

FUJITSU Software

NetCOBOL V11.0

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

CGI Subroutines User's Guide

Windows

B1WD-3361-01ENZ0(00)
August 2015

Preface

Purpose of this manual

This manual describes how to create, execute, and debug COBOL programs that use the NetCOBOL CGI subroutines 3.1.

Also, refer to the *NetCOBOL User's Guide* for explanations of the creation, compilation, and execution of COBOL programs.

For the debugging of COBOL programs, refer to the *NetCOBOL Debugging Guide*.

For an explanation of COBOL syntax rules, refer to the *NetCOBOL Language Reference*.

To create Web applications that will operate with Unicode, refer to the *Appendix B* in the *Web Guide*.

Supported Environments

The *NetCOBOL CGI subroutines 3.1* are 32-bit applications that operate on WWW servers in the 32-bit mode of the following systems:

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Trademarks

The following trademarks and registered trademarks are used in this manual:

- 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 registered trademarks of Microsoft Corporation in the United States and other countries.
- Netscape, Netscape Navigator, and Netscape Enterprise Server are trademarks or registered trademarks of Netscape Communications Corporation in the United States and other countries.
- Other brand and product names are trademarks or registered trademarks of their respective owners.
- The use of screens herein is permitted by Microsoft Corporation of the United States.

Product names

The names of products described in this manual are abbreviated as follows:

Product Name	Abbreviation
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 Foundation Microsoft® Windows Server® 2008 R2 Standard Microsoft® Windows Server® 2008 R2 Enterprise Microsoft® Windows Server® 2008 R2 Datacenter	Windows Server 2008 R2
Microsoft(R) Internet Information Services	IIS

Product Name	Abbreviation
Microsoft(R) Internet Explorer	IE

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.

The contents of this manual may be revised without prior notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Fujitsu Limited.

August 2015

Copyright 2009 - 2015 FUJITSU LIMITED

Contents

Chapter 1 Common Gateway Interface (CGI).....	1
1.1 Processing Flow of CGI.....	1
1.2 What are NetCOBOL CGI Subroutines?.....	2
Chapter 2 Creating Web Applications Using CGI Subroutines.....	3
2.1 System Requirements.....	3
2.2 Resource Requirements.....	3
2.2.1 COBOL Programs Using CGI Subroutines.....	3
2.2.2 Web Page for Invoking Application.....	6
2.2.3 Web Page for Processing Result Output.....	6
2.2.4 Conditions for CGI Application Creation Using COBOL.....	6
2.3 Use of Assist Functions for Web Application Development.....	7
Chapter 3 Use of CGI Subroutines.....	8
3.1 CGI Subroutines.....	8
3.1.1 File Upload Function.....	10
3.2 Interface with CGI Subroutines.....	13
3.2.1 Library File.....	13
3.2.2 Calling a CGI Subroutine.....	14
3.3 Set the Work Environment and Acquire Web Parameters.....	14
3.3.1 COBW3_INIT.....	14
3.4 Manipulate Web Parameters.....	15
3.4.1 COBW3_GET_VALUE, COBW3_GET_VALUE_XX, COBW3_GET_VALUE_NX, COBW3_GET_VALUE_XN, and COBW3_GET_VALUE_NN.....	15
3.4.2 COBW3_CHECK_VALUE, COBW3_CHECK_VALUE_X and COBW3_CHECK_VALUE_N.....	17
3.5 Output Processing Results.....	18
3.5.1 COBW3_PUT_HEAD.....	18
3.5.2 COBW3_PUT_HTML.....	19
3.5.3 COBW3_SET_CNV, COBW3_SET_CNV_XX, COBW3_SET_CNV_NX, COBW3_SET_CNV_XN, and COBW3_SET_CNV_NN.....	22
3.5.4 COBW3_DEL_CNV, COBW3_DEL_CNV_X and COBW3_DEL_CNV_N.....	26
3.5.5 COBW3_INIT_CNV.....	26
3.5.6 COBW3_SET_REPEAT, COBW3_SET_REPEAT_XX, COBW3_SET_REPEAT_NX, COBW3_SET_REPEAT_XN, and COBW3_SET_REPEAT_NN.....	27
3.5.7 COBW3_DEL_REPEAT, COBW3_DEL_REPEAT_X and COBW3_DEL_REPEAT_N.....	30
3.5.8 COBW3_INIT_REPEAT.....	31
3.5.9 COBW3_PUT_TEXT.....	31
3.6 Execute System Commands.....	32
3.6.1 COBW3_SYSTEM.....	32
3.7 Release CGI Execution Environment Resources.....	32
3.7.1 COBW3_FREE.....	32
3.8 Acquire Request Information.....	33
3.8.1 COBW3_RECEIVE_HEADER.....	33
3.8.2 COBW3_GET_REQUEST_INFO.....	33
3.8.3 COBW3_GET_AUTHORIZE.....	34
3.9 Manipulate Cookie Data.....	35
3.9.1 COBW3_SET_COOKIE, COBW3_SET_COOKIE_XX, COBW3_SET_COOKIE_NX, COBW3_SET_COOKIE_XN, and COBW3_SET_COOKIE_NN.....	35
3.9.2 COBW3_DEL_COOKIE, COBW3_DEL_COOKIE_X and COBW3_DEL_COOKIE_N.....	38
3.9.3 COBW3_INIT_COOKIE.....	39
3.9.4 COBW3_GET_COOKIE, COBW3_GET_COOKIE_XX, COBW3_GET_COOKIE_NX, COBW3_GET_COOKIE_XN, and COBW3_GET_COOKIE_NN.....	39
3.10 Manipulate Uploaded Files.....	41
3.10.1 COBW3_GET_UPLOADFILE_INFO, COBW3_GET_UPLOADFILE_INFO_X and COBW3_GET_UPLOADFILE_INFO_N.....	41

3.10.2 COBW3_GEN_UPLOADFILE,COBW3_GEN_UPLOADFILE_X and COBW3_GEN_UPLOADFILE_N.....	43
3.10.3 COBW3_DEL_UPLOADFILE.....	44
3.11 Other Subroutines.....	45
3.11.1 COBW3_NAME.....	45
3.11.2 COBW3_VALUE.....	46
3.11.3 COBW3_CNV_SET.....	46
3.11.4 COBW3_CNV_DEL.....	48
3.11.5 COBW3_CNV_INIT.....	48
3.12 Quantitative Restrictions on CGI Subroutines.....	48
Chapter 4 Creating and Executing Web Applications.....	50
4.1 Execution Procedure.....	50
4.2 Compilation and Linkage.....	50
4.3 Setting the Environment Variables of CGI Subroutines.....	51
4.3.1 Information That Must Be Set in System Environment Variables.....	51
4.3.2 Information That Must be Set In System Environment Variables Or Execution Initialization File.....	51
4.4 Executing Web Applications.....	52
Chapter 5 Confirming Application Operation.....	53
5.1 Examine the Log Data.....	53
5.2 Checking Operation by Interactive Debugger.....	53
5.3 Confirmation of Errors Detected by CGI Subroutines.....	54
5.4 Confirmation of In-execution Data in Presentation Format.....	54
Chapter 6 Sample.....	55
6.1 Use of the Sample Program.....	55
Appendix A Questions and Answers.....	57
Appendix B Error Handling.....	62
Appendix C Conceptual Diagram of Web Application Creation.....	67

Chapter 1 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a standard for data transfer between WWW browsers (clients) and WWW servers. CGI provides a mechanism for clients (typically using a WWW Browser) to invoke executable programs on a WWW server.

Data summation, data retrieval, file access, or database access in a business system that was constructed using the Internet or running on a local Intranet cannot be achieved when using only static HTML documents (Web pages) with a WWW server.

To support the above requirements, external COBOL applications or other tools must be used. Such external COBOL applications running in conjunction with a WWW Server are called Web applications.

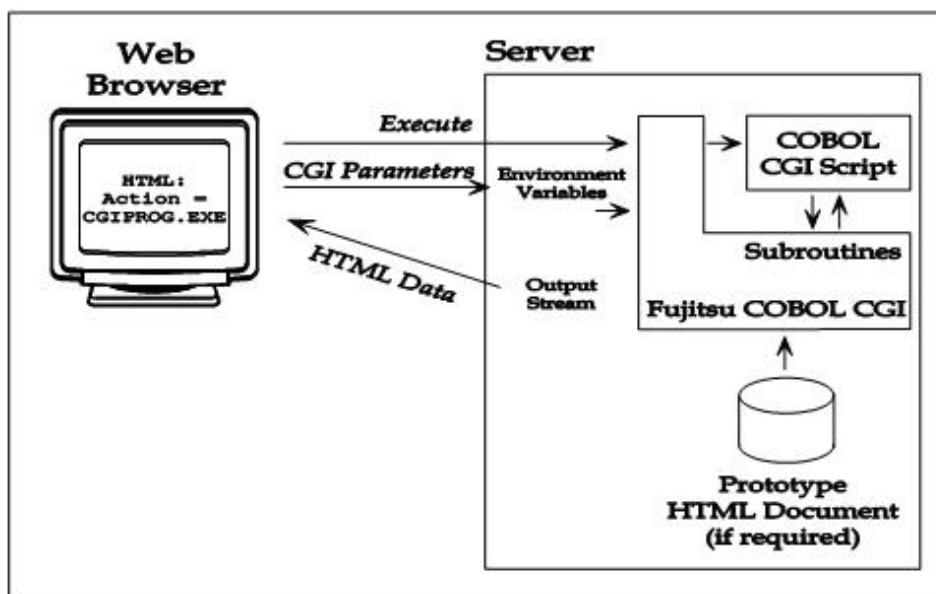
A Web application may consist of one or many programs. When using CGI, a main program is invoked on a WWW Server as a result of a request received from a client. This main program must be in .EXE format. When using NetCOBOL, you may, however, segment a Web application by using .DLL files as subprograms. DLL's, however, cannot be invoked directly from a client using an HTTP page.

This chapter:

- describes the CGI processing flow, and
- gives an overview of NetCOBOL CGI subroutines.

1.1 Processing Flow of CGI

This section explains the processing flow using CGI.



When a URL (Uniform Resource Locator) is entered in the WWW browser (client), an HTML document (later called "Web page for invoking application") that was previously registered in the WWW server is typically returned to the client and displayed.

When the Web application Submit button on the application-invoking Web page that is displayed on the WWW browser is clicked, the request is sent back to the WWW server and it generates a new process and activates the Web application. When text input items are included on the application-invoking Web page, the user-input data is passed as transfer data (also called "Web parameters") to the Web application invoked on the WWW server.

The Web application executes the following processing:

- Execution of main program and execution of initialization code
- Retrieval of Web parameters
- Execution of business logic
- Execution of code that outputs processing results as HTML documents (also called "Web page" for processing result output)

- Execution of termination code that terminates the Web application
- When the processing of the Web application terminated, the process also terminates.

1.2 What are NetCOBOL CGI Subroutines?

NetCOBOL CGI subroutines, which are also simply called CGI subroutines, are Web application building block subroutines that use CGI. Using CGI subroutines, a Web application can be easily created in COBOL.

Web applications that use CGI subroutines can operate on any type of WWW server, as CGI is the most commonly support Web application API.

Chapter 2 Creating Web Applications Using CGI Subroutines

This chapter describes how to use CGI subroutines to create web applications. It includes these topics:

- system requirements,
- resource requirements, and
- how to use assist functions in the COBOL Project Manager.

2.1 System Requirements

The following systems requirements must be met in order to create and execute Web applications that use CGI subroutines:

WWW Server

The WWW Server must be set up for executing Web applications and must support the CGI API.

NetCOBOL

NetCOBOL is required to compile, link, and execute COBOL Web applications.

Also, the NetCOBOL runtime system, which comes as NetCOBOL Server Run-time is required for production usage on a production Web server.

For details on the installation and settings of the above products, consult the system administrator of the WWW Server you are using.

2.2 Resource Requirements

The following resources must be available to create COBOL Web applications using CGI subroutines:

- COBOL application program(s) using CGI subroutines
- HTML documents
 - Web pages for invoking applications (for activating the Web applications)
 - Web pages for processing result output (only when necessary)

These resources are explained below.

2.2.1 COBOL Programs Using CGI Subroutines

Create one or more COBOL programs that comprise the Web application.

These COBOL programs can use all of the standard COBOL functions other than the screen input-output functions.

For conditions regarding COBOL program creation, see "*Conditions for CGI application creation using COBOL*" in Chapter 2.

Sample Program

The sample program shown below comes with NetCOBOL and is contained in the CGISMP01 subdirectory under the SAMPLES folder. It is a program that contains a simple Web application. NetCOBOL CGI application programs can call other COBOL programs as well, although this simple sample does not require such. The following figure shows the source code for the sample program named SAMPLE01, which is part of the CGISMP01 project:

Example

```
*-----*
* All Rights Reserved, Copyright(C) FUJITSU LIMITED 1997-2002      *
*                                                                     *
* - Module Name      SAMPLE01                                       *
* - Outline          CGI Sample Program showing data input and output *
*-----*
```



```

*-----*
IDENTIFICATION   DIVISION.
PROGRAM-ID       SAMPLE01.
ENVIRONMENT      DIVISION.
CONFIGURATION    SECTION.
INPUT-OUTPUT     SECTION.
DATA             DIVISION.
WORKING-STORAGE  SECTION.
    COPY COBW3.
*
01 WORK-CHARA-LENGTH      PIC S9(4) COMP-5 .
01 WORK-CHARA             PIC X(50) VALUE SPACE .
01 WORK-COMMA             PIC X(2)  VALUE ", " .
*
PROCEDURE         DIVISION.
*
SAMPLE01-START.
*
* Initialize CGI work environment
  CALL "COBW3_INIT"      USING COBW3 .
*
* Obtain the text entered by the user
  MOVE "ENTRY1"          TO COBW3-SEARCH-DATA .
  CALL "COBW3_GET_VALUE" USING COBW3 .
*
* Set the GET-TEXT variable in the prototype HTML to the data entered
  MOVE "GET-TEXT"        TO COBW3-CNV-NAME .
  MOVE COBW3-GET-DATA    TO COBW3-CNV-VALUE .
  CALL "COBW3_SET_CNV"   USING COBW3 .
*
* Obtain the setting of the radio button
  MOVE "RADIO1"          TO COBW3-SEARCH-DATA .
  CALL "COBW3_GET_VALUE" USING COBW3 .
*
* Set the GET-RADIO variable to the value of the selected button
  MOVE "GET-RADIO"       TO COBW3-CNV-NAME .
  MOVE COBW3-GET-DATA    TO COBW3-CNV-VALUE .
  CALL "COBW3_SET_CNV"   USING COBW3 .
*
* Obtain the setting of the check boxes
  MOVE "CHECK1"          TO COBW3-SEARCH-DATA .
  MOVE 1                 TO WORK-CHARA-LENGTH .

  PERFORM WITH TEST AFTER
    VARYING COBW3-NUMBER FROM 1 BY 1
      UNTIL COBW3-NUMBER > 4

    CALL "COBW3_GET_VALUE" USING COBW3
    IF COBW3-SEARCH-FLAG-EXIST THEN
      IF COBW3-NUMBER > 1 THEN
        MOVE WORK-COMMA TO WORK-CHARA(WORK-CHARA-LENGTH:2)
        ADD 2           TO WORK-CHARA-LENGTH
      END-IF
      MOVE COBW3-GET-DATA(1:COBW3-GET-LENGTH)
        TO WORK-CHARA(WORK-CHARA-LENGTH:COBW3-GET-LENGTH)
      ADD COBW3-GET-LENGTH TO WORK-CHARA-LENGTH
    END-IF

  END-PERFORM .
*
* Output "?" if nothing selected
  IF WORK-CHARA-LENGTH = 1 THEN
    MOVE "?" TO WORK-CHARA

```

```

END-IF .

* Set the GET-CHECK variable to the list of check box values
MOVE "GET-CHECK" TO COBW3-CNV-NAME .
MOVE WORK-CHARA TO COBW3-CNV-VALUE .
CALL "COBW3_SET_CNV" USING COBW3 .

* Setup the prototype HTML file name
MOVE "sample01_1.htm" TO COBW3-HTML-FILENAME .

* Send the completed HTML to the user
CALL "COBW3_PUT_HTML" USING COBW3 .

* Send a final message to the user
MOVE "<CENTER>Thank you for cooperation. </CENTER></BODY></HTML>"
      TO COBW3-PUT-STRING.
MOVE 58 TO COBW3-PUT-STRING-LENGTH.
CALL "COBW3_PUT_TEXT" USING COBW3 .

* Release the resources obtained by NetCOBOL CGI
CALL "COBW3_FREE" USING COBW3 .

*
SAMPLE01-END.
*
EXIT PROGRAM.

```

.....

For the flow of the processing of the sample program, see "*Appendix C. Conceptual Diagram of Web Application Creation*".

The following explains the CGI specific program code added to the SAMPLE01 COBOL:

IDENTIFICATION DIVISION

None

ENVIRONMENT DIVISION

None

DATA DIVISION

Using a standard COPY statement, the copy library that is to be used as the interface to the CGI subroutines must be added into the WORKING-STORAGE SECTION.

<pre> WORKING-STORAGE SECTION. COPY COBW3. </pre>

This copy library (COBW3.cbl) is already installed in the folder in which NetCOBOL was installed.

PROCEDURE DIVISION

To initialize the environment for the CGI subroutines and to acquire Web parameters, COBW3_INIT must be called first. To terminate the processing of the CGI subroutines and release acquired resources, the last step of the CGI COBOL application should always be to call COBW3_FREE. Other CGI subroutines may be executed using the standard CALL statement when necessary. The Specific CGI subroutines used in the sample application (in the order in which they appear) include:

COBW3_INIT

initializes the CGI subroutines environment

COBW3_GET_VALUE

returns the value entered/selected by the user in the input HTML form for the input item (control) whose name has been placed in COBW3-SEARCH-DATA (a field contained in the COBW3.cbl copy library file)

COBW3_SET_CNV

sets the value of an HTML variable in the prototype HTML to be output as a result

COBW3_PUT_HTML

sends the response HTML Web page back to the client

COBW3_PUT_TEXT

sends additional HTML back to the client

COBW3_FREE

Releasing of resources that were acquired by CGI subroutines

See Chapter 3, "Use of CGI Subroutines," for additional details of Web parameter reception and referencing, and processing result output.

2.2.2 Web Page for Invoking Application

A Web page for invoking an application is an HTML document that is to be used to start a Web application. To start a Web application that uses CGI, use the <FORM> or <A> tags. For example, when an execution-format file uses a main program as the entry of the Web application, the FORM tag must be specified as follows:

```
<FORM METHOD="POST" ACTION="cgismp01.exe">
```



Note

When a Web application was started using the <A> tag, no Web parameters can be passed. For explanations of the other HTML tags, see *Appendix A of the NetCOBOL Web Guide*.

2.2.3 Web Page for Processing Result Output

A Web page for processing result output is an HTML document with which the Web application execution results are returned to the WWW browser. Because response data can be directly returned from a Web application, the Web page for processing result output does not have to exist as a static file. However, when a Web page for processing result output does exist as a static file, return data is highly independent of the program, and flexibility and maintainability are improved. For example, the layout of processing result can be changed without recompiling the Web application.

The contents of the web page for processing result output can be dynamically changed and output.

When a static HTML document (that is, an HTML document whose output contents do not change according to the processing results) must be output, an HTML document that was created by a standard HTML creation tool can be used.

When a dynamic HTML document (that is, an HTML document whose output contents change according to the processing results) must be output, data can be dynamically changed and output by specifying an item name (conversion name) enclosed with "*//COBOL//*" on the processing result output Web page. Because all such data in the processing result output Web page can be replaced, even individual characters in the processing result output Web page can be changed as well as some tags such as the background color specification. See *COBW3_PUT_HTML* in Chapter 3, "Output of processing results."

A processing result page to be output can be divided into two or more HTML documents (e.g., title, text, and header documents) or can be combined with plain text.

2.2.4 Conditions for CGI Application Creation Using COBOL

The following conditions must be satisfied when using the CGI subroutines from COBOL:

- A COBOL program that is to be used by a Web application must not use the screen I/O functions listed below. For details, see "*NetCOBOL User's Guide*" in Chapter 21.
 - Presentation file module (screen handling function)
 - Screen handling module
 - ACCEPT/DISPLAY function (However, environment variables and date/time operation function can be used.)

- REPLACING statement must not be specified in a COPY statement that is used to read the COBW3.cbl library that is supplied by the CGI subroutines. In addition, a REPLACE statement that replaces the actual COPY statement must not be specified.
- If data was initialized in the DATA DIVISION by using the VALUE clause, the initial values that are related to two or more requests to start a CGI process with a COBOL application may not be assured in some operating environments. When initial values must be assured each time a request is issued from a WWW browser, the data must always be initialized by the PROCEDURE DIVISION to be executed to insure these initial values.

2.3 Use of Assist Functions for Web Application Development

The COBOL Project Manager provides assist functions for Web application development. These assist functions will actually generate portions of Web applications directly within the COBOL Project Manager.

A COBOL program that uses CGI subroutines can be easily created using the Web application wizard in the COBOL Project Manager. Using the Web application wizard, Web applications such as COBOL programs that use CGI subroutines can be automatically generated. Because the Web application wizard is linked with the COBOL Project Manager, the created programs can be generated, edited, compiled, linked, and maintained easily.

For details of the assist functions for Web application development, refer to the Assist Functions in the *Web Application Development User's Guide*.

Chapter 3 Use of CGI Subroutines

This chapter lists and describes each of the NetCOBOL CGI subroutines.

3.1 CGI Subroutines

Table 3.1 CGI Subroutines (COBW3_XXXXX)

Function	Subroutine name	Purpose
Work environment initialization and Web parameter acquisition	COBW3_INIT	Initialization of CGI subroutine work environment, and acquisition of Web parameters
Web parameter operation	COBW3_GET_VALUE COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX COBW3_GET_VALUE_XN COBW3_GET_VALUE_NN	Retrieval of NAME in Web parameters, and acquisition of VALUE
	COBW3_CHECK_VALUE COBW3_CHECK_VALUE_X COBW3_CHECK_VALUE_N	Retrieval of VALUE in Web parameters
Processing result output	COBW3_PUT_HEAD	Output of header
	COBW3_PUT_HTML	Output of web page for processing result output
	COBW3_SET_CNV COBW3_SET_CNV_XX COBW3_SET_CNV_NX COBW3_SET_CNV_XN COBW3_SET_CNV_NN	Registration of conversion data in Web page for processing result output
	COBW3_DEL_CNV COBW3_DEL_CNV_X COBW3_DEL_CNV_N	Deletion of conversion data from Web page for processing result output
	COBW3_INIT_CNV	Initialization of conversion data in Web page for processing result output
	COBW3_SET_REPEAT COBW3_SET_REPEAT_XX COBW3_SET_REPEAT_NX COBW3_SET_REPEAT_XN COBW3_SET_REPEAT_NN	Registration of repeated conversion data in Web page for processing result output
	COBW3_DEL_REPEAT COBW3_DEL_REPEAT_X COBW3_DEL_REPEAT_N	Deletion of repeated conversion data from Web page for processing result output
	COBW3_INIT_REPEAT	Initialization of repeated conversion data in Web page for processing result output
	COBW3_PUT_TEXT	Output of data (with addition of line feed code)

Function	Subroutine name	Purpose
System command execution	COBW3_SYSTEM	Execution of specified system command
CGI execution environment resource release	COBW3_FREE	Releasing of resources that were acquired by CGI subroutines
Request information acquisition	COBW3_RECEIVE_HEADER	Acquisition of HTTP header
	COBW3_GET_REQUEST_INFO	Acquisition of request information
	COBW3_GET_AUTHORIZE	Acquisition of authorization information
Cookie data operation	COBW3_SET_COOKIE	Registration of cookie data
	COBW3_SET_COOKIE_XX	
	COBW3_SET_COOKIE_NX	
	COBW3_SET_COOKIE_XN	
	COBW3_SET_COOKIE_NN	
	COBW3_DEL_COOKIE	Deletion of registered cookie data
	COBW3_DEL_COOKIE_X	
	COBW3_DEL_COOKIE_N	
	COBW3_INIT_COOKIE	Initialization of all registered cookie data
	COBW3_GET_COOKIE	Acquisition of cookie data from Web parameter
COBW3_GET_COOKIE_XX		
COBW3_GET_COOKIE_NX		
COBW3_GET_COOKIE_XN		
COBW3_GET_COOKIE_NN		

It is necessary to use the following subroutines in the ASCII environment:

- COBW3_GET_VALUE
- COBW3_CHECK_VALUE
- COBW3_SET_CNV
- COBW3_DEL_CNV
- COBW3_SET_REPEAT
- COBW3_DEL_REPEAT
- COBW3_SET_COOKIE
- COBW3_DEL_COOKIE
- COBW3_GET_COOKIE

It is necessary to use subroutines such as COBW3_XXXXX_XX and COBW3_XXXXX_N in the Unicode environment. Moreover, the meaning is as follows:

Subroutine name	Meaning of "X" and "N"
COBW3_XXXXX_XX	The NAME and VALUE parameters are regarded as alphanumeric data items.
COBW3_XXXXX_NX	The NAME parameter is regarded as a national data item. The VALUE parameter is regarded as an alphanumeric data item.
COBW3_XXXXX_XN	The NAME parameter is regarded as an alphanumeric data item. The VALUE parameter is regarded as a national data item.
COBW3_XXXXX_NN	The NAME and VALUE parameters are regarded as national data items.

Subroutine name	Meaning of "X" and "N"
COBW3_XXXXX_X	An input parameter is regarded as an alphanumeric data item.
COBW3_XXXXX_N	An input parameter is regarded as a national data item.

Any code system can use subroutines not listed above.

The operation of the conventional subroutines listed below is assured only when the related Web applications operate in an environment using ASCII code. If a Web application that will operate with a code system other than ASCII is to be developed, its alternative subroutine listed in the following table must be used.

Conventional subroutine	Alternative subroutine
COBW3_NAME	COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX COBW3_GET_VALUE_XN COBW3_GET_VALUE_NN
COBW3_VALUE	COBW3_CHECK_VALUE_X COBW3_CHECK_VALUE_N
COBW3_CNV_SET	COBW3_SET_CNV_XX COBW3_SET_CNV_NX COBW3_SET_CNV_XN COBW3_SET_CNV_NN
COBW3_CNV_DEL	COBW3_DEL_CNV_X COBW3_DEL_CNV_N
COBW3_CNV_INIT	COBW3_INIT_CNV

3.1.1 File Upload Function

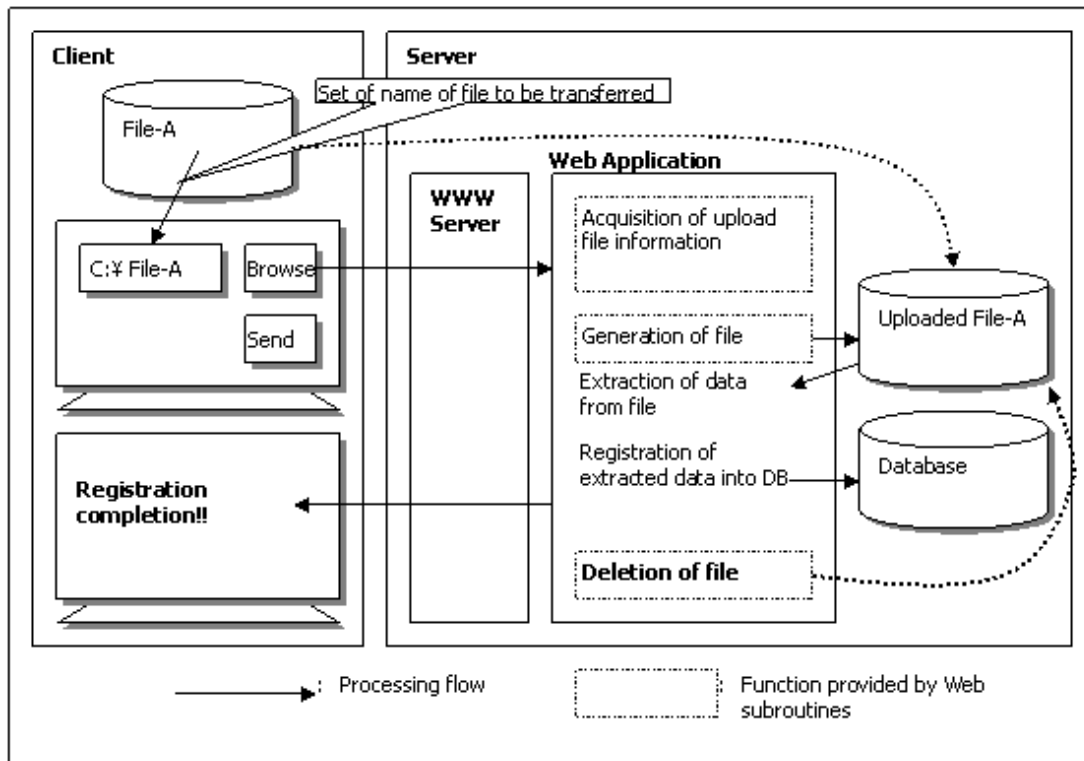
The file upload function transfers client-side-generated data (files) to the server.

This function is valid in the following cases:

- When a large amount of data that may cause timer expiration in online processing is to be entered.
- When a mobile environment with meter-rate accounting is to be used.
- When a data entry tool is already prepared at the client side and server linkage using the tool output data (files) is requested instead of a request to change the data entry tool into a Web tool.
- When only the HTTP ports can be accessed in file transfer because of security conditions. (File transfer in conventional systems is done using FTP.)

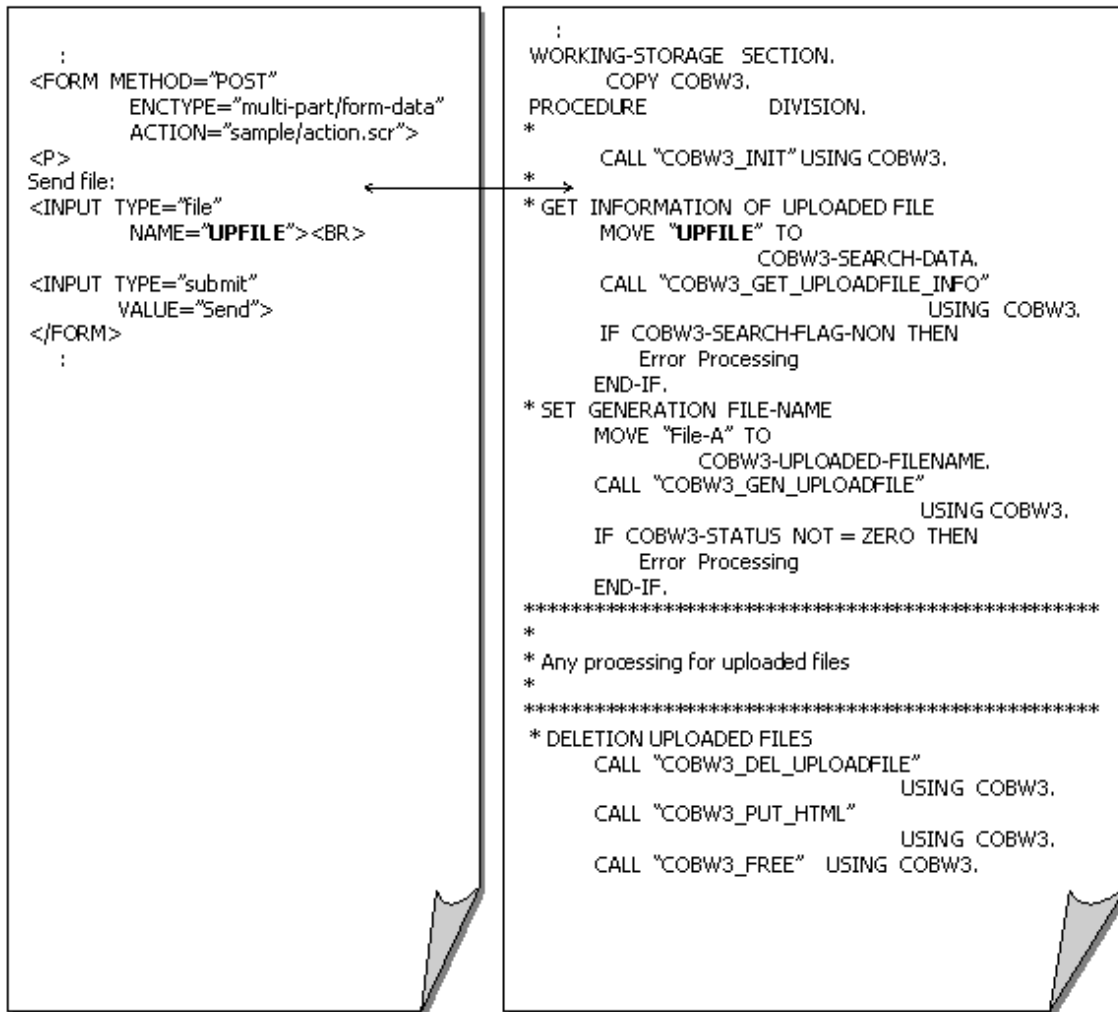
Using the file upload function, file transfer can be associated with applications. Therefore, data can be checked and registered immediately after file transfer.

The concept chart of processing which uses the file upload function is shown below:



Web page for invoking the applications

Web application (COBOL source)



Web subroutines provide the following functions to easily create such application programs:

Function	Subroutine name	Usage
File upload	COBW3_GET_UPLOADFILE_INFO	Acquires the information on an upload file.
	COBW3_GET_UPLOADFILE_INFO_X	
	COBW3_GET_UPLOADFILE_INFO_N	
	COBW3_GEN_UPLOADFILE	Generates an uploaded file on the server.
COBW3_GEN_UPLOADFILE_X		
COBW3_GEN_UPLOADFILE_N		
	COBW3_DEL_UPLOADEDFILE	Deletes an uploaded file.

To upload a file, the following values must be written on the Web page for invoking an application:

Tag name	Attributes	Value
FORM	METHOD	POST
	ENCTYPE	multipart/form-data
INPUT	TYPE	file

For details of these tags and attributes, see the "NetCOBOL Web Guide" and other documents or home pages that explain HTML.

The above tags can be used together with the INPUT tag of other types (that is, other than "file") in the same FORM. The data that corresponds to these INPUT tags can be acquired by a subroutine such as COBW3_GET_VALUE regardless of whether the file upload function is used.

 **Note**

The NAME value specified in the INPUT tag (TYPE="file") for uploading files cannot be used in the INPUT tag of other types (that is, other than "file"). Otherwise, the normal operation of the Web subroutine is not assured.

 **Example**

```

:
<INPUT TYPE="text" NAME="FILE1">
<INPUT TYPE="file" NAME="FILE1" & Do not specify
:

```

3.2 Interface with CGI Subroutines

The following subsections explain the interface that is required for using CGI subroutines. Each processing rule is explained under the following titles:

Format: Explains the format required to call a specific CGI subroutine.

Data setting at calling: Explains the information (Call parameters) that must be set before the CGI subroutine is called.

- When a condition-name was written, a value must be set using the SET statement.
- When a condition-name is underlined, the value is an initial value.
- Calling a CGI subroutine does not change a data value that was set at calling time.

Processing result data: Explains the processing results that are accomplished by executing the specific CGI subroutine.

- When a condition-name is written, the condition must be checked using the condition-name condition.
- The contents of an error or warning detected in the CGI subroutines can be referenced by using COBW3-STATUS.

Value	Explanation
0	Normal end
Other than 0	Error number

For an explanation of the error number, see Appendix B, "Error Handling."

 **Note**

If an area other than the data area that was opened at data setting for the subroutine calling is changed before the subroutine was called, the operation or data contents are not assured.

Before COBW3_INIT is called, COBW3 in the interface area must be initialized using LOW-VALUE.

3.2.1 Library File

A library file to be copied into each NetCOBOL CGI program is supplied as the interface with the call parameters and return parameters of the CGI subroutines. To call a CGI subroutine, the library file listed below must be read into the WORKING-STORAGE SECTION using the COPY statement. This library exists in the folder in which NetCOBOL was previously installed.

Library name

COBW3.cbl

Format

Specify this library as follows:

```
WORKING-STORAGE SECTION.  
    COPY COBW3.
```



Note

The REPLACING option must not be specified in the COPY statement for reading the CGI subroutine library nor should the REPLACE statement be specified for replacing this COPY statement.

If the library file is changed, the operation is not assured.

3.2.2 Calling a CGI Subroutine

To set the CGI subroutine environment or to acquire Web parameters, *COBW3_INIT* must be called first. To terminate the CGI subroutine processing at the end of the execution phase, *COBW3_FREE* must always be called to release the CGI subroutine resources.

Other subroutines must be called using the CALL statement when necessary.

To call another subroutine from a program other than the program that called the *COBW3_INIT*, the COBW3 interface area that was defined by the program that originally called *COBW3_INIT* must be passed between the related programs.



Example

Parent program A calls *COBW3_INIT*, and child program calls *COBW3_FREE*.

Parent program

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.      A.  
DATA            DIVISION.  
WORKING-STORAGE SECTION.  
    COPY COBW3.  
PROCEDURE      DIVISION.  
    MOVE LOW-VALUE TO COBW3.  
    CALL "COBW3_INIT" USING COBW3.  
    :  
    CALL "E"          USING COBW3.  
    STOP RUN.
```

Child program

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.      B.  
DATA            DIVISION.  
LINKAGE         SECTION.  
    COPY COBW3.  
PROCEDURE      DIVISION USING COBW3.  
    :  
    CALL "COBW3_FREE" USING COBW3.  
    EXIT PROGRAM.
```

3.3 Set the Work Environment and Acquire Web Parameters

3.3.1 COBW3_INIT

This subroutine initializes the CGI subroutine work environment, and acquires Web parameters and sets them in the work area to be used by the CGI subroutines.



Note

Before calling the *COBW3_INIT*, the COBW3 in the interface area must be initialized using LOW-VALUE.

Format

```
CALL "COBW3_INIT" USING COBW3.
```

Data setting at calling

COBW3-DMODE [Optional]

Using this parameter, specify whether to output error messages of the CGI subroutines.

The COBW3-DMODE is valid only when the operation code system is ASCII.

Condition name	Value	Explanation
<u>COBW3-DMODE-NODBG</u>	LOW-VALUE	Does not output error messages.
COBW3-DMODE-DBG	"1"	Output error messages to the browser.

Processing result data

None



If the *COBW3_INIT* is called two or more times consecutively, the acquired Web parameters may be initialized and the subsequent operation is not assured. To prevent this, the program must be constructed so that the *COBW3_INIT* is called only once during the period from WWW browser connection to the calling of another CGI subroutine.

3.4 Manipulate Web Parameters

3.4.1 COBW3_GET_VALUE, COBW3_GET_VALUE_XX, COBW3_GET_VALUE_NX, COBW3_GET_VALUE_XN, and COBW3_GET_VALUE_NN

These subroutines retrieve an arbitrary name (NAME) from the Web parameters that were acquired using *COBW3_INIT*, and acquire the corresponding value (VALUE). The functions of these subroutines are as follows:

ASCII environment

COBW3_GET_VALUE

Retrieves an alphanumeric character string name (NAME), and returns the alphanumeric character string that is the corresponding value (VALUE).

Unicode environment

COBW3_GET_VALUE_XX

Retrieves an alphanumeric character string name (NAME), and returns the alphanumeric character string that is the corresponding value (VALUE).

COBW3_GET_VALUE_NX

Retrieves a national character string name (NAME), and returns the alphanumeric character string that is the corresponding value (VALUE).

COBW3_GET_VALUE_XN

Retrieves an alphanumeric character string name (NAME), and returns the national character string that is the corresponding value (VALUE).

COBW3_GET_VALUE_NN

Retrieves a national character string name (NAME), and returns the national character string that is the corresponding value (VALUE).

To store the acquired value (VALUE), the en-size blank is added as a padding character.

Format

```
CALL "COBW3_GET_VALUE" USING COBW3.
```

```
CALL "COBW3_GET_VALUE_XX" USING COBW3.
```

```
CALL "COBW3_GET_VALUE_NX" USING COBW3.
```

```
CALL "COBW3_GET_VALUE_XN" USING COBW3.
```

```
CALL "COBW3_GET_VALUE_NN" USING COBW3.
```

Data setting at calling

COBW3-SEARCH-DATA or COBW3-SEARCH-DATA-N

Specify a retrieval name (NAME) that was already specified in the "name" in the HTML document (Web page used to invoke the application).

When COBW3_GET_VALUE, COBW3_GET_VALUE_XX or COBW3_GET_VALUE_XN are used, specify the name in COBW3-SEARCH-DATA.

When COBW3_GET_VALUE_NX or COBW3_GET_VALUE_NN are used, specify the name in COBW3-SEARCH-DATA-N.

COBW3-SEARCH-LENGTH [Optional]

To retrieve a name (NAME) that has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a name up to the last character other than the blank.
1 to 1024	Retrieves a name of the specified character string length.

COBW3-NUMBER [Optional]

When two or more identical names (NAME) are specified in the CGI parameter, specify the occurrence sequence of the retrieval targets.

Condition name	Value	Explanation
COBW3-NUMBER-INIT	1	Retrieves the first matching name.
-	2 to 9999	Retrieves the name with the specified occurrence sequence.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-NON	"0"	The retrieved name was not found.
COBW3-SEARCH-FLAG-EXIST	"1"	The retrieved name was found.

COBW3-GET-DATA or COBW3-GET-DATA-N

The value (VALUE) that corresponds to the retrieval name (NAME) is set.

When COBW3_GET_VALUE, COBW3_GET_VALUE_XX or COBW3_GET_VALUE_NX are used, the found value is set in COBW3-GET-DATA.

When COBW3_GET_VALUE_XN or COBW3_GET_VALUE_NN are used, the found value is set in COBW3-GET-DATA-N.

COBW3-GET-LENGTH

The character string length (byte length) of the value (VALUE) corresponding to the retrieval name (NAME) is set.

3.4.2 COBW3_CHECK_VALUE, COBW3_CHECK_VALUE_X and COBW3_CHECK_VALUE_N

These subroutines retrieve an arbitrary value (VALUE) from the Web parameters that were acquired using *COBW3_INIT*. These subroutines are used for the VALUE check of the item which was checked on the WWW browser when these subroutines are called from an application-invoking Web page with a check box. The functions of these subroutines are as follows:

ASCII environment

COBW3_CHECK_VALUE

Retrieves an alphanumeric character string value (VALUE).

Unicode environment

COBW3_CHECK_VALUE_X

Retrieves an alphanumeric character string value (VALUE).

COBW3_CHECK_VALUE_N

Retrieves a national character string value (VALUE).

Format

```
CALL "COBW3_CHECK_VALUE" USING COBW3 .
CALL "COBW3_CHECK_VALUE_X" USING COBW3 .
CALL "COBW3_CHECK_VALUE_N" USING COBW3 .
```

Data setting at calling

COBW3-SEARCH-DATA or COBW3-SEARCH-DATA-N

Specify a retrieval value (VALUE).

When COBW3_CHECK_VALUE and COBW3_CHECK_VALUE_X are used, specify the value in COBW3-SEARCH-DATA.

When COBW3_CHECK_VALUE_N is used, specify the value in COBW3-SEARCH-DATA-N.

COBW3-SEARCH-LENGTH [Optional]

To retrieve a value (VALUE) that has a valid blank at the end, specify the byte length of the blank-containing value.

Value	Explanation
0	Retrieves a value up to the last character other than the blank.
1 to 1024	Retrieves a value of the specified character string length.

COBW3-NUMBER [Optional]

When two or more identical values (VALUE) are specified in the CGI parameter, specify the occurrence sequence of the retrieval targets.

Condition name	Value	Explanation
<u>COBW3-NUMBER-INIT</u>	1	Retrieves the first matching value.
-	2 to 9999	Retrieves the value with the specified occurrence sequence.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-NON	"0"	The retrieved value was not found.
COBW3-SEARCH-FLAG-EXIST	"1"	The retrieved value was found.

3.5 Output Processing Results

3.5.1 COBW3_PUT_HEAD

This subroutine outputs an arbitrary HTTP header that is specified.

When an HTML document for a processing result output Web page is output, the CGI subroutine outputs its header as "text/html." Therefore, the header need not be output by using this subroutine. However, when a non-HTML document such as plain text is to be output or an arbitrary HTTP header is to be output to a WWW server, this subroutine must be used.



Note

This subroutine must be used before COBW3_PUT_HTML or COBW3_PUT_TEXT is used. If this subroutine is used after COBW3_PUT_HTML or COBW3_PUT_TEXT are executed, the specified header is invalidated.

If a value other than the default value is specified in COBW3-CONTENT-TYPE, COBW3-STATUS-CODE, or COBW3-CONTENT-LENGTH, the specified value must not be changed later.

Format

```
CALL "COBW3_PUT_HEAD" USING COBW3.
```

Data setting at calling

COBW3-PUT-HEAD

Specify a character string that is to be output as a HTTP header.

COBW3-PUT-HEAD-LENGTH [Optional]

To output a header that is a character string containing a valid blank at the end, specify the total character string length (byte length) that contains the blank.

Value	Explanation
<u>0</u>	Outputs a character string up to the last character other than the blank. When the COBW3-PUT-HEAD character string consists of only blanks, the character string length is assumed to be 0 (only line feed occurs).
1 to 512	Outputs a character string of the specified character string length.

COBW3-CONTENT-TYPE [Optional]

Specify the file type attribute (Content-type) of the response data.

If this subroutine was called two or more times, only the first specified Content-type is valid.

Condition name	Value	Explanation
-	<u>LOW-VALUE</u>	Outputs an HTML document.
COBW3-CONTENT-TYPE-NON	HIGH-VALUE	Does not output the Content-type information.
COBW3-CONTENT-TYPE-HTML	"text/html"	Outputs an HTML document.

Condition name	Value	Explanation
COBW3-CONTENT-TYPE-TEXT	"text/plain"	Outputs a text file.
-	Arbitrary character string	Specifies an arbitrary character string in the Content-type field of the HTTP header.

COBW3-STATUS-CODE [Optional]

Specify a status code (Status-code).

If this subroutine was called two or more times, only the first specified Status-code is valid.

Condition name	Value	Explanation
-	<u>LOW-VALUE</u>	Outputs a normal end code ("200") as the Status-code.
COBW3-STATUS-CODE-NON	HIGH-VALUE	Does not output a Status-code.
COBW3-STATUS-CODE-200	"200"	Outputs a normal end code ("200") as the Status-code.
-	Arbitrary code	Outputs an arbitrary Status-code.

Processing result data

None

3.5.2 COBW3_PUT_HTML

This subroutine outputs a processing result output Web page (HTML document) to a WWW browser after a Web application is executed. In this case, when the processing result output Web page has a conversion name enclosed in "*//COBOL//*" or "*//COBOL_REPEAT//*", the conversion name must be previously converted into a conversion result (conversion character string) which was registered using COBW3_SET_CNV or COBW3_SET_REPEAT.

Format

```
CALL "COBW3_PUT_HTML" USING COBW3.
```

Data setting at calling

COBW3-HTML-FILENAME

Specify the file name of the Web page for processing result output.

Processing result data

None

Specification of conversion name in a Web page for processing result output

In the target output position in the Web page for processing result output, specify a character string (conversion name) that is to be converted and output when the Web application is executed. This conversion name must be enclosed in "*//COBOL//*".



Note

"*//COBOL//*" must not be written so that it is split across two or more lines.

```
. . . //COBOL//conversion-name//COBOL// . . .
```

The conversion processing depends on the specification format of the conversion name as follows:

Specification format of conversion name	Processing
This is a //COBOL//NAME//COBOL// test	When conversion name "NAME" is already registered, "//COBOL//NAME//COBOL//" is replaced with the registered conversion result (conversion character string).
This is a //COBOL// test	An error occurs because the conversion name is specified erroneously.
This is a //COBOL test	The conversion name is output without conversion.
This is a //COBOL//NAME//COBOL// test	An error occurs because the conversion name is specified erroneously.
This is a //COBOL//NAME//COBOL	An error occurs because the conversion name is specified erroneously.
This is a //COBOL///COBOL//	An error occurs because the conversion name is specified erroneously.
This is a //cobol//NAME//cobol//.	The conversion name is output without conversion.



Example

Usage example

[a.htm (Web page for processing result output)]

```

:
<BODY>
<BR>
Name: //COBOL//GET-NAME//COBOL//
<BR>
:

```

[COBOL program]

```

:
* Name setting
MOVE "GET-NAME" TO COBW3-CNV-NAME.
MOVE "Fujitsu Taro" TO COBW3-CNV-VALUE.
* Registration of conversion data contained on Web page for processing result output
CALL "COBW3_SET_CNV~" USING COBW3.
:
* Setting of file name of Web page for processing result output
MOVE "a.htm" TO COBW3-HTML-FILENAME.
* Output of Web page for processing result output
CALL "COBW3_PUT_HTML" USING COBW3.
:

```

[Processing result output to WWW browser]

```

:
Name: Fujitsu Taro
:

```

Specification of repetition range in a Web page for processing result output

In the target repetition output position on the Web page for processing result output, specify a repetition range by using "*//COBOL_REPEAT_START//*" and "*//COBOL_REPEAT_END//*." The repetition count must be the same as the number of registered repetitive conversion results (conversion character strings) to be converted.

- Start of repetition range

At the start position of the data repetition range, write "*//COBOL_REPEAT_START//*."

- Repetition item

Describe the data that is to be repeatedly output. When a character string is to be converted and output when the Web application is executed, describe a conversion name.

To output a different conversion result (different conversion character string) depending on the repetition time, enclose the conversion name in "//COBOL_REPEAT//." To output a permanent conversion character string in the repetition, enclose the conversion name (character string) in "//COBOL//."

- End of repetition range

At the end position of the data repetition range, write "//COBOL_REPEAT_END//."

Note

A conversion character string that corresponds to a conversion name enclosed in "//COBOL_REPEAT//" must always be registered at least once.

If a conversion name that is enclosed in "//COBOL_REPEAT//" is not described in a repetition item, that repetition item is output only once.

"//COBOL//" or "//COBOL_REPEAT//" must not be written so that it is split across two or more lines.

The lines that contain "//COBOL_REPEAT_START//" or "//COBOL_REPEAT_END//" for repetition range specification are deleted when the processing results are output to the WWW browser. To prevent problems, HTML tags for data display must not be written on these lines.

All of the repetitive conversion results (conversion character strings) that were written in a repetition range must be registered the same number of times. If these characters are registered different numbers of times, an error occurs because some conversion names cannot acquire the corresponding registered data in the repetition.

If a repetition range start is not described, no repetition occurs and descriptions other than "//COBOL//" are output as data.

If a repetition range end is not described, an error occurs after the Web page for processing result output is output up to the end.

```
//COBOL_REPEAT_START//  
. . //COBOL_REPEAT//conversion-name//COBOL_REPEAT// . .  
. . //COBOL//conversion-name//COBOL// . .  
//COBOL_REPEAT_END//
```

Conversion processing of conversion names depends on the specification format of the conversion names in the same way as in the processing of the Web page for processing result output.

Example

Usage example

[b.htm (Web page for processing result output)]

```
<TABLE>
  <TR>
    <TD>Name</TD>
    <TD>Age</TD>
    <TD>Company</TD>
  </TR>
  // COBOL_REPEAT_START//
  <TR>
    <TD>// COBOL_REPEAT// GET-NAME// COBOL_REPEAT//</TD>
    <TD>// COBOL_REPEAT// GET-AGE// COBOL_REPEAT</TD>
    <TD>// COBOL// GET-OFFICE// COBOL//</TD>
  </TR>
  // COBOL_REPEAT_END//
</TABLE>
```

[COBOL program]

```
PERFORM UNTIL EMD-FLAG="END"
* Setting of name
  MOVE "GET-NAME" TO COBW3-CNV-NAME
  MOVE Name TO COBW3-CNV-VALUE
* Registration of repetition conversion data contained on Web page for processing result output
  CALL "COBW3_SET_REPEAT" USING COBW3
* Setting of age
  MOVE "GET-AGE" TO COBW3-CNV-NAME
  MOVE Age TO COBW3-CNV-VALUE
* Registration of repetition conversion data contained on Web page for processing result output
  CALL "COBW3_SET_REPEAT" USING COBW3
:
END-PERFORM
:
* Setting of company name
  MOVE "GET-OFFICE" TO COBW3-CNV-NAME.
  MOVE "Fujitsu" TO COBW3-CNV-VALUE.
* Registration of conversion data contained on Web page for processing result output
  CALL "COBW3_SET_CNV" USING COBW3.
:
* Setting of file name of Web page for processing result output
  MOVE "b.htm" TO COBW3-HTML-FILENAME.
* Output of Web page for processing result output
  CALL "COBW3_PUT_HTML" USING COBW3.
:
```

[Processing result output to WWW browser]

Name	Age	Company
Suzuki	35	Fujitsu
Sato	26	Fujitsu
Tanaka	23	Fujitsu

3.5.3 COBW3_SET_CNV, COBW3_SET_CNV_XX, COBW3_SET_CNV_NX, COBW3_SET_CNV_XN, and COBW3_SET_CNV_NN

These subroutines register a conversion result (conversion character string) that corresponds to a conversion name that is enclosed in "*COBOL*" on the processing result output Web page that is to be output by *COBW3_PUT_HTML*.

The registered information is referenced during the execution of *COBW3_PUT_HTML*. The conversion name in the processing result output Web page is converted according to the registered conversion data.

These subroutines have the following functions:

ASCII environment

COBW3_SET_CNV

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

Unicode environment

COBW3_SET_CNV_XX

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

COBW3_SET_CNV_NX

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is a national character string.

COBW3_SET_CNV_XN

Registers, as a national character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

COBW3_SET_CNV_NN

Registers, as a national character string, the conversion character string that corresponds to the conversion name, which is a national character string.



If two or more different conversion names are specified in a Web page for processing result output, conversion character strings must be registered by calling *COBW3_SET_CNV*, etc. for each of these conversion names.

Format

```
CALL "COBW3_SET_CNV" USING COBW3.
```

```
CALL "COBW3_SET_CNV_XX" USING COBW3.
```

```
CALL "COBW3_SET_CNV_NX" USING COBW3.
```

```
CALL "COBW3_SET_CNV_XN" USING COBW3.
```

```
CALL "COBW3_SET_CNV_NN" USING COBW3.
```

Data setting at calling

COBW3-CNV-NAME and COBW3-CNV-NAME-N

Specify a conversion name that is to be converted.

When *COBW3_SET_CNV*, *COBW3_SET_CNV_XX* or *COBW3_SET_CNV_XN* are used, specify the name in *COBW3-CNV-NAME*.

When *COBW3_SET_CNV_NX* or *COBW3_SET_CNV_NN* are used, specify the name in *COBW3-CNV-NAME-N*.

COBW3-CNV-NAME-LENGTH [Optional]

To register a conversion name that has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a conversion name up to the last character other than the blank. However, if the COBW3-CNV-NAME or COBW3-CNV-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Retrieves a conversion name of the specified character string length.

COBW3-CNV-VALUE and COBW3-CNV-VALUE-N

Specify a conversion result (conversion character string).

When COBW3_SET_CNV, COBW3_SET_CNV_XX or COBW3_SET_CNV_NX are used, specify the conversion character string in COBW3-CNV-VALUE.

When COBW3_SET_CNV_XN or COBW3_SET_CNV_NN are used, specify the conversion character string in COBW3-CNV-VALUE-N.

COBW3-CNV-VALUE-LENGTH [Optional]

When the conversion character string has a valid blank at the end, specify the byte length of the character string containing the blank.

Value	Explanation
0	Registers a conversion character string up to the last character other than the blank. However, if the COBW3-CNV-VALUE or COBW3-CNV-VALUE-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 1024	Registers a conversion character string of the specified character string length.

COBW3-CNV-MODE [Optional]

Specify a conversion mode.

Condition name	Value	Explanation
<u>COBW3-CNV-MODE-ADDREP</u>	LOW-VALUE	Adds conversion data if the specified conversion name was not registered.
COBW3-CNV-MODE-REPLACE	"1"	Replaces conversion data. If the specified conversion name was not registered, an abnormal end occurs (the conversion name is not converted).
COBW3-CNV-MODE-ADD	"2"	Adds conversion data. If the specified conversion name was already registered, an abnormal end occurs (the conversion data is not added).

COBW3_SANITIZE_CNV [Optional]

If characters that are vulnerable to a cross site scripting attack are found in conversion data, those characters are automatically replaced. This process is referred to as "sanitizing".

For more details on cross site scripting, refer to Appendix P, Security, in the NetCOBOL User's Guide.

COBW3_SANITIZE_CNV is valid when either COBW3_SET_CNV_XX or COBW3_SET_CNV_NX is used. However, if the code set is Unicode, COBW3_SANITIZE_CNV is also valid when COBW3_SET_CNV_XN or COBW3_SET_CNV_NN is used.

Condition name	Value	Explanation
<u>COBW3-SANITIZE-CNV-OFF</u>	LOW-VALUE	Does not sanitize.
COBW3-SANITIZE-CNV-ON	"1"	Sanitize.

Note

The sanitization procedure replaces the five characters that are vulnerable to a cross site scripting attack (&, <, >, ", ') with the following escape characters:

```
& -> &amp;
< -> &lt;
> -> &gt;
" -> &quot;
' -> &#39;
```

As a result, a single character is being replaced with 4-6 characters, increasing the length of the sanitized data. This means that, depending on the content of the unsanitized data, sanitizing data may cause the maximum data length (1024 bytes) set in the Web parameter VALUE to be exceeded. If this happens, the sanitized data is truncated automatically at 1024 bytes.

It is also possible that the escape characters themselves may be truncated. In this case, the vulnerable character is deleted, not replaced. An example is given below.

Example

The unsanitized data area is 1021 bytes long. The first 1020 bytes contain n characters that do not require sanitizing, but the last character is an ampersand (&).

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&    <- Last character is "&"
|                               |
|           1021 bytes         |
```

After sanitizing, this becomes:

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&amp;
|                               |
|           1025 bytes         |
```

This exceeds the 1024-byte limit, but truncating the data to 1024 bytes would interrupt the escape character string (&). For this reason, the ampersand is deleted instead of being replaced.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    <- Trailing "&" is deleted
|                               |
|           1020 bytes         |
```

As a result, the actual length of the data is reduced to 1020 bytes.

Processing result data

COBW3_SANITIZED_CNV_FLAG

If COBW3_SANITIZE_CNV is set, a value that indicates whether or not the replacement procedure (sanitizing) actually took place is set.

COBW3_SANITIZED_CNV_FLAG is valid when either COBW3_SET_CNV_XX or COBW3_SET_CNV_NX is used. However, if the code set is Unicode, COBW3_SANITIZED_CNV_FLAG is also valid when COBW3_SET_CNV_XN or COBW3_SET_CNV_NN is used.

Condition name	Value	Explanation
COBW3-SANITIZED-CNV-NON	"0"	Did not sanitize.
COBW3-SANITIZED-CNV-EXIST	"1"	Sanitized.

3.5.4 COBW3_DEL_CNV, COBW3_DEL_CNV_X and COBW3_DEL_CNV_N

These subroutines delete the conversion data that was registered by COBW3_SET_CNV, etc. These subroutines have the following functions:

ASCII environment

COBW3_DEL_CNV

Deletes the conversion data that was registered so that it corresponded to the conversion name which was an alphanumeric character string.

Unicode environment

COBW3_DEL_CNV_X

Deletes the conversion data that was registered so that it corresponded to the conversion name, which was an alphanumeric character string.

COBW3_DEL_CNV_N

Deletes the conversion data that was registered so that it corresponded to the conversion name, which was a national character string.

Format

```
CALL "COBW3_DEL_CNV" USING COBW3.
```

```
CALL "COBW3_DEL_CNV_X" USING COBW3.
```

```
CALL "COBW3_DEL_CNV_N" USING COBW3.
```

Data setting at calling

COBW3-CNV-NAME and COBW3-CNV-NAME-N

Specify a conversion name that is to be deleted.

When COBW3_DEL_CNV or COBW3_DEL_CNV_X is used, specify the conversion name in COBW3-CNV-NAME.

When COBW3_DEL_CNV_N is used, specify the name in COBW3-CNV-NAME-N.

COBW3-CNV-NAME-LENGTH [Optional]

When the conversion name has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a conversion name up to the last character other than the blank. However, if the COBW3-CNV-NAME or COBW3-CNV-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Retrieves a conversion name of the specified character string length.

Processing result data

None

3.5.5 COBW3_INIT_CNV

This subroutine initializes the conversion data that was registered by COBW3_SET_CNV, etc.

Format

```
CALL "COBW3_INIT_CNV" USING COBW3.
```

Data setting at calling

None

Processing result data

None

3.5.6 COBW3_SET_REPEAT, COBW3_SET_REPEAT_XX, COBW3_SET_REPEAT_NX, COBW3_SET_REPEAT_XN, and COBW3_SET_REPEAT_NN

These subroutines register a repetitive conversion result (conversion character string) that corresponds to a conversion name that is enclosed in `"//COBOL_REPEAT/"` on the processing result output Web page that is to be output by `COBW3_PUT_HTML`.

The registered information is referenced during the execution of `COBW3_PUT_HTML`. The conversion name that is specified in the repetition range in the processing result output Web page is converted according to the number of repetition conversion character strings registered for the same conversion data.

These subroutines have the following functions:

ASCII environment

COBW3_SET_REPEAT

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

Unicode environment

COBW3_SET_REPEAT_XX

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

COBW3_SET_REPEAT_NX

Registers, as an alphanumeric character string, the conversion character string that corresponds to the conversion name, which is a national character string.

COBW3_SET_REPEAT_XN

Registers, as a national character string, the conversion character string that corresponds to the conversion name, which is an alphanumeric character string.

COBW3_SET_REPEAT_NN

Registers, as a national character string, the conversion character string that corresponds to the conversion name, which is a national character string.



.....
If two or more different conversion names are specified in a Web page for processing result output, repetition conversion character strings must be registered by calling `COBW3_SET_REPEAT`, etc. for each of these conversion names.
.....

The conversion character strings are converted in the same sequence as the sequence of the registration performed by calling `COBW3_SET_REPEAT`, etc.

The repetition conversion character strings that correspond to the conversion names contained in the same repetition range must be registered the same number of times.

Format

```
CALL "COBW3_SET_REPEAT" USING COBW3 .
```



```
CALL "COBW3_SET_REPEAT_XX" USING COBW3.
```

```
CALL "COBW3_SET_REPEAT_NX" USING COBW3.
```

```
CALL "COBW3_SET_REPEAT_XN" USING COBW3.
```

```
CALL "COBW3_SET_REPEAT_NN" USING COBW3.
```

Data setting at calling

COBW3-CNV-NAME and COBW3-CNV-NAME-N

Specify a conversion name that is to be converted.

When COBW3_SET_REPEAT, COBW3_SET_REPEAT_XX or COBW3_SET_REPEAT_XN are used, specify the conversion name in COBW3-CNV-NAME.

When COBW3_SET_REPEAT_NX or COBW3_SET_REPEAT_NN are used, specify the conversion name in COBW3-CNV-NAME-N.

COBW3-CNV-NAME-LENGTH [Optional]

To register a conversion name that has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a conversion name up to the last character other than the blank. However, if the COBW3-CNV-NAME or COBW3-CNV-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Retrieves a conversion name of the specified character string length.

COBW3-CNV-VALUE and COBW3-CNV-VALUE-N

Specify a conversion result (conversion character string).

When COBW3_SET_REPEAT, COBW3_SET_REPEAT_XX or COBW3_SET_REPEAT_NX are used, specify the conversion result in COBW3-CNV-VALUE.

When COBW3_SET_REPEAT_XN or COBW3_SET_REPEAT_NN are used, specify the conversion result in COBW3-CNV-VALUE-N.

COBW3-CNV-VALUE-LENGTH [Optional]

When the conversion character string has a valid blank at the end, specify the byte length of the character string containing the blank.

Value	Explanation
0	Registers a conversion character string up to the last character other than the blank. However, if the COBW3-CNV-VALUE or COBW3-CNV-VALUE-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 1024	Registers a conversion character string of the specified character string length.

To specify a blank as the conversion result, specify blank characters as the conversion character string and specify zero as the conversion character string length, and register the repetition conversion data.

COBW3_SANITIZE_CNV [Optional]

If characters that are vulnerable to a cross site scripting attack are found in conversion data, those characters are automatically replaced. This process is referred to as "sanitizing".

For more details on cross site scripting, refer to Appendix P, Security, in the NetCOBOL User's Guide.

COBW3_SANITIZE_CNV is valid when either COBW3_SET_REPEAT_XX or COBW3_SET_REPEAT_NX is used. However, if the code set is Unicode, COBW3_SANITIZE_CNV is also valid when COBW3_SET_REPEAT_XN or COBW3_SET_REPEAT_NN is used.

Condition name	Value	Explanation
<u>COBW3-SANITIZE-CNV-OFF</u>	LOW-VALUE	Does not sanitize.
COBW3-SANITIZE-CNV-ON	"1"	Sanitize.

Note

The sanitization procedure replaces the five characters that are vulnerable to a cross site scripting attack (&, <, >, ", ') with the following escape characters:

```
& -> &amp;
< -> &lt;
> -> &gt;
" -> &quot;
' -> &#39;
```

As a result, a single character is being replaced with 4-6 characters, increasing the length of the sanitized data. This means that, depending on the content of the unsanitized data, sanitizing data may cause the maximum data length (1024 bytes) set in the Web parameter VALUE to be exceeded. If this happens, the sanitized data is truncated automatically at 1024 bytes.

It is also possible that the escape characters themselves may be truncated. In this case, the vulnerable character is deleted, not replaced. An example is given below.

Example

The unsanitized data area is 1021 bytes long. The first 1020 bytes contain n characters that do not require sanitizing, but the last character is an ampersand (&).

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&    <- Last character is "&"
|                               |
|           1021 bytes           |
```

After sanitizing, this becomes:

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&amp;
|                               |
|           1025 bytes           |
```

This exceeds the 1024-byte limit, but truncating the data to 1024 bytes would interrupt the escape character string (&). For this reason, the ampersand is deleted instead of being replaced.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    <- Trailing "&" is deleted
|                               |
|           1020 bytes           |
```

As a result, the actual length of the data is reduced to 1020 bytes.

Processing result data

COBW3_SANITIZED_CNV_FLAG

If COBW3_SANITIZED_CNV is set, a value that indicates whether or not the replacement procedure (sanitizing) actually took place is set.

COBW3_SANITIZED_CNV_FLAG is valid when either COBW3_SET_REPEAT_XX or COBW3_SET_REPEAT_NX is used. However, if the code set is Unicode, COBW3_SANITIZED_CNV_FLAG is also valid when COBW3_SET_REPEAT_XN or COBW3_SET_REPEAT_NN is used.

Condition name	Value	Explanation
<u>COBW3-SANITIZED-CNV-NON</u>	"0"	Did not sanitize.
COBW3-SANITIZED-CNV-EXIST	"1"	Sanitized.

3.5.7 COBW3_DEL_REPEAT, COBW3_DEL_REPEAT_X and COBW3_DEL_REPEAT_N

These subroutines delete the repetition conversion data that was registered by COBW3_SET_REPEAT, etc. These subroutines have the following functions:

ASCII environment

COBW3_DEL_REPEAT

Deletes the repetition conversion data that was registered so that it corresponded to the conversion name, which was an alphanumeric character string.

Unicode environment

COBW3_DEL_REPEAT_X

Deletes the repetition conversion data that was registered so that it corresponded to the conversion name, which was an alphanumeric character string.

COBW3_DEL_REPEAT_N

Deletes the repetition conversion data that was registered so that it corresponded to the conversion name, which was a national character string.



Note

When two or more repetition conversion data are registered for a specified conversion name, this subroutine deletes all of the registered data.

Format

```
CALL "COBW3_DEL_REPEAT" USING COBW3 .
```

```
CALL "COBW3_DEL_REPEAT_X" USING COBW3 .
```

```
CALL "COBW3_DEL_REPEAT_N" USING COBW3 .
```

Data setting at calling

COBW3-CNV-NAME and COBW3-CNV-NAME-N

Specify a conversion name that is to be deleted.

When COBW3_DEL_REPEAT and COBW3_DEL_REPEAT_X are used, specify the conversion name in COBW3-CNV-NAME.

When COBW3_DEL_REPEAT_N is used, specify the name in COBW3-CNV-NAME-N.

COBW3-CNV-NAME-LENGTH [Optional]

When the conversion name has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a conversion name up to the last character other than the blank. However, if the COBW3-CNV-NAME or COBW3-CNV-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Retrieves a conversion name of the specified character string length.

Processing result data

None

3.5.8 COBW3_INIT_REPEAT

This subroutine initializes all the repetition conversion data that was registered using COBW3_SET_REPEAT, etc.

Format

```
CALL "COBW3_INIT_REPEAT" USING COBW3.
```

Data setting at calling

None

Processing result data

None

3.5.9 COBW3_PUT_TEXT

This subroutine adds line feed codes to a specified character string, and outputs the created data to a WWW browser.

Format

```
CALL "COBW3_PUT_TEXT" USING COBW3.
```

Data setting at calling

COBW3-PUT-STRING

Specify a character string that is to be output.

COBW3-PUT-STRING-LENGTH [Optional]

When the character string has a valid blank at the end, specify the byte length of the character string containing the blank.

Value	Explanation
0	Outputs a character string up to the last character other than the blank. However, if the COBW3-PUT-STRING character string consists of only blank characters, the character string length is assumed to be 0 (only line feed is performed).
1 to 1024	Outputs a character string of the specified character string length.

Processing result data

None

Output of character string data (including line feed codes) to WWW browser

The following figure shows a description example for outputting the character string "<CENTER>Thank you very much for your cooperation. </CENTER>" to a WWW browser:



Example

Usage example

[COBOL program]

```
      :  
000360* Character string set to be output  
000370  MOVE "<CENTER> Thank you very much for your cooperation. </CENTER>"  
000390      TO COBW3-PUT-STRING.  
000400* Output of character string data  
000410  CALL "COBW3_PUT_TEXT" USING COBW3.  
      :
```

[Processing result output to WWW browser]

```
      :  
Thank you very much for your cooperation.  
      :
```

3.6 Execute System Commands

3.6.1 COBW3_SYSTEM

This subroutine executes an arbitrary system command from a Web application.

Format

```
CALL "COBW3_SYSTEM" USING COBW3.
```

Data setting at calling

COBW3-SYSTEMINFO

Specify a target system command and its parameters.

Processing result data

None



A third person may enter the system improperly and destroy the system. To prevent this, do not design COBOL programs that allow a command to be directly entered and executed from a client (WWW browser) by using COBW3_SYSTEM.

3.7 Release CGI Execution Environment Resources

3.7.1 COBW3_FREE

This subroutine releases the resources that were acquired by CGI subroutines. This subroutine must always be called at the end of an application's execution.

Format

```
CALL "COBW3_FREE" USING COBW3.
```

Data setting at calling

None

Processing result data

None

3.8 Acquire Request Information

3.8.1 COBW3_RECEIVE_HEADER

This subroutine acquires an HTTP header that corresponds to a request.

Format

```
CALL "COBW3_RECEIVE_HEADER" USING COBW3.
```

Data setting at calling

COBW3-HEADER-NAME

Specify a header name that is to be acquired.

COBW3-HEADER-NAME-LENGTH [Optional]

When the header name has a valid blank at the end, specify the byte length of the blank-containing header name (character string).

Value	Explanation
0	Retrieves a header name up to the last character other than the blank.
1 to 64	Retrieves a header name of the specified character string length.

Usually header names do not have a blank at the end. Therefore, specify "0."

Processing result data

COBW3-HEADER-VALUE

The header value that corresponds to the requested header name is set.

COBW3-HEADER-VALUE-LENGTH

The byte length of the header value (character string) corresponding to the requested header name is set.

The header names that can be acquired depend on the WWW browser and WWW server being used.

3.8.2 COBW3_GET_REQUEST_INFO

This subroutine acquires various information about a request.

Format

```
CALL "COBW3_GET_REQUEST_INFO" USING COBW3.
```

Data setting at calling

COBW3-REQUEST-INFO-TYPE

Specifies the type of information to be acquired (returned).

Condition name	Value	Explanation
COBW3-URI	LOW-VALUE	Returns a request URI.
COBW3-URL	"1"	Returns a request URL.
COBW3-SERVER-TYPE	"2"	Returns a server type.

Condition name	Value	Explanation
(COBW3-LISTENERTYPE)		
COBW3-VIRTUALPATH	"3"	Returns a virtual path name for the request.
COBW3-PHYSICALPATH	"4"	Returns a physical path name corresponding to the virtual path name for the request.
COBW3-QUERYSTRING	"5"	Returns Web parameters.
COBW3-LANGUAGE	"6"	Returns a list of languages that can be received by the WWW browser.
COBW3-ENCODING	"7"	Returns a list of encoding that can be received by the WWW browser.
COBW3-REQMIMETYPE	"8"	Returns the MIME type of the request.
COBW3-REQUEST-METHOD	"9"	Returns a request method.
COBW3-PATH-INFO	"A"	Returns an extended path name that is sent from the WWW browser.
COBW3-PATH-TRANSLATED	"B"	Returns a physical path name corresponding to an extended path.
COBW3-REMOTE-ADDR	"C"	Returns the IP address of the client.
COBW3-REMOTE-HOST	"D"	Returns the host name of the client.
COBW3-GATEWAY-INTERFACE	"E"	Returns the revision number of the supported CGI.
COBW3-SERVER-NAME	"F"	Returns the host name of the server.
COBW3-SERVER-PORT	"G"	Returns a server-connected port number.
COBW3-SERVER-PROTOCOL	"H"	Returns a server-used protocol.
COBW3-SERVER-PORT-SECURE	"I"	Returns 1 (during secure communication) or 0 (in other cases).

Note

- The MIME type of the request cannot be acquired.
- As the host name of the client, an IP address may be returned.
- Whether the secure communication information can be acquired depends on the WWW server.

Processing result data

COBW3-REQUEST-INFO

The acquired request information is set.

COBW3-REQUEST-INFO-LENGTH

The byte length of the acquired request information (character string) is set.

3.8.3 COBW3_GET_AUTHORIZE

This subroutine acquires authorization information from a WWW browser. The user ID, password, and client IP address are acquired according to the authorization request.

Format

```
CALL "COBW3_GET_AUTHORIZE" USING COBW3.
```

Data setting at calling

None

Processing result data

COBW3-USERID

The acquired user ID is set.

COBW3-USERID-LENGTH

The byte length of the user ID (character string) is set.

COBW3-PASSWORD

The acquired user password is set.

COBW3-PASSWORD-LENGTH

The byte length of the user password (character string) is set.

COBW3-IP-ADDRESS

The acquired IP address of the client is set.

COBW3-IP-ADDRESS-LENGTH

The byte length of the IP address (character string) is set.

COBW3-AUTH-TYPE

The acquired authorization type information is set.

COBW3-AUTH-TYPE-LENGTH

The byte length of the authorization type information (character string) is set.



When the authorization type is anonymous authorization, information other than the IP address is not acquired. When the authorization type is basic authorization, all of the information is acquired and "Basic" is reported as the authorization type. When the authorization type is IIS Microsoft encrypted authorization, all of the information other than user password is acquired and "NTLM" is reported as the authorization type.

3.9 Manipulate Cookie Data

3.9.1 COBW3_SET_COOKIE, COBW3_SET_COOKIE_XX, COBW3_SET_COOKIE_NX, COBW3_SET_COOKIE_XN, and COBW3_SET_COOKIE_NN

These subroutines register the cookie data that is to be sent to a WWW browser. The registered cookie data is used at header output.

These subroutines have the following functions:

ASCII environment

COBW3_SET_COOKIE

Registers, as an alphanumeric character string, the cookie value that corresponds to the cookie name, which is an alphanumeric character string.

Unicode environment

COBW3_SET_COOKIE_XX

Registers, as an alphanumeric character string, the cookie value that corresponds to the cookie name, which is an alphanumeric character string.

COBW3_SET_COOKIE_NX

Registers, as an alphanumeric character string, the cookie value that corresponds to the cookie name, which is a national character string.

COBW3_SET_COOKIE_XN

Registers, as a national character string, the cookie value that corresponds to the cookie name, which is an alphanumeric character string.

COBW3_SET_COOKIE_NN

Registers, as a national character string, the cookie value that corresponds to the cookie name, which is a national character string.



Note

These subroutines must be used before COBW3_PUT_HTML or COBW3_PUT_TEXT is used. If these subroutines are used after COBW3_PUT_HTML or COBW3_PUT_TEXT are executed, the specified information is invalidated.

The character code of the cookie to be sent to the WWW browser is ASCII without regard to the COBOL operation code system. A cookie is sent or received by applications other than COBOL applications, and is mainly used to send or receive key information. Therefore, the cookie should be set using ASCII characters.

Format

```
CALL "COBW3_SET_COOKIE" USING COBW3.
```

```
CALL "COBW3_SET_COOKIE_XX" USING COBW3.
```

```
CALL "COBW3_SET_COOKIE_NX" USING COBW3.
```

```
CALL "COBW3_SET_COOKIE_XN" USING COBW3.
```

```
CALL "COBW3_SET_COOKIE_NN" USING COBW3.
```

Data setting at calling

COBW3-COOKIE-NAME and COBW3-COOKIE-NAME-N

Specify a cookie name that is to be registered.

When COBW3_SET_COOKIE, COBW3_SET_COOKIE_XX or COBW3_SET_COOKIE_XN are used, specify the cookie name in COBW3-COOKIE-NAME.

When COBW3_SET_COOKIE_NX or COBW3_SET_COOKIE_NN are used, specify the cookie name in COBW3-COOKIE-NAME-N.

COBW3-COOKIE-NAME-LENGTH [Optional]

To register a cookie name that has a valid blank at the end, specify the byte length of the blank-containing cookie name (character string).

Value	Explanation
0	Registers a cookie name up to the last character other than the blank. However, if the COBW3-COOKIE-NAME or COBW3-COOKIE-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 64	Registers a cookie name of the specified character string length.

COBW3-COOKIE-VALUE and COBW3-COOKIE-VALUE-N

Specify a cookie value.

When COBW3_SET_COOKIE, COBW3_SET_COOKIE_XX or COBW3_SET_COOKIE_NX are used, specify the cookie value in COBW3-COOKIE-VALUE.

When COBW3_SET_COOKIE_XN or COBW3_SET_COOKIE_NN are used, specify the cookie value in COBW3-COOKIE-VALUE-N.

COBW3-COOKIE-VALUE-LENGTH [Optional]

When the cookie value has a valid blank at the end, specify the byte length of the blank-containing cookie value (character string).

Value	Explanation
0	Registers a cookie value up to the last character other than the blank. However, if the COBW3-COOKIE-VALUE or COBW3-COOKIE-VALUE-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 1024	Registers a cookie value of the specified character string length.

COBW3-COOKIE-EXPIRES [Optional]

Specify an expiry date (year, month, day, hours, minutes, and seconds) of a cookie in Greenwich Mean Time (GMT) mode.

- COBW3-COOKIE-EXPIRES-YEAR (year): 1582 to 9999
- COBW3-COOKIE-EXPIRES-MONTH (month): 01 to 12
- COBW3-COOKIE-EXPIRES-DAY (day): 01 to 31. The end day depends on the month.
- COBW3-COOKIE-EXPIRES-HOUR (hours): 00 to 23
- COBW3-COOKIE-EXPIRES-MIN (minutes): 00 to 59
- COBW3-COOKIE-EXPIRES-SEC (seconds): 00 to 59

Note

If the expiry date was omitted, the related cookie data is deleted at the end of the WWW browser. Cookie data that is being held by the WWW browser can be deleted by specifying a past date.

The difference between Japan Standard Time (JST) and GMT is nine hours. Therefore, GMT is acquired by subtracting 9 from JST.

To specify an expiry date, use a date after October 15 in 1582. If an earlier date is specified, the day of the week cannot be automatically calculated and the specified expiry date is invalidated.

COBW3-COOKIE-DOMAIN [Optional]

Specify a domain in which the specified cookie is valid.

COBW3-COOKIE-PATH [Optional]

Specify a virtual path on which the specified cookie is valid.

COBW3-COOKIE-SECURE [Optional]

Specify the security status. When the security is valid, the WWW browser sends the cookie data only while SSL is being used.

Condition name	Value	Explanation
<u>COBW3-COOKIE-SECURE-OFF</u>	LOW-VALUE	Invalidates the security.
COBW3-COOKIE-SECURE-ON	"1"	Validates the security.

COBW3-COOKIE-MODE [Optional]

Specify the registration mode of the cookie data.

Condition name	Value	Explanation
<u>COBW3-COOKIE-MODE-ADDREP</u>	LOW-VALUE	Adds cookie data if the specified cookie name was not registered.
COBW3-COOKIE-MODE-REPLACE	"1"	Replaces cookie data. If the specified cookie name was not registered, an abnormal end occurs (the cookie data is not replaced).
COBW3-COOKIE-MODE-ADD	"2"	Adds cookie data. If the specified cookie name was already registered, an abnormal end occurs (the cookie data is not added).

Processing result data

None

3.9.2 COBW3_DEL_COOKIE, COBW3_DEL_COOKIE_X and COBW3_DEL_COOKIE_N

These subroutines delete the cookie data that was registered by COBW3_SET_COOKIE, etc.

These subroutines have the following functions:

ASCII environment

COBW3_DEL_COOKIE

Deletes the cookie data that corresponds to the cookie name, which is an alphanumeric character string.

Unicode environment

COBW3_DEL_COOKIE_X

Deletes the cookie data that corresponds to the cookie name, which is an alphanumeric character string.

COBW3_DEL_COOKIE_N

Deletes the cookie data that corresponds to the cookie name, which is a national character string.



Note

These subroutines must be used before COBW3_PUT_HTML or COBW3_PUT_TEXT is used. If these subroutines are executed after COBW3_PUT_HTML or COBW3_PUT_TEXT are used, the specified information is invalidated.

Format

```
CALL "COBW3_DEL_COOKIE" USING COBW3.
```

```
CALL "COBW3_DEL_COOKIE_X" USING COBW3.
```

```
CALL "COBW3_DEL_COOKIE_N" USING COBW3.
```

Data setting at calling

COBW3-COOKIE-NAME and COBW3-COOKIE-NAME-N

Specify a cookie name that is to be deleted.

When COBW3_DEL_COOKIE or COBW3_DEL_COOKIE_X are used, specify the cookie name in COBW3-COOKIE-NAME.

When COBW3_DEL_COOKIE_N is used, specify the cookie name in COBW3-COOKIE-NAME-N.

COBW3-COOKIE-NAME-LENGTH [Optional]

When the cookie name has a valid blank at the end, specify the byte length of the blank-containing cookie name (character string).

Value	Explanation
0	Deletes a cookie name up to the last character other than the blank. However, if the COBW3-COOKIE-NAME or COBW3-COOKIE-NAME-N character string consists of only blank characters, the character string length is assumed to be 0.
1 to 64	Deletes a cookie name of the specified character string length.

Processing result data

None

3.9.3 COBW3_INIT_COOKIE

This subroutine initializes all the cookie data that was registered by COBW3_SET_COOKIE, etc.



Note

This subroutine must be used before COBW3_PUT_HTML or COBW3_PUT_TEXT is used. If this subroutine is used after COBW3_PUT_HTML or COBW3_PUT_TEXT are executed, the specified information is invalidated.

Format

```
CALL "COBW3_INIT_COOKIE" USING COBW3.
```

Data setting at calling

COBW3-COOKIE-INIT-MODE [Optional]

Specify the initialization mode of the cookie data.

Condition name	Value	Explanation
<u>COBW3-COOKIE-INIT-MODE-NORMAL</u>	LOW-VALUE	Initializes all the cookie data.
COBW3-COOKIE-INIT-MODE-REQUEST	"1"	Initializes the cookie data contained in the request data.

Processing result data

None

3.9.4 COBW3_GET_COOKIE, COBW3_GET_COOKIE_XX, COBW3_GET_COOKIE_NX, COBW3_GET_COOKIE_XN, and COBW3_GET_COOKIE_NN

These subroutines retrieve an arbitrary cookie name according to the request cookie data that were acquired using the *COBW3_INIT*, and acquire the cookie value that corresponds to the cookie name.

The functions of these subroutines are as follows:

ASCII environment

COBW3_GET_COOKIE

Retrieves a cookie name that is an alphanumeric character string, and returns the corresponding cookie value, which is an alphanumeric character string.

Unicode environment

COBW3_GET_COOKIE_XX

Retrieves a cookie name that is an alphanumeric character string, and returns the corresponding cookie value, which is an alphanumeric character string.

COBW3_GET_COOKIE_NX

Retrieves a cookie name that is a national character string, and returns the corresponding cookie value, which is an alphanumeric character string.

COBW3_GET_COOKIE_XN

Retrieves a cookie name that is an alphanumeric character string, and returns the corresponding cookie value, which is a national character string.

COBW3_GET_COOKIE_NN

Retrieves a cookie name that is a national character string, and returns the corresponding cookie value, which is a national character string.

To store the acquired cookie value, the en-size blank is added as a padding character.

Format

```
CALL "COBW3_GET_COOKIE" USING COBW3.
```

```
CALL "COBW3_GET_COOKIE_XX" USING COBW3.
```

```
CALL "COBW3_GET_COOKIE_NX" USING COBW3.
```

```
CALL "COBW3_GET_COOKIE_XN" USING COBW3.
```

```
CALL "COBW3_GET_COOKIE_NN" USING COBW3.
```

Data setting at calling

COBW3-COOKIE-NAME and COBW3-COOKIE-NAME-N

Specify a cookie name that is to be retrieved.

When COBW3_GET_COOKIE, COBW3_GET_COOKIE_XX or COBW3_GET_COOKIE_XN are used, specify the name in COBW3-COOKIE-NAME.

When COBW3_GET_COOKIE_NX or COBW3_GET_COOKIE_NN are used, specify the name in COBW3-COOKIE-NAME-N.

COBW3-COOKIE-NAME-LENGTH [Optional]

To retrieve a cookie name that has a valid blank at the end, specify the byte length of the blank-containing cookie name (character string).

Value	Explanation
0	Retrieves a cookie name up to the last character other than the blank. When the COBW3-COOKIE-NAME or COBW3-COOKIE-NAME-N character string consists of only blanks, the character string length is assumed to be 0.
1 to 64	Retrieves a cookie name of the specified character string length.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-NON	"0"	The retrieved name was not found.

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-EXIST	"1"	The retrieved name was found.

COBW3-COOKIE-VALUE and COBW3-COOKIE-VALUE-N

The cookie value that corresponds to the retrieved cookie name is set.

When COBW3_GET_COOKIE, COBW3_GET_COOKIE_XX or COBW3_GET_COOKIE_NX are specified, the found value is set in COBW3-COOKIE-VALUE.

When COBW3_GET_COOKIE_XN or COBW3_GET_COOKIE_NN are specified, the found value is set in COBW3-COOKIE-VALUE-N.

COBW3-COOKIE-VALUE-LENGTH

The character string length (byte length) of the cookie value that corresponds to the retrieved cookie name is set.

3.10 Manipulate Uploaded Files

3.10.1 COBW3_GET_UPLOADFILE_INFO, COBW3_GET_UPLOADFILE_INFO_X and COBW3_GET_UPLOADFILE_INFO_N

These subroutines acquire the information regarding an uploaded file. Details of these subroutines are as follows:

ASCII environment

COBW3_GET_UPLOADFILE_INFO

Retrieves the name of an uploaded file (alphanumeric character string specified in NAME), and acquires information about an uploaded file.

Unicode environment

COBW3_GET_UPLOADFILE_INFO_X

Retrieves the name of an uploaded file (alphanumeric character string specified in NAME), and acquires the information about an uploaded file.

COBW3_GET_UPLOADFILE_INFO_N

Retrieves the name of an uploaded file (national language character strings specified in NAME), and acquires the information about the found uploaded file.

How to write

```
CALL "COBW3_GET_UPLOADFILE_INFO" USING COBW3.
```

```
CALL "COBW3_GET_UPLOADFILE_INFO_X" USING COBW3.
```

```
CALL "COBW3_GET_UPLOADFILE_INFO_N" USING COBW3.
```

Data setting for calling

COBW3-SEARCH-DATA and COBW3-SEARCH-DATA-N

Set the name (NAME) of the upload file to be retrieved (The name specified in "name" of the HTML document (Web page for invoking application) must be set instead of the file name).

For COBW3_GET_UPLOADFILE_INFO and COBW3_GET_UPLOADFILE_INFO_X, set the name to COBW3-SEARCH-DATA.

For COBW3_GET_UPLOADFILE_INFO_N, set the name to COBW3-SEARCH-DATA-N.

COBW3-SEARCH-LENGTH [optional]

If the name (NAME) has a valid blank at the end, set the string length (byte length) of the name including the blank.

Value	Meaning
0	Searches the upload file name using the length up to the last character excluding the blank. However, if COBW3-SEARCH-DATA or COBW3-SEARCH-DATA-N is completely blank, the string length is set to zero for processing.
1 to 1024	Searches the upload file name using the specified string length.

COBW3-NUMBER [optional]

If multiple entities with the same name (NAME) exist in Web parameters, set the order of appearance of names to be searched.

Condition name	Value	Meaning
COBW3-NUMBER-INIT	1	Searches the name that matches first.
-	2 to 9999	Searches the names in the specified order of appearance.

Processing result data:

COBW3-SEARCH-FLAG

Condition name	Value	Meaning
COBW3-SEARCH-FLAG-NON	"0"	The name to be searched does not exist, or size of the uploaded file is 0.
COBW3-SEARCH-FLAG-EXIST	"1"	The name to be searched exists.

COBW3-UPLD-CL-FILE-PATH

Sets the path name of the upload file in the client that corresponds to the retrieved name (NAME).

COBW3-UPLD-CL-FILE-PATH-LENGTH

Sets the character string length (byte length) of the path name of the upload file in the client that corresponds to the retrieved name (NAME).

COBW3-UPLD-CL-FILE-NAME

Sets the file name of the upload file in the client that corresponds to the retrieved name (NAME).

COBW3-UPLD-CL-FILE-NAME-LENGTH

Sets the character string length (byte length) of the file name of the upload file in the client that corresponds to the retrieved name (NAME).

COBW3-UPLD-CONTENT-TYPE

Sets the character string that indicates the Content-type of the upload file that corresponds to the retrieved name (NAME).

COBW3-UPLD-CONTENT-TYPE-LENGTH

Sets the character string length (byte length) of the character string that indicates the content type of the upload file that corresponds to the retrieved name (NAME).

COBW3-UPLD-FILE-SIZE

Sets the uploaded file size (byte length).

Note

When "\" is included in the uploaded file name when the client is UNIX, the file name and the path name cannot be correctly received.

Example

[c.htm (Web page for invoking application)]

```

:
<FORM METHOD="POST" ENCTYPE="multipart/form-data"
  ACTION="sample/action.script">
<P>
Your Name:<INPUT TYPE="text" NAME="NAME1"><BR>
Send file 1:<INPUT TYPE="file" NAME="FILE1"><BR>
<INPUT TYPE="submit" VALUE="Send">
<INPUT TYPE="reset" VALUE="Reset">
</FORM>
:

```

[COBOL source program]

```

:
* Name setup
  MOVE "FILE1" TO COBW3-SEARCH-DATA.
* Acquisition of upload file information
  CALL "COBW3_GET_UPLOADFILE_INFO_X" USING COBW3.
:
  IF COBW3-SEARCH-FLAG-EXIST THEN
* Setup of name of file to be generated
  MOVE "d.tmp" TO COBW3-UPLOADED-FILENAME
* File generation
  CALL "COBW3_GEN_UPLOADFILE_X" USING COBW3
:
* File data processing
:
* Deletion of generated file
  CALL "COBW3_DEL_UPLOADEDFILE" USING COBW3
  END-IF.
:

```

3.10.2 COBW3_GEN_UPLOADFILE, COBW3_GEN_UPLOADFILE_X and COBW3_GEN_UPLOADFILE_N

These subroutines generate an uploaded file by using a specified file name. The details of these subroutines are as follows:

ASCII environment

COBW3_GEN_UPLOADFILEX

Searches the name (NAME) of an alphanumeric character string, and generates the found uploaded file by using the specified file name.

Unicode environment

COBW3_GEN_UPLOADFILE_X

Searches the name (NAME) of an alphanumeric character string, and generates the found uploaded file by using the specified file name.

COBW3_GEN_UPLOADFILE_N

Searches the name (NAME) of a national character string, and generates the found uploaded file by using the specified file name.



Note

Uploaded files must not be generated as executable files because of the threat to security. Especially, uploaded files must not be generated as files that can be automatically executed by the system.

How to write

```
CALL "COBW3_GEN_UPLOADFILE" USING COBW3.
```

```
CALL " COBW3_GEN_UPLOADFILE_X" USING COBW3.
```

```
CALL " COBW3_GEN_UPLOADFILE_N" USING COBW3.
```

Data setting for calling

COBW3-SEARCH-DATA and COBW3-SEARCH-DATA-N

Set the name (NAME) of the uploaded file to be searched (The name specified in "name" of the HTML document [Web page for invoking application] must be set instead of the file name).

For COBW3_GEN_UPLOADFILE and COBW3_GEN_UPLOADFILE_X, set the name to COBW3-SEARCH-DATA.

For COBW3_GEN_UPLOADFILE_N, set the name to COBW3-SEARCH-DATA-N.

COBW3-SEARCH-LENGTH [optional]

If the value (VALUE) has a valid blank at the end, set the string length (byte length) of the value including the blank.

Value	Meaning
0	Searches the name using the length up to the last character excluding the blank.
1 to 1024	Searches the name using the specified string length.

COBW3-NUMBER [optional]

Sets the order of appearance of NAMES to be searched when there are multiple NAMES with the same name in the Web parameters.

Condition name	Value	Meaning
COBW3-NUMBER-INIT	1	Searches the name that matches first.
-	2 to 9999	Searches the name in the specified order of appearance.

COBW3-UPLOADED-FILENAME

Specify the name of the file to be generated on the server.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Meaning
COBW3-SEARCH-FLAG-NON	"0"	The name to be searched does not exist.
COBW3-SEARCH-FLAG-EXIST	"1"	The name to be searched exists.

3.10.3 COBW3_DEL_UPLOADFILE

This subroutine deletes an uploaded file that was generated using a subroutine such as COBW3_GEN_UPLOADFILE.

This subroutine can delete a file that was generated using COBW3_GEN_UPLOADFILE, COBW3_GEN_UPLOADFILE_X or COBW3_GEN_UPLOADFILE_N within the same request. File generation or deletion among two or more requests (two or more pages) cannot be performed.

How to write

```
CALL "COBW3_DEL_UPLOADFILE" USING COBW3.
```

Data setting for calling

COBW3-UPLOADED-FILENAME

Set the name of the file to be deleted.

Processing result data

None

3.11 Other Subroutines

The following subroutines are supported to assure compatibility.



The operation of the following subroutines is assured only when the operation code system of the application is ASCII.

3.11.1 COBW3_NAME

These subroutines retrieve an arbitrary name (NAME) from the Web parameters that were acquired using the *COBW3_INIT*.

These subroutines also acquire the value (VALUE) that corresponds to the name (NAME). To store the acquired value (VALUE), the en-size blank is added as a padding character.

Format

```
CALL "COBW3_NAME" USING COBW3.
```

Data setting at calling

COBW3-SEARCH-DATA

Specify a retrieval name (NAME) that was specified in the "name" in the HTML document (Web page used for invoking the application).

COBW3-SEARCH-LENGTH [Optional]

To retrieve a name (NAME) that has a valid blank at the end, specify the byte length of the name containing the blank (character string).

Value	Explanation
0	Retrieves a name up to the last character other than the blank.
1 to 1024	Retrieves a name of the specified character string length.

COBW3-NUMBER [Optional]

When two or more identical names (NAME) are specified in the Web parameter, specify the occurrence sequence of the retrieval targets.

Condition name	Value	Explanation
<u>COBW3-NUMBER-INIT</u>	1	Retrieves the first matching name.
-	2~9999	Retrieves the name with the specified occurrence sequence.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-NON	"0"	The retrieved name was not found.

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-EXIST	"1"	The retrieved name was found.

COBW3-GET-DATA

The value (VALUE) that corresponds to a retrieval name (NAME) is set.

COBW3-GET-LENGTH

The character string length (byte length) of the value (VALUE) corresponding to the retrieval name (NAME) is set.

3.11.2 COBW3_VALUE

This subroutine retrieves an arbitrary name (NAME) according to the Web parameters that were acquired using the *COBW3_INIT*. When this subroutine is called from an application-invoking Web page having a check box, the subroutine checks, by using VALUE, the item that was confirmed by the WWW browser.

Format

```
CALL "COBW3_VALUE" USING COBW3.
```

Data setting at calling

COBW3-SEARCH-DATA

Specify a retrieval value (VALUE) that is to be retrieved.

COBW3-SEARCH-LENGTH [Optional]

When the retrieval value has a valid blank at the end, specify the byte length of the blank-containing retrieval value (character string).

Value	Explanation
0	Retrieves a value up to the last character other than the blank.
1 to 1024	Retrieves a value of the specified character string length.

COBW3-NUMBER [Optional]

When two or more of the same values (VALUE) are specified in the Web parameter, specify the occurrence sequence of the retrieval targets.

Condition name	Value	Explanation
<u>COBW3-NUMBER-INIT</u>	1	Retrieves the first matching VALUE.
-	2 to 9999	Retrieves the VALUE with the specified occurrence sequence.

Processing result data

COBW3-SEARCH-FLAG

Condition name	Value	Explanation
COBW3-SEARCH-FLAG-NON	"0"	The retrieved VALUE was not found.
COBW3-SEARCH-FLAG-EXIST	"1"	The retrieved VALUE was found.

3.11.3 COBW3_CNV_SET

This subroutine registers the conversion data that is specified in the processing result output Web page that is to be output by *COBW3_PUT_HTML*.

The registered information is referenced at the execution of *COBW3_PUT_HTML*. The conversion name in the processing result output Web page is converted according to the registered conversion data.



If two or more different conversion names are specified in the Web page for processing result output, conversion data must be registered by calling *COBW3_CNV_SET* for each of these conversion names.

Format

```
CALL "COBW3_CNV_SET" USING COBW3.
```

Data setting at calling

COBW3-CNV-NAME

Specify a conversion name that is to be converted.

COBW3-CNV-NAME-LENGTH [Optional]

To register a conversion name that has a valid blank at the end, specify the byte length of the blank-containing conversion name (character string).

Value	Explanation
0	Registers a conversion name up to the last character other than the blank. If the COBW3-CNV-NAME character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Retrieves a conversion name of the specified character string length.

COBW3-CNV-VALUE

Specify a conversion result (conversion character string).

COBW3-CNV-VALUE-LENGTH [Optional]

When the conversion character string has a valid blank at the end, specify the byte length of the character string containing the blank.

Value	Explanation
0	Registers a conversion character string up to the last character other than the blank. However, if the COBW3-CNV-VALUE character string consists of only blank characters, the character string length is assumed to be 0.
1 to 1024	Registers a conversion character string of the specified character string length.

COBW3-CNV-MODE [Optional]

Specify a conversion mode.

Condition name	Value	Explanation
<u>COBW3-CNV-MODE-ADDREP</u>	LOW-VALUE	Adds conversion data if the specified conversion name was not registered.
COBW3-CNV-MODE-REPLACE	"1"	Replaces conversion data. If the specified conversion name was not registered, an abnormal end occurs (the conversion name is not converted).
COBW3-CNV-MODE-ADD	"2"	Adds conversion data. If the specified conversion name was already registered, an abnormal end occurs (the conversion data is not added).

Processing result data

None

3.11.4 COBW3_CNV_DEL

This subroutine deletes the conversion data that was registered by COBW3_CNV_SET.

Format

```
CALL "COBW3_CNV_DEL" USING COBW3.
```

Data setting at calling

COBW3-CNV-NAME

Specify a conversion name that is to be deleted.

COBW3-CNV-NAME-LENGTH [Optional]

When the conversion name has a valid blank at the end, specify the byte length of the blank-containing conversion name (character string).

Value	Explanation
0	Registers a conversion name up to the last character other than the blank. If the COBW3-CNV-NAME character string consists of only blank characters, the character string length is assumed to be 0.
1 to 30	Registers a conversion name of the specified character string length.

Processing result data

None

3.11.5 COBW3_CNV_INIT

This subroutine initializes the conversion data that was registered by COBW3_CNV_SET.

Format

```
CALL "COBW3_CNV_INIT" USING COBW3.
```

Data setting at calling

None

Processing result data

None

3.12 Quantitative Restrictions on CGI Subroutines

Data to be used by CGI subroutines must satisfy the following quantitative restrictions:

Item	Value
Maximum length of Web parameter (*1)	1073741822 bytes
Maximum character string length of Web parameter "NAME"	1024 bytes
Maximum character string length of Web parameter "VALUE"	1024 bytes
Maximum record length of processing result output Web page (file) specified in COBW3-HTML-FILENAME (*2)	1024 bytes
Maximum line length of processing result output Web page after conversion data replacement	3072 bytes

*1 - The data format of the Web parameter is different according to the specification of ENCTYPE of the FORM tag. The data of the file up-loading etc. besides NAME and VALUE are included in the Web parameter. Moreover, when limited memory is available, data might not be able to be received even when the size of data does not exceed the fixed quantity limitation because the Web parameter is acquired using the memory of the machine running the WWW Server. Therefore, you should display the size of the file that can be up-loaded to the Web page for the invoking application, or display not to up-load a file that exceeds the limitation on the Web page for invoking the application if it is necessary. Refer to a book or other resource documenting HTML for details of the data format of the Web parameter.

*2 - The byte length of one line in an HTML document that was created using an HTML editor sold on the market may exceed 1,024 bytes. In this case, shorten the line length to less than 1,024 bytes by using a text editor such as Notepad.

Chapter 4 Creating and Executing Web Applications

This chapter explains the basic method for compilation, linkage, and execution of Web applications using CGI subroutines.

For details of compile options and link options that are not explained in this chapter, refer to the *NetCOBOL User's Guide*.

4.1 Execution Procedure

Create and execute a Web application in the following procedure:

- Compile and link the created programs.

For details, see "*Compilation and Linkage*" in Chapter 4.

- Set the environment.

For details, see the manual of the WWW server to be used.

- Set up the environment for the CGI subroutines.

For details, see "Setting the Environment Variables of CGI Subroutines" in Chapter 4.

- Execute the Web application.

Call the application-invoking Web page (HTML document) that was registered in the WWW server, and activate the Web page.

For details, see "Executing Web Applications" in Chapter 4.

4.2 Compilation and Linkage

Compile and link the COBOL Web application as explained below. When the COBOL Project Manager is used, use the Build utility to perform compilation and linkage.

- Compilation of COBOL source programs

COBOL source programs can be specifically compiled by either of two methods: (1) Execute the WINCOB command and operate the window, or (2) Enter the compile command after the command prompt. The following example explains compilation using the compile command.

- COBOL32 -M MAIN-PROGRAM.COB
- COBOL32 INITIAL-PROGRAM.COB
- COBOL32 BUSINESS-PROGRAM.COB
- COBOL32 END-PROGRAM.COB

For details of compilation using the WINCOB command and for details of compilation commands, refer to the *NetCOBOL User's Guide*.

- Linkage of COBOL object programs

Object programs which are created by compiling COBOL source programs can be linked with each other by either of two methods: (1) Execute the WINLINK command and operate the window, or (2) Enter the link command after the command prompt. The following example explains linkage using the link command.

For details of linkage using the WINLINK command and for details of link commands, refer to the *NetCOBOL User's Guide*.

- Creation (static linkage) of two or more object files into one execution-format file (.EXE file)

```
LINK MAIN-PROGRAM.OBJ INITIAL-PROGRAM.OBJ BUSINESS-PROGRAM.OBJ END-PROGRAM.OBJ  
F3BICIMP.LIB MSVCRT.LIB F3BICWSR.LIB /OUT:EXECUTION-FORMAT-NAME.EXE
```

- Creation (dynamic linkage) of two or more object files into one execution-format file (.EXE) and two or more dynamic link libraries (.DLL)

```
LINK INITIAL-PROGRAM.OBJ F3BICIMP.LIB MSVCRT.LIB /DLL /OUT:INITIAL-PROGRAM.DLL
```

```
LINK BUSINESS-PROGRAM.OBJ F3BICIMP.LIB MSVCRT.LIB F3BICWSR.LIB /DLL /OUT:BUSINESS-PROGRAM.DLL
```

```
LINK END-PROGRAM.OBJ F3BICIMP.LIB MSVCRT.LIB /DLL /OUT:END-PROGRAM.DLL
```

```
LINK MAIN-PROGRAM.OBJ INITIAL-PROGRAM.LIB BUSINESS-PROGRAM.LIB END-PROGRAM.LIB F3BICIMP.LIB MSVCRT.LIB /OUT:EXECUTION-FORMAT-NAME.EXE
```

For details of linkage using the WINLINK command and for details of link commands, refer to the NetCOBOL User's Guide.



Note

The CGI subroutines do not have backward compatibility. Therefore, libraries (COBW3.cbl) whose version or level is later than that of the CGI subroutines to be used cannot be used. CGI subroutines whose version or level is earlier than that of the libraries that were copied into Web applications must not be linked.

4.3 Setting the Environment Variables of CGI Subroutines

To execute a Web application that uses CGI subroutines, the necessary execution environment information must be previously set in system environment variables or in the standard execution initialization file (COBOL85.CBR).

4.3.1 Information That Must Be Set in System Environment Variables

PATH

The path to any dynamic library (.DLL) that is to be called by the Web application must be specified.

4.3.2 Information That Must be Set In System Environment Variables Or Execution Initialization File

@CBR_ATTACHE_TOOL=TEST [Start parameter]

It is specified to start the debugger from the program which wants to be debugged. When the Web application made in COBOL is debugged by using the debugger, it is necessary to set this environment variable.

Please refer to Chapter 5, "Confirming Application Operation" for details.

@MessOutFile=file-name

Specifies a file into which execution-time messages that are output by the COBOL run time system are to be stored. When this file name is specified, message box display on the screen is suppressed. A file name can be specified with an absolute path or relative path. When a relative path is specified, the path is assumed to be a relative path from the current folder. If there is an existing file having the same name, output messages are written to the existing file.

@WinCloseMsg=OFF

Specifies whether a confirmation message is to be displayed (ON) or not (OFF) when the window is closed. To execute a Web application, specify "OFF" (no display of confirmation message).

@CBR_CGI_LOGFILE=log-file-name

Specifies a file in which the log information that is output by CGI subroutines is to be recorded. This log file is useful to investigate error causes, etc. The log file name must be specified with an absolute path on the server. If the log file name is not specified, no log data is output.

@CBR_CGI_SEVERITY=severity-level

Specifies the severity level of the error messages log data that can be output by CGI subroutines. The values listed in the following table can be specified. If a value other than the specifiable values is specified or if no value is specified, the severity level is regarded as 0.

Severity level	Message output to log file
0	Only fatal error data is output.
1	Fatal error data and error data are output.
2	Fatal error data, error data, and warning data are output.
3	Fatal error data, error data, warning data, and CGI-subroutine trace data are output.

 **Note**

When a larger value is specified, the speed of an application operation slows significantly. Therefore, specify a value so that only the data of the required severity level is output during operation. Especially, level 3 should only be specified when investigating error causes.

For details of other execution environment information, refer to the *NetCOBOL User's Guide*.

4.4 Executing Web Applications

Place the created Web application in the physical path that corresponds to the Web Server's virtual path, and activate the application by using the Web page created for invoking the application.

Chapter 5 Confirming Application Operation

The operation of a Web application can be confirmed when testing or you can check if a problem occurs by the following methods:

- Examine the log data.
- Check the operation by using the interactive debugger.
- Confirm the error that is detected by the CGI subroutine.
- Confirm the in-execution data in the presentation format.

This chapter explains each confirmation method.

5.1 Examine the Log Data

Each CGI subroutine has a feature for outputting log data to a log file. This log data is very useful because it can be used to change environment variables without changing the application. For an explanation of the environment variable settings, see "*Setting the Environment Variables of CGI Subroutines*" in Chapter 4. Because this log data is collected from CGI subroutines, the log data does not indicate the trace data on the application itself, but indicates the approximate status of the application. The configuration of the log file is as follows:

```
process-id thread-id year-month-day hours:minutes:seconds severity-level message
```

An example of the contents of a log file follows:

```
0000000188 0000000187 1999-02-02 11:48:07 02 COB-06310: COBW3: The specified conversion name is
already registered. The already registered conversion data is validated.
0000000188 0000000081 1999-02-02 11:48:10 01 COB-04470: COBW3: The specified conversion data cannot
be converted because it was not previously registered.
```

To acquire the trace data from the application, use the COBOL TRACE function. For details of the TRACE function, refer to the "*NetCOBOL Debugging Guide*".

5.2 Checking Operation by Interactive Debugger

To debug a COBOL Web application by using the interactive debugger, the debugger must be activated from the program that is to be debugged.

The Web application can be remotely debugged from the client which starts a WWW browser by using a remote debugger. Please refer to "*NetCOBOL Debugging Guide*" for the usage of a remote debugger.

Start debugging by using a debugger in the following procedure:

1. Compile and link a Web application for debug.

Create a Web application by specifying a compile option and a link option for debugging. For explanations of the compilation and linkage for debugging, refer to the "*NetCOBOL Debugging Guide*".

2. Set the run-time environment information.

To start the debugger from a program that is to be debugged, the run-time environment information shown below must be previously set in the system environment variable or in the run-time initialization file (COBOL85.CBR). When the run-time environment information is set in the system environment variable, the system must be restarted.

```
@CBR_ATTACH_TOOL=TEST [start-parameter]
```

Specifies whether the debugger will start (TEST) at application execution. After the "TEST," a debugger start parameter can be specified. For an explanation of the start parameters, refer to the "*NetCOBOL Debugging Guide*".

3. Execute a log-on to the computer in which the Web application to be debugged is to be executed.
4. Start the Web application from a WWW browser. This WWW browser need not operate on the computer in which the Web application to be debugged is to be executed.

5. When the Web application is started, the debugger is also automatically started. After the debugger has started, specify a debugging information file storage folder and necessary information from the "Start Debugging" dialog, and start the debugging.

The subsequent operation is the same as that of the ordinary debugging using a debugger. For an explanation of the debugger, refer to the "NetCOBOL Debugging Guide".

Note

While you are debugging a Web application using the debugger, any operation of the Web application takes longer than usual. Thus, a timeout may occur during debugging depending on the WWW browser. Set an appropriate timeout value for the WWW Browser if possible. Set an appropriate timeout value for the WWW Server if possible.

5.3 Confirmation of Errors Detected by CGI Subroutines

To confirm an error that is detected by a CGI subroutine, debug mode must be set in the program as follows immediately before `COBW3_INIT` is called:

```
000190      :
000190      SET COBW3-DMODE-DBG TO TRUE.
000200      CALL "COBW3_INIT" USING COBW3.
000200      :
```

This setting displays an error message in the WWW browser if an error is detected in the CGI subroutine.

If a header output parameter other than `COBW3-CONTENT-TYPE-NON` is set, `COBW3-CONTENT-TYPE-HTML` is assumed and `COBW3_INIT` declares the Content-type.

To set the debug mode, the program must be modified and the Web application must be created again. When the debug mode is set, the error messages that are displayed in the WWW browser can be confirmed also using log data. Therefore, to confirm error messages without modifying the application, log data should be used.

Note

When the operation code system of the Web application is Unicode, the debug mode setting is invalid. Check the operation by confirming the log data, etc.

5.4 Confirmation of In-execution Data in Presentation Format

To confirm the in-execution data in the presentation format, use `COBW3_PUT_TEXT`. That is, specify the target data in `COBW3_PUT_TEXT`, and display the data in the WWW browser. Then, by checking the displayed data, debug the application.

Note

During debugging by `COBW3_PUT_TEXT`, data of other than presentation format (e.g., binary data) is not displayed. Moreover, debugging efficiency is low because compilation and execution are repeated frequently. To solve these problems, the application should be debugged by confirming the in-execution data by using the interactive debugger.

Chapter 6 Sample

This chapter explains the supplied sample program.

This sample shows the basic use of a Web application that receives Web parameters by using CGI subroutines, processes the received parameters, and outputs the processing result HTML page.

6.1 Use of the Sample Program

Simple questionnaire using CGI subroutines

This example is a program that exchanges information between a WWW Server and a WWW Browser by using CGI subroutines.

Outline

The Entry Field, Check Box, and Radio Button values are received from the WWW browser, a Web page for processing result output is created on the WWW server according to the received information, and the created Web page is displayed on the WWW browser.

Supplied programs

- CGIBASIC.COB (COBOL source program)
- SAMPLE01.HTM (Web page for invoking application)
- SAMPLE01_1.HTM (Web page for processing result output)
- COBOL85.CBR (initial file for execution)

CGI subroutines used

COBW3_INIT, COBW3_GET_VALUE, COBW3_SET_CNV, COBW3_PUT_HTML, COBW3_PUT_TEXT, and COBW3_FREE

Program compilation

Compile the program by using the COBOL32 command in which the -M option and -I option (for specifying the folder for storing the CGI subroutine library) are specified.

The -I option need not be specified if the CGI subroutine library file (COBW3.cbl) was already copied from the COBOL-installed folder to the current folder.

```
COBOL32 -M -I"C:\Program Files\Fujitsu NetCOBOL for Windows\COBOL" CGIBASIC.COB
```

Program linkage

Import libraries F3BICWSR.LIB, F3BICIMP.LIB, and MSVCRT.LIB are added.

```
LINK /OUT:CGISMP01.EXE CGIBASIC.OBJ F3BICWSR.LIB F3BICIMP.LIB MSVCRT.LIB
```

Conditions for program execution

The path to the Web application for which the FORM tag ACTION attribute is specified in the Web page for invoking the application (CGISMP01.HTM) must be changed to a virtual path on the WWW Server to be actually used.

Program execution

Copy the CGISMP01.EXE, SAMPLE01.HTM, SAMPLE01_1.HTM, and COBOL85.CBR files to the specified folder on the WWW Server.

From the WWW browser, enter the URL of the WWW Server and specify SAMPLE01.HTM. A simple questionnaire screen is displayed.

Enter or select necessary data, and click the [Invoke CGI Program] button. The entered data is sent to the WWW Server, and the processing result is displayed on the WWW Browser.

To cancel the entered data before the data is sent to the WWW server, click the [Reset Values] button.

The processing result HTML Web page should be displayed noting which data you entered and which check box and radio button you selected.

Appendix A Questions and Answers

This appendix explains questions and answers regarding the use of CGI subroutines.

Q1.

An error message was displayed in the WWW browser. What action should be taken?

A1.

Check that the WWW Server setting and Web application activation is the same as the program name specified in the FORM tag (ACTION attribute) on the application-invoking Web page.

Also check that the WWW Server is correctly set up, the program name is correct, and the storage location of the processing result output Web page is correct.

Q2.

Should data be protected from outside people?

A2.

In communication between a client and server, data may be read or changed by outside people. To prevent such problems that may affect system operation, an appropriate network security system such as Secure Socket Layer (SSL) should be installed.

Consult with the server administrator.

Q3.

A debugger was activated for execution time debugging, but the debugger screen was not displayed on the client. Why?

A3.

The debugger screen is usually displayed on the Web Server monitor, and is not displayed on the client.

Q4.

Files were not correctly input or output by a Web application. Why?

A4.

To read or write a file, the authorizations for reading and/or writing the file and the folder containing the file must be previously assigned to the ID of the user who executes the Web application.

Q5.

How can I dynamically change the display contents of a Web application?

A5.

(1) Use COBW3_SET_CNV, etc.

Write "//COBOL//conversion-name//COBOL//" in the variable part on the Web page for processing result output. Register the conversion character string by using CGI subroutine COBW3_SET_CNV, etc. Then, change the contents of the Web page by using COBW3_PUT_HTML.

(2) Use COBW3_PUT_TEXT.

Usually, the data that is to be written to an output file such as an HTML document can be written using COBW3_PUT_TEXT.

Therefore, prepare different COBW3_PUT_TEXT's for respective conditions in the program. Processing result data can be dynamically output by specifying the data name in the COBW3_PUT_TEXT.

(3) Divide one page into two or more files.

An output file can be divided into two or more files.

For example, call COBW3_PUT_HTML by specifying the file name of the first half of the output Web page. To process the last half of the output Web page, prepare various files that are described according to various conditions. Then, call COBW3_PUT_HTML by changing the file name according to the conditions in the program.

Thus, the last half of the Web page can be dynamically changed. When a page is divided into parts, two or more similar files need not be prepared.

Further, pages depending on various conditions can be created by combining these functions, i.e., (1) to (3).

Q6.

Why does the Web application not operate normally?

A6.

Check that COBW3_INIT is specified at the start of the Web application.

Then, collect log data and check the error information. For an explanation of the operation check method, see Chapter 5, "*Confirming Application Operation*."

Q7.

The following message was displayed in the WWW Browser. Why?

Server Error

The server encountered an internal error or mis-configuration and was unable to complete your request.

Alternatively, the Web application returned only part of the Web header. Why?

A7.

There may be an error in the Content type. Review the COBW3-CONTENT-TYPE value that was specified in COBW3_PUT_HEAD.

When an HTML document is output

COBW3_PUT_HEAD is not called. Or, specify either of the default value (LOW-VALUE) or COBW3-CONTENT-TYPE-HTML for a value of COBW3-CONTENT-TYPE if you call COBW3_PUT_HEAD.

When text data is output

COBW3_PUT_HEAD is called by specifying COBW3-CONTENT-TYPE-TEXT.

Q8.

When is a Content-type declaration message displayed in a WWW browser?

A8.

When COBW3-DMODE-DBG is specified in COBW3_INIT, the header information such as Content-type is displayed, as debug data, in the WWW Browser.

Such a message is not displayed unless COBW3-DMODE is specified.

Q9.

When a Web application was executed, the message from the file download was displayed. Why?

A9.

If a Web application has a Content-type declaration error, such a message may be output.

Check that the Content-type was declared correctly.

Q10.

Are there any conditions that must be satisfied when creating COBOL programs for Web applications?

A10.

Web applications can use almost all of the functions of COBOL. However, note that the following functions for screen operation cannot be used:

Presentation file module (Screen handling module)

Screen handling module

ACCEPT/DISPLAY function

(The operation function of environment variables, date, and time can be used, however.)

For details of the Web parameter reception and referencing, and processing result output, see Chapter 3, "*Use of CGI Subroutines*."

Q11.

Can the COBOL debugger be used?

A11.

The COBOL debugger can be used also in the Web environment. See "*Checking Operation by Interactive Debugger*" in Chapter 5.

Q12.

A Status-code that was sent from the WWW Server was displayed in the WWW browser. What is its meaning?

A12.

See "HTTP Basics" in Appendix A, "*For beginners Developing Web Applications*" in the "*NetCOBOL Web Guide*."

Q13.

Can the expiration time (timeout period) be controlled?

A13.

Expiration time control is a function of the WWW Server settings. See the manual of the WWW Server.

Q14.

Can a Web application using the frame function be used in a Web application?

A14.

To use the frame function, special settings or special processing need not be described in the Web application.

If the WWW Browser supports the frame function, the frame function can be used by describing the frame.

Q15.

What information does the IPP interface file contain?

A15.

It contains script names, for example.

For details, see the manual of the IPP.

Q16.

Why can't COBW3_CHECK_VALUE, etc. correctly retrieve the VALUE specified with the <INPUT> tag?

A16.

If NAME was omitted in the INPUT tag, the VALUE of the INPUT tag is not set in the Web parameter in some WWW Browsers. If the INPUT tag requires the VALUE, always specify the NAME.

Q17.

Why can't NAME or VALUE be correctly retrieved by COBW3_GET_VALUE or COBW3_CHECK_VALUE, etc?

A17.

If the length of the retrieval-target character string that is contained in the Web parameter exceeds the quantitative limit, the CGI subroutine processes the character string within the range of the quantitative limit. Therefore, the expected result may not be obtained.

Q18.

A file that was specified with an absolute path could not be accessed. Why?

A18.

Check that the storage destination of the COBOL file to be used when the Web application is executed is specified correctly according to the drive configuration of the server machine to be used.

Q19.

Can resources in the network environment be accessed?

A19.

At the time of the Web application execution, the network drive configuration when the user logs in is invalid.

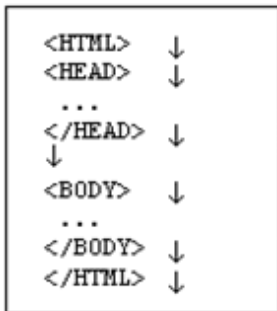
The network environment resources to be used at execution must be accessed by specifying UNC. For details, see "Programs *Running under a Service*," of the *NetCOBOL User's Guide Chapter 21*.

Q20.

Error message "No data is contained in the document." was displayed on the WWW browser. Why?

A20.

Always add one blank line before the <BODY> tag on the Web page to which the processing result is to be output from the Web application.



Q21.

How can the environment variables required to start the WWW Server be set?

A21.

Press the [Start] button, select [Settings], [Control Panel], [System], and then [Environment]. Add the necessary environment variable to the "System environment variable (S):" field, and reboot the system.

Alternatively, prepare the execution initial file (COBOL85.CBR) in which the necessary execution environment information is described, and then place the file in the folder that contains the COBOL Web application (.exe).

Q22.

Error message "F3BIPRCT.DLL was not found." is displayed in the server. Why?

A22.

Check that the NetCOBOL run-time system was installed correctly.

When a Web application is executed, the specified user environment variables are not validated.

To solve this problem, previously specify the installation folder of the COBOL run-time system in the system environment variable "PATH" of the server machine.

To validate the setting of the system environment variable, reboot the system.

Q23.

How can the display of a specified command on a WWW browser be suppressed when starting a Web application from a batch file?

A23.

Specify "@ECHO OFF" at the head of the batch file.

Q24.

What should be done if no response is received from the WWW Server?

A24.

The WWW Server may have a window input wait state or message box input wait state.

To execute a Web application on the WWW Server, describe the environment variable information (listed below) in the execution initial file or in the system environment variables.

Specifying the environment variable information prevents the window or message box from being displayed. Therefore, the operator input wait state is prevented.

Environment variable information	Explanation
@MessOutFile=File name	Outputs a COBOL execution-time message to a specified file.
@WinCloseMsg=OFF	Does not display a window closing message.

For details regarding various environment variables, refer to the NetCOBOL User's Guide.

The following functions that will display a window must not be used from a COBOL program:

- Presentation file module (screen handling function)
- Screen handling module
- ACCEPT/DISPLAY function

(The operation function of environment variables, date, and time can be used, however.)

Q25.

The COBOL debugger cannot be activated. Why?

A25.

Check the following conditions:

Environment variable @CBR_ATTACH_TOOL=TEST was set.

The COBOL Tool Attaching Service was started.

Q26.

The COBOL debugger started, but debugging did not start. Why?

A26.

Display the [Debugging information] page in the [Start debugging] dialog box, and check that [Debugging information file storage folders] is correctly set.

Appendix B Error Handling

If an error is detected during execution of a CGI subroutine, the CGI subroutine handles the error.

An error message is output in the following format:

```
COB-message-number:COBW3:message-text
```

COB-message-number: Indicates a message serial number.

message-text: Indicates the contents of the message.

The following table lists the message numbers and text of messages which may be output:

Message number	Message text
01100	It is thought that the work area or the object was destroyed. Check the object of the Web subroutine and the usage of the area.
01500	The Web subroutine cannot operate correctly in the calling COBOL application operation code system.
01501	The operation state of the calling application is abnormal. The Web subroutine cannot operate normally.
02050	Search character string length specified as a negative value. The processing continues as if 0 was specified.
02051	A value was specified that exceeds the limit for the character string length of the search string. The processing continues assuming that the maximum length was specified.
03000	Web application was called by a method without GET or POST.
03001	COBW3_INIT was called two times or more. Please call COBW3_FREE a second time before the call.
03002	The work area of the Web subroutine could not be acquired.
03003	The length of data retrieved by the GET method exceeds the limit. The characters are truncated at the maximum.
03020	The data that was passed by POST or GET was not acquired successfully.
03030	Failed in the acquisition of the work area of the Web subroutine.
03040	Uploaded file information could not be acquired.
03041	The Web parameter can't be processed because failed to acquire header information.
03042	multipart/form-data is specified for ENCTYPE of the FORM tag though the method is things except POST. The Web parameter and uploaded file information could not be acquired.
03045	An error in processing the output to the browser.
03065	Environment variable @CBR_WEB_OUT_CODE was set incorrectly. It is assumed that the conversion code was not set
03200	Conversion from UTF-8 to UCS-2 failed.
03201	Conversion from UCS-2 to UTF-8 failed.
03700	A code was specified that could not be recognized as a character. Processing continues.
03701	EUC and SJIS character strings are intermingled. Processing continues.
03850	Cookie was not acquired successfully.
04000	There is no Web parameter.
04001	The Web parameter length is 0.
04006	A negative value or "0" was specified for COBW3-NUMBER. The processing continues as if 1 was specified.
04008	The length of VALUE data exceeds the limit. The string is truncated to the supported length.
04009	The data passed with POST or GET method was nothing.

Message number	Message text
04400	The conversion name is not specified. Please set the conversion name.
04401	A negative value was specified for character string length of the conversion name. The processing continues as if 0 was specified.
04402	A value was specified that exceeds the limit for character string length of the conversion name. The processing continues assuming the maximum length was specified.
04408	The length of VALUE data exceeds the limit. The string is truncated to the supported length.
04450	Failed to delete the variable name specified in COBW3-CNV-NAME because it could not be found in the registered information.
04470	Failed to change the variable name specified in COBW3-CNV-NAME because it could not be found in the registered information.
04480	Failed to delete the variable name specified in COBW3-CNV-NAME.
05000	Failed in the execution of the system command.
05001	The system command is not specified.
05500	The file name is not specified. Please set the file name.
05501	There is no uploaded file information.
05510	Failed to acquire the path name in the client of the uploaded file.
05511	The character string length of the path name in the client of the uploaded file exceeded the limitation. The string is truncated to the supported length.
05520	Failed to acquire the file name in the client of the uploaded file.
05521	The character string length of the file name in the client of the uploaded file exceeded the limitation. The string is truncated to the supported length.
05530	Failed to acquire the content type of the uploaded file.
05531	The character string length of the content type of the uploaded file exceeded the limitation. The string is truncated to the supported length.
05550	The file was not generated because the specified file name had already existed.
05551	Failed to generate the file.
05552	Failed to generate data in the subroutine. The generated file can't be deleted with COBW3_DEL_UPLOADEDFILE.
05580	The specified file name is not a file generated in the same thread. The file was not deleted.
05581	Failed to delete the specified file.
06100	An I-O error occurred in the OPEN process of the specified HTML document file.
06101	An I-O error occurred in the READ process of the specified HTML document file.
06102	The length of the specified HTML document file is 0.
06105	A conversion name in the HTML document was not registered using COBW3_CNV_SET. Please register the conversion data using COBW3_CNV_SET.
06106	Failed in the output because the output work area of the HTML document was insufficient. Please confirm the conversion data length corresponding to the conversion name.
06107	Failed in the output because an error was found in the specified form of the conversion name within the HTML document.
06108	The string length of the specified conversion name is too long. Check the conversion name.
06120	Failed to output the HTML document to the WWW server.

Message number	Message text
06130	The number of characters in one line of the HTML document exceeds the limit. The string is truncated to the supported length.
06300	The conversion name is not specified. Please set the conversion name.
06301	A negative value was specified for character string length of the conversion name. The processing is continued regarding for 0 to have been specified.
06302	A value was specified that exceeds the limit for character string length of the conversion name. The processing continues assuming the maximum length was specified.
06303	The conversion character string is not specified. Please confirm the conversion character string.
06304	A negative value was specified for character string length of the conversion character string. The processing continues assuming 0 was specified.
06305	A value was specified that exceeds the limit for character string length of the conversion character string. The processing continues assuming the maximum length was specified.
06310	The specified variable name has already been registered. The old conversion information remains effective.
06320	Failed in registration because the registration work area for the conversion information was insufficient. Please register again after calling COBW3_CNV_INIT and initializing conversion information.
06330	Failed to delete conversion information.
06502	Unexpected error occurred in writing data on standard output.
06503	A negative value was specified for the length of the string to be output. The processing continues assuming 0 was specified.
06504	A value was specified that exceeds the limit for output character string length. The processing continues assuming the maximum length was specified.
06700	COBW3_PUT_HTML or COBW3_PUT_TEXT has already been called. The output request for the header is invalid.
06704	A negative value was specified for COBW3-PUT-HEAD-LENGTH. The processing continues assuming 0 was specified.
06706	A value was specified that exceeded the limit for COBW3-PUT-HEAD-LENGTH. The processing continues assuming the maximum length was specified.
06708	The object output as a header was not specified.
06710	A value that had already been output and a different value were specified for COBW3-CONTENT-TYPE.
06714	A value that had already been output and a different value were specified for COBW3-STATUS-CODE.
06730	An error not anticipated in the declaration of Content-type occurred.
06734	An error not anticipated in the declaration of Status-code occurred.
06736	Unexpected error occurred in declaration of specified header.
06738	An error not anticipated in the declaration of Content-length occurred.
06742	Failed in the output of the specified header.
06746	The output of the header could not be completed.
07010	The user name was not able to be acquired.
07011	The character string length of the user name exceeded the limitation. The maximum length is made effective.
07020	The password was not able to be acquired.
07021	The character string length of the password exceeded the limitation. The maximum length is made effective.
07030	The IP address was not able to be acquired.
07031	The character string length of the IP address exceeded the limitation. The maximum length is made effective..

Message number	Message text
07040	The authentication type was not able to be acquired.
07041	The character string length of the authentication type exceeded the limitation. The maximum is made effective.
08004	A negative value was specified for COBW3-HEADER-NAME-LENGTH. The processing continues assuming 0 was specified.
08005	The script-name could not be acquired.
08006	A value was specified that exceeded the limit for COBW3-HEADER-NAME-LENGTH. The processing continues assuming the maximum length was specified.
08008	COBW3-HEADER-NAME is not specified.
08010	Failed to acquire the HTTP header.
08020	The length of COBW3_HEADER_VALUE exceeds the limit. The string is truncated to the supported length.
08024	The HTTP header specified for COBW3_HEADER_NAME could not be found.
08030	The length of the corresponding value of the key-word exceeds the limit. The maximum length is used.
08101	The key-word was not specified. Please set the key-word.
08105	The key-word to be retrieved was not found in the interface file.
08108	The name of the interface file could not be acquired.
08110	An I-O error occurred by OPEN process of the interface file.
08114	An I-O error occurred by READ process of the interface file.
08118	An I-O error occurred by CLOSE process of the interface file.
08130	The length of the corresponding value of the key-word exceeds the limit. The maximum length is used.
08150	An I-O error occurred in Web subroutine.
08500	COBW3_PUT_HTML or COBW3_PUT_TEXT has already been called. Operation to cookie is invalid.
08510	The cookie name is not specified. Set the cookie name.
08512	A negative value was specified for the string length of the cookie name. The processing continues assuming 0 was specified.
08514	A value was specified that exceeded the limit for the string length of the cookie name. The processing continues assuming the maximum length was specified.
08520	The cookie value is not specified. Set the cookie value.
08522	A negative value was specified for the string length of the cookie value. The processing continues assuming 0 was specified.
08524	A value was specified that exceeded the limit for the string length of the cookie value. The processing continues assuming the maximum length was specified.
08530	Failed to change the specified cookie data because it was not registered.
08532	The specified cookie name has already been registered. The old cookie data remains effective.
08540	A date before Oct 15, 1582 is not valid as a cookie period. The period is assumed not to have been specified.
08541	The year for the cookie period is not specified correctly. The period is assumed not to have been specified.
08542	The month for the cookie period is not specified correctly. The period is assumed not to have been specified.
08543	The day for the cookie period is not specified correctly. The period is assumed not to have been specified.
08544	The hour for the cookie period is not specified correctly. The period is assumed not to have been specified.
08545	The minute for the cookie period is not specified correctly. The period is assumed not to have been specified.
08546	The second for the cookie period is not specified correctly. The period is assumed not to have been specified.

Message number	Message text
08550	An invalid value was specified for the cookie security. The security is assumed not to have been specified.
08560	An invalid value was specified for the cookie registration type. The registration type is assumed not to have been specified.
08570	Failed in registration because the registration work area for the cookie data was insufficient. Please register again after calling COBW3_INIT_COOKIE to initialize conversion information.
08610	Failed to delete the specified cookie data because it was not registered.
08620	Failed to delete the specified cookie data.
08710	An invalid value was specified for the initialization type of cookie information. The initialization type is assumed not to have been specified.
08720	Failed in initialization of the request data by cookie information because the registration work area for the cookie information was insufficient.
08810	The length of the cookie value exceeded the limit. The string is truncated to the supported length.
08820	Cookie data was not sent from the client.
09001	The correct value is not specified for COBW3_REQUEST_INFO_TYPE.
09010	Failed to acquire the request.
09020	The length of COBW3_REQUEST_INFO exceeded the limit. The string is truncated to the supported length.
09500	COBW3_CNV_SET can't be used in the Unicode environment.
09501	COBW3_CNV_DEL can't be used in the Unicode environment.
09502	COBW3_NAME can't be used in the Unicode environment.
09504	COBW3_VALUE can't be used in the Unicode environment.
09520	This subroutine can't be used in the ASCII environment.
99999	Can't open message catalog.

Appendix C Conceptual Diagram of Web Application Creation

