FUJITSU Software
NetCOBOL V11.1.0

Release Notes

Linux(64)

Preface

This book explains Release Note of NetCOBOL.

About the alias of the product

The name of the product described in this book is written for short as follows.

Product Name	Abbreviation
Red Hat(R) Enterprise Linux(R) 7 (for Intel64) Red Hat(R) Enterprise Linux(R) 6 (for Intel64)	Linux
Microsoft(R) Windows Server® 2016 Datacenter Microsoft(R) Windows Server® 2016 Standard	Windows Server 2016
Microsoft(R) Windows Server(R) 2012 R2 Datacenter Microsoft(R) Windows Server(R) 2012 R2 Standard Microsoft(R) Windows Server(R) 2012 R2 Essentials Microsoft(R) Windows Server(R) 2012 R2 Foundation	Windows Server 2012 R2
Microsoft(R) Windows Server(R) 2012 Datacenter Microsoft(R) Windows Server(R) 2012 Standard Microsoft(R) Windows Server(R) 2012 Essentials Microsoft(R) Windows Server(R) 2012 Foundation	Windows Server 2012
Microsoft(R) Windows Server(R) 2008 R2 Foundation Microsoft(R) Windows Server(R) 2008 R2 Standard Microsoft(R) Windows Server(R) 2008 R2 Enterprise Microsoft(R) Windows Server(R) 2008 R2 Datacenter	Windows Server 2008 R2
Windows(R) 10 Home Windows(R) 10 Pro Windows(R) 10 Enterprise Windows(R) 10 Education	Windows 10
Windows(R) 8.1 Windows(R) 8.1 Pro Windows(R) 8.1 Enterprise	Windows 8.1
Windows(R) 7 Home Premium Windows(R) 7 Professional Windows(R) 7 Enterprise Windows(R) 7 Ultimate	Windows 7
Microsoft(R) Visual C++(R) development system	Visual C++
Microsoft(R) Visual Basic(R) programming system	Visual Basic

Microsoft Windows products listed in the table above are referred to in this manual as "Windows".

Purpose of this documentation

This documentation explains the additional function, the bug fix and the interchangeable information from the old edition.

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

Documentation audience

This document is for customers who are using old edition of this products and intended for shift or shifting to this product.

Positioning of this documentation

This document explains only fragments of the additional function, the bug fix, and the interchangeable information from the old edition. For details and to get the overall look of the product, refer to each product manual.

Related manuals

The manual of this product and a related product includes the following manuals besides this document.

- NetCOBOL Language Reference

- NetCOBOL Syntax Samples

- NetCOBOL User's Guide

- NetCOBOL COBOL File Access Subroutines User's Guide

- NetCOBOL CBL Subroutines User's Guide

- NetCOBOL Messages

- PowerFORM Runtime Reference

- J Adapter Class Generator User's Guide

- PowerBSORT User's Guide

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 and Windows Server 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.

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	V11.0.0	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 is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 1.2 Outline of additional function of 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.2.16 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	ODBC support	The database access by ODBC is supported.	NetCOBOL User's Guide - Chapter 16 Database (SQL) - A.2.46 SQLGRP(SQL host variable definition expansion) NetCOBOL Language Reference - Chapter 8 Database (SQL)
3	V11.1.0	ltrace support	When the shared library is created specifying optional PLT, this library of the dynamic linking structure can be traced by using the ltrace command. Moreover, the processing time of the external program can be measured.	NetCOBOL User's Guide - A.2.36 PLT (Specifies to trace the external program calls that are included in a shared library) - Appendix M ltrace Command
4	V11.1.0	PRINTER_n support	In the ASSIGN clause, PRINTER_n phrase is supported. n for PRINTER-n is an integer from 1 through 99.	NetCOBOL Language Reference

N O.	V/L	Function name	Content	Location in Manual
			- 2 -	- 4.3.1.3 ASSIGN Clause (Sequential File, Relative File, and Indexed File) - 11.8.6.17 The WRITE Statement (Object-Oriented Programming)
5	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.	man manual of the cobmkmf command.
6	V11.1.0	CBL subroutines support	Support of CBL subroutines that are compatible with COBOL CBL subroutines from Micro Focus Co., Ltd.	NetCOBOL CBL Subroutines User's Guide
7	V11.0.0	UTF-32 support	Encoding form UTF-32 and Shift-JIS can now be used.	NetCOBOL User's Guide - Chapter 18 Unicode
8	V11.0.0	Hadoop support	The Hadoop integration function can now be used.	Hadoop Integration Function User's Guide
9	V11.0.0	Named pipes of COBOL File Access Routines	Named pipes of COBOL File Access Routines is supported.	COBOL File Access Subroutines User's Guide - 3.27 Named pipe

Table 1.3 Outline of additional function of NetCOBOL V10

NO.	V/L	Function name	Content	Description place of manual
1	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 - 6.8 Other File Functions - A.2.2 ARITHMETIC NetCOBOL Messages
2	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 - 1.3.5.2 Slack Bit - 2.1 Nucleus - 2.3 Input-Output Facility - Inter-program Communication Module

NO.	V/L	Function name	Content	Description place of manual
				- 2.3.3 External Attribute and Internal Attribute
				- 2.3.5 Program Activation
				- 2.3.7 Initial State of a Program
				- 2.3.8 Recursive Attribute of the Program
				- 3.1.1 Program-Id Paragraph
				- 4.3.1.1 ACCESS MODE Clause
				- 4.3.1.3 ASSIGN Clause (Sequential File, Relative File, and Indexed File)
				- 4.3.1.8 FILE STATUS Clause
				- 4.3.1.9 FORMAT Clause
				- 4.3.1.10 GROUP Clause
				- 4.3.1.23 PROCESSING MODE Clause
				- 4.3.1.30 SELECTED FUNCTION Clause
				- 4.3.1.32 SYMBOLIC DESTINATION Clause
				- 4.3.1.33 UNIT CONTROL clause
				- 5.1 Composition of the Data Division
				- 5.2.9 RECORD Clause
				- 5.4 Data Description Entry
				- 5.4.3 CHARACTER TYPE Clause
				- 5.4.13 SYNCHRONIZED Clause
				- 5.4.17 VALUE Clause
				- 6.4.7 CALL Statement
				- 8.2.1 Embedded SQL Declare Section
				- 10.2.3 Data Description Entry
				- Appendix A List of Reserved Words
				NetCOBOL User's Guide
				- 8.1 Calling Relationship Types
3	V10.4.0	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
				- 4.1.1 Runtime Environment
				- 6.8.1.2 High-speed File Processing - Batch specification

NO.	V/L	Function name	Content	Description place of manual
				NetCOBOL Messages
4	V10.4.0	Syslog output	The Syslog can be specified as the output destination for DISPLAY statements.	NetCOBOL User's Guide - 4.1.1 Runtime Environment - 9.1 ACCEPT/DISPLAY Function - 9.1.9 Programs Using Syslog - Appendix E Environment Variable List 　- CBR_DISPLAY_CONSOLE _SYSLOG_LEVEL 　- CBR_DISPLAY_CONSOLE _SYSLOG_IDENT 　- CBR_DISPLAY_CONSOLE _OUTPUT 　- CBR_DISPLAY_SYSERR_S YSLOG_LEVEL 　- CBR_DISPLAY_SYSERR_S YSLOG_IDENT 　- CBR_DISPLAY_SYSERR_O UTPUT 　- CBR_DISPLAY_SYSOUT_S YSLOG_LEVEL 　- CBR_DISPLAY_SYSOUT_S YSLOG_IDENT 　- CBR_DISPLAY_SYSOUT_ OUTPUT
5	V10.4.0	CHECK function suppress option	Runtime options are provided to suppress individual CHECK functions. 　- nocb 　- noci 　- nocn 　- nocp	NetCOBOL User's Guide - 4.2.2 Specifying Runtime Options - 4.2.2.2 [ccount | { noc | nocb | noci | nocn | nocp }] - 5.3.2 Output Messages
6	V10.4.0	TRACE function suppress option	Runtime option is provided that suppresses the TRACE function. 　- nor	NetCOBOL User's Guide - 4.2.2 Specifying Runtime Options - 4.2.2.1 [r count | nor] (Set the trace data limit, and suppress the TRACE function)
7	V10.4.0	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 User's Guide - 4.1.1 Runtime Environment - 5.2.2 Trace Information

1.3 PowerFORM

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 1.4 Outline of additional function of PowerFORM V11

NO.	V/L	Function name	Content	Location in Manual
1	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
2	V11.0.0	Form exporting	The Form can now be outputted to PDF.	PowerFORM Runtime Reference - Chapter 4. Extended Functions - Appendix C. Printer Information File
3	V11.0.0	UTC (Coordinated Universal Time) support	UTC (Coordinated Universal Time) can be specified.	PowerFORM Runtime Reference - Appendix C. Printer Information File
4	V11.0.0	Enhanced print features	The Form Descriptor (free frame) can be used.	PowerFORM Runtime Reference - Chapter 4. Extended Functions
5	V11.0.0	Enhanced print features	The new function added in PowerFORM V11 can be used.	PowerFORM Runtime Reference - What's New in PowerFORM V11?

1.4 PowerBSORT

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Outline of additional function of PowerBSORT V7

NO.	V/L	Function name	Content	Location in Manual
1	V7.0.0	Data format	The Unicode UTF-32 form was supported as data format.	User's Guide - 1.4 Environment variables and Startup file - 1.6.3 Data format - 1.6.4 Data forms that can be specified in each field - 3.2.16 Input code system option (-q) - 3.2.26 COBOL file index specification option (-X) - 4.2.7.3 icode operand - 4.2.8.10 idxkey operand - 6.5.1 BSRTPRIM structure

NO.	V/L	Function name	Content	Location in Manual
				- 6.5.10 BSIDXKEY structure
2	V7.0.0	Record summation option	The length of data formats that can be specified for the binary file has been enhanced	User's Guide - 1.6.4.2 Data formats that can be specified in the summation field
3	V7.0.0	Record selection option	The length of data formats that can be specified by a literal value has been enhanced.	User's Guide - 1.6.4.3 Data formats that can be specified in the selection field
4	V7.0.0	Record reconstruction option	The length of data formats that can be specified by a literal value has been enhanced.	User's Guide - 1.6.4.4 Data formats that can be specified in the literal value of the reconstruction field
5	V7.0.0	File types	The physical sequential file format supported by COBOL85 that operated on the Fujitsu mainframe was supported.	User's Guide - 1.4.2 Startup file - 1.5 PowerBSORT Input Output Environment - 3.2.3 I/O file system option (-F) - 4.2.5.3 filesys operand - 6.5.8 BSFSYS structure - B.8 Important Notes about the NetCOBOL File system

Outline of additional function of PowerBSORT V6

NO.	V/L	Function name	Content	Location in Manual
1	V6.1.0	Data format	The external decimal number of Micro Focus COBOL form was supported as data format.	User's Guide - 3.2.16 Input code system option (-q) - 4.2.7.3 icode operand - 6.5.1 BSRTPRIM structure

1.5 J Adapter Class Generator

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Outline of additional function of J Adapter Class Generator V11

NO.	V/L	Function name	Content	Location in Manual
1	V11.0.0	Encoding form Shift-JIS support.	Encoding form Shift-JIS can now be used.	J Adapter Class Generator User's Guide

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 is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

2.1.1 Enhanced compiler check of a national item, national edited item and NATIONAL function

Content

- V11.0 or earlier

  In the following locales, the COBOL compiler is able to compile the program that contains the national item, national edited item and NATIONAL function as described.

  - C

- V11.1 or later

  In the following locales, when the COBOL compiler compiles the program that the national item, national edited item and NATIONAL function are described, the following message is output.

  - C

  ```
  JMN1859I-S
  '@1@' cannot be specified in the coded character set being used. '@1@' is ignored.
  ```

  ```
  Parameter explanation
  @1@: national item, national edited item or NATIONAL function
  ```

Action

  You will modify the source in the following way.

  - Remove or rewrite the cause of the error (national item, national edited item and NATIONAL function).

Note

J Adapter Class Generator

Java-lang-String class generated with the java2cob command (in V11.0 or earlier) contains the national item.

In the locale environments other than xx_XX.UTF-8, when this adapter class is compiled by using the compiler V11.1 or later, the above error is output. In this case, you should regenerate the adapter class by using java2cob command (in V11.1 or later).

In the locale environments other than xx_XX.UTF-8, if you use the adapter class, the adapter class should be the one generated with the following conditions.

- Use the java2cob command with -cs option

- Specify "Option Code=SJIS" in the optional file, and use the java2cob command.

2.1.2 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.

2.1.3 Interchangeable information regarding bug fixes

Here, it explains bug fixes corrected by NetCOBOL V10.1.0 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 of change
1	V10.1.0 to V10.4.0	PG76651	Under the following conditions, when a COBOL program is executed, 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 to V10.4.0	PH02265	Under the following conditions, a compilation error may not be generated.
			
			1. Concatenation expression is described in the external name of program, class or method.
			
			2. And, one of the following is described at the beginning of the concatenation expression.
			  - Character constant
			  - National character constant
			
			3. And, when one of the following is described in the concatenation expression.
			  - Hexadecimal character constant
			  - National hexadecimal character constant

3	V10.1.0	PG77383	Under the following conditions, the execution results are incorrect.
			
			1. The OPTIMIZE compiler option is specified (*A).
			
			2. And, there is a statement that sets packed decimal item to numeric edited data item.
			
			3. And, the number of digits for integer part of packed decimal item and numeric edited data item are the same and neither have a decimal-part.
			
			4. And, the numeric edited data item specifies only zero suppression for the edit (Only '9', 'Z', and '*' are used for PICTURE character-string).
			
			5. And, before the statement in 2 (*B), there is a statement that sets data item (or intermediate result) to packed decimal item.
			
			6. And, 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</pre>

NO.	V/L(*)	P number	Content of change
| | | | ```
                   4                              5
                   6                              7
                   8                              9
                  10                             11
                  12                             13
                  14                             15
                  16                             17

         -------------------------------------------------
``` |
| | | | 7. And, the data item in 5 (or intermediate result) and the packed decimal item in 2 don't have the decimal-part.

*A: The default is NOOPTIMIZE.

*B: MOVE statement, COMPUTE statement, and so on. |
| 4 | 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]

      - FUNCTION MAX

      - FUNCTION MIN

      - FUNCTION MEAN

      - FUNCTION MEDIAN

      - FUNCTION RANGE

    [B]

      - FUNCTION ANNUITY

      - FUNCTION NUMVAL

      - FUNCTION NUMVAL-C

      - FUNCTION RANDOM

  3. And, the arguments of the function in 2) above are fixed point of 9 digits or less.

  4. And, the number of the arguments in [A] above is 4 or more. |
| 5 | V10.1.0 | PG75097 | Under the following conditions, the COBOL application running under Red Hat Enterprise Linux (for Intel64) cannot be remotely debugged from the NetCOBOL Studio of Windows edition. "Error occurred in debugger or engine. (65535)" is displayed.

  1. The server is Red Hat Enterprise Linux (for Intel64) or Red Hat Enterprise Linux (for Itanium).

  2. And, installed the following products in the server.

    - Alchemy NetCOBOL Enterprise Edition V10.1.0 for Linux (64bit)

  3. And, gdb used on the server is gdb-7.0(*) or later.

*: Red Hat Enterprise Linux 5.5 is installed or gdb-7.0 is installed alone. |
| 6 | 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, |

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

			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.

7	V11.0.0	PH04570	Under the following [Conditions], the following [Phenomenon 1] or [Phenomenon 2] might occur.
			
			[Phenomenon 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
			
			[Phenomenon 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

NO.	V/L(*)	P number	Content of change
			- 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.
8	V10.1.0 to V11.0.0	PH05538	Under either of the following conditions, the COBOL program execution result is incorrect or terminates abnormally (Segmentation Fault). [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 OPTIMIZE. [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.
9	V10.1.0 to V11.0.0	PH05861	Under the following conditions, the following execution results are incorrect. - When moving from reference modification of an zoned decimal item to numeric-edited items or floating-point items, the reference modification range of the sending side is shifted 1-digit to the left. 1. There is a MOVE statement. (*A) - The sending side is a signed zoned decimal item - The receiving side is a numeric-edited item or floating-point item 2. And, in the sending side of 1, a SIGN IS TRAILING SEPARATE CHARACTER clause is being specified. 3. And, there is a reference modification of the sending side. 4. And, the length of reference modification of 3 is being specified with a numeric literal. *A: Includes the implicit MOVE statement. Example:

```
WORKING-STORAGE SECTION.
01  SND    PIC S9(13) SIGN TRAILING SEPARATE
                          VALUE -0098765432100.
01  RCV    PIC Z(10)9.
```

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

```
PROCEDURE DIVISION.
    MOVE SND (3:11) TO RCV. *> Sending value is 98765432100
    DISPLAY "RCV= " RCV.
```

```
Result:
  RCV=  9876543210
```

NO.	V/L(*)	P number	Content of change
10	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.

Supplement)

  - 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.

[Example of condition 1]

```
-calling program--------------
 :
WORKING-STORAGE SECTION.
```

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

```
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]
```

```
VAR=000  <-  A correct result is "   ".
```

*: 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 Linux (64bit)

2.2.1 About the BOM of the UTF-8 file

Content

Operation differs when input the UTF-8 file with BOM.

- Before V11.0

  BOM is read as part of the data when it is included in the UTF-8 file.

  When a valid keyword is described at the first line of the input file, but the file is read with BOM added at the beginning of the keyword, the keyword might be invalid.

- After V11.0

  BOM is identified as UTF-8 identification code and it is skipped.

  When a valid keyword is described at the first line of the input file, the keyword is considered as valid as the BOM in the beginning of the file is skipped.

Influence

When the input resource file is UTF-8 with BOM, the content described in the first line is valid after V11.0 however the content described in the first line is invalid before V11.0.

There is no change in the operation of the application when the first line is a comment line or null line. The input resource when executing it becoming an object is as follows.

- Runtime initialization file

- Entry information file

- Class information file

- Print information file

- Printer information file

Action

Please delete or make the first line a comment.

2.2.2 Process ID notified by the subroutine (COB_GET_PROCESSID)

Content

The process ID notified by the subroutine (COB_GET_PROCESSID) was changed.

- V11.0 or earlier

  When an executable file was started from the shell script and service, etc., process group ID (process ID in the process that had started first) was notified.

  Example: When an executable file is started from the shell script, process ID of the shell script is notified.

- V11.1 or later

  When an executable file was started from the shell script and service, etc., process ID (of the process that calls this subroutine) is notified.

2.2.3 Interchangeable information regarding bug fixes

Here, it explains bug fixes corrected by NetCOBOL V10.1.0 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 of change
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	PG72597	Under the following conditions, the WRITE AFTER ADVANCING PAGE is not valid at execution time. 1. The line sequential file is used. 2. And, the line sequential file is linked with the external file handler. 3. And, after an OPEN statement is executed for file in 2, the WRITE statement of the AFTER ADVANCING PAGE phrase is executed.

3	V10.1.0	PG73346	Under the following conditions, at 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/JMP0320I-U
			
			1. File operation using COBOL file system (except BSAM specification) is done.
			
			2. And, either of the following I/O statements is executed.
			
			  - READ statement
			
			  - WRITE statement
			
			  - REWRITE statement
			
			  - DELETE statement
			
			  - START statement
			
			3. And, the execution of I/O statement failed.

4	V10.1.0	PG75097	Under the following conditions, the COBOL application running under Red Hat Enterprise Linux (for Intel64) cannot be remotely debugged from the NetCOBOL Studio (Windows) edition. "Error occurred in debugger or engine. (65535)" is displayed. 1. The server is Red Hat Enterprise Linux (for Intel64) or Red Hat Enterprise Linux (for Itanium). 2. And, installed the following products in the server. - Alchemy NetCOBOL Enterprise Edition V10.1.0 for Linux (64bit) 3. And, gdb used on the server is gdb-7.0(*) or later. *: Red Hat Enterprise Linux 5.5 is installed or gdb-7.0 is installed alone.
5	V10.1.0	PG87520	Under the following conditions, the MOVE result by internal Bool item is incorrect. Note: The move result is different according to the state of the area allocated just behind the sending data item.

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

			[Condition 1]
			
			1. Move from internal Boolean item to internal Boolean item.
			
			2. And, data start position (*) of the sending data item is different from data start position of the receiving data item.
			
			3. And,
			[a] + [b] > 32, or [c] + [d] > 32
			
			a: Data start position (*) of the sending data item
			
			b: The length of sending data item or the length of receiving data item, whichever is smaller.
			
			c: Data start position (*) of the receiving data item
			
			d: The length of receiving data item
			
			4. And, the receiving data item is subordinate to the group item.
			
			5. And, data start position of the receiving data item is not one byte alignment of the group item in 4.
			
			6. And, "Bit length of sending data item" < " Bit length from bit data starting location on the receiving side to the first byte boundary "
			
			7. And, bit data of sender data item is outside the byte boundary or Byte boundary exist between the start position of the sender's data item of paragraphs 6(*) (bit length till the byte boundary of the first recipient).
			
			*: relative bit position in byte
			
			Example of Condition 1
			
			<pre> DATA DIVISION.
			      WORKING-STORAGE SECTION.
			      01 DATA1.
			        02 DATA1-1 PIC 1(5) BIT.
			        02 DATA1-2 PIC 1(4) BIT.
			        02 DATA1-3 PIC 1(7) BIT.
			      01 DATA2.
			        02 DATA2-1 PIC 1(1) BIT.
			        02 DATA2-2 PIC 1(32) BIT.
			      01 ANS-DATA PIC 1(32) BIT VALUE B"1111".
			      PROCEDURE DIVISION.
			          MOVE X"FFFF" TO DATA1.
			          MOVE DATA1-2 TO DATA2-2.
			          IF DATA2-2 = ANS-DATA
			            THEN DISPLAY "OK"
			            ELSE DISPLAY "NG" *> B"11111110-00000000"
			          END-IF.</pre>
			[Condition 2]
			
			1. Move from internal Boolean item to internal Boolean item.
			
			2. And, the sending data item or the receiving data item is referred by the subscript.
			
			3. And, "The length of receiving data item " > 1
			
			4. And, it is either the following.
			
			  - "The length of sending data item" > 1
			
			  - The receiving data item is referred by the subscript.
			
			  - " The length of receiving data item " > 25

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

			5. And, There is not the data starting location of the receiving data item on byte boundary.
			
			6. And, "Bit length of sending data item" < " Bit length from bit data start position of first byte of the receiving data item
			
			7. And, the bit data of the sending data item overlaps the byte boundary, or there is byte boundary between from start position of the sending data item to "Bit length from bit data start position of the receiving side to the first byte boundary" in 6.
			
			Example of Condition 2
			
			<pre> DATA DIVISION.
			            WORKING-STORAGE SECTION.
			            01 DATA1.
			              02 DATA1-1 PIC 1(5) BIT.
			              02 DATA1-2 PIC 1(4) BIT.
			              02 DATA1-3 PIC 1(7) BIT.
			            01 DATA2.
			              02 DATA2-1 PIC 1(10) BIT OCCURS 5 TIMES.
			            01 ANS-DATA PIC 1(10) BIT VALUE B"1111".
			            01 CNT PIC 9 VALUE 2.
			            PROCEDURE DIVISION.
			                MOVE X"FFFF" TO DATA1.
			                MOVE DATA1-2 TO DATA2-1(CNT).
			                IF DATA2-1(CNT) = ANS-DATA
			                  THEN DISPLAY "OK"
			                  ELSE DISPLAY "NG" *> B"11111110-00000000"
			                END-IF.</pre>

6	V10.4.0 to V11.0.0	PH08017	Under the following conditions, the data cannot be read from the input file of the redirection specification by the execution of the ACCEPT statement. 1. Compile option RCS(SJIS) is specified. 2. The ACCEPT statement of the system standard input is executed by the redirection specification. 3. And, the line feed code is not included in the specified input file.

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

2.3 PowerFORM

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

There is no information on interchangeability.

2.4 PowerBSORT

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (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	V6.0.0	PG76059	Under the following conditions, PowerBSORT may cause result error, infinite loop or abnormal termination. 1. The bsort command, the bsortex command or the BSORT function is used. 2. And, sort function is used. 3. And, the text file CSV format or text file TSV format is used. 4. And, the record summation option or the record selection option for the output files is specified. 5. And, the summation field or selection field is enclosed with a double quotation.
2	V6.0.0	PG76539	Under the following conditions, PowerBSORT may cause result error or abnormal termination. 1. The bsortex command is used. 2. And, the text file CSV format or text file TSV format is used. 3. And, the record reconstruction option for the output files is used.
3	V6.0.0 to 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 is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

There is no information on interchangeability.

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 is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 3.1 Program fix list of NetCOBOL Development Environment

NO.	V/L(*)	P number	Content of change
1	V10.1.0 to V11.0.0	PG81806	Under the following conditions, the COBOL compiler terminates abnormally. 1. Any of the following statements is described. - CALL STATEMENT - CANCEL STATEMENT - INVOKE STATEMENT - STRING statement/UNSTRING statement specified with a CSV-FORMAT 2. And, the statement in 1 above is being described in the conditional statement. 3. And, the hierarchy (*A) of the conditional statement in 2 above is more than 180. *A: In the following code, hierarchy 1 is the EVALUATE statement and hierarchy 2 is the COMPUTE statement. However, depending on the amount of computer memory, there may be no abnormal termination.
2	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 shared object file 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. - Subprogram is called from the CALL statement for which the data-name is specified.

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

3	V11.0.0	PH03777	Under the following conditions, the COBOL compiler terminates abnormally.
			
			    1. There is either of the following items.
			        - Alpha Numeric 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 either of the following move statements.
			        - MOVE statement
			        - 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

4	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.

5	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 ``` System error 'errno=0xXXX' occurred on 'open REP,xxx'. ```

NO.	V/L(*)	P number	Content of change
| | | | ```
JMN0013I-S The repository file cannot be opened. The object
file cannot be used.
``` |

- 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 : 320bytes

    b. Text-name literal: 240bytes(*)

*: 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.

NO.	V/L(*)	P number	Content of change
6	V11.0.0	PH04570	Under the following [Conditions], the following [Phenomenon 1] or [Phenomenon 2] might occur.

[Phenomenon 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

[Phenomenon 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

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

			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
		PH05538	Under either of the following conditions, the COBOL program execution result is incorrect or terminates abnormally (Segmentation Fault).
			
			[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 OPTIMIZE.
			
			[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.

8	V10.1.0
	to
	V11.0.0
		PH05861	Under the following conditions, the following execution results are incorrect.
			
			- When moving from reference modification of an zoned decimal item to numeric-edited items or floating-point items, the reference modification range of the sending side is shifted 1-digit to the left.
			
			1. There is a MOVE statement. (*A)
			
			   - The sending side is a signed zoned decimal item
			
			   - The receiving side is a numeric-edited item or floating-point item
			
			2. And, in the sending side of 1, a SIGN IS TRAILING SEPARATE CHARACTER clause is being specified.
			
			3. And, there is a reference modification of the sending side.
			
			4. And, the length of reference modification of 3 is being specified with a numeric literal.
			
			*A: Includes the implicit MOVE statement.

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

```
WORKING-STORAGE SECTION.
01   SND    PIC S9(13) SIGN TRAILING SEPARATE
                              VALUE -0098765432100.
01   RCV    PIC Z(10)9.
PROCEDURE DIVISION.
    MOVE SND (3:11) TO RCV. *> Sending value is 98765432100
    DISPLAY "RCV= " RCV.
```

```
Result:
  RCV=  9876543210
```

| 9 | 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

*: End position depicts the location from start of the record to the offset + Item length.

```
 Record including the item of 1.
```



```
 *:End position of  Item 1
```

```
 **:End position of the record including the items of 1 item
```
 |

NO.	V/L(*)	P number	Content of change
10	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.

Supplement)

- 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.

[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.
-----------------------------
```

NO.	V/L(*)	P number	Content of change
| | | | ```
-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]
``` |
| | | | ```
VAR=000  <-  A correct result is "   ".
``` |
| 11 | V10.1.0 to V11.0.0 | PH08595 | Under of 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 |

NO.	V/L(*)	P number	Content of change
			*A: Default is OPTIMIZE.
12	V11.0.0	PH10340	Under of 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.

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

3.2 NetCOBOL Runtime Environment

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 3.2 Program fix list of NetCOBOL Runtime Environment

NO.	V/L(*)	P number	Content of change
1	V11.0.0	PG99231	Under the following conditions, the opening of the file fails in the execution of the COBOL file access subroutine. - Error Number: FA_EBADFILE(105) - I-O Status : 90 1. The COBOL file access subroutine reads and writes the index file. 2. The index key is UTF-32 little endian in the national data item. 3. As for API function (*A), the NULL pointer is specified for the record key list. (*B) 4. The API function in 3 is executed. *A: cobfa_open or cobfa_openW *B: If the NULL pointer is specified for the record key list to open an indexed file, this function automatically identifies the key list structure, record format, and record length of an existing file, and opens the file.
2	V11.0.0	PH05486	Under the following conditions, the message of the Hadoop integration function (*A) might be output to the Reduce task log, and the output record from the Reduce application might be lost. (*B) *A: EX0084: Job has been killed by the user. Attempted to write out the input to the command. *B: Number of lost records is less than the number of output error messages. 1. Satisfied either of the following requirements. - Reduce application is used. - Auto sort specification for the Map output file is specified. 2. And, two or more Reduce input files are used. 3. And, the number of records of either of Reduce input files is more than 1000 records.
3	V10.4.0 to V11.0.0	PH08017	Under the following conditions, the data cannot be read from the input file of the redirection specification by the execution of the ACCEPT statement. 1. Compile option RCS(SJIS) is specified.

NO.	V/L(*)	P number	Content of change
			2. The ACCEPT statement of the system standard input is executed by the redirection specification.
			3. And, the line feed code is not included in the specified input file.
4	V10.1.0 to V11.0.0	PH08039	Under the following conditions, COBOL application terminates abnormally and outputs the following message.
			``` JMP0099I-U FORCED TERMINATIONCODE=9999 ```
			1. In COBOL source, there is a code that receives an OSIV system runtime parameter.
			2. And, compiler options MAIN and RCS(SJIS) are specified.
			3. And, the main program is compiled by NetCOBOL for Linux(x64) V10.4.
			4. And, the runtime system at execution time is V11.0.
			5. And, the environment variable CBR_CONVERT_CHARACTER is not specified, or CBR_CONVERT_CHARACTER=ICONV is specified.
			6. And, COBOL application is executed specifying the OSIV system runtime parameter.
5	V10.1.0 to V11.0.0	PH10544	Under the following [Conditions], the following [Phenomenon 1] or [Phenomenon 2] might occur.
			[Phenomenon 1]
			Under the following conditions, the opening of the file fails in the execution of the COBOL file access subroutine.
			- Error Number: FA_EOTHER(999)
			- I-O Status : 90
			[Phenomenon 2]
			Under the following conditions, the OPEN statement fails and outputs the following the runtime message.
			``` JMP0400I-U ACCESS ENVIRONMENT ERROR OCCURRED. ```
			[Conditions]
			In case of phenomenon 1:
			1. The OPEN statement of COBOL file is executed by large capacity file specified (*).
			2. And, in the same process as 1, the cobfa_open function of COBOL file access routine was executed.
			In case of phenomenon 2:
			1. The cobfa_open function of COBOL file access routine was executed.
			2. And, in the same process as 1, the OPEN statement of COBOL file is executed by large capacity file specified.
			*: In assigning files, the character string "LFS" or "LBSAM" is specified following the file name.

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

3.3 PowerFORM

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 3.3 Program fix list of PowerFORM

NO.	V/L(*)	P number	Content of change
1	V11.0.0	PH11272	Under the following conditions, the wrong time is output to the time field of the form. 1. The time field is set to the form. 2. And, the time difference from UTC (Coordinated Universal Time) is specified by key word "TIMEZONE" of the printer information file. 3. And, two or more forms are output at the same time.

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

3.4 PowerBSORT

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Table 3.4 Program fix list of PowerBSORT

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

1	V6.0.0 to 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	V6.0.0 to V7.0.0	PH07992	Under the following conditions, PowerBSORT might loop infinitely. 1. Sort function is executed. 2. And, when the use of a temporary file is required (When the input data size is larger than the usable memory size).
3	V6.0.0 to	PH10071	Under the following conditions, PowerBSORT outputs the error message "Error occurred during the file (%s1) writing.(%s2)" and terminates abnormally. %s1: Temporary file name.

NO.	V/L(*)	P number	Content of change
	V7.0.0		%s2: 122 is output for the Linux version. The sorting processing cannot be continued by using temporary files other than the temporary file that exceeds capacity when two directories or more that make the temporary file or two temporary files or more are specified. 1. Sort function is specified. 2. And, the capacity of the directory that makes the temporary file is limited by quota. 3. And, when the capacity of the disk used exceeds the capacity limitation by the output to the temporary file.
4	V6.0.0 - V7.0.0	PH10293	Under the following conditions, PowerBSORT outputs the error message "Error occurred during the file (%s1) writing.(%s2)" and terminates abnormally. %s1: Output file name. %s2: 122 is output for the Linux version. When two output files or more are specified, and the output file exceeds the available size of the file system, the output file cannot be switched. 1. Two output files or more are specified. 2. And, the capacity of the directory that makes the output file is limited by quota. 3. And, when the capacity of the disk used exceeds the capacity limitation by the output to the output file.

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

3.5 J Adapter Class Generator

Information described here is applied to the following products.

- NetCOBOL Enterprise Edition for Linux (64bit)

Program fix list of J Adapter Class Generator

NO.	V/L(*)	P number	Content of change
1	V11.0.0	PH11308	Under the following conditions, the java2cob command outputs a non-english message. 1. The japanese locale different from the environment variable LC_CTYPE is set in the environment variable LC_MESSAGE. 2. And, the java2cob command is executed. 3. And, when the adapter class is generated, a Japanese message is output.

*: 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 Guideline

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 Languages

Language used in the messages for compiler, runtime, tools etc., can be selected in accordance with the current locale setting.

Setting of the current locale is valid for the value of the first environment variable detected in the following order.

1. LC_ALL

2. LC_MESSAGES

3. LANG

Messages for the following Japanese locales are provided with this product.

- ja_JP.UTF-8

Messages are generated in English if the specified locale is not supported.

4.3 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.

4.3.1 Related products

The following NetCOBOL family products are not supported.

- FORM

- FORM Overlay Option

- MeFt/Web

- SIMPLIA

The following Fujitsu products are not supported.

- Interstage Charset Manager

- Interstage List Works

- Interstage Business Application Server

- Interstage JOB Workload Server

- Interstage Big Data Parallel Processing Server

- PrintWalker/LXE

- PrintWalker/LXE-EX

- INTARFRM

🔔 Note

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

Global version	Japanese version
PowerFORM RTS	MeFt
PowerBSORT	PowerSORT

4.3.2 Specifications

The following specifications are different with the Japanese version.

4.3.2.1 Currency symbol

In this compiler, $ (x'24') is used as currency symbol.

When any other character is used, the following method is used.

- Specify with the compile option CURRENCY.

📚 See

Please refer "A.2.10 CURRENCY (currency symbol handling)" of NetCOBOL User's Guide.

- Specify in the source program by using CURRENCY SIGN clause.

📚 See

Please refer 4.2.3.7 CURRENCY SIGN clause" of "NetCOBOL Language Reference."

🔔 Note

- Character composed of multiple bytes cannot be used as currency symbol. Regarding characters that can be used as currency symbol, please refer "4.2.3.7 CURRENCY SIGN clause" of "NetCOBOL Language Reference."

- If data is passed between compilation units having different currency symbol, the result might differ from the intended/expected result. Please specify compile option CURRENCY or CURRENCY SIGN clause and use same currency symbol character.

4.3.2.2 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.

📚 See

Refer to "NSP(Handling of spaces related to national item)" of "NetCOBOL User's Guide" for details.

> **📒 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.
> ···

4.3.2.3 Printing

It is recommend that PowerFORM Runtime use PDF file output by using the form descriptor created with PowerFORM when developing both a Japanese version and a global version.

The difference between a Japanese version and a global version is shown as follows.

Please consult "4.3.1 Related products" regarding supported situations for the related products.

Data stream type

Data stream type UVPI is not supported in a global version however it is supported in a Japanese version.

> **📑 See**
> ···
> Refer to "4.3.1 Related products".
> ···

Paper size

The default value of the size of the form is A4 in a Japanese version. LETTER is used in a global version.

When the size of the form is changed, it is specified on the print information file or I control record.

Font for print

The default value of the print font is Minchou/Gothic in a Japanese version. COURIER is used in a global version.

When the print font is changed, the font table is used.

Characters that can be written

It is a range of Unicode and SJIS in a Japanese version. ASCII is used in a global version.

Form descriptor

It corresponds to SMD and PMD in a Japanese version. Only PMD is used in a global version.

> **📑 See**
> ···
> Refer to "4.3.1 Related products".
> ···

4.3.3 PowerFORM RTS

When the Form is output in a Global version, there are the following notes.

- Only ASCII character can be used.

- When the paper size is omitted, it is printed on LETTER size.

- Only the print to the PostScript printer and the PDF output are possible.