomp option of mpiexec to specify a sequential program.	MPI User's Guide "4.1 Execution Command Formats"
		Support of the execution definition file specification	Add --app option of mpiexec to specify an execution definition file.	MPI User's Guide "4.1 Execution Command Formats"
		Performance improvement of MPI_Scatterv	MCA parameter coll_tuned_scatterv_use_linear_sync is added to use MPI_Scatterv whose performance was enhanced by improving algorithms.	MPI User's Guide "4.2 MCA Parameters"

VL	Generation Number	Function	Description	Reference
		Support of the non-contiguous mode	Add support for a non-contiguous mode.	MPI User's Guide

3.3.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L20	14	Support of MPI-3.0 Standard	Add support for the MPI-3.0 Standard.	MPI User's Guide
V2.0L10	13	Support of the sequential program execution	Add --nomp option of mpiexec to specify a sequential program.	MPI User's Guide "4.1 Execution Command Formats"
		Support of the execution definition file specification	Add --app option of mpiexec to specify an execution definition file.	MPI User's Guide "4.1 Execution Command Formats"
	12	Support of MPI-3.0 Standard	Add support for a part of the MPI-3.0 Standard.	MPI User's Guide
		For NUMA system	MCA Parameter plm_ple_memory_allocation_policy that sets the memory policy of MPI process is added.	MPI User's Guide "4.2 MCA Parameters" and "4.6 Settings in NUMA system"
V1.0L30	11	Support of MPI-2.2 Standard	Add support for the MPI-2.2 Standard.	MPI User's Guide
		Outputs stack trace information	MCA parameter orte_abort_print_stack is added that outputs stack trace information when the MPI program ends abnormally.	MPI User's Guide "4.2 MCA Parameters"
V1.0L20	09	Output improvement of MPI Statistical Information Section Specifying	The MPI library is improved to accumulate the execution time when the FJMPI_COLLECTION_START/FJMPI_COLLECTION_STOP function is looped two or more times.	MPI User's Guide "5.1 MPI Statistical Information Section Specifying Interface"

3.4 Programming Workbench

3.4.1 Debugger

3.4.1.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L30	09	Addition of Thread enable option	Thread enable option of Debugger start setting Window has been added. If user selects it, the thread information is displayed in the Debugger. Otherwise it is not displayed.	Debugger User's Guide "3.1 Debugger start setting Window"

3.4.2 Profiler

3.4.2.1 FX10 System

VL	Generation Number	Function	Description	Reference
V1.0L30	09	Measurement of detailed information on shared library in the Instant Profiler	Detailed information on the shared library can be measured is added.	Profiler User's Guide "Table 2.1 fipp command options"
		Event in the Advanced Profiler direct	The function to specify the measurement event number of the hardware monitor information is added.	Profiler User's Guide "Table 3.4 fapp command options" and "Table 3.22 Output form of measurement event number specification"
		Highly precise measurement of Hardware monitor information in the Advanced Profiler	Highly precise measurement function is added to the Hardware monitor information.	Profiler User's Guide "Table 3.4 fapp command"
		Measured Information in the Advanced Profiler (GUI format)	The following item of "Measured Information" is added. - Information is measured	Profiler User's Guide "Figure 2.6 Measured Information window" and "Table 2.5 Measured Information"
		I/O trace function Memory trace function	The I/O trace function and the memory trace function are added to the tracer.	Profiler User's Guide "4.3.4 I/O trace" and "4.3.5 Memory trace"
V1.0L20	05	The function of text format output in the profiler	In the advanced profiler, the function of text format output is added.	Profiler User's Guide "3.1 Overview of Advanced profiler", "3.2.6 fapppx command", "3.3 Advanced profiler information", and "3.4 Advanced profiler information (Text/CVS format) "

3.4.2.2 FX100 System

VL	Generation Number	Function	Description	Reference
V2.0L20	03	Instant Profiler	COARRAY feature is supported.	Profiler User's Guide "A.1 Instant Profiler"
		Advanced Profiler	COARRAY feature is supported.	Profiler User's Guide "A.2 Advanced Profiler"
			MPI 3.0 function is supported.	Profiler User's Guide "3.4.4 MPI information"

3.4.2.3 PC Cluster

VL	Generation Number	Function	Description	Reference
V2.0L30	15	Instant Profiler	COARRAY feature is supported.	Profiler User's Guide "A.1 Instant Profiler"
		Advanced Profiler	COARRAY feature is supported.	Profiler User's Guide "A.2 Advanced Profiler"
V2.0L20	14	Advanced Profiler	MPI 3.0 function is supported.	Profiler User's Guide "Table 3.10 Formulas of message length"
V1.0L30	11	Measurement of detailed information on shared library in the Instant Profiler	Detailed information on the shared library can be measured is added.	Profiler User's Guide "Table 2.1 fipp command options (Information gathering)"
		Addition of measurement event of hardware monitor information in the Advanced Profiler	The measurement function of information including the AVX instruction is added to the measurement event of hardware monitor information.	Profiler User's Guide "Table 3.3 fapp command options (Information gathering)" and "Table 3.11 Items of AVX"
		Measured Information in the Advanced Profiler (GUI format)	The following item of "Measured Information" is added. - Information is measured	Profiler User's Guide "Figure 2.6 Measured Information window" and "Table 2.5 Measured Information"
		I/O trace function Memory trace function	The I/O trace function and the memory trace function are added to the tracer.	Profiler User's Guide "4.3.4 I/O trace" and "4.3.5 Memory trace"

3.4.3 User GUI

3.4.3.1 PC Cluster

VL	Generation Number	Function	Description	Reference
V1.0L20	09	User GUI for Mac	The GUI of FSDT for Macintosh is offered.	None

Chapter 4 Compatibility Information

This chapter explains incompatible items changed from the previous version on this system and which manual each of them is detailed in.

For information on a way of dealing with the incompatible item, read relevant manuals as needed.

Refer to "[1.3.2 Component Packages](#)" for details.

4.1 Migrating to V2.0L30 (FX10 System Generation Number:13, FX100 System Generation Number:04, PC Cluster Generation Number:15)

4.1.1 Fortran Compiler

4.1.1.1 FX10 System

Compatible Item	Relevant Manual
The diagnostic message jwe0323i-w is output at execution time for incorrect undefined variable reference when the compiler option -Hx is effective	Fortran User's Guide
Error check in Fortran 2003 and OpenMP standard at compilation time	Fortran User's Guide
Value of throughput is added to Fortran Input-Output Information	Runtime Information Output Function User's Guide

4.1.1.2 FX100 System

Compatible Item	Relevant Manual
The diagnostic message jwe0323i-w is output at execution time for incorrect undefined variable reference when the compiler option -Hx is effective	Fortran User's Guide
Error check in Fortran 2003 and OpenMP standard at compilation time	Fortran User's Guide
Value of throughput is added to Fortran Input-Output Information	Runtime Information Output Function User's Guide

4.1.1.3 PC Cluster

Compatible Item	Relevant Manual
The diagnostic message jwe0323i-w is output at execution time for incorrect undefined variable reference when the compiler option -Hx is effective	Fortran User's Guide
Error check in Fortran 2003 and OpenMP standard at compilation time	Fortran User's Guide
Value of throughput is added to Fortran Input-Output Information	Runtime Information Output Function User's Guide

4.1.2 C++ Compiler

4.1.2.1 FX10 System

Compatible Item	Relevant Manual
The language specification level is changed when the compiler options -Xg -std={c++98 c++03 c++11} are valid	C++ User's Guide

Compatible Item	Relevant Manual
The GNU C++ compatible version is changed when the compiler options -Xg -std={c++11 gnu++11} are valid	C++ User's Guide
The C++ standard template library(STL) with the C++11 mode is changed	C++ User's Guide

4.1.2.2 FX100 System

Compatible Item	Relevant Manual
The language specification level is changed when the compiler options -Xg -std={c++98 c++03 c++11} are valid	C++ User's Guide
The GNU C++ compatible version is changed when the compiler options -Xg -std={c++11 gnu++11} are valid	C++ User's Guide
The C++ standard template library(STL) with the C++11 mode is changed	C++ User's Guide

4.1.2.3 PC Cluster

Compatible Item	Relevant Manual
The language specification level is changed when the compiler options -Xg -std={c++98 c++03 c++11} are valid	C++ User's Guide
The GNU C++ compatible version is changed when the compiler options -Xg -std={c++11 gnu++11} are valid	C++ User's Guide
The C++ standard template library(STL) with the C++11 mode is changed	C++ User's Guide

4.1.3 MPI Library

4.1.3.1 FX10 System

Compatible Item	Relevant Manual
On FEFS, the error class returned by the MPI_File_open function when a non-existent file and MPI_MODE_RDONLY are specified for the function respectively is changed	MPI User's Guide

4.1.3.2 FX100 System

Compatible Item	Relevant Manual
On FEFS, the error class returned by the MPI_File_open function when a non-existent file and MPI_MODE_RDONLY are specified for the function respectively is changed	MPI User's Guide

4.1.3.3 PC Cluster

Compatible Item	Relevant Manual
On FEFS, the error class returned by the MPI_File_open function when a non-existent file and MPI_MODE_RDONLY are specified for the function respectively is changed	MPI User's Guide

4.2 Migrating to V2.0L20 (FX10 System Generation Number:12, FX100 System Generation Number:03, PC Cluster Generation Number:14)

4.2.1 Fortran Compiler

4.2.1.1 FX10 System

Compatible Item	Relevant Manual
Error check in OpenMP standard at compilation time	Fortran User's Guide
Error check in Fortran 2003 standard at compilation time	Fortran User's Guide

4.2.1.2 FX100 System

Compatible Item	Relevant Manual
Error check in OpenMP standard at compilation time	Fortran User's Guide
Error check in Fortran 2003 standard at compilation time	Fortran User's Guide
Enhancement intrinsic procedures and intrinsic module	Fortran User's Guide
Change of the compilation message when the compiler option -K{FLTLD NOFLTLD} is specified	Fortran User's Guide

4.2.1.3 PC Cluster

Compatible Item	Relevant Manual
Error check in OpenMP standard at compilation time	Fortran User's Guide
Error check in Fortran 2003 standard at compilation time	Fortran User's Guide
Enhancement intrinsic procedures and intrinsic module	Fortran User's Guide

4.2.2 C Compiler

4.2.2.1 FX100 System

Compatible Item	Relevant Manual
Change of the compilation message when the compiler option -K{FLTLD NOFLTLD} is specified	C User's Guide

4.2.3 C++ Compiler

4.2.3.1 FX100 System

Compatible Item	Relevant Manual
Change of the compilation message when the compiler option -K{FLTLD NOFLTLD} is specified	C++ User's Guide

4.2.4 MPI Library

4.2.4.1 FX100 System

Compatible Item	Relevant Manual
Behavior of mpiexec(1) when a same MCA parameter is specified more than once is changed	MPI User's Guide
Value of MPI_ERR_LASTCODE is changed	MPI User's Guide
The behavior when --version option or -V option is specified for the mpiexec command is changed	MPI User's Guide
The behavior when an invalid parameter is specified for -showme option of the compilation/linkage command is changed	MPI User's Guide
The output destination of version information and a help message is changed	MPI User's Guide

4.2.4.2 PC Cluster

Compatible Item	Relevant Manual
Behavior of mpiexec(1) when a same MCA parameter is specified more than once is changed	MPI User's Guide
Value of MPI_ERR_LASTCODE is changed	MPI User's Guide
The behavior when --version option or -V option is specified for the mpiexec command is changed	MPI User's Guide
The behavior when an invalid parameter is specified for -showme option of the compilation/linkage command is changed	MPI User's Guide
The output destination of version information and a help message is changed	MPI User's Guide

4.3 Migrating to V2.0L10 (FX10 System Generation Number:11, FX100 System Generation Number:02, PC Cluster Generation Number:13)

4.3.1 Fortran Compiler

4.3.1.1 FX10 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	Fortran User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	Fortran User's Guide

4.3.1.2 FX100 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	Fortran User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	Fortran User's Guide

4.3.1.3 PC Cluster

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	Fortran User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	Fortran User's Guide

4.3.2 C Compiler

4.3.2.1 FX10 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C User's Guide

4.3.2.2 FX100 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C User's Guide

4.3.2.3 PC Cluster

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C User's Guide

4.3.3 C++ Compiler

4.3.3.1 FX10 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C++ User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C++ User's Guide

4.3.3.2 FX100 System

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C++ User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C++ User's Guide

4.3.3.3 PC Cluster

Compatible Item	Relevant Manual
The function for canceling the program compilation that is forecasted to take a long time.	C++ User's Guide
The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}	C++ User's Guide

4.4 Migrating to V2.0L10 (FX10 System Generation Number:10, PC Cluster Generation Number:12)

4.4.1 Fortran Compiler

4.4.1.1 FX10 System

Compatible Item	Relevant Manual
The -Komitfp option is induced by -Kfast, and maintenance of the trace back information	Fortran User's Guide
Change of value of macro along with the compiler version up	Fortran User's Guide
Change of the name of temporary object file and the directory to store	Fortran User's Guide
Extend specifications about comment in namelist input file	None

4.4.1.2 PC Cluster

Compatible Item	Relevant Manual
Change of value of macro along with the compiler version up	Fortran User's Guide
Change of the name of temporary object file and the directory to store	Fortran User's Guide
Change the -KOMITFP option to -Komitfp	Fortran User's Guide
Extend specifications about comment in namelist input file	None

4.4.2 C Compiler

4.4.2.1 FX10 System

Compatible Item	Relevant Manual
The -Komitfp option is induced by -Kfast, and maintenance of the trace back information	C User's Guide
Change of value of macro along with the compiler version up	C User's Guide
Change of the name of temporary object file and the directory to store	C User's Guide

4.4.2.2 PC Cluster

Compatible Item	Relevant Manual
Change of value of macro along with the compiler version up	C User's Guide
Change of the name of temporary object file and the directory to store	C User's Guide
Change the -KOMITFP option to -Komitfp	C User's Guide

4.4.3 C++ Compiler

4.4.3.1 FX10 System

Compatible Item	Relevant Manual
The -Komitfp option is induced by -Kfast, and maintenance of the trace back information	C++ User's Guide
Change of value of macro along with the compiler version up	C++ User's Guide
Change of the name of temporary object file and the directory to store	C++ User's Guide

4.4.3.2 PC Cluster

Compatible Item	Relevant Manual
Change of value of macro along with the compiler version up	C++ User's Guide
Change of the name of temporary object file and the directory to store	C++ User's Guide
Change the -KOMITFP option to -Komitfp	C++ User's Guide

4.4.4 MPI Library

4.4.4.1 FX10 System

Compatible Item	Relevant Manual
Change of Extended RDMA Interface Error Message	MPI User's Guide
The default value of MCA parameter orte_abort_print_stack is changed from 0 to 1	MPI User's Guide
There are some mpiexec return values passed by the parallel execution functions of the Job Operation Software. By the version up from V1.0L30 to V2.0L10, some return values passed by the Job Operation Software are changed. Therefore the corresponding mpiexec return values are also changed. Refer to "Job Operation Software Release Notes" for details.	Job Operation Software Release Notes

4.4.4.2 PC Cluster

Compatible Item	Relevant Manual
The default value of MCA parameter orte_abort_print_stack is changed from 0 to 1	MPI User's Guide
There are some mpiexec return values passed by the parallel execution functions of the Job Operation Software. By the version up from V1.0L30 to V2.0L10, some return values passed by the Job Operation Software are changed. Therefore the corresponding mpiexec return values are also changed. Refer to "Job Operation Software Release Notes" for details.	Job Operation Software Release Notes

4.5 Migrating to V1.0L30 (FX10 System Generation Number:09, PC Cluster Generation Number:11)

4.5.1 Fortran Compiler

4.5.1.1 FX10 System

Compatible Item	Relevant Manual
Change of values of macro and named constant along with the support of OpenMP API version 3.1 specifications	Fortran User's Guide
Enhancement Fortran 2008 intrinsic procedures and intrinsic modules Fortran 2008	Fortran User's Guide
Bound remapping list and pointer target of two rank or more in pointer assignment statement	Fortran User's Guide
Non pure final subroutine referred from pure procedure	Fortran User's Guide
Array argument that is not CONTIGUOUS in intrinsic module function C_LOC	Fortran User's Guide
Runtime message (jwe1007i-s) change	Fortran User's Guide

4.5.1.2 PC Cluster

Compatible Item	Relevant Manual
Change of values of macro and named constant along with the support of OpenMP API version 3.1 specifications	Fortran User's Guide
Enhancement Fortran 2008 intrinsic procedures and intrinsic modules Fortran 2008	Fortran User's Guide
Bound remapping list and pointer target of two rank or more in pointer assignment statement	Fortran User's Guide
Non pure final subroutine referred from pure procedure	Fortran User's Guide
Array argument that is not CONTIGUOUS in intrinsic module function C_LOC	Fortran User's Guide
Runtime message (jwe1007i-s) change	Fortran User's Guide

4.5.2 C Compiler

4.5.2.1 FX10 System

Compatible Item	Relevant Manual
Change of value of macro along with the support of OpenMP API version 3.1 specifications	C User's Guide

4.5.2.2 PC Cluster

Compatible Item	Relevant Manual
Change of value of macro along with the support of OpenMP API version 3.1 specifications	C User's Guide

4.5.3 C++ Compiler

4.5.3.1 FX10 System

Compatible Item	Relevant Manual
Change of value of macro along with the support of OpenMP API version 3.1 specifications	C++ User's Guide

4.5.3.2 PC Cluster

Compatible Item	Relevant Manual
Change of value of macro along with the support of OpenMP API version 3.1 specifications	C++ User's Guide

4.5.4 MPI Library

4.5.4.1 FX10 System

Compatible Item	Relevant Manual
Change of Communication Library Error Message	MPI User's Guide

4.5.4.2 PC Cluster

Compatible Item	Relevant Manual
Change of Communication Library Error Message	MPI User's Guide

4.5.5 Programming Workbench

4.5.5.1 FX10 System

Compatible Item	Relevant Manual
The -c option of the vtunifypx command and vtunify-mpi application of the tracer is abolished	Profiler User's Guide
VT_MAX_MPI_COMMS/VT_MAX_MPI_WINS of the environment variable for execution is abolished	Profiler User's Guide
The record in the trace data of the MPI_Address function is abolished	Profiler User's Guide

4.5.5.2 PC Cluster

Compatible Item	Relevant Manual
The -c option of the vtunify command and vtunify-mpi application of the tracer is abolished	Profiler User's Guide
VT_MAX_MPI_COMMS/VT_MAX_MPI_WINS of the environment variable for execution is abolished	Profiler User's Guide
The record in the trace data of the MPI_Address function is abolished	Profiler User's Guide

4.6 Migrating to V1.0L20 (FX10 System Generation Number:07, PC Cluster Generation Number:10)

4.6.1 Fortran Compiler

4.6.1.1 FX10 System

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	Fortran User's Guide

4.6.1.2 PC Cluster

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	Fortran User's Guide

4.6.2 C Compiler

4.6.2.1 FX10 System

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	C User's Guide

4.6.2.2 PC Cluster

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	C User's Guide

4.6.3 C++ Compiler

4.6.3.1 FX10 System

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	C++ User's Guide

4.6.3.2 PC Cluster

Compatible Item	Relevant Manual
Error message output when specifying parallelization options are omitted at linking	C++ User's Guide

4.7 Migrating to V1.0L20 (FX10 System Generation Number:06, PC Cluster Generation Number:09)

4.7.1 Fortran Compiler

4.7.1.1 FX10 System

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	Fortran User's Guide

4.7.1.2 PC Cluster

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	Fortran User's Guide

4.7.2 C Compiler

4.7.2.1 FX10 System

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	C User's Guide

4.7.2.2 PC Cluster

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	C User's Guide

4.7.3 C++ Compiler

4.7.3.1 FX10 System

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	C++ User's Guide

4.7.3.2 PC Cluster

Compatible Item	Relevant Manual
Message output for unrecognized compiler option	C++ User's Guide

4.8 Migrating to V1.0L20 (FX10 System Generation Number:05, PC Cluster Generation Number:08)

4.8.1 Fortran Compiler

4.8.1.1 FX10 System

Compatible Item	Relevant Manual
Runtime message (jwe0220i-e) change	Fortran User's Guide
Changes to the specification when SOURCE= specifier appears and an allocatable component is "allocated" in ALLOCATE statement	Fortran User's Guide
Support Fortran 2008 intrinsic procedures	Fortran User's Guide
Change when specifying the -O option before specifying the -g option	Fortran User's Guide

4.8.1.2 PC Cluster

Compatible Item	Relevant Manual
Runtime message (jwe0220i-e) change	Fortran User's Guide
Changes to the specification when SOURCE= specifier appears and an allocatable component is "allocated" in ALLOCATE statement	Fortran User's Guide
Support Fortran 2008 intrinsic procedures	Fortran User's Guide
Change when specifying the -O option before specifying the -g option	Fortran User's Guide

4.8.2 C Compiler

4.8.2.1 FX10 System

Compatible Item	Relevant Manual
Change in method of specifying the -M option to linker	C User's Guide
Change when specifying the -O option before specifying the -g option	C User's Guide

4.8.2.2 PC Cluster

Compatible Item	Relevant Manual
Change in method of specifying the -M option to linker	C User's Guide
Change when specifying the -O option before specifying the -g option	C User's Guide

4.8.3 C++ Compiler

4.8.3.1 FX10 System

Compatible Item	Relevant Manual
Change in method of specifying the -M option to linker	C++ User's Guide
Change when specifying the -O option before specifying the -g option	C++ User's Guide

4.8.3.2 PC Cluster

Compatible Item	Relevant Manual
Change in method of specifying the -M option to linker	C++ User's Guide
Change when specifying the -O option before specifying the -g option	C++ User's Guide

4.8.4 MPI Library

4.8.4.1 FX10 System

Compatible Item	Relevant Manual
Changes to retrieved values of the predefined attributes	MPI User's Guide

4.8.4.2 PC Cluster

Compatible Item	Relevant Manual
Changes to retrieved values of the predefined attributes	MPI User's Guide

Chapter 5 Migrating from former Language Package V3

This chapter provides notes on migrating from former Language Package V3 to this system.

Fortran, C, and C++ programs

At the source code level, Fortran, C, and C++ programs are basically forward-compatible, but there are incompatibilities at the object program and executable program level. Use this system to perform recompilation.

MPI programs

MPI programs are compatible at the source code level except for a small percentage of specifications that depend on the processing system implementation, such as info argument values. There are, however, incompatibilities at the object program and executable program level. Use this system to perform recompilation.

XPFortran programs

XPFortran programs are compatible at the source code level except for some language specifications that have changed; there are incompatibilities at the object program and executable program level. Use this system to perform recompilation.

Other

In addition, there have been some specification changes, such as command names, option names, and environment variable names, for component functions provided by this system.

Chapter 6 Notes on migration from FX10 system to FX100 system

This chapter provides notes on migrating from FX10 system (Generation Number:09 or later) to FX100 system.

If migrating from FX10 system (Generation Number:08 or earlier), refer to "[Chapter 4 Compatibility Information](#)" also.

For detail of "Generation Number", refer to "[1.3.2 Component Packages](#)".

6.1 Common guidelines

Change in source code

- Fortran/C/C++ language specification may need to correct the source program.
Refer to "Enhancement intrinsic procedures and intrinsic module" in the "Fortran User's Guide (PRIMEHPC FX100)" for details.
- Rewriting might become necessary so that the optimizing control line specified to draw out the feasibility capability on the source code and the FX10 system of the Fortran/C/C++ program made for the FX10 system may draw out the performance on the FX100 system.

Re-creating of executable program

- Compile again by the compiler corresponding to the hardware new function of the FX100 system (4SIMD etc.) to draw out the hardware ability of the FX100 system enough.

Difference of result because of accuracy error margin

- The result in the FX100 system and the result in the FX10 system might be different in the accuracy range of error according to the influence of optimization for the FX100 system.

6.2 Modification of each component

The notes on migration to FX100 system are as follows by component.

The titles in each table are presented below.

- FX10 System Generation Number
Means Generation Number of FX10 system at the migration origin.
- Modification
Presents a summary of changes.
- Related Manual
Means the name of manuals for detailed information about "Modification".

6.2.1 Compiler

6.2.1.1 Fortran Compiler

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	The following compiler options are added. <ul style="list-style-type: none">- -K{ nf nonf }- -K{ fed nofed }- -Ksimd=auto- -K{ FLTLD NOFLTLD }	Fortran User's Guide

FX10 System Generation Number	Modification	Related Manuals
	<ul style="list-style-type: none"> - -K{ HPC_ACE HPC_ACE2 } - -K{ lto nolto } - -K{ simd_separate_stride simd_noseparate_stride } - -N{ coarray nocoarray } - -K{simd_reduction_product simd_noreduction_product} 	
	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - { NF NONF } - { FLTLD NOFLTLD } - SIMD[(ALIGNED UNALIGNED)] - { SIMD_REDUNDANT_VL(<i>n</i>) SIMD_NOREDUNDANT_VL } - UXSIMD[(ALIGNED UNALIGNED)] 	Fortran User's Guide
	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - SIMD - Instruction Scheduling (Software Pipelining) 	None
	<p>Abolition of compilation options and the optimization control specifiers</p> <ul style="list-style-type: none"> - -Kcpu option - -K{prefetch_double_line_L2 prefetch_nodouble_line_L2} options and the optimization control specifiers {PREFETCH_DOUBLE_LINE_L2 PREFETCH_NODOUBLE_LINE_L2} - -K{simd_region_constant nosimd_region_constant} options 	Fortran User's Guide
	<p>The function for multi-operation function is added.</p>	Fortran User's Guide
	<p>SIMD optimization information is improved ("VL: <i>length</i>" is adding displayed).</p>	Fortran User's Guide
	<p>The category of hardware monitor information is changed.</p>	Runtime Information Output Function User's Guide
	<p>Change the default value of the -Ksimd[=<i>level</i>] option</p>	Fortran User's Guide
	<p>Specification change of the optimization control specifiers SIMD and UXSIMD</p>	Fortran User's Guide
	<p>Enhancement intrinsic procedures and intrinsic module</p>	Fortran User's Guide
	<p>Change of the compilation message when the compiler option -K{FLTLD NOFLTLD} is specified.</p>	Fortran User's Guide
	<p>The option which the compiler option -Kuxsimd needs is changed.</p>	Fortran User's Guide
	<p>The following specifications are added.</p> <ul style="list-style-type: none"> - Extend PROCEDURE statement in interface block - Parameterized derived types - BLOCK construct - Implied shape array - ERROR STOP statement - Extend STOP statement 	Fortran Language Reference

FX10 System Generation Number	Modification	Related Manuals
	- Double colon after PROCEDURE in interfaceblock	
	COARRAY function is supported.	Fortran User's Guide Additional Volume COARRAY Fortran Language Reference
	" [=kind]" is added to the -Kifunc option.	Fortran User's Guide
	The following environment variable is added. - LIBRARY_PATH	Fortran User's Guide
	Add the user-defined derived type input/output specification	Fortran User's Guide Fortran Language Reference
	The diagnostic message jwe0323i-w is output at execution time for incorrect undefined variable reference when the compiler option -Hx is effective	Fortran User's Guide
	Error check in Fortran 2003 and OpenMP standard at compilation time	Fortran User's Guide
	Value of throughput is added to Fortran Input-Output Information	Runtime Information Output Function User's Guide
11 or earlier	The following compiler options are added. - -K{ openmp_tls openmp_notls } - -K{ intentopt nointentopt }	Fortran User's Guide
	The following functions are improved. - Automatic Parallelization - General Optimization	None
	Error check in OpenMP standard at compilation time	Fortran User's Guide
	Error check in Fortran 2003 standard at compilation time	Fortran User's Guide
	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
10 or earlier	The following compiler options are added. - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation}	Fortran User's Guide
	The function for canceling the program compilation that is forecasted to take a long time	Fortran User's Guide
	The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}.	Fortran User's Guide
09	The following compiler options are added. - -K{loop_versioning loop_noversioning} - -K{parallel_fp_precision parallel_nofp_precision} - -K{omitfp noomitfp} - -g0	Fortran User's Guide
	Abolition of compilation options and the optimization control specifiers	Fortran User's Guide

FX10 System Generation Number	Modification	Related Manuals
	- -K{FLTLD NOFLTLD} options and the optimization control specifiers {FLTLD NOFLTLD}	
	The following optimization control specifiers are added. - CLONE - {LOOP_VERSIONING LOOP_NOVERSIONING}	Fortran User's Guide
	The execution performance of an array assignment statement is improved in the FORALL construct.	None
	The inline expansion function is improved when compiler option -Knoalias=s is effective.	None
	The execution performance of the threadprivate directive of OpenMP is improved.	None
	The -Komitfp option is induced by -Kfast, and maintenance of the trace back information.	Fortran User's Guide
	Change of value of macro along with the compiler version up	Fortran User's Guide
	Change of the name of temporary object file and the directory to store	Fortran User's Guide
	For Fortran 2008 specification (Data object has the implicit SAVE attribute in declaration part of the module)	Fortran Language Reference

6.2.1.2 C Compiler

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	The following compiler options are added. - -K{nf nonf} - -K{fed nofed} - -Ksimd=auto - -K{FLTLD NOFLTLD} - -K{HPC_ACE HPC_ACE2} - -K{lto nolto} - -K{simd_separate_stride simd_noseparate_stride} - -K{simd_reduction_product simd_noreduction_product}	C User's Guide
	The following optimization control specifiers are added. - {nf nonf} - {fltld nofltld} - simd [aligned unaligned] - {simd_redundant_vl simd_noredundant_vl} - uxsimd [aligned unaligned]	C User's Guide
	The following functions are improved. - SIMD - Instruction Scheduling (Software Pipelining)	None
	Abolition of compilation options and the optimization control specifiers	C User's Guide

FX10 System Generation Number	Modification	Related Manuals
	<ul style="list-style-type: none"> - <i>-Kcpu</i> option - <i>-K{prefetch_double_line_L2 prefetch_nodouble_line_L2}</i> options and the optimization control specifiers { <i>prefetch_double_line_L2 prefetch_nodouble_line_L2</i>} - <i>-K{simd_region_constant nosimd_region_constant}</i> options 	
	SIMD optimization information is improved (" <i>VL: length</i> " is adding displayed).	C User's Guide
	The category of hardware monitor information is changed.	Runtime Information Output Function User's Guide
	Change the default value of the <i>-Ksimd[=<i>level</i>]</i> option	C User's Guide
	Change of the compilation message when the compiler option <i>-K{FLTLD NOFLTLD}</i> is specified	C User's Guide
	The option which the compiler option <i>-Kuxsimd</i> needs is changed.	C User's Guide
	Specification change of the optimization control specifiers <i>simd</i> and <i>uxsimd</i>	C User's Guide
	SIMD built-in functions are supported.	C User's Guide
	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C User's Guide
	[= <i>{loop procedure}</i>] is added to the <i>-Kifunc</i> option.	C User's Guide
	The following environment variables are added. <ul style="list-style-type: none"> - CPATH - C_INCLUDE_PATH - LIBRARY_PATH 	C User's Guide
	The following Built-in Functions are added. <ul style="list-style-type: none"> - Built-in functions with names that start with '<i>__atomic</i>' - <i>__builtin_index</i> - <i>__builtin_isinf</i> - <i>__builtin_isinff</i> - <i>__builtin_isinfl</i> - <i>__builtin_isnan</i> - <i>__builtin_isnanf</i> - <i>__builtin_isnanl</i> - <i>__builtin_memchr</i> - <i>__builtin_rindex</i> 	C User's Guide
11 or earlier	GNU C Extensions <ul style="list-style-type: none"> - Assembler instructions with C expression operands are supported when the <i>-Xa</i> option is effective 	C User's Guide
	The following compiler option is added. <ul style="list-style-type: none"> - <i>-K{openmp_tls openmp_notls}</i> 	C User's Guide
	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization 	None

FX10 System Generation Number	Modification	Related Manuals
	- General Optimization	
	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
10 or earlier	The following compiler options are added. - -K{openmp_assume_norecurrence openmp_noassume_norecurrence} - -N{cancel_overtime_compilation nocancel_overtime_compilation}	C User's Guide
	The function for canceling the program compilation that is forecasted to take a long time	C User's Guide
	The compiler option -K{ordered_omp_reduction noordered_omp_reduction} is changed to -K{openmp_ordered_reduction openmp_noordered_reduction}.	C User's Guide
09	The following compiler options are added. - -K{loop_versioning loop_noverioning} - -K{parallel_fp_precision parallel_nofp_precision} - -K{omitfp noomitfp} - -g0	C User's Guide
	Abolition of compilation options and the optimization control specifiers - -K{FLTLD NOFLTLD} options and the optimization control specifiers {fltld nofltld}	C User's Guide
	The following optimization control specifiers are added. - clone - {loop_versioning loop_noverioning }	C User's Guide
	The execution performance of schedule(static) of OpenMP loop construct is improved.	None
	The -Komitfp option is induced by -Kfast, and maintenance of the trace back information.	C User's Guide
	Change of value of macro along with the compiler version up	C User's Guide
	Change of the name of temporary object file and the directory to store	C User's Guide

6.2.1.3 C++ Compiler

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	The following compiler options are added. - -K{nf nonf} - -K{fed nofed} - -Ksimd=auto - --linkcoarray - -K{FLTLD NOFLTLD} - -K{HPC_ACE HPC_ACE2} - -K{lto nolto} - -K{simd_separate_stride simd_noseparate_stride}	C++ User's Guide

FX10 System Generation Number	Modification	Related Manuals
	<ul style="list-style-type: none"> - -Nstl=libc++ - -stdlib=<i>stl_kind</i> - -K{simd_reduction_product simd_noreduction_product} 	
	<p>The following optimization control specifiers are added.</p> <ul style="list-style-type: none"> - {nf nonf} - {fltld nofltld} - simd [aligned unaligned] - {simd_redundant_v1 simd_noredundant_v1} - uxsimd [aligned unaligned] 	C++ User's Guide
	<p>The following functions are improved.</p> <ul style="list-style-type: none"> - SIMD - Instruction Scheduling (Software Pipelining) 	None
	<p>Abolition of compilation options and the optimization control specifiers</p> <ul style="list-style-type: none"> - -Kcpu option - -K{prefetch_double_line_L2 prefetch_nodouble_line_L2} options and the optimization control specifiers {prefetch_double_line_L2 prefetch_nodouble_line_L2} - -K{simd_region_constant nosimd_region_constant} options 	C++ User's Guide
	SIMD optimization information is improved (" <i>VL: length</i> " is adding displayed).	C++ User's Guide
	The category of hardware monitor information is changed.	Runtime Information Output Function User's Guide
	Change the default value of the -Ksimd[= <i>leve</i>] option	C++ User's Guide
	Change of the compilation message when the compiler option -K{FLTLTD NOFLTLTD} is specified.	C++ User's Guide
	The option which the compiler option -Kuxsimd needs is changed.	C++ User's Guide
	Specification change of the optimization control specifiers simd and uxsimd	C++ User's Guide
	SIMD built-in functions are supported.	C++ User's Guide
	The file that suffixes are ".h" and ".H" is appended as an input file for the compile command.	C++ User's Guide
	[={loop procedure}] is added to the -Kifunc option.	C++ User's Guide
	<p>The following environment variables are added.</p> <ul style="list-style-type: none"> - CPATH - CPLUS_INCLUDE_PATH - LIBRARY_PATH 	C++ User's Guide
	The C++11 specifications are supported.	C++ User's Guide
	<p>The following Built-in Functions are added.</p> <ul style="list-style-type: none"> - Built-in functions with names that start with '__atomic' - __builtin_index 	C++ User's Guide

FX10 System Generation Number	Modification	Related Manuals
	<ul style="list-style-type: none"> - __builtin_isinf - __builtin_isinff - __builtin_isinfl - __builtin_isnan - __builtin_isnanf - __builtin_isnanl - __builtin_memchr - __builtin_rindex 	
	The language specification level is changed when the compiler options <code>-Xg -std={c++98 c++03 c++11}</code> are valid	C++ User's Guide
	The GNU C++ compatible version is changed when the compiler options <code>-Xg -std={c++11 gnu++11}</code> are valid	C++ User's Guide
	The C++ standard template library (STL) with the C++11 mode is changed	C++ User's Guide
11 or earlier	The C++11 specifications(part) are supported further.	C++ User's Guide
	GNU C++ Extensions <ul style="list-style-type: none"> - Assembler instructions with C expression operands are supported when the <code>-Xa</code> option is effective 	C++ User's Guide
	The following GNU C++ compatible options are supported. <ul style="list-style-type: none"> - <code>-std={c++03 c++11 gnu++03 gnu++11}</code> 	C++ User's Guide
	The following compiler options are added. <ul style="list-style-type: none"> - <code>-K{openmp_tls openmp_notls}</code> 	C++ User's Guide
	The following functions are improved. <ul style="list-style-type: none"> - Automatic Parallelization - General Optimization 	None
	The execution performance of the collapse clause of the loop construct of OpenMP is improved.	None
10 or earlier	The following compiler options are added. <ul style="list-style-type: none"> - <code>-K{openmp_assume_norecurrence penmp_noassume_norecurrence}</code> - <code>-N{cancel_overtime_compilation nocancel_overtime_compilation}</code> 	C++ User's Guide
	The function for canceling the program compilation that is forecasted to take a long time	C++ User's Guide
	The compiler option <code>-K{ordered_omp_reduction noordered_omp_reduction}</code> is changed to <code>-K{openmp_ordered_reduction openmp_noordered_reduction}</code> .	C++ User's Guide
09	The following compiler options are added. <ul style="list-style-type: none"> - <code>-K{loop_versioning loop_noversioning}</code> - <code>-K{parallel_fp_precision parallel_nofp_precision}</code> - <code>-K{omitfp noomitfp}</code> - <code>-K{remove_inlinefunction noremove_inlinefunction}</code> - <code>-g0</code> 	C++ User's Guide

FX10 System Generation Number	Modification	Related Manuals
	Abolition of compilation options and the optimization control specifiers - -K{FLTLD NOFLTLD} options and the optimization control specifiers {fltld nofltld}	C++ User's Guide
	The following optimization control specifiers are added. - clone - {loop_versionning loop_noversionning }	C++ User's Guide
	The execution performance of schedule(static) of OpenMP loop construct is improved.	None
	The -Komitfp option is induced by -Kfast, and maintenance of the trace back information.	C++ User's Guide
	Change of value of macro along with the compiler version up	C++ User's Guide
	Change of the name of temporary object file and the directory to store	C++ User's Guide

6.2.1.4 XPFortran Translator

None

6.2.2 Mathematical Libraries

6.2.2.1 SSL II

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	New routine of SSL II Thread-Parallel Capabilities DM_VMVSCCC, DM_VRANU5, DM_VSCLU, DM_VSCLUX, DM_VSCS, DM_VSSSLU, SM_VSSSLUX, DM_VSSSS	FUJITSU SSL II Thread-Parallel Capabilities User's Guide

6.2.2.2 C-SSL II

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	New routine of C-SSL II Thread-Parallel Capabilities c_dm_vjdhecr, c_dm_vjdnhr, c_dm_vlcspcxr1, c_dm_vlspcxr2, c_dm_vradau5, c_dm_vsrlu, c_dm_vsrlux, c_dm_vsrs, c_dm_vmvscce, c_dm_vranu5, c_dm_vsclu, c_dm_vsclux, c_dm_vscs	FUJITSU C-SSL II Thread-Parallel Capabilities User's Guide

6.2.2.3 BLAS

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	CBLAS, which is C interface to the BLAS, is supported.	BLAS LAPACK ScaLAPACK User's Guide
	The performance of DGEMM is improved when following conditions are met.	None

FX10 System Generation Number	Modification	Related Manuals
	<ol style="list-style-type: none"> 1. TRANSA='T'. And, 2. The number of threads is 16 or 32. 	

6.2.2.4 LAPACK

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	LAPACK version 3.5.0 of Netlib is supported.	BLAS LAPACK ScaLAPACK User's Guide
	PLASMA, which is parallelized LAPACK with pthread, is supported.	BLAS LAPACK ScaLAPACK User's Guide

6.2.3 MPI Library

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	On FEFS, the error class returned by the MPI_File_open function when a non-existent file and MPI_MODE_RDONLY are specified for the function respectively is changed.	MPI User's Guide
	The behavior when --version option or -V option is specified for the mpiexec command is changed.	MPI User's Guide
	The behavior when an invalid parameter is specified for -showme option of the compilation/linkage command is changed.	MPI User's Guide
	The output destination of version information and a help message is changed.	MPI User's Guide
	Behavior of mpiexec(1) when a same MCA parameter is specified more than once is changed.	MPI User's Guide
	Value of MPI_ERR_LASTCODE is changed.	MPI User's Guide
	Support of MPI-3.0 Standard	
	Change of the "Threshold Value" for Switching between Eager Protocol and Rendezvous Protocol	MPI User's Guide
09	Change of Extended RDMA Interface Error Message	MPI User's Guide
	The default value of MCA parameter orte_abort_print_stack is changed from 0 to 1.	MPI User's Guide

6.2.4 Programming Workbench

6.2.4.1 Debugger

FX10 System Generation Number	Modification	Related Manuals
10 or earlier	Start condition of Debugger is changed.	Debugger User's Guide

6.2.4.2 Profiler

FX10 System Generation Number	Modification	Related Manuals
12 or earlier	COARRAY function is supported. - Instant Profiler - Advanced Profiler	Profiler User's Guide
	MPI 3.0 function is supported. - Advanced Profiler	Profiler User's Guide
	Measured information of Hardware monitor information in Instant Profiler is changed.	Profiler User's Guide
	Measured information of Hardware monitor information in Advanced Profiler is changed.	Profiler User's Guide
	Frequency of the collection data and the analyzing data for the precision PA visibility function (Excel format) in Advanced Profiler is changed.	Profiler User's Guide
	Presentation item of the precision PA visibility function (Excel format) in Advanced Profiler is changed.	Profiler User's Guide