FUJITSU Software
NetCOBOL V11.1.0

FUJITSU

Release Notes

Windows(64)

Preface

This book explains Release Note of NetCOBOL.

Abbreviations

The following abbreviations are used in this manual:

Product Name	Abbreviation
Microsoft® Windows Server® 2016 Datacenter Microsoft® Windows Server® 2016 Standard Microsoft® Windows Server® 2016 Essentials	Windows Server 2016
Microsoft® Windows Server® 2012 R2 Datacenter Microsoft® Windows Server® 2012 R2 Standard Microsoft® Windows Server® 2012 R2 Essentials Microsoft® Windows Server® 2012 R2 Foundation	Windows Server 2012 R2
Microsoft® Windows Server® 2012 Datacenter Microsoft® Windows Server® 2012 Standard Microsoft® Windows Server® 2012 Essentials Microsoft® Windows Server® 2012 Foundation	Windows Server 2012
Microsoft® Windows Server® 2008 R2 Datacenter Microsoft® Windows Server® 2008 R2 Enterprise Microsoft® Windows Server® 2008 R2 Standard Microsoft® Windows Server® 2008 R2 Foundation	Windows Server 2008 R2
Windows® 10 Education Windows® 10 Home Windows® 10 Pro Windows® 10 Enterprise	Windows 10(x64)
Windows® 8.1 Windows® 8.1 Pro Windows® 8.1 Enterprise	Windows 8.1(x64)
Windows® 7 Home Premium Windows® 7 Professional Windows® 7 Enterprise Windows® 7 Ultimate	Windows 7(x64)
Microsoft(R) Visual C++(R) development system	Visual C++
Microsoft(R)Visual Basic(R) programming system	Visual Basic
Oracle Solaris	Solaris

- In this manual, when all the following products are indicates, it is written as "Windows" or "Windows(x64)".

    - Windows Server 2016

    - Windows Server 2012 R2

    - Windows Server 2012

- Windows Server 2008 R2

- Windows 10(x64)

- Windows 8.1(x64)

- Windows 7(x64)

Purpose of this book

This book explains the function addition, the trouble correction from the old edition, and information on interchangeability according to them.

It has aimed can the smoother shift of the customer who was using the old edition to this product.

Object reader in this book

An old product is introduced, and it is targeted for the shift to this product to be examined, and to be shifting.

Location of this book

Please refer to the manual of each program for a whole image or detailed information because it is fragmentary information though this book explains the function addition, the trouble correction, and information on interchangeability according to them.

About the registered trademarks

- NetCOBOL is a trademark or registered trademark of Fujitsu Limited or its subsidiaries in the United States or other countries or in both.

- Microsoft, Windows, Windows Server and Windows Vista are trademarks or registered trademarks of Microsoft Corporation.

- Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Oracle Solaris might be described as Solaris, Solaris Operating System, or Solaris OS.

- Other brand and product names are trademarks or registered trademarks of their respective owners.

- Trademark indications are omitted for some system and product names described in this manual.

Export Regulation

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

February 2017

Contents

Chapter 1 Outline of Additional Functionality

New product functionality and content is outlined below by version and level.

1.1 All Products

The following Information applies to all products.

Table 1.1 Outline of additional functionality in NetCOBOL V11

NO.	V/L	Function Name	Content	Location in Manual
1	V11.1.0	OS support	Windows Server 2016 is supported.	-
2	V11.0.1	OS support	Windows 10 is supported.	-
3	V11.0.0	OS support	Windows 8.1 and Windows Server 2012 R2 are supported.	-
3	V11.0.0	Fujitsu Common Tools	Fujitsu Middleware Installation System and FJQSS (Information Collection Tool)	- Overview Software Release Guide - FJQSS User's Guide

📙 **Note**

. . . .

The V/L column lists the version and level of NetCOBOL Enterprise Edition.

. . . .

1.2 NetCOBOL

Information described here applies to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

Table 1.2 Outline of additional functionality in NetCOBOL V11

NO.	V/L	Function Name	Content	Location in Manual
1	V11.1.0	Enhanced file-identifier and file-identifier literal	Up to 30 characters can now be specified for file-identifier. The file-identifier literal can be treated as an environment variable name.	NetCOBOL User's Guide - A.2.17 FILELIT(file-identifier literal handling) NetCOBOL Language Reference - 4.3.1.3 ASSIGN Clause (Sequential File, Relative File, and Indexed File)
2	V11.1.0	Micro Focus synonym support	The Micro Focus native synonyms can be used in NetCOBOL by specifying the compiler option MF.	NetCOBOL User's Guide - A.2.30 MF (Micro Focus compatible mode) NetCOBOL Language Reference - 10.8 Synonym Compatibility Mode
3	V11.1.0	PRINTER_n support	In the ASSIGN clause, PRINTER_n phrase is supported. n for	NetCOBOL Language Reference

NO.	V/L	Function Name	Content	Location in Manual
			PRINTER_n is an integer from 1 through 99.	- 4.3.1.3 ASSIGN Clause (Sequential File, Relative File, and Indexed File) - 11.8.6.17 The WRITE Statement (Object-Oriented Programming)
4	V11.1.0	Enhanced cobmkmf command	The extensions other than default (*.cob, *.cobol) can be used as an extension of the source files that the cobmkmf command handles. Moreover, the files in the subdirectory can be included in the build target.	NetCOBOL User's Guide - I.3 cobmkmf Command
5	V11.0.1	Eclipse 4.3 workbench support	Eclipse 4.3(*) workbench can now be used in NetCOBOL Studio, in which view layout can be changed and recent plug-in can be used. *: Using custom installation, NetCOBOL Studio based on Eclipse 3.4 can be installed instead of NetCOBOL Studio based on Eclipse 4.3. However, NetCOBOL Studio based on Eclipse 3.4 does not support operation on Windows 10. If you use NetCOBOL Studio on Windows 10, you use NetCOBOL Studio based on Eclipse 4.3.	NetCOBOL Studio User's Guide
6	V11.0.1	Free format COBOL editor support (Only NetCOBOL Studio (Eclipse 4.3))	Free format style COBOL editor can now be used in NetCOBOL Studio.	NetCOBOL Studio User's Guide - 5.10 Reference Formats - A.1.45 SRF compile option
7	V11.0.1	Security enhancement in remote development (Only NetCOBOL Studio (Eclipse 4.3))	SSH Port Forwarding enhances the security in remote developing on Linux(x64).	NetCOBOL Studio User's Guide - 9.2.1.1 NetCOBOL Remote Development Service
8	V11.0.0	UTF-32 support	Encoding form UTF-32 can now be used.	User's Guide - 20 Unicode
9	V11.0.0	COBOL Resource Project function support	A "COBOL resource project" is used for management of the library file and the descriptor file on the NetCOBOL Studio.	NetCOBOL Studio User's Guide - 3.1.2.2 COBOL Resource Project
10	V11.0.0	COBOL Solution Project function support	A "COBOL solution project" is used for management of multiple projects on the NetCOBOL Studio.	NetCOBOL Studio User's Guide - 3.1.2.3 COBOL Solution Project
11	V11.0.0	Project configuration conversion command	Provide the project configuration conversion command as a transfer support tool in order to convert the project in Project Manager to the project in NetCOBOL Studio.	NetCOBOL Studio User's Guide - E.3 Transition of Project According to Project Configuration Conversion Command

NO.	V/L	Function Name	Content	Location in Manual
12	V11.0.0	The default value of paper size and printing format specification support	Paper size and printing format can now be specified in print information file.	NetCOBOL User's Guide - 7.1.9 Print Information File
13	V10.4.0	31-digit support	Up to 31 digits can now be used for numeric items.	NetCOBOL Language Reference - 1.2.3.1 Numeric Literal - 1.7 Operation Mode - 5.4.8 PICTURE Clause - 5.4.16 USAGE Clause - 8.2.2 Host Variable Definitions - Appendix D Intermediate Results NetCOBOL User's Guide - 6.9 How to Use Other File Systems - A.2.2 ARITHMETIC NetCOBOL Messages
14	V10.4.0	Recursive calls	Programs can now be called recursively. Data items used by the recursive program are defined in the LOCAL-STORAGE SECTION.	NetCOBOL Language Reference - 2.3.8 Recursive Attribute of the Program - 3.1.1 Program-Id Paragraph - 6.4.7 CALL statement NetCOBOL User's Guide - 8.1.1 Calling Relationship Forms
15	V10.4.0	FETCH FIRST/LAST Statement support	FIRST and LAST can now be specified for embedded SQL FETCH statements.	NetCOBOL Language Reference - 8.6.4 FETCH Statement
16	V10.4.0	Extension of the file size limit	The maximum file size of a record sequential file, a line sequential file and an index file can now be extended to the system limit.	NetCOBOL User's Guide - 6.9 How to Use Other File Systems
17	V10.2.0	FETCH NEXT/PRIOR Statement support	NEXT and PRIOR can be specified for the embedded SQL FETCH Statement.	Language Reference - 8.6.4 FETCH Statement NetCOBOL User's Guide - Acquisition of data by FETCH PRIOR statement
18	V10.2.0	IBM DB2 SQLCA compatibility	SQLSTATE, SQLCODE, and SQLERRD included in SQLCA can be compiled.	Language Reference - 8.2.4 SQLSTATE/ SQLCODE

NO.	V/L	Function Name	Content	Location in Manual
				- 8.2.6 SQLERRD
19	V10.2.0	Unified specification of high-speed file processing	High-speed file processing can be specified for record sequential and line sequential files with one configuration specification as opposed to requiring specification for each file.	NetCOBOL User's Guide - High-Speed File Processing
20	V10.2.0	Event Log output	The Windows Event Log can be specified as the output destination for DISPLAY statements.	NetCOBOL User's Guide - ACCEPT/DISPLAY Function - @CBR_DISPLAY_CONSOLE_EVENTLOG_LEVEL (Specify the Event Type for DISPLAY UPON CONSOLE output to the Event Log) - @CBR_DISPLAY_CONSOLE_EVENTLOG_SRCNAME (Specify the Event Source Name for DISPLAY UPON CONSOLE output to the Event Log) - @CBR_DISPLAY_CONSOLE_OUTPUT (Specify the Event Log for DISPLAY UPON CONSOLE output) - @CBR_DISPLAY_SYSERR_EVENTLOG_LEVEL (Specify the Event Type for DISPLAY UPON SYSERR output to the Event Log) - @CBR_DISPLAY_SYSERR_EVENTLOG_SRCNAME (Specify the Event Source Name for DISPLAY UPON SYSERR output to the Event Log) - @CBR_DISPLAY_SYSERR_OUTPUT (Specify the Event Log for DISPLAY UPON SYSERR output) - @CBR_DISPLAY_SYSOUT_EVENTLOG_LEVEL (Specify the Event Type for DISPLAY UPON SYSOUT output to the Event Log) - @CBR_DISPLAY_SYSOUT_EVENTLOG_SRCNAME (Specify the Event Source Name for DISPLAY UPON

NO.	V/L	Function Name	Content	Location in Manual
				SYSOUT output to the Event Log)
				- @CBR_DISPLAY_SYSOU T_OUTPUT (Specify the Event Log for DISPLAY UPON SYSOUT output)
21	V10.2.0	CHECK function suppress option	Runtime options are provided to suppress individual CHECK functions. - nocb - noci - nocl - nocn - nocp	NetCOBOL User's Guide - Format of Runtime Options
22	V10.2.0	TRACE function suppress option	Runtime option is provided that suppresses the TRACE function. - nor	NetCOBOL User's Guide - Format of Runtime Options
23	V10.2.0	TRACE information file for separate processes	Separate TRACE information files can be generated for different processes to avoid conflicts in writing to the file.	NetCOBOL Debugging Guide - Trace Information
24	V10.2.0	Simpler identification of application errors in COBOL Error Report	When an application error occurs and COBOL Error Report starts, message JMW0099I-U is output. This makes such application errors easy to identify in the output.	NetCOBOL Messages - COBOL Error Report Messages
25	V10.2.0	Suppression of SYSERR of DISPLAY statement	The message output to SYSERR of the DISPLAY statement is suppressed.	NetCOBOL User's Guide - @NoMessage(Set to Suppress Runtime Messages)
26	V10.2.0	NetCOBOL Studio	NetCOBOL Studio includes development features for COBOL programming based on Eclipse, an open-source GUI development environment.	NetCOBOL Studio User's Guide
27	V10.2.0	Remote development functions	COBOL programs for the Solaris, Linux(Itanium), Linux(x64), or Windows(x64) servers can be developed remotely. After performing standalone tests on a local computer, add the settings for remote development to the project for a smooth transition to test builds and links on a server.	NetCOBOL Studio User's Guide - Remote Development Function
28	V10.2.0	Debug functions	The interactive debugger can help programmers detect logic errors in program processing. It can be used to set breakpoints to halt execution of a program, and to verify execution of a	NetCOBOL Studio User's Guide - Debugging Function

NO.	V/L	Function Name	Content	Location in Manual
			program by confirming data item values. The debugging of multithreaded programs is supported; however, the debugger does not support multiple debug sessions in a single workspace.	

29	V10.2.0	Screen Handling Module	Screen handling module is supported.	Language Reference
				
				  - 2.8 Screen Handling Module
				
				NetCOBOL User's Guide
				
				  - Using Screens

📖 **Note**

· ·

The Location in Manual column shows which sections of the documentation are impacted by the change.

· ·

1.3 PowerFORM

Information described here applies to the following products.

  - NetCOBOL Enterprise Edition for Windows (64bit)

Table 1.3 Outline of additional functionality in PowerFORM

NO.	V/L	Function Name	Content	Location in Manual

1	V11.1.0	Enhanced print features	Printer paper can now be specified when the specified paper was not supported with the output printer.	PowerFORM Runtime Reference
				
				  - Appendix C.
				    Printer Information File

2	V11.0.0	Unicode(UTF-32) support	UTF-32 data can now be used in COBOL applications.	PowerFORM Runtime Reference - How to Use PowerFORM RTS

3	V11.0.0	Encoding systems for print information file	Print Information File in UTF-8 with BOM can now be used in user created Unicode COBOL applications.	PowerFORM Runtime Reference
				
				  - How to Use PowerFORM RTS
				
				  - Chapter 5.
				    Printer Information File
				
				  - Appendix A.
				    Troubleshooting Guide
				
				  - Appendix E.
				    Error Messages

4	V11.0.0	31-digit support	Up to 31 digits can now be used for numeric fields in COBOL applications.	PowerFORM Runtime Reference - Output field declarations
5	V11.0.0	Form exporting	The Form can now be outputted to PDF. And The character (surrogate	PowerFORM Runtime Reference

NO.	V/L	Function Name	Content	Location in Manual
			pair) added by character-code standard "JIS X 0213:2004(JIS2004)" can be output.	- Chapter 4. Extended Functions - Appendix C. Printer Information File
6	V11.0.0	Enhanced print features	Print options are provided to define the behavior when output printer is omitted and when it fails in opening the specified output printer.	PowerFORM Runtime Reference - Appendix C. Printer Information File
7	V11.0.0	Enhanced print features	Print options are provided to define the behavior when the specified paper was not supported with the printer device.	PowerFORM Runtime Reference - Appendix C. Printer Information File
8	V11.0.0	Enhanced print features	Print options are provided to define the behavior when the specified paper feed was not supported with the printer device.	PowerFORM Runtime Reference - Appendix C. Printer Information File
9	V11.0.0	Enhanced print features	The processing time at expansion/ reduction print and the print preview can be shortened.	PowerFORM Runtime Reference - Appendix C. Printer Information File
10	V11.0.0	Enhanced print features	The processing time of the Form output using the fixed-width font can be shortened.	PowerFORM Runtime Reference - Appendix C. Printer Information File
11	V11.0.0	Support locale	The following 4 locales outputs are supported. - Japanese - English - Chinese (simplified Chinese) - Portuguese	Release Notes - 4.4.4 PowerFORM RTS
12	V11.0.0	UTC (Coordinated Universal Time) support	UTC (Coordinated Universal Time) can be specified.	PowerFORM Runtime Reference - Appendix C. Printer Information File
13	V11.0.0	Enhanced print features	When the user character-code is UNICODE, the table of full-width/ half-width character used by outputting the UNICODE character can be specified, and customize at the character-code level can be specified.	PowerFORM Runtime Reference - Appendix C. Printer Information File
14	V11.0.0	Enhanced print features	The Form Descriptor (free frame) can be used.	PowerFORM Runtime Reference - Chapter 4. Extended Functions
15	V11.0.0	Enhanced print features	The new function added in PowerFORM V11 can be used.	PowerFORM Getting Started

NO.	V/L	Function Name	Content	Location in Manual
				- What's New in PowerFORM V11?

1.4 PowerBSORT

Information described here applies to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

There is no additional function.

1.5 J Adapter Class Generator

Information described here applies to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

Table 1.4 Outline of additional functionality in J Adapter Class Generator

NO.	V/L	Function Name	Content	Location in Manual
1	V11.0.0 (V11.0.0)	Unicode character for file path character string	The Unicode character can be used for the file path character string specified for a command line argument and an optional file.	-
2	V10.0.0 (V10.2.0)	J Adapter Class Generator	J Adapter Class Generator is a tool that generates a COBOL class (adapter class) used to call a Java class. Using the generated adapter class makes Java class libraries available from COBOL.	J Adapter Class Generator User's Guide

NOTE: The version in parentheses shows the version level in the NetCOBOL series.

Chapter 2 Information on interchangeability

Here, information on the interchangeability changed from the version and level before is described.

2.1 NetCOBOL Development Environment

Information described here applies to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

2.1.1 Change the file association by using PRINTER_n phrase

Content

- V11.0 or earlier

  The ASSIGN clause with PRINTER_n phrase associates file-name with a physical file.

- V11.1 or later

  The ASSIGN clause with PRINTER_n phrase associates file-name with a printing device.

Action

Change PRINTER_n to another name.

2.1.2 Change default link option /SUBSYSTEM

Content

The default of link option /SUBSYSTEM was changed.

Impact

When the following functions are used, the output character string might not be displayed in the window with this link option.

- COBOL Console Window

- Screen Function

Action

If you use COBOL console window or screen function, please specify following LINK option when you link main programs.

- When compile option MAIN(WINMAIN) is specified for the main program, it is WinMain type.

  When compile option MAIN(MAIN) is specified for the main program, it is main type.

  [WinMain type]

  /SUBSYSTEM:WINDOWS and 5.02

  [main type]

  /SUBSYSTEM:CONSOLE and 5.02

- If you use screen function it is possible to display it correctly by specifying environment variable @ScrnSize and changing size of logical screen.

2.1.3 Reference format setting in NetCOBOL Studio

Content

In NetCOBOL Studio (the Eclipse 4.3 workbench), SRF and TAB compile option setting is not consistent with the applicable editor setting in V11.0.

Action

To restore the consistency as NetCOBOL Studio (the Eclipse 3.4 workbench), select "SRF and TAB compile option setting to be consistent with the applicable editor setting" checkbox.

2.1.4 Runtime code-set when compilation option ENCODE is specified

Content

Added the compilation option ENCODE to define the encoding form of the data item.

With the specification of compilation option ENCODE, the runtime code set is changed.

- When the compilation option RCS is explicitly specified.

    - The runtime code-set becomes the code-set that is specified with the RCS compilation option.

- When the compilation option RCS is not specified explicitly.

    - When the compilation option ENCODE is specified explicitly, then the runtime code-set becomes the Unicode.

    - When the compilation option ENCODE is not specified explicitly, then the runtime code-set becomes the ANSI code page.

Conditions

1. In compiler versions prior to 10.5.0, the compilation option RCS (SJIS) is specified explicitly or implicitly and it is a program asset.

2. And, in compiler versions 11.0.0 and forward, without specifying the compilation option RCS (SJIS) explicitly, for the program assets of 1 above, the compilation option ENCODE(SJIS[,SJIS]) is described explicitly and recompiled.

Impact

The impact corresponding to the above conditions is as follows.

- An error (JMP0081I-U) occurs in the following situations.

    - When the recompiled program is called from a program that is not recompiled.

    - When a program that is not recompiled is called from a program that is recompiled.

- During execution after the main program is recompiled, the source becomes the Unicode.

For details, refer to " Unicode" in the " NetCOBOL User's Guide".

Action

Create the target program without explicitly specifying the compilation option ENCODE.

Moreover, in regards to the repository, keep the compilation option RCS and the compilation option ENCODE specified on the reference and remote side the same.

2.1.5 Location of SQLCODE/SQLMSG/SQLERRD definition

SQLCODE, SQLMSG, and SQLERRD can now be defined outside of the SQL declaration section. Under the conditions listed below, SQL statement runtime information was not stored in V10.1.0. In V10.2.0 and later, runtime information is stored in SQLCODE, SQLERRD, and SQLMSG.

1. SQLSTATE is described inside the SQL declaration section, and

2. SQLCODE, SQLERRD, or SQLMSG are described outside the SQL declaration section, and

3. SQL statements are executed, and

4. SQLCODE, SQLERRD, or SQLMSG are referred to in the PROCEDURE DIVISION, and

5. Conditions 1 through 4 are occurred in the same compilation unit.

In V10.2.0 and later, SQLCODE, SQLERRD, and SQLMSG are reserved names used to fetch SQL statement runtime information. If you have variables by those names that are used for any other purpose, the results are unpredictable. In this case, change the names of those variables.

2.1.6 Result of addition and subtraction of zoned decimal item outside regulations

Under the following conditions, the execution result of V10.1.0 and that of V10.2.0 and later are different.

1. There is a binary operation of addition or the subtraction.

2. The number of identifier of the result is 1.

3. [NOT] ON SIZE ERROR is not specified.

4. ROUNDED is not specified.

5. Two operands are zoned decimal and numbers of digits are from 16 to 18.

6. The accuracy of intermediate result (*)of binary operation is from 17 to 19 digits.

7. Either of representation of two operands is wrong as zoned decimal.

   * : Refer to "COBOL Language Reference" about the accuracy of intermediate result.

Example

```
       WORKING-STORAGE SECTION.
       01 DATA-ZONE1    PIC S9(18).
       01 DATA-ZONE2    PIC S9(18).
       01 DATA-2 REDEFINES  DATA-ZONE2 PIC X(18).
       01 DATA-ZONE3    PIC S9(18).
       01 DATA-3 REDEFINES  DATA-ZONE3 PIC X(18).

       PROCEDURE DIVISION.
          MOVE X"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" TO DATA-2  DATA-3  .....(a)
          COMPUTE DATA-ZONE1 = DATA-ZONE2 + DATA-ZONE3
          DISPLAY DATA-ZONE1
```

In the above program, save area of alphanumeric item is allocated in zoned decimal. When (a) is executed, wrong representation (0xFFFF...FFFF) is set to DATA-ZONE2 and DATA-ZONE3 as zoned decimal.

- V10.1.0

```
    +333333333333333330
```

- V10.2.0 and later

```
    +545555555455555554
```

The result of the arithmetic statement including the operand to which wrong internal format is set as zoned decimal is not provided for. Please correct the part where wrong internal format is set.

The outline of the representation that zoned decimal is correct is as follows.

Data item	Zone part	Numeric part	Sign part
Zoned decimal (SEPARATE)	3	0 to 9	2B,2D
Zoned decimal (no SEPARATE)	3		4,5

2.1.7 Interchangeable information regarding bug fixes

Here, it explains bug fixes corrected by NetCOBOL V11 and later version in which the operation of the NetCOBOL development environment changes in the following tables.

Table 2.1 Interchangeable information on bug fixes of NetCOBOL Development Environment

No.	V/L(*)	P number	Content
1	V10.1.0 to V10.4.0	PG76651	When all of the following conditions are met while executing a COBOL program, a size comparison of national items varying in length (national language edited item, including the intrinsic function) was incorrect. 1. Compile option RCS (UCS2,LE) or RCS (UTF16,LE) is specified. 2. And, there is a size comparison of a National item and a National item or National language character constant. 3. And, at least one side is an item with the ANY LENGTH clause is specified or reference modification. 4. And, the length of the compared targets differs.

2	V10.1.0	PG77383	Under the following conditions, the execution results are incorrect.
			
			1. The OPTIMIZE compiler option is specified (*1).
			
			2. There is a statement that sets packed decimal item to numeric edited data item.
			
			3. The number of digits for integer part of packed decimal item and numeric edited data item are the same and neither both of them doesn't have decimal-part.
			
			4. The numeric edited data item specifies only zero suppression for the edit (Only '9', 'Z', and '*' are used for PICTURE character-string).
			
			5. Before the statement in 2 (*2), there is a statement that sets data item (or intermediate result) to packed decimal item.
			
			6. The relation between number of digits for the data item in 5 (or intermediate result) and number of digits for the packed decimal part in 2 is the following :
			
			<pre>-- 	
			Data item (or intermediate result) Packed decimal item
			------------------------------------ ---------------
			          2 3
			          4 5
			          6 7
			          8 9
			         10 11
			         12 13
			         14 15
			         16 17
			--</pre> 	
			7. The data item in 5.(or intermediate result) and the packed decimal item in 2 don't have the decimal-part.
			
			*1: The default is NOOPTIMIZE.
			
			*2: MOVE statement, COMPUTE statement, and so on.

3	V10.1.0	PG78440	Under the following conditions, the execution results are incorrect. 1. The BINARY(BYTE) or BINARY(WORD, MLBOFF) compiler option is specified, along with 2. The following intrinsic functions: [A]

No.	V/L(*)	P number	Content
			- FUNCTION MAX
			- FUNCTION MIN
			- FUNCTION MEAN
			- FUNCTION MEDIAN
			- FUNCTION RANGE
			[B]
			- FUNCTION ANNUITY
			- FUNCTION NUMVAL
			- FUNCTION NUMVAL-C
			- FUNCTION RANDOM
			3. The arguments of the function in 2) above are fixed point of 9 digits or less.
			4. The number of the arguments in [A] above is 4 or more.
4	V10.1.0 to V11.0.0	PH05538	Under either of the following conditions, the COBOL program execution result is incorrect or terminates abnormally (ACCESS VIOLATION). [Condition 1] 1. The data item declared to based-storage section is used by either of the following methods. - There is a subscript. or - There is a reference modifier. The high-order-end-character-position of the reference modifier is a variable. 2. And, compile option OPTIMIZE(*A) is effective *A: Default is NOOPTIMIZE. [Condition 2] 1. In the based-storage section, the declared data items are specified in any of the following statements. - INSPECT statement - STRING statement (Nucleus) - UNSTRING statement (Nucleus) [Condition 3] 1. In the based-storage section, the data items specified the OCCURS clause are being declared. 2. And, the data items of 1 are being specified in the INITIALIZE statement.
5	V10.0.1 to V11.0.0	PH05861	Under the following conditions, the result of the move is incorrect. The reference modification range of the sending side is shifted one digit to the left. 1. The following MOVE sentences (*1) are described. - The sending side is a signed zoned decimal item. - And, the receiving side is a numeric edited data item or a floating point item. 2. And, the SIGN IS TRAILING SEPARATE CHARACTER clause is specified in the sending data item.

No.	V/L(*)	P number	Content
			3. And, there is a reference modification of the sending data item.
			4. And, the length of reference modification of (3) is specified with a numeric literal.
			*1: Includes the implicit MOVE statement.
6	V11.0.0	PH04570	Under the following [Conditions], the following [Situation 1] or [Situation 2] might occur.
			[Situation 1]
			The COBOL compiler terminates normally (*A) without giving an error message.
			<pre>JMN2038I-S Length of the literal in the VALUE clause must not exceed the length of the item. The extra characters at the right end of the literal are truncated</pre>
			*A: In the created object program, the following literal values are set to the data item of the object program.
			- The extra characters at the right end of the literal are truncated to match the item length
			[Situation 2]
			The COBOL compiler outputs the following error message to a correct syntax.
			<pre>JMN2106I-S The value of the literal following 'THROUGH' in the VALUE clause of the condition-name must be greater than the value of the literal preceding 'THROUGH'.</pre>
			[Conditions]
			1. There is a national item or national edited item that either encodes the following.
			- UTF32
			- UTF32LE
			- UTF32BE
			2. And, the VALUE phrase is specified as follows.
			- data item of (1), or
			- Condition-name of which conditional variable is data item of (1)
			3. And, in the VALUE phrase of (2), the national nonnumeric literal (*B) is specified to meet one of the following.
			- National nonnumeric literal is larger than the sizes specified by the PICTURE phrase
			- The value of the national nonnumeric literal following 'THROUGH' is greater than the value of the national nonnumeric literal preceding 'THROUGH'
			4. And, national nonnumeric literal of (3) are from 41 to 80 characters.
			*B: The concatenation expression of national nonnumeric literal is contained.
7	V10.1.0 to V11.0.0	PH02265	Under the following [Conditions 1] or [Condition 2], even if there is an error in the program, the compiler may not output the error messages (*A).
			(*A)

			<pre>JMN1775I-S A nonnumeric literal or a national literal must be specified after an AS phrase. The compiler skips to the next paragraph or division.
			JMN1107I-S An invalid character-string is specified in the program-id, or the program-id is missing. The program-name of the next paragraph or division is used.</pre>

No.	V/L(*)	P number	Content

			JMN1292I-S An invalid character-string is specified as the program-name, or the program-name is missing.
			JMN5526I-S The method-name specified in the INVOKE statement must be an identifier, nonnumeric literal, or national character literal. The INVOKE statement is ignored.
			JMN5561I-S The method-name specified in an in-line method invocation must be a nonnumeric literal or a national character literal. The in-line method invocation is ignored.
			The object program that has been created can be executed correctly.
			
			[Conditions 1]
			
			    1. The concatenation expression is described in the following.
			
			        - AS phrase of program-name
			
			        - AS phrase of class-name
			
			        - AS phrase of method-name
			
			        - AS phrase of property-name
			
			        - AS phrase of program-name-literal
			
			        - Method-name specified for INVOKE statement or in-line method invocation
			
			    2. And, the high order end of concatenation expression of (1) is a nonnumeric literal.
			
			    3. And, the concatenation expression of (1) contains a hexadecimal nonnumeric literal.
			
			    4. And, the COBOL source program code set and the runtime code set are the following combinations.
			
			        a. The COBOL source program code set is SJIS and the runtime code set is SJIS.
			
			        b. Or, the COBOL source program code set is UTF-8 and the runtime code set is Unicode.
			
			[Conditions 2]
			
			    1. The concatenation expression is described in the following.
			
			        - AS phrase of program-name
			
			        - AS phrase of class-name
			
			        - AS phrase of method-name
			
			        - AS phrase of property-name
			
			        - Method-name specified for INVOKE statement or in-line method invocation
			
			    2. And, the high order end of concatenation expression of (1) is a national language nonnumeric literal.
			
			    3. And, the concatenation expression of (1) contains a national hexadecimal nonnumeric literal.
			
			    4. And, the COBOL source program code set and the runtime code set are the following combinations.
			
			        a. The COBOL source program code set is SJIS and the runtime code set is SJIS.
			
			Supplementary Explanation:
			
			A literal that specifies in the following must be a nonnumeric literal or a national language nonnumeric literal.

No.	V/L(*)	P number	Content
			- AS phrase of program-name
			- AS phrase of class-name
			- AS phrase of method-name
			- AS phrase of property-name
			- AS phrase of program-name-literal
			- Method-name specified for INVOKE statement or in-line method invocation
			The above conditions are in violation of the syntax rules.

*: V/L is a scope where the bug exists.

2.2 NetCOBOL Runtime Environment

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

2.2.1 Character-code conversion result of data and imperfect character outside character-code range

Content

The result of the character-code conversion of the data stored in the data item of the DISPLAY statement, the STRING statement (writing 2), the UNSTRING statement (writing 2), and the WRITE statement of the print file changed to prevent an unjustified data error.

- Character-code conversion of data beyond the limits of character-code set conversion origin

  - Before V11.0

  It is converted into an alternative character.

  - After V11.0

  It becomes a code conversion error when executing it, and the following messages (detailed code 42 (0x2a)) are output.

    - DISPLAY statement: JMP0086I-W

    - STRING statement (writing 2) and the UNSTRING statement (writing 2): An overflow condition has occurred. JMP0260I-U when ON OVERFLOW is not specified

    - WRITE statement of the print file: JMP0310I-I/U and JMP0320I-I/U (It is executed along with the execution result when the input-output error of the file occurs).

- Character-code conversion of imperfect character

  When a surrogate pair of the character-code is a high rank or only one of the subordinate positions is stored or the multi byte character is lacked. The result of converting such an imperfect character is different.

  - Before V11.0

  The character is converted into the replacement character.

  - After V11.0

  It becomes a code conversion error during execution, and the following messages (detail code 22 (0x16) or 42 (0x2a)) are output.

    - DISPLAY statement: JMP0086I-W

    - STRING statement (writing 2) and the UNSTRING statement (writing 2): An overflow condition has occurred. JMP0260I-U when ON OVERFLOW is not specified.

- WRITE statement of the print file: JMP0310I-I/U and JMP0320I-I/U (It is executed along with the execution result when the input-output error of the file occurs).

Action

Correct to store a correct character-code in the data item that refers to a detailed message code when executing it.

Specify the execute environment variable @CBR_CONVERT_CHARACTER=SYSTEM expressly when you want to obtain the result similar to before V11.0.

However, it is not possible to specify it by the program using the specification after V11.0.

Refer to environment variable information @CBR_CONVERT_CHARACTER on the user's guide for details.

2.2.2 Replacement character used by code conversion

Content

The replacement character used has changed when the character-code conversion corresponding to the character-code in the conversion origin does not exist.

- Before V10.4

  Normal-width underscore "_"

- After V11.0

  Normal-width question "?"

Action

Specify the execute environment variable information @CBR_CONVERT_CHARACTER=SYSTEM expressly.

However, it is not possible to specify it by the program using the specification after V11.0.

Refer to environment variable information @CBR_CONVERT_CHARACTER on the user's guide for details.

2.2.3 Change in severity code of message when application is executed

Content

The severity code of the message changed when the following applications were executed.

- Before V11.0

  JMP0086I-E CHARACTER CODE CONVERSION FAILED. $1 $2

- After V11.0

  JMP0086I-W CHARACTER CODE CONVERSION FAILED. $1 $2

Influence

The Return code (PROGRAM-STATUS) of the COBOL program changes with a change in the severity code.

Action

Refer to the Messages manual for additional details.

2.2.4 Interchangeable information regarding bug fixes

Here, it explains bug fixes corrected by NetCOBOL V10 or later version in which the operation of the NetCOBOL development environment changes in the following tables.

Table 2.2 Interchangeable information on bug fixes of NetCOBOL Runtime Environment

No.	V/L(*)	P number	Content
1	V10.1.0 to V10.4.0	PG97090	The problem of not correctly replacing the alternative character specified for argument-2 was corrected when there was no national character corresponding to the character converted by the NATIONAL-OF function when the COBOL program was executed under the following conditions. 1. It is a program compiled using compiler option RCS(UTF16,BE). 2. And, the NATIONAL-OF function is used. 3. And, Argument-2 is specified for the function of No.2. And, data that is not alphanumeric characters is specified for argument-1 in No.2 above, and the code conversion error (there is no corresponding National character) occurs internally.
2	V10.1.0 to V10.4.0	PG76651	When all of the following conditions are met while executing a COBOL program, a size comparison of national items varying in length (national language edited item, including the intrinsic function) was incorrect. 1. Compiler option RCS(UCS2,LE) or RCS(UTF16,LE) is specified. 2. And, there is a size comparison of National item and National item or National language character constant. 3. And, at least one side is an item with the ANY LENGTH clause is specified or reference modification. 4. And, the length of the compared targets differs.
3	V10.1.0	PG77099	Under the following conditions, the execution environment variable @CBR_SSIN_FILE=THREAD doesn't become effective, and one input file is shared by file input of ACCEPT statement in the process. (*) 1. It is an application that operates by multi-thread. 2. The execution environment variable @CBR_SSIN_FILE=THREAD is specified. 3. ACCEPT statement is executed, and data is input from the file. * : When the execution environment variable @CBR_SSIN_FILE=THREAD is specified, input file of each thread can be opened by file input of ACCEPT statement.

4	V10.1.0	PG73346	Under the following conditions, at the execution, Unnecessary zero string is set to four subordinate position digits of the hexadecimal character of eight digits buried under execution time message JMP0320I-I/U.
			
			1. File operation using COBOL file system (*) is done.
			
			2. Either of the following I/O statements is executed.
			
			  - READ statement
			
			  - WRITE statement
			
			  - REWRITE statement
			
			  - DELETE statement
			
			  - START statement
			
			3. The execution of I/O statement failed.
			
			* : It doesn't correspond when BSAM is specified.

5	V10.1.0	PG79852	Under the following conditions, when the message of the application error or execution time is generated in the running of COBOL application, the message that the following do not anticipate before outputting diagnostic report is output in COBOL Error Report.

No.	V/L(*)	P number	Content
			`There is no disk in the drive. Please insert the disk in the drive.`
			1. Environment variable @CBR_JUSTINTIME_DEBUG=NO is not specified.
			2. The load module that operates in the process to be diagnosed makes DEBUG information file (*) (PDB file).
			3. In the folder where the load module is stored in 2, there is no DEBUG information in 2.
			4. The system that uses NetCOBOL drives the state that cannot be referred by the un-mount of virtual CD drive.
			5. The drive character of the drive that makes DEBUG information file in 2 and the drive that cannot be referred in 4 is corresponding.
			* : DEBUG information file (PDB file) is generated when the linkage option "/DEBUG" is specified.
6	V10.1.0	PG79859	Under the following conditions, when COBOL application is debugged by remote debugging facility of NetCOBOL Studio, the message that the following do not anticipate while debugging it is output.
			`There is no disk in the drive. Please insert the disk in the drive.`
			1. COBOL application is debugged by remote debugging facility of NetCOBOL Studio.
			2. The load module that operates in the process to be diagnosed makes DEBUG information file (*) (PDB file).
			3. In the folder where the load module is stored in 2, there is no DEBUG information in 2.
			4. The system that uses NetCOBOL drives the state that cannot be referred by the un-mount of virtual CD drive.
			5. The drive character of the drive that makes DEBUG information file in 2 and the drive that cannot be referred in 4 is corresponding.
			* : DEBUG information file (PDB file) is generated when the linkage option "/DEBUG" is specified.
7	V10.1.0 to V11.0.0	PH01026	Under the following conditions, the character comparison cannot be compared correctly.
			1. The COBOL program compiled with the compile option NSPCOMP(ASP) is executed.
			2. And, the encoding of the data item is Shift-JIS for the following.
			- The compile option RCS is omitted.
			- Or, the compile option RCS(SJIS) is specified.
			- Or, the compile option ENCODE(SJIS,SJIS) is specified.
			3. And, either of the following character comparisons is done.
			- National character comparison with a national data item as an operand
			- Character comparison with a group item as an operand
			However, the following conditions are excluded.
			- Comparison between group items that do not include any national data items
			- Comparison between group items including an item whose attribute does not specify explicit or implicit display
			4. And, in character comparison of (3), the character code range is the following.

No.	V/L(*)	P number	Content
			- X"8181" to X"819F" - X"81E0" to X"81FC" 5. And, in either of the object of comparisons, the character immediately after the character of (4) is the national spaces(X"8140"). 6. And, in either of (5), - there is an ANK space (X"2020") of 2 bytes in the same character position as a national space of (5). or - the character position of (4) is an end of data item.
8	V11.0.0	PH03200	Under the following conditions, when the COBOL application is executed, the record is not written as the collating sequence of the index key item. Moreover, the intended record might not be read. 1. When the program is compiled, the compile option ENCODE(SJIS,SJIS) is specified. 2. And, the item attribute of the prime record key or alternate record key is a national item. 3. And, the WRITE statement is executed.
9	V10.1.0 to V11.0.0	PH06622	Under the following conditions, the record is not printed according to the specified printing attributes or printing position. For a Unicode application, additionally, the following phenomena occurs. (*A) - The runtime system message "JMP320I 'CNVER=xx'" is output. - The printing results of the national item are garbled. *A: It depends on the data stored in the national item. 1. The print file without a FORMAT clause is used. 2. And, this program is output directly to the printer without using the related product. 3. And, the record item specified for the WRITE statement or the data item specified for the FROM phrase of WRITE statement is a group item that is subordinate to the data item with the REDEFINES clause. 4. And, the following either is described in the group item of (3) that specifies a REDEFINES clause. - There is a group item subordinate. - There is a national item subordinate. - An effective CHARACTER TYPE clause is specified for the data item subordinate. (*B) - An effective PRINTING POSITION clause is specified for the data item subordinate. (*B) 5. And, after the group item of (3), either of the following is described in the same level-number as (3). - There is an elementary item or a group item that specifies a CHARACTER TYPE clause. - There is an elementary item or a group item that specifies a PRINTING POSITION clause. - There is a group item that has the data item subordinate with an effective CHARACTER TYPE clause.

No.	V/L(*)	P number	Content

			- There is a group item that has the data item subordinate with a PRINTING POSITION clause.
			
			*B: When the REDEFINES clause is specified for either of the following items, during the compilation of a COBOL program, the JMN2224I-W message is outputted. It is a warning that the CHARACTER TYPE clause or the PRINTING POSITION clause specified for the item is invalid.
			
			  - The data item with CHARACTER TYPE clause or the PRINTING POSITION clause specified
			
			  - The group item that has the subordinate data item with effective CHARACTER TYPE clause or PRINTING POSITION clause
			
			  - **Example program showing what occurs:**
			
			<pre>DATA DIVISION.
			WORKING-STORAGE SECTION.
			01 DATA1.
			  03 DATA2.
			    05 DATA31.
			      07 DATA31A PIC X(10).
			    05 DATA32 REDEFINES DATA31.
			      07 DATA32A. *> Conditions of (4)
			        09 DATA32A1 PIC X(5).
			        09 DATA32A2 PIC X(5).
			    05 DATA33. *> Conditions of (5)
			      07 DATA33A PIC N(5) MODE-1.
			
			PROCEDURE DIVISION.
			
			    WRITE PRINT-REC FROM DATA1 AFTER PAGE. *> Conditions of (3)</pre>

*: V/L is a scope where the bug exists.

2.3 PowerFORM

Information described here applies to the following products.

  - NetCOBOL Enterprise Edition for Windows (64bit)

There is no information on interchangeability.

2.4 PowerBSORT

Information described here applies to the following products.

  - NetCOBOL Enterprise Edition for Windows (64bit)

2.4.1 Interchangeable information regarding bug fixes

Here, it explains bug fixes corrected by PowerBSORT V7 and later version in which the operation of the PowerBSORT changes in the following tables.

Table 2.3 Interchangeable information on bug fixes of PowerBSORT

NO.	V/L(*)	P number	Content of change

1	V7.0.0	PH05442	Under the following conditions, PowerBSORT cannot correctly check whether the reconstruction field and the selection field exist in the record.
			As a result of an error in usage or implementation, the output may produce unpredictable or abnormal results.
			
			1. The bsortex command is used.
			
			2. And, the record format is one of the following.
			
			   - Binary file variable-length record form (-record recform=var), or
			
			   - Text file fixed field specification (-record recform=txtfix)
			
			3. And, two output file information options (-output) or more are specified.
			
			4. And, the following options are specified by two output file information options (-output) or more.
			
			   - Record reconstruction option (-output reconst=...)
			
			   - Record selection option (-output include=.../omit=.../case=...)
			
			5. And, the field in the record is specified for a reconstruction field for the record reconstruction option (form of "pos.len" or "pos.END").
			
			6. And, the maximum position of the reconstruction field specified by each output file information option (-output) is different. or,
			The maximum position of the selection field specified by each output file information option (-output) is different.

*: V/L is a scope where the bug exists.

2.5 J Adapter Class Generator

Information described here applies to the following products.

- NetCOBOL Enterprise Edition for Windows (64bit)

2.5.1 Code conversion when application is executed

Content

The replacement character used has changed when Shift JIS is specified as code set during execution and data that is out of Shift JIS is included in character strings that are got by using GET-STRING-X method or GET-STRING-N method.

- Before V10.1

  Normal-width underscore "_"

- After V11.0

  If the data before conversion is alphanumeric character, Normal-width underscore.

  If the data before conversion is Japanese character, Double-byte underscore.

Action

Specify the execute environment variable information @CBR_CONVERT_CHARACTER=SYSTEM expressly.

However, it is not possible to specify it by the program using the specification after V11.0.

Refer to environment variable information @CBR_CONVERT_CHARACTER on the User's Guide for details.

2.5.2 Error message output when code conversion error occurs

Content

In the conversion of data using the java-lang-String class method (such as GET-STRING-X, GET-STRING-N, NEW-STRING-X and NEW-STRING-N), if the data is incorrect, a code conversion error will occur.

Then, in NetCOBOL V11.0 or later, the following error message is to be output.

```
CHARACTER CODE CONVERSION FAILED. ERRNO:DETAIL
```

Action

Modify incorrect data items to store character codes correctly.

2.5.3 Characters that the results of code conversion are different

Content

In operation of an application with Shift_JIS, some Unicode characters that are passed to Java have changed as follows.

Shift_JIS	Unicode before V10.1	Unicode after V11
8160	U+301C	U+FF5E
8161	U+2016	U+2225
817C	U+2212	U+FF0D
8191	U+00A2	U+FFE0
8192	U+00A3	U+FFE1
81CA	U+00AC	U+FFE2

Action

Change the setting that is before V10.1 by specifying the following environment variable.

```
COBJNI_CONVERT=SJIS
```

Chapter 3 Program Fix List

In this software, the trouble that occurs by the version and level before is corrected.

This document explains the content of the correction corrected by this software of this version level.

3.1 NetCOBOL Development Environment

Information described here applies to the following products.

- NetCOBOL Enterprise Edition V11.1.0 for Windows (64bit)

Table 3.1 Program fix list for the NetCOBOL Development Environment

No.	V/L(*)	P number	Content

1	V10.1.0 to V11.0.0	PG81806	Under the following conditions, the COBOL compiler terminates abnormally.
			
			1. Any of the following statements are described.
			
			  - CALL STATEMENT
			
			  - CANCEL STATEMENT
			
			  - INVOKE STATEMENT
			
			  - STRING statement/UNSTRING statement specified with a CSV-FORMAT
			
			2. And, the statement of (1) is described in a conditional statement.
			
			3. And, the hierarchy of the conditional statement of (2) is 180 or more. (*A).
			
			*A: In the following code
			
			- The EVALUATE statement is hierarchy 1.
			
			- The COMPUTE statement is hierarchy 2.
			
			However, depending on the amount of computer memory, there may be no abnormal termination.

2	V10.1.0 to V11.0.0	PG98620	Under the following conditions when compiling, the COBOL program terminates abnormally without giving an error message. 1. Compiler option OPTIMIZE is specified (*A). 2. And, the CALL statement in which the called program is specified in the identifier is described. 3. And, the USING phrase in the CALL statement of (2). 4. And, the RETURNING phrase in the CALL statement of (2). 5. And, pointer data item is set or referred to. 6. And, an internal program exists. 7. And, when the USING phrase is omitted in the header of the Procedure Division of (6).

No.	V/L(*)	P number	Content
			*A: Default is NOOPTIMIZE.
3	V10.1.0 to V11.0.0	PH01579	Under the following conditions, the COBOL compiler creates an object program that terminates abnormally at execution time. However, no message is output. 1. The section-name or the paragraph-name is described in the USING phrase of a CALL statement.
4	V10.2.0 to V11.0.0	PH01938	Under the following conditions, for files that were sent to the server from the NetCOBOL Studio, some of the characters in the COBOL source file are converted to "?". 1. In a COBOL Project created with NetCOBOL Studio, the text file encode "MS932" is specified. 2. And, in the Server Information dialog box (*1), the "ShiftJIS" is specified in "COBOL Source character-code on the server side" item. 3. And, in the Remote Development page (*2) of a COBOL project, the server name of 2. is specified in the "Server name". 4. And, the following characters are included in the COBOL source file. - "〜" 0x8160(ShiftJIS)(WAVE DASH) - "∥" 0x8161(ShiftJIS)(DOUBLE VERTICAL LINE) - "−" 0x817C(ShiftJIS)(MINUS SIGN) - "¢" 0x8191(ShiftJIS)(CENT SIGN) - "£" 0x8192(ShiftJIS)(POUND SIGN) - "¬" 0x81CA(ShiftJIS)(NOT SIGN) - "￤" 0xFA55(ShiftJIS)(FULLWIDTH BROKEN BAR) 5. And, when COBOL sources are sent to the server, either of the following remote development functions is used. - Makefile Creation - Build - Rebuild *1: Refer to "9.3.1 Setting the NetCOBOL Studio operating environment" in "NetCOBOL Studio User's Guide." *2: Refer to "9.3.2 Setting server information for a COBOL project" in the "NetCOBOL Studio User's Guide."
5	V11.0.0	PH01985	Under the following conditions, during the compilation of a COBOL program, the compiler terminates abnormally. No message is output. 1. SYMBOLIC CHARACTER clause is specified. 2. And, the alphabet-name specified for (1) is associated with the following by the ALPHABET clause. - SJIS - UTF8 - UTF16 - UTF16LE - UTF16BE

No.	V/L(*)	P number	Content
			- UTF32
			- UTF32LE
			- UTF32BE
			Information This condition is incorrect as syntax rules. Refer to Syntax Rule 5 of "4.2.3.12 SYMBOLIC CHARACTERS Clause" in the "NetCOBOL Language Reference".

6	V11.0.0	PH01986	Under the following conditions, during the compilation of a COBOL program, the compiler terminates abnormally.
			
			   1. The class including a method prototype definition is compiled with NetCOBOL V10.4 or earlier.
			
			   2. And, a separated method definition is compiled with NetCOBOL V11.0 or later.
			
			Supplementary Explanation:
			
			The method prototype definition and a separated method definition must be compiled with the same VL compiler. With the above conditions, the compiler outputs the following error message.
			
			<pre>JMN1768I-S The repository file for the CLASS 'xxx' that defined
			the PROTOTYPE METHOD must be created with the same VL compiler.
			The compiler skips to the next element.</pre>

7	V10.1.0 to V11.0.0	PH02032	Under the following conditions, during the compilation of a COBOL program, the compiler outputs the following message for the correct continuation line of the literal.
			
			<pre>JMN1022I-S The ending quotation mark of the literal is missing.
			The characters at the end of Area B are assumed to be a literal.
			JMN1021I-S The continuation line of the literal must have a hyphen
			in the indicator area and must start with a quotation mark. The
			continuation of the literal is assumed.</pre>
			   1. REPLACE statement is used.
			
			   2. And, there is the following description in replacement range (*A) by REPLACE statement of (1).
			
			      a. COPY statement is used.
			
			      b. And, the character (*B) before the COPY statement of (a) is not one of the following characters.
			
			         - period
			
			         - left parentheses
			
			         - right parentheses
			
			         - colon
			
			      c. And, continuation of lines of the literal (*C) is described after the COPY statement of (a).
			
			      d. And, continuation of lines of the COBOL words (*D) is not described between the COPY statement of (a) and the continuation lines of the literal of (c).
			
			*A: Until the next REPLACE statement appears or up to the end of the separately compiled program
			
			*B: The space and the comment line are excluded

No.	V/L(*)	P number	Content
			*C: Nonnumeric literal, hexadecimal nonnumeric literal, national nonnumeric literal or boolean literal
			*D: PICTURE character-string or COBOL words excluding national language user-defined word
| | | | ```
Example)
      IDENTIFICATION DIVISION.
      PROGRAM-ID. SAMPLE1.
          REPLACE ==ABC== BY ==DEF==. *>Start of replacement
      ENVIRONMENT DIVISION.
      CONFIGURATION SECTION.
      SPECIAL-NAMES.
          SYMBOLIC CONSTANT
            COPY SYMCON.   *> The character before the COPY
                              *> statement is not period, parentheses
                                 *> or colon.
          .
      DATA DIVISION.
      WORKING-STORAGE SECTION.
      01 MSG-01 PIC X(100) VALUE "ABCD...
    -                          "XYZ".    *> Continuation of lines
                                            *> of the literal
``` |
| 8 | V11.0.0 | PH02110 | Under the following conditions, during the compilation of a COBOL program, the compiler cannot read the screen form descriptor. The compiler then outputs the following message. |
| | | | ```
JMN1857I-S When UTF32 Encoding format is specified in the screen
form descriptor, then there is a need to specify the screen form
descriptor that is converted for UTF32. Stop the inclusion of the
screen form descriptor.
``` |
			1. The format 3 COPY statement is specified with XMDLIB in the COBOL source program.
			2. And, the encoding of the national character of screen form descriptor is UTF-32 by specifying any of the following.
			- Compiler option ENCODE(UTF8,UTF32) is specified.
			- The alphabet-name is associated with UTF32, UTF32LE or UTF32BE in the following. (*A)
			a. ENCODING phrase of the COPY statement of (1).
			b. Or, ENCODING phrase of the record that has the COPY statement of (1).
			c. Or, ENCODING phrase of the file description entry that has the record with the COPY statement of (1).
			3. And, character string "SMD" or "PMD" is specified in the environment variable SMED_SUFFIX.
			*A: If a different encoding is specified, the relation of strength becomes a.> b.> c.
9	V10.1.0 to V11.0.0	PH02168	Under the following conditions, during the compilation of a COBOL program, the compiler outputs a wrong error message (*A) and the COBOL program cannot be compiled.
			(*A)
| | | | ```
JMN1479I-S The concatenation expression is too long. A literal is
created using the first parts of the expression so that it does
not exceed the maximum length.
JMN0124I-U This source file has an invalid character.
``` |

No.	V/L(*)	P number	Content
			<pre>(substepname=JMNxxx, line-number=nn.) The compilation is
terminated.</pre> 1. Encoding of COBOL source is UTF-8. 2. And, concatenation expression is described for national language nonnumeric literal (*B) 3. And, the number of characters concatenated at (2) is 54-80 characters. (*B) Including symbolic constant and named literal.
10	V10.1.0 to V11.0.0	PH02265	Under the following [Conditions 1] or [Condition 2], even if there is an error in the program, the compiler may not output the error messages (*A). (*A) <pre>JMN1775I-S A nonnumeric literal or a national literal must be
specified after an AS phrase. The compiler skips to the next
paragraph or division.
JMN1107I-S An invalid character-string is specified in the
program-id, or the program-id is missing. The program-name of the
next paragraph or division is used.
JMN1292I-S An invalid character-string is specified as the
program-name, or the program-name is missing.
JMN5526I-S The method-name specified in the INVOKE statement must
be an identifier, nonnumeric literal, or national character
literal. The INVOKE statement is ignored.
JMN5561I-S The method-name specified in an in-line method
invocation must be a nonnumeric literal or a national character

literal. The in-line method invocation is ignored.</pre>
The object program that has been created can be executed correctly.
[Conditions 1]
1. The concatenation expression is described in the following.
  - AS phrase of program-name
  - AS phrase of class-name
  - AS phrase of method-name
  - AS phrase of property-name
  - AS phrase of program-name-literal
  - Method-name specified for INVOKE statement or in-line method invocation
2. And, the high order end of concatenation expression of (1) is a nonnumeric literal.
3. And, the concatenation expression of (1) contains a hexadecimal nonnumeric literal.
4. And, the COBOL source program code set and the runtime code set are the following combinations.
  a. The COBOL source program code set is SJIS and the runtime code set is SJIS.
  b. Or, the COBOL source program code set is UTF-8 and the runtime code set is Unicode.
[Conditions 2]
1. The concatenation expression is described in the following.
  - AS phrase of program-name

No.	V/L(*)	P number	Content
			- AS phrase of class-name
			- AS phrase of method-name
			- AS phrase of property-name
			- Method-name specified for INVOKE statement or in-line method invocation
			2. And, the high order end of concatenation expression of (1) is a national language nonnumeric literal.
			3. And, the concatenation expression of (1) contains a national hexadecimal nonnumeric literal.
			4. And, the COBOL source program code set and the runtime code set are the following combinations.
			a. The COBOL source program code set is SJIS and the runtime code set is SJIS.
			Supplementary Explanation:
			A literal that specifies in the following must be a nonnumeric literal or a national language nonnumeric literal.
			- AS phrase of program-name
			- AS phrase of class-name
			- AS phrase of method-name
			- AS phrase of property-name
			- AS phrase of program-name-literal
			- Method-name specified for INVOKE statement or in-line method invocation
			The above conditions are in violation of the syntax rules.
11	V11.0.0	PH02798	Under of the following conditions, during the execution of the COBOL program, the execution result of STRING statement (CSV) or UNSTRING statement (CSV) are incorrect.
			The character-strings of execution result are garbled.
			1. The runtime code set is Unicode.
			2. And, the following sentence is described.
			- STRING statement of format 2
			- Or, UNSTRING statement of format 2
			3. And, in the statement of (2), a national item or a national edited item is specified in the following,
			- Sending item of STRING statement
			- Or, receiving item of UNSTRING statement
			4. And, the encoding form of item (3) is one of the following.
			- SJIS
			- UTF16LE (*A)
			- UTF16BE (*A)
			- UTF32
			- UTF32LE

No.	V/L(*)	P number	Content

			- UTF32BE
			
			5. And, the encoding form of item (3) is different from the runtime code set.
			
			6. And, item (3) is a reference modified item.
			
			7. And, the length of the reference modification of (6) is specified with the data-name.
			
			*A: In the following cases only.
			
			  - The endian (LE/BE) of the item (3) is different from the endian of the compiler option RCS or ENCODE is specified.
			
			  - And, the encoding form of the item is not "UTF16".
			
			[Example]
			
			`OK : STRING  WK1(1:5)     INTO RCVDATA  BY CSV-FORMAT`
			`NG : STRING  WK1(1:POS1)   INTO RCVDATA  BY CSV-FORMAT`

12	V10.2.0 to V11.0.0	PH03188	Under the following conditions, the characters of the file path included in the COBOL project is garbled, and the following situation might occur.
			
			[Situation 1]
			
			If the character-strings of the source file path that was set in the source file folder are garbled, the source file is removed from the building target.
			
			[Situation 2]
			
			Build error occurs because the compiler and the linker are not able to find the reference files. An example would be when the character-strings of the path of the compiler option LIB are garbled and the library files are not read. Then the compiler outputs the following message.
			
			`JMN1057I-S An I/O error occurred while processing the COBOL`
			`library text. Processing of the library text was terminated.`
			
			1. NetCOBOL Studio is used.
			
			2. And, the non- ASCII code is used in the file path or folder path specified for setting COBOL project.
			
			3. And, the COBOL project of (2) is imported into a workspace.

13	V10.1.0 to V11.0.0	PH03685	Under the following conditions, when compiled with the 64bit compiler, the memory size to be deleted by the execution of the CANCEL statement is different from that of the 32bit compiler.
			
			1. Compiler option THREAD(SINGLE) is specified.(*A)
			
			2. And, a dynamic link library (DLL) of the subprogram is created by either of the following methods.
			
			  - Subprogram is statically linked. (simple structure)(*B).
			
			  - Or, subprogram is linked by a dynamic link structure.(*B)
			
			3. And, the subprogram of (2) is called by either of the following using the dynamic program structure.
			
			  - Subprogram is called from the program for which compiler option DLOAD is specified.
			
			  - Or, subprogram is called from the CALL statement for which the data-name is specified.
			
			4. And, the CANCEL statement was executed to the subprogram of (2).

No.	V/L(*)	P number	Content
			*A: Default is THREAD(SINGLE)
			*B: For the details on the simple structure, dynamic link structure and dynamic program structure refer to "NetCOBOL User's Guide."
14	V11.0.0	PH03777	Under the following conditions, the COBOL compiler terminates abnormally. 1. There is either of the following items. - Alphanumeric edited data item - National edited data item 2. And, PICTURE character-string of item of (1) is more than 46 characters.(*A) 3. And, the item of (1) is specified in the receiving side of the following move statements. - MOVE statement - Or, statement executing the implicit MOVE statement *A: Here, it means the character count described in the PICTURE clause. [Example] The following editing formats are the same. However, the count of character is different. PIC XBXXXXXXXX = 10 character PIC XBX(8) = 6 character
15	V11.0.0	PH04215	Under the following conditions, the execution results are incorrect. In the execution of the INVOKE statement and the in-line method invocation, the parameter is cut or garbled. 1. The runtime code set is Unicode. 2. And, the national item or the national language nonnumeric literal is specified for either of the following. - USING or RETURNING parameter of an INVOKE statement - Parameter of in-line method invocation 3. And, in the formal parameter of the method invoked by (2), the following is defined. - National item for which ANY LENGTH phrase is specified 4. And, either of the following conditions exist. a. National item is specified for (2). And, the encode mode of the parameter of (2) and the parameter of (3) are different.(*A) Or b. National language nonnumeric literal is specified for (2). And, the encode mode of national item of (3) is UTF32. *A: In this situation, even though the syntax is wrong, the compile error is not output.
16	V10.1.0 to V11.0.0	PH04233	Under the following conditions, the compiler terminates abnormally and outputs the following message. - V10.4 or earlier <pre>System error 'errno=0xXXX' occurred on 'open REP,xxx'.</pre> <pre>JMN0013I-S The repository file cannot be opened. The object file cannot be used.</pre>

No.	V/L(*)	P number	Content

			- V11.0 or later
			
			``
			System error 'errno=0xXXX' occurred on 'iconv : error code=7,
			code no=2 from=sjisms to=utf8_4_ms'.
			``
			
			``
			JMN0124I-U This source file has an invalid character. The
			compilation is terminated. (substep-name=xxx, linenumber=xxx.)
			``
			
			1. The encoding of the COBOL source file and library file is Shift_JIS.
			
			2. And, the en-size kana characters are included in either of the following the nonnumeric literal.
			
			    a. AS literal in CLASS-ID paragraph
			
			    b. Text-name literal(*)
			
			3. And, the literal of 2 exceeds the following length in UTF-8 representation.
			
			    a. AS literal in CLASS-ID paragraph : 320 bytes
			
			    b. Text-name literal: 240 bytes(*A)
			
			*A: When the product version you are using is a V10.4 or earlier, (2.b) and (3.b) do not apply to the occurrence conditions.

17	V11.0.0	PH04570	Under the following [Conditions], the following [Situation 1] or [Situation 2] might occur.
			
			[Situation 1]
			
			The COBOL compiler terminates normally (*A) without giving an error message.
			
			``
			JMN2038I-S Length of the literal in the VALUE clause must not
			exceed the length of the item. The extra characters at the right
			end of the literal are truncated
			``
			
			*A: In the created object program, the following literal values are set to the data item of the object program.
			
			  - The extra characters at the right end of the literal are truncated to match the item length
			
			[Situation 2]
			
			The COBOL compiler outputs the following error message to a correct syntax.
			
			``
			JMN2106I-S The value of the literal following 'THROUGH' in the
			VALUE clause of the condition-name must be greater than the value
			of the literal preceding 'THROUGH'.
			``
			
			[Conditions]
			
			1. There is a national item or national edited item that either encodes the following.
			
			  - UTF32
			
			  - UTF32LE
			
			  - UTF32BE
			
			2. And, the VALUE phrase is specified as follows.
			
			  - data item of (1), or
			
			  - Condition-name of which conditional variable is data item of (1)
			
			3. And, in the VALUE phrase of (2), the national nonnumeric literal (*B) is specified to meet one of the following.

No.	V/L(*)	P number	Content
			- National nonnumeric literal is larger than the sizes specified by the PICTURE phrase
			- The value of the national nonnumeric literal following 'THROUGH' is greater than the value of the national nonnumeric literal preceding 'THROUGH'
			4. And, national nonnumeric literal of (3) are from 41 to 80 characters.
			*B: The concatenation expression of national nonnumeric literal is contained.
| 18 | V10.1.0 to V11.0.0 | PH04722 | Under the following conditions, the following message is displayed in the **Problems** view. The build of the COBOL project failed.<br><br>```
Target "cobol_buildtool-clean" does not exist in the project
"XXX".
It is used from target "clean_build".
```<br><br>XXX: project name<br><br>1. The "Dynamic-link library" is specified in **Target type** of the COBOL project.<br><br>2. And, the COBOL project of (1) is imported into the workspace.<br><br>3. And, in the **Rename** of context menu item for the **Navigator** view, the project name of the COBOL project of (2) is renamed. |
| 19 | V10.1.0 to V11.0.0 | PH04922 | Under the following conditions, during the debugging of the COBOL application, the following message is displayed.<br><br>[Example: Debugging sample program "SALES" of PowerFORM]<br><br>```
SALES: SALES.EXE - Entry Point Not Found
* The procedure entry point Iser32.CharNextExa could not be
 located in the dynamic link library api-ms-win-downlevel-user32-
 l1-1-0.dll
```<br><br>1. The operating system is Windows 7.<br><br>2. And, Internet Explorer version is 10.<br><br>3. And, the dll importing DLL of MinWin(*).<br><br>*: It is the dll that the file name beginning with "api-ms-win". This file is stored in system32 folder of Windows. |
| 20 | V10.1.0 to V11.0.0 | PH05538 | Under either of the following conditions, the COBOL program execution result is incorrect or terminates abnormally (ACCESS VIOLATION).

[Condition 1]

1. The data item declared to based-storage section is used by either of the following methods.

- There is a subscript.
or

- There is a reference modifier. The high-order-end-character-position of the reference modifier is a variable.

2. And, compile option OPTIMIZE(*A) is effective

*A: Default is NOOPTIMIZE.

[Condition 2]

1. In the based-storage section, the declared data items are specified in any of the following statements.

- INSPECT statement |

No.	V/L(*)	P number	Content
			- STRING statement (Nucleus)
			- UNSTRING statement (Nucleus)
			[Condition 3]
			1. In the based-storage section, the data items specified the OCCURS clause are being declared.
			2. And, the data items of 1 are being specified in the INITIALIZE statement.
21	V10.0.1 to V11.0.0	PH05861	Under the following conditions, the result of the move is incorrect.
			The reference modification range of the sending side is shifted one digit to the left.
			1. The following MOVE sentences (*1) are described.
			- The sending side is a signed zoned decimal item.
			- And, the receiving side is a numeric edited data item or a floating point item.
			2. And, the SIGN IS TRAILING SEPARATE CHARACTER clause is specified in the sending data item.
			3. And, there is a reference modification of the sending data item.
			4. And, the length of reference modification of (3) is specified with a numeric literal.
			*1: Includes the implicit MOVE statement.
22	V10.1.0 to V11.0.0	PH06489	Under the following conditions, during the compilation of COBOL program, the compiler terminates abnormally. No message is output.
			1. In the item length, internal boolean of multiple of 8 is declared.
			2. And, in the item (1), SYNCHRONIZED clause is being specified.
			3. And, in the item (1), VALUE clause is being specified.
			4. And, in the item (1), any of the following condition is met.
			a. Level number of item in (1) is 01 or 77. Or
			b. Length from the end position of item of (1) to the end position of the record including (1) is shorter than the length of the following.
			- In the item of (1), when OCCURS clause is not being specified then :1 byte
			- In the item of (1), when OCCURS clause is being specified: 1 byte x repeat count of maximum value
			5. And, when items of (1) are being described in any of the section.
			- constant section of program definition
			- working-storage section of object definition of class
			- constant section of object definition of class
			- constant section of method definition of class
			- working-storage section of program definition
			- working-storage of factory definition of class
			- constant section of factory definition of class
			- working-storage section of separated method definition
			- constant section of separated method definition

No.	V/L(*)	P number	Content

			*: End position depicts the location from start of the record to the offset + Item length.
			
			`Record including the item of (1).`
			
			[diagram: a record box with a section labeled "Item of (1)", with markers • and ••]
			
			`*:End position of  Item (1)`
			
			`**:End position of the record including the items of (1) item`

23	V10.1.0 to V11.0.0	PH06787	Under the following [Condition 1] or [Condition 2], when a COBOL program is executed, there are times when correct value cannot be passed to the temporary parameter, [Condition 1] 1. When calling of any of the following is being used. - Method calling with the INVOKE statement - In-line calling of the method 2. And, in the calling of (1), any of the following parameter is specified. - Parameter which specifies BY CONTENT - data item of the constant section - nonnumeric literal 3. And, JUSTIFIED clause is being specified in the temporary parameter of the calling destination of (1). 4. And, when length of parameter of (2) and length of temporary parameter of (3) varies. [Condition 2] 1. When calling of any of the following is being used. - Method calling by INVOKE statement - In-line calling of the method 2. And, in the calling of (1). , any of the following parameters are specified. - Parameter specified with BY CONTENT - data item of the constant section - nonnumeric literal 3. And, in the temporary parameter of calling destination of (1), BLANK WHEN ZERO clause is being specified. 4. And, the result of move from parameter of (2) to the temporary parameter of (3) becomes 0. **Supplementary Explanation:** - The temporary parameter specifies the data items specified by the RETURNING or USING of the header of the procedure division - Moving to the temporary parameters is performed in the parameter passing specified with BY CONTENT. For details refer to NetCOBOL Language Reference.

No.	V/L(*)	P number	Content

[Example of condition 1]

```
-calling program--------------
 :
WORKING-STORAGE SECTION.
01 VAR PIC X(3).
01 OBJ OBJECT REFERENCE CCC.
PROCEDURE DIVISION.
     MOVE "ABC" TO VAR.
     INVOKE CCC "NEW" RETURNING OBJ.
     INVOKE OBJ "MMM" USING BY CONTENT VAR.
------------------------------

-called program--------------
CLASS-ID. CCC INHERITS FJBASE.
:
METHOD-ID. MMM.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
01 VAR PIC X JUSTIFIED.
PROCEDURE DIVISION USING VAR.
     DISPLAY "VAR=" VAR.
END METHOD MMM.
------------------------------

 [execution result]
```

```
VAR=A     <-  A correct result is "C".
```

[Example of condition 2]

```
-calling program--------------
 :
WORKING-STORAGE SECTION.
01 OBJ OBJECT REFERENCE CCC.
01 RET-VAL PIC 9.
PROCEDURE DIVISION.
     INVOKE CCC "NEW" RETURNING OBJ.
     MOVE OBJ::"MMM" ("000") TO RET-VAL.
END PROGRAM PPP.
------------------------------

-called program--------------
CLASS-ID. CCC INHERITS FJBASE.
:
METHOD-ID. MMM.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
01 VAR PIC 9(3) BLANK WHEN ZERO.
01 RET-VAL PIC 9.
PROCEDURE DIVISION USING VAR RETURNING RET-VAL.
     DISPLAY "VAR=" VAR.
END METHOD MMM.
------------------------------

 [execution result]
```

No.	V/L(*)	P number	Content
			``` VAR=000 <- A correct result is " ". ```
24	V10.1.0 to V11.0.0	PH07796	Under the following conditions, during the debugging of the COBOL application, the value of a data item is not displayed in a tooltip. 1. For the reference format in the COBOL editor of the NetCOBOL Studio, the fixed format is specified. 2. And, the execution of the program is halted at a breakpoint of the debugger function. 3. And, the mouse cursor is positioned on a data item displayed in the COBOL editor.

25	V10.1.0
	to
	V11.0.0
		PH08595	Under the following conditions, the COBOL compiler terminates abnormally.
			
			1. Compile option OPTIMIZE (*A) is effective.
			
			2. And, the repeat processing is described.
			
			3. And, the following are described in the repeat processing of (2).
			
			   - index-name
			
			   - reference modification
			
			   - subscripting item
			
			   - multiplication
			
			*A: Default is NOOPTIMIZE.

26	V10.1.0 to V11.0.0	PH08887	Under the following conditions, during the debugging of the COBOL application, the breakpoint is displayed in the incorrect position. 1. The source file contains more than 65535 lines. 2. And, debugging a program using the NetCOBOL Studio. 3. And, the position of the breakpoint is at a line number greater than 65535.

27	V10.1.0
	to
	V11.0.0
		PH09195	Under the following conditions, the compiler terminates abnormally. The following message might be output.
			
			``
			The instruction 0xfffffffd referenced memory at 0xfffffffd. The
			memory could not be written."
			``
			
			1. The ACCEPT statement of screen operation is described.
			
			2. And, the WITH phrase is specified in the ACCEPT statement of (1).
			
			3. And, immediately after the WITH phrase of (2), one or more of the following (*A) are described.
			
			   - LEFT-JUSTIFY
			
			   - RIGHT-JUSTIFY
			
			   - SPACE-FILL
			
			   - TRAILING-SIGN
			
			   - UPDATE
			
			4. And, immediately after (3), the PROMPT phrase is described without a CHARACTER phrase(*A).
			
			5. And, immediately after (4), one or more of the following are described.
			
			   - AUTO
			
			   - BELL

No.	V/L(*)	P number	Content
			- BLINK
			- FULL
			- GRID
			- HIGHLIGHT
			- LOWLIGHT
			- LEFTLINE
			- OVERLINE
			- REQUIRED
			- REVERSE-VIDEO
			- SECURE
			- SIZE
			- UNDERLINE
			- FOREGROUND-COLOR
			- BACKGROUND-COLOR
			- ZERO-FILL

			*A: These phrases are treated as comments.
			
			Example)
			
			`ACCEPT WDATA WITH UPDATE PROMPT AUTO.`
			`                        (3)    (4)      (5) -- condition`

28	V10.1.0 to V11.0.0	PH10340	Under the following conditions, when the COBOL program outputs runtime message (JMP0024I-U) and it terminates abnormally, a wrong line number is displayed in the runtime message. 1. There is a program that calls deadlock exit schedule subroutine (COB_DEADLOCK_EXIT). 2. And, there is no deadlock exit in a program that called the subroutine or the high-level program.

29	V10.1.0 to V11.0.0	PH11312	Under the following conditions, the compiler terminates abnormally and outputs the following message.
			
			`JMN0102I-U The compilation process cannot be continued. If other diagnostic messages have been generated, correct those errors and try the compilation again. (substep-name=JMNCOU, module-name=SABEMAC, detailcode=6692, line-number=nnnn)`
			
			1. The command option -P is specified when compiling.
			
			2. And, the compiler option LIST is specified.
			
			3. And, the UTF-8 is specified as the code set of the COBOL source file and the library file.
			
			4. And, in either of the following, the name is more than 63 bytes in length.
			
			  - procedure-name
			
			  - section-name
			
			  - paragraph-name

No.	V/L(*)	P number	Content
30	V10.1.0 to V11.0.0	PH11342	Under the following conditions, the compiler terminates abnormally and outputs the following message. ``` JMN0124I-U This source file has an invalid character. The compilation is terminated. (substep-name=JMN220, linenumber=nn.) ``` 1. The compiler option SCS(UTF8) is specified. 2. And, a SQL statement is described. 3. And, the statement length of (2) is more than 242 bytes 4. And, the SQL statement of (2) contains non- ASCII characters.

*: V/L is a scope where the bug exists.

3.2 NetCOBOL Runtime Environment

Information described here applies to the following products.

- NetCOBOL Enterprise Edition V11.1.0 for Windows (64bit)

Table 3.2 Program fix list for the NetCOBOL Runtime Environment

No.	V/L(*)	P number	Content
1	V10.1.0 to V11.0.0	PG90530	Under the following conditions, in the COBOL Error Report, the value of the eflag register is incorrect. (The value of the 'R15' register is set to the eflag register.) ``` > register : R14=0000000000000005 R15=000000001C007000 EFL=1C007000 ``` 1. When the COBOL Error Report outputs diagnostic information for the NetCOBOL program.
2	V10.1.0 to V11.0.0	PH00632	Under the following conditions, when the COBOL Error Report is used, it outputs the JMW0013I-W message and the dump file is not output. 1. When the COBOL application is executed, the application error or the U level runtime error (JMP0009I-U, JMP0010I-U, or JMP0370I-U) occurred. 2. And, the COBOL Error Report is started. 3. And, ten dump files are located in the dump file created folder.

3	V10.1.0 to V11.0.0	PH01026	Under the following conditions, the character comparison cannot be compared correctly.
			
			1. The COBOL program compiled with the compile option NSPCOMP(ASP) is executed.
			
			2. And, the encoding of the data item is Shift-JIS for the following.
			
			   - The compile option RCS is omitted.
			
			   - Or, the compile option RCS(SJIS) is specified.
			
			   - Or, the compile option ENCODE(SJIS,SJIS) is specified.
			
			3. And, either of the following character comparisons is done.
			
			   - National character comparison with a national data item as an operand
			
			   - Character comparison with a group item as an operand
			
			   However, the following conditions are excluded.
			
			     - Comparison between group items that do not include any national data items

No.	V/L(*)	P number	Content
			- Comparison between group items including an item whose attribute does not specify explicit or implicit display
			4. And, in character comparison of (3), the character code range is the following.
			- X"8181" to X"819F"
			- X"81E0" to X"81FC"
			5. And, in either of the object of comparisons, the character immediately after the character of (4) is the national spaces(X"8140").
			6. And, in either of (5),
			- there is an ANK space (X"2020") of 2 bytes in the same character position as a national space of (5). or
			- the character position of (4) is an end of data item.
4	V11.0.0	PH03200	Under the following conditions, when the COBOL application is executed, the record is not written as the collating sequence of the index key item. Moreover, the intended record might not be read. 1. When the program is compiled, the compile option ENCODE(SJIS,SJIS) is specified. 2. And, the item attribute of the prime record key or alternate record key is a national item. 3. And, the WRITE statement is executed.
5	V10.1.0 to V11.0.0	PH03685	Under the following conditions, when compiled with the 64bit compiler, the memory size to be deleted by the execution of the CANCEL statement is different from that of the 32bit compiler. 1. Compiler option THREAD(SINGLE) is specified.(*A) 2. And, a dynamic link library (DLL) of the subprogram is created by either of the following methods. - Subprogram is statically linked. (simple structure)(*B). - Or, subprogram is linked by a dynamic link structure.(*B) 3. And, the subprogram of (2) is called by either of the following using the dynamic program structure. - Subprogram is called from the program for which compiler option DLOAD is specified. - Or, subprogram is called from the CALL statement for which the data-name is specified. 4. And, the CANCEL statement was executed to the subprogram of (2). *A: Default is THREAD(SINGLE) *B: For the details on the simple structure, dynamic link structure and dynamic program structure refer to "NetCOBOL User's Guide."
6	V10.1.0 to V11.0.0	PH06337	Under the following conditions, when the COBOL Error Report is used, the COBOL Error Report terminates abnormally. 1. When the COBOL application is executed, the application error or the U level runtime error occurred.

No.	V/L(*)	P number	Content
			2. And, the environment variable value of 5120 bytes or more is set to the environment variable.
			3. And, the COBOL Error Report is started.
7	V10.1.0 to V11.0.0	PH06622	Under the following conditions, the record is not printed according to the specified printing attributes or printing position.

Under the following conditions, the record is not printed according to the specified printing attributes or printing position.

For a Unicode application, additionally, the following phenomena occurs. (*A)

- The runtime system message "JMP320I 'CNVER=xx'" is output.

- The printing results of the national item are garbled.

*A: It depends on the data stored in the national item.

1. The print file without a FORMAT clause is used.

2. And, this program is output directly to the printer without using the related product.

3. And, the record item specified for the WRITE statement or the data item specified for the FROM phrase of WRITE statement is a group item that is subordinate to the data item with the REDEFINES clause.

4. And, the following either is described in the group item of (3) that specifies a REDEFINES clause.

    - There is a group item subordinate.

    - There is a national item subordinate.

    - An effective CHARACTER TYPE clause is specified for the data item subordinate. (*B)

    - An effective PRINTING POSITION clause is specified for the data item subordinate. (*B)

5. And, after the group item of (3), either of the following is described in the same level-number as (3).

    - There is an elementary item or a group item that specifies a CHARACTER TYPE clause.

    - There is an elementary item or a group item that specifies a PRINTING POSITION clause.

    - There is a group item that has the data item subordinate with an effective CHARACTER TYPE clause.

    - There is a group item that has the data item subordinate with a PRINTING POSITION clause.

*B: When the REDEFINES clause is specified for either of the following items, during the compilation of a COBOL program, the JMN2224I-W message is outputted. It is a warning that the CHARACTER TYPE clause or the PRINTING POSITION clause specified for the item is invalid.

- The data item with CHARACTER TYPE clause or the PRINTING POSITION clause specified

- The group item that has the subordinate data item with effective CHARACTER TYPE clause or PRINTING POSITION clause

- **Example program showing what occurs:**

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01  DATA1.
```

No.	V/L(*)	P number	Content
| | | | ```
03  DATA2.
  05  DATA31.
    07  DATA31A   PIC X(10).
  05  DATA32  REDEFINES DATA31.
    07  DATA32A.                        *> Conditions of (4)
      09  DATA32A1  PIC X(5).
      09  DATA32A2  PIC X(5).
  05  DATA33.                           *> Conditions of (5)
    07  DATA33A    PIC N(5) MODE-1.


PROCEDURE DIVISION.

    WRITE PRINT-REC FROM DATA1 AFTER PAGE.  *> Conditions of
(3)
``` |
| 8 | V11.0.0 | PH08039 | Under the following conditions, when the COBOL application is executed, an unfixed value (*A) is set in the data item that receives an OSIV system runtime parameter.

1. In COBOL source, there is a code that receives an OSIV system runtime parameter.

2. And, compiler option RCS is not specified. Or, the compiler option RCS(SJIS) is specified.

3. And, the main program is compiled by the following.

  - NetCOBOL for Windows (64bit) V10.1 to V10.5

4. And, the runtime system at execution time is V11.0 or later.

5. And, the environment variable @CBR_CONVERT_CHARACTER is not specified, or @CBR_CONVERT_CHARACTER=ICONV is specified.

6. And, the characters (*B) that are outside the SJIS range are specified in the OSIV system runtime parameter then the application is executed.

*A: To the position in which the substitution character (_) is stored, unintended character is stored. The value depends on the state of the memory.

*B: Unicode native characters |
| 9 | V10.1.0 to V11.0.0 | PH08330 | Under the following conditions, during the execution of the COBOL program, the application terminates abnormally (EXCEPTION_ACCESS_VIOLATION).

The incidence depends on the state of the memory.

1. The ACCEPT statement (screen operation) or DISPLAY statement (screen operation) of the Micro Focus native function is used.

2. And, the data item is subscripted.

3. And, the data item of (2) is specified for the ACCEPT statement or DISPLAY statement of (1). |

*: V/L is a scope where the bug exists.

3.3 PowerFORM

Information described here applies to the following products.

- NetCOBOL Enterprise Edition V11.1.0 for Windows (64bit)

Table 3.3 Program fix list for the PowerFORM

No.	V/L(*)	P number	Content

1	V10.1.0 to V11.0.0	PH09964	Under the following conditions, an error of "44" or "09" might occur in the preview or the print functions. After the error occurs, the following occurs:
			
			 - For the preview, the print preview window is not output.
			
			 - For the print function, the image portion is not output and form is ejected.
			
			 1. The print or the preview is done.
			
			 2. And, either of the following is specified.
			
			     - Any of TIFF, PNG, JPEG, and GIF is specified for the bitmap field.
			
			     - Any of an expansion/reduction print, a compact print, and the LP mode print is done.
			
			 The output might not be generated whenever the above-mentioned requirements are met, and then the output can occur normally.

*: V/L is a scope where the bug exists.

3.4 PowerBSORT

Information described here applies to the following products.

- NetCOBOL Enterprise Edition V11.1.0 for Windows (64bit)

Table 3.4 Program fix list of PowerBSORT

NO.	V/L(*)	P number	Content of change

1	V7.0.0	PH05442	Under the following conditions, PowerBSORT cannot correctly check whether the reconstruction field and the selection field exist in the record.
			 As a result of an error in usage or implementation, the output may produce unpredictable or abnormal results.
			
			 1. The bsortex command is used.
			
			 2. And, the record format is one of the following.
			
			     - Binary file variable-length record form (-record recform=var), or
			
			     - Text file fixed field specification (-record recform=txtfix)
			
			 3. And, two output file information options (-output) or more are specified.
			
			 4. And, the following options are specified by two output file information options (-output) or more.
			
			     - Record reconstruction option (-output reconst=...)
			
			     - Record selection option (-output include=.../omit=.../case=...)
			
			 5. And, the field in the record is specified for a reconstruction field for the record reconstruction option (form of "pos.len" or "pos.END").
			
			 6. And, the maximum position of the reconstruction field specified by each output file information option (-output) is different. or,
			 The maximum position of the selection field specified by each output file information option (-output) is different.

2	V7.0.0	PH07992	Under the following conditions, PowerBSORT might loop infinitely. 1. Sort function is executed.

NO.	V/L(*)	P number	Content of change
			2. And, when the use of a temporary file is required (When the input data size is larger than the usable memory size).

*: V/L is a scope where the bug exists.

3.5 J Adapter Class Generator

Information described here applies to the following products.

- NetCOBOL Enterprise Edition V11.1.0 for Windows (64bit)

Table 3.5 Program fix list for the J Adapter Class Generator

No.	V/L(*)	P number	Content
1	V10.2.0 to V11.0.0	PH02733	Under the following conditions, the last character of the class name or interface name of a COBOL source code that is generated by the java2cob command might become a hyphen, and the following error message be generated. `JMN1123I-S An invalid word '-' was found. The invalid word and any following code is ignored until the next valid paragraph, section or division.` - The 28th character of a Java class name (*A) that generates an adapter class is a period or a dollar sign. Or - The 28th character of a Java interface name (*A) that generates an adapter class is a period or a dollar sign. *A: The package name is excluded.
2	V10.1.0 to V11.0.0	PH04723	Under the following environment and conditions, a character string that is returned from a Java class is converted into the alternative character "?" when the getStringX method or getStringN method of the java.lang.String class is used. [Environment] NetCOBOL for Windows is installed in any other Windows locales except for Japanese. [Condition] 1. ASCII code is used when an application that uses an adapter class is run. 2. And, when any of the following characters are gotten by using a method of the java.lang.String class. - cent sign: U+00A2 - pound sign: U+00A3 - not sign: U+00AC

*: V/L is a scope where the bug exists.

Chapter 4 Notes to consider when a Japanese native application is converted to run globally

This chapter describes what is involved in taking a Japanese native application global.

4.1 Assumption

The program resources and run time code-set are made by Unicode.

🔖 See

....
Refer to "Unicode" of "NetCOBOL User's Guide" for details.
....

4.2 Environments

There are following notes in the environments.

- This product must only be installed in a folder which has the only ASCII character set.

- When cobmkmf is used, the character outside ASCII cannot be used for the file name and the folder name of program resources.

4.3 Languages

The language used by this product (GUI, messages, etc.) is determined according to the setting of the display language of the execution environment of the Windows system.

By default, if the set display language cannot be used, English is selected as the default

The default of the display language of the user account can be set depending on the following criteria.

- For Windows 8.1 or later and beyond, Windows Server 2012 or later

  "Language" of Control Panel

- Windows 7 Enterprise/Ultimate, Windows Server 2008 R2

  "Keyboards and Languages" tab of "Region and Language" of Control Panel

- The other edition of Windows 7

  Note that for the other editions of Window 7, the display language default cannot be set by the Control Panel. Usually, the display language of the Windows system is the default of the display language of the user account.

Compilation messages

In this compiler, in accordance with the UI language during compilation, the COBOL command messages are set to Japanese or English, accordingly.

Runtime messages

The language of the Runtime Error Messages is decided when the runtime environment is established.

If there is no message of an applicable language, a Runtime Error Message is displayed in English.

NetCOBOL Studio

- When you use NetCOBOL Studio, define the applicable language of OS when the NetCOBOL product is installed, the system locale (This is set on the "Administrative" of "Region and Language" dialog box of the Control Panel), the display language (This is set on the "display language" of the Control Panel), and the language of the Format (This is set on the "Formats" of "Region and Language" dialog box of the Control Panel).

- Match the language of the client to the language of the server when you use the remote development function of NetCOBOL Studio. When the language of the client and the server is different, the message of the server might not be able to be displayed with NetCOBOL Studio of the client.

PowerFORM

When you use PowerFORM, define the same OS language for the system locale (This is set on the "Administrative" of "Region and Language" dialog box of the Control Panel) and the display language (This is set on the "display language" of the Control Panel) when the NetCOBOL product is installed.

4.4 Feature difference with Japanese version

In NetCOBOL, there are a Japanese version and a Global version. This product is a Global version.

This section explains the feature differences between a Japanese version and a Global version.

4.4.1 Related products

The following NetCOBOL family products are not supported.

- FORM

- MeFt/Web

The following Fujitsu products are not supported.

- Interstage Charset Manager

- Interstage List Works

- Interstage Business Application Server

- PowerRDBconnector

- INTARFRM

 Note
. . . .
For the following products, the name is different with the Japanese version.

Global version	Japanese version
PowerFORM RTS	MeFt
PowerBSORT	PowerSORT

. . . .

4.4.2 Specifications

The following specifications are different with the Japanese version.

Currency symbol

In this compiler, currency sign of OS is used as Currency Symbol character.

If character other than the currency sign of OS is used, the method of specifying may differ based on the character intended to use.

- Characters with same code as $(X'24') or \(X'5C')

    Specified with/based on compile option CURRENCY.

- 1 byte character other than this

  CURRENCY SIGN clause is used, specified inside the source program.

 Note
. . . .

- Characters consisting of multiple bytes cannot be used as currency symbol character. For possible characters that can be used as currency symbol, please refer "CURRENCY SIGN clause" of COBOL Language Reference.

- If currency sign of OS cannot be used as currency symbol, in that case \ (X'5C') will be taken as currency symbol.

- If data is passed between compilation units with different currency symbols, results might not be as expected. Specify the compilation option CURRENCY or CURRENCY SIGN clause to use the same currency symbol.

. . . .

Handling of National item spaces

In this compiler, National item space (Trailing Blank and Figurative Constant SPACE) of Unicode encoding is handled as alphabetic spaces (U+0020).

Specify compilation option NSP, if national spaces are to be changed.

 Note
. . . .

If data is passed between compilation units with different National item spaces, results might not be as expected. Specify the compilation option NSP, and use the same national item space.

. . . .

Printing font

If you use a print file without a FORMAT clause, a default font name according to the setting of the display language of the execution environment of the Windows system will be used.

When the display language is Japanese, the default font name is "MS Mincho, MS Gothic" and with any other language, the default font name is "Courier New".

In this scenario, the printing font value can be changed by the environment variables.

4.4.3 PowerFORM

- The following function is not supported in Global version.

    - Making of Form Descriptors (PXD) for Class Interface.

    - Selection of Target System of the overlay pattern table. Target System becomes Windows fixed.

    - Overlay conversion of Template Image.

- When PowerFORM is used, the only ASCII characters can be used.

- When Form Descriptors is edited, the initial values of the size of the form and the font-name, etc. are different.

4.4.4 PowerFORM RTS

When the Form is output in a Global version, it is necessary to specify the following.

- Specify Unicode for an application code.

See

Refer to "How to Use PowerFORM RTS" of "PowerFORM Runtime Reference" for details.

- Specify the font name by an English name.

See

Refer to "Appendix C. Printer Information File" of "PowerFORM Runtime Reference" for details.

    - MAPFONT (Map font names)

- Specify the printer information file to which BOM(UTF-8) is added.

See

Refer to "Chapter 5. Printer Information File" of "PowerFORM Runtime Reference" for details.

- Specify "USECHARTYPE UNI" with the printer information file. Adjust each character-code by "UNICODEN" and "UNICODEW" when the output result is not correct.

See

Refer to "Appendix C. Printer Information File" of "PowerFORM Runtime Reference" for details.

    - USECHARTYPE (Unicode full-width/half-width character table)

    - UNICODEW (Unicode full-width character range)

    - UNICODEN (Unicode half-width character range)

- The default value of the font name is different according to the locale.

 See

Refer to "Appendix C. Printer Information File" of "PowerFORM Runtime Reference" for details.

- MAPFONT (Map font names)