



Microsoft® Windows® 95
Microsoft® Windows® 98
Microsoft® Windows® Me

Microsoft® Windows NT®
Microsoft® Windows® 2000
Microsoft® Windows® XP

B1JW-5391-01Z2

NetCOBOL for Windows V7.0

Webサブルーチン使用手引書



Net  COBOL

 FUJITSU

まえがき

“COBOL Webサブルーチン 3.1”は、以下に示すシステムの32Bitモードで動作し、WWWサーバで動作する32Bitアプリケーションの開発環境および運用環境を提供します。

Microsoft(R) Windows NT(R) Server Network operating system Version 4.0

Microsoft(R) Windows NT(R) Server Network operating system,Enterprise Edition
Version 4.0

Microsoft(R) Windows(R) 2000 Server operating system

Microsoft(R) Windows(R) 2000 Advanced Server operating system

製品の呼び名について

本書に記載されている製品の名称を、以下のように略して表記します。

「Microsoft(R) Windows NT(R) Server Network operating system Version 4.0」

「Windows NT(R)」または「Windows NT(R) 4.0」

「Microsoft(R) Windows NT(R) Server Network operating system,Enterprise Edition
Version 4.0」

「Windows NT(R)」または「Windows NT(R) 4.0」

「Microsoft(R) Windows(R) 2000 Server operating system」

「Windows(R) 2000」

「Microsoft(R) Windows(R) 2000 Advanced Server operating system」

「Windows(R) 2000」

「Microsoft(R) Internet Information Server 4.0」

「IIS」または「IIS 4.0」

「Microsoft(R) Internet Information Server 5.0」

「IIS」または「IIS 5.0」

「Microsoft(R) Internet Explorer」

「IE」

「Netscape Enterprise Server」

「NES」

「Fujitsu Interstage Application Server」

「Interstage」

「Fujitsu Interstage Application Server InfoProvider Pro」

「IPP」

本書の目的

本書は、COBOL Webサブルーチン 3.1を利用したCOBOLプログラムの作成、そのプログラムの実行およびデバッグの方法について説明しています。

COBOLプログラムの作成、翻訳、実行およびデバッグについては、“NetCOBOL 使用手引書”をあわせてお読みください。

COBOLの文法規則については、“COBOL文法書”をお読みください。

その他の注意事項

本書で使用している“Interstage”は、Fujitsu INTERSTAGE Application Server V4.0以前では“INTERSTAGE”に置き換えてお読みください。

登録商標について

本書に記載されている登録商標を、以下に示します。

Microsoft、Windows、Windows NTは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

Netscape、Netscape Navigator、Netscape Enterprise Serverは、米国Netscape Communications Corporationの米国およびその他の国における商標または登録商標です。

Oracle、Web Request Brokerは、米国ORACLE Corporationの商標または登録商標です。
INTERSTAGE、Interstageは、富士通株式会社の登録商標です。
その他の会社名または製品名は、それぞれ各社の商標または登録商標です。

Microsoft Corporationのガイドラインに従って画面写真を使用しています。

2002年7月

All Rights Reserved, Copyright(C) 富士通株式会社 1997-2002

目次

第1章	COBOL Webサブルーチンについて	1
1.1	COBOL CGIサブルーチン	2
1.1.1	CGIの概要	2
1.1.2	CGIの処理の流れ	2
1.1.3	COBOL CGIサブルーチンとは	3
1.2	COBOL ISAPIサブルーチン	4
1.2.1	ISAPIの概要	4
1.2.2	ISAの構成	4
1.2.3	COBOL ISAPIサブルーチンとは	5
1.3	COBOL SAFサブルーチン	6
1.3.1	SAFの概要	6
1.3.2	COBOL SAFサブルーチンとは	6
第2章	CGIアプリケーションの作成と実行	7
2.1	必要なもの	8
2.2	作成するもの	9
2.2.1	CGIサブルーチンを使用したCOBOLプログラム	9
2.2.2	呼出し用ページ	12
2.2.3	結果出力用ページ	13
2.3	作成する際の注意点	14
2.4	実行手順	15
2.4.1	翻訳およびリンク	15
2.4.2	CGIサブルーチンの環境変数設定	16
2.4.3	CGIアプリケーションの実行	17
第3章	ISAPIアプリケーションの作成と実行	19
3.1	必要なもの	20
3.2	作成するもの	21
3.2.1	GetExtensionVersion	21
3.2.2	HttpExtensionProc	22
3.2.3	TerminateExtension	24
3.2.4	呼出し用ページ	25
3.2.5	結果出力用ページ	26
3.3	ISAを作成する際の注意点	27
3.4	実行手順	28
3.4.1	翻訳およびリンク	28
3.4.2	IISの設定	29
3.4.3	ISAPIサブルーチンの環境変数設定	32
3.4.4	ISAPIアプリケーション(ISA)の実行	33
第4章	SAFアプリケーションの作成と実行	35
4.1	必要なもの	36
4.2	作成するもの	37
4.2.1	SAFサブルーチンを使用したCOBOLプログラム	37
4.2.2	呼出し用ページ	38
4.2.3	結果出力用ページ	39
4.3	SAFを作成する際の注意点	40
4.4	実行手順	41
4.4.1	翻訳およびリンク	41
4.4.2	NESの設定	42
4.4.3	SAFサブルーチンの環境変数設定	46
4.4.4	SAFアプリケーションの実行	47

4.5	SAFディレクタ リファレンス	49
4.5.1	COBOL_Init	49
4.5.2	COBOL_Perform	50
第5章	Webアプリケーション開発支援機能の利用	53
5.1	Webアプリケーション開発支援機能の概要	54
5.2	Webアプリケーション開発支援機能を使用した作業の流れ	57
5.3	Webアプリケーションウィザード	58
5.3.1	Webアプリケーション	58
5.3.2	Webアプリケーションウィザードの使用方法	60
5.3.3	Webアプリケーション雛形の構成	66
5.4	Webパラメタ取得プログラム生成	73
5.4.1	Webパラメタ取得プログラム	73
5.4.2	Webパラメタ取得プログラム生成ツールの使用方法	75
5.4.3	Webパラメタ取得プログラムの生成規則	77
5.5	結果ページ出力プログラム生成	81
5.5.1	結果ページ出力プログラム	81
5.5.2	結果ページ出力プログラム生成ツールの使用方法	84
5.5.3	結果ページ出力プログラムの生成規則	86
5.6	Webアプリケーション開発のヒント	89
5.6.1	プロジェクトマネージャからのHTML文書の操作	89
5.6.2	エディタテンプレートの活用	89
第6章	WebアプリケーションにおけるUnicodeの利用	91
6.1	Unicodeの利用	92
6.2	WWWブラウザ	93
6.3	WWWブラウザからの入力コード系	94
6.4	WWWブラウザへの出力コード系	95
6.5	翻訳オプション	96
6.6	使用可能なサブルーチン	97
6.7	呼出し用ページに使用できるHTML文書のコード系	98
6.8	結果出力用ページに使用できるHTML文書のコード系	99
6.9	ログファイルのコード系	100
6.10	コード系の異なるCOBOLアプリケーションの混在	101
第7章	COBOL Webサブルーチンの使い方	103
7.1	COBOL Webサブルーチンの機能	104
7.1.1	基本機能	104
7.1.2	ファイルアップロード機能	105
7.1.3	セッション管理機能	107
7.1.4	SAF固有機能	108
7.2	Webサブルーチンのインタフェース	109
7.2.1	登録集	109
7.2.2	Webサブルーチンの呼出し	110
7.2.3	Webサブルーチンの定量制限	150
7.3	Webサブルーチンの提供するクラス	152
第8章	動作確認	155
8.1	ログ情報を参照する	156
8.2	対話型デバグで動作確認する	157
8.2.1	CGIアプリケーションの動作確認	157
8.2.2	ISAPIアプリケーションの動作確認	157
8.2.3	SAFアプリケーションの動作確認	161
8.3	Webサブルーチンが検出したエラーを参照する	163
8.4	実行中のデータを表示形式で参照する	164
付録A	Q&A集	165

付録B	エラー処理	173
付録C	WebアプリケーションをCOBOLで作成する場合の概念図	179
C.1	CGIアプリケーションをCOBOLで作成する場合の概念図	179
C.2	ISAをCOBOLで作成する場合の概念図	180
C.3	SAFをCOBOLで作成する場合の概念図	180
付録D	CGIからISAPIへの移行の手引き	181
D.1	CGIからISAPIへの移行	181
D.2	アプリケーションの形式	182
D.2.1	呼出し用ページ	182
D.2.2	翻訳、リンク方法について	182
D.3	実行単位	183
D.3.1	翻訳方法について	183
D.3.2	共有資源へのアクセス	184
D.4	モジュール構成	184
D.5	サーバとのインタフェース領域	184
D.6	復帰コード	185
D.7	環境変数操作	186
D.8	CGI環境変数の操作	187
付録E	CGIからSAFへの移行の手引き	189
E.1	CGIからSAFへの移行	189
E.2	アプリケーションの形式	190
E.2.1	呼出し用ページ	190
E.2.2	翻訳、リンク方法について	190
E.3	実行単位	191
E.3.1	翻訳方法について	191
E.3.2	共有資源へのアクセス	191
E.4	サーバとのインタフェース領域	191
E.5	環境変数操作	192
E.6	CGI環境変数の操作	194
付録F	CGIサブルーチン V1.0からの移行時の注意事項について	197
F.1	COBOLプログラムおよび関連資産の修正	197
F.2.1	登録集の変更	197
F.2.2	サブルーチン名およびパラメタ名の変更	197
F.2.3	サブルーチンの非互換に関する変更	199
F.2	COBOLプログラムの翻訳およびリンク	200
索引		201

第1章 COBOL Webサブルーチンについて

1.1 COBOL CGIサブルーチン

1.1.1 CGIの概要

CGI(Common Gateway Interface)とは、WWWブラウザ(クライアント)とWWWサーバとのやりとりに関する規格です。

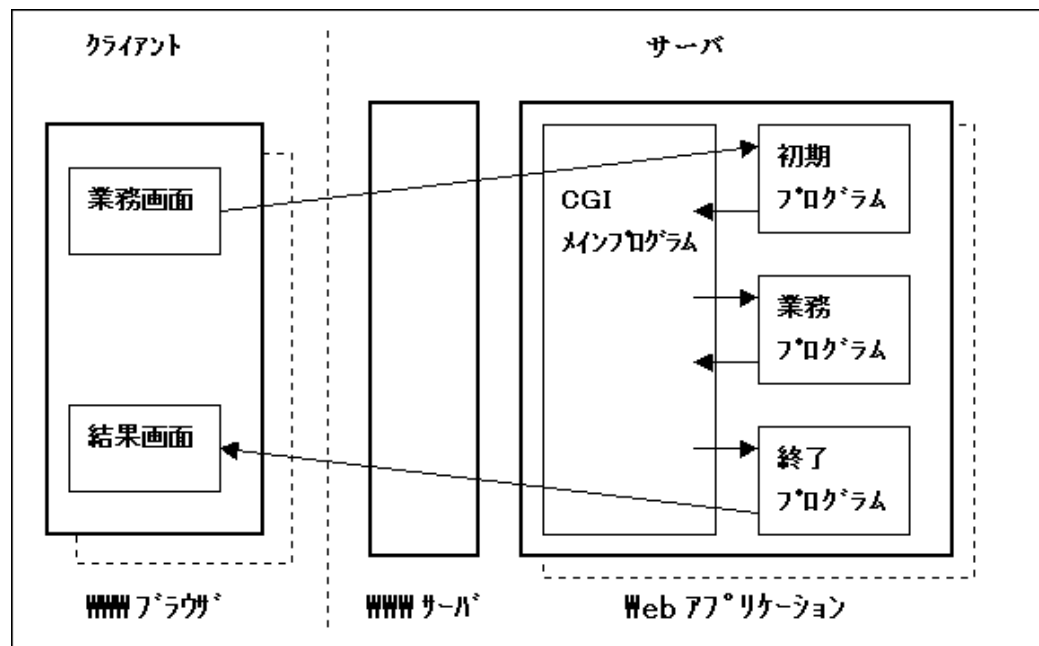
インターネット/イントラネットを利用して業務を構築する場合に、WWWサーバ上のHTML文書(ホームページなど)だけでは集計、検索などのデータ処理、ファイルアクセスおよびデータベースアクセスを実現できません。

このようなサービスを実現するためには、外付けのCOBOLアプリケーションなどを利用する必要があります。このCOBOLアプリケーションをWebアプリケーションと呼びます。

Webアプリケーションは、メインプログラム、初期プログラム、業務プログラムおよび終了プログラムで構成されています。

1.1.2 CGIの処理の流れ

CGIを利用した場合の処理の流れを説明します。



→ : 処理の流れ

WWWブラウザ(クライアント)にURLを入力すると、あらかじめWWWサーバに登録しているHTML文書の業務画面(以降、呼出し用ページと呼びます)が表示されます。

WWWブラウザに表示された呼出し用ページに用意されたWebアプリケーション実行ボタンをクリックすると、WWWサーバがプロセスを生成し、Webアプリケーションを起動します。また、呼出し用ページにテキスト入力項目などを用意した場合、利用者が入力した情報は、受渡しデータ(以降、Webパラメタと呼びます)としてWWWサーバ側のWebアプリケーションに渡されます。

Webアプリケーションでは、以下の処理を行います。

初期処理を行う初期プログラムの実行

Webパラメタの受け取り

手続きの実行

処理結果をHTML文書(以降、結果出力用ページと呼びます)としてWWWブラウザに出力する

業務プログラムの実行

終了処理を行う終了プログラムの実行
Webアプリケーションの処理が終了するとプロセスも終了します。

1.1.3 COBOL CGIサブルーチンとは

COBOL CGIサブルーチン(以降、CGIサブルーチンと略します)は、CGIを使用したWebサブルーチンです。このサブルーチンを使用することで、WebアプリケーションをCOBOLによって簡単に作成できます。

CGIを使用したWebアプリケーションは、WWWサーバの種類を特定せずに動作させることができます。また、CGIサブルーチンはWebサブルーチンの共通機能で構成されているので、CGIを使用したWebアプリケーションはわずかな修正で、他のサブルーチンを使用したWebアプリケーションに移行できます。

1.2 COBOL ISAPIサブルーチン

1.2.1 ISAPIの概要

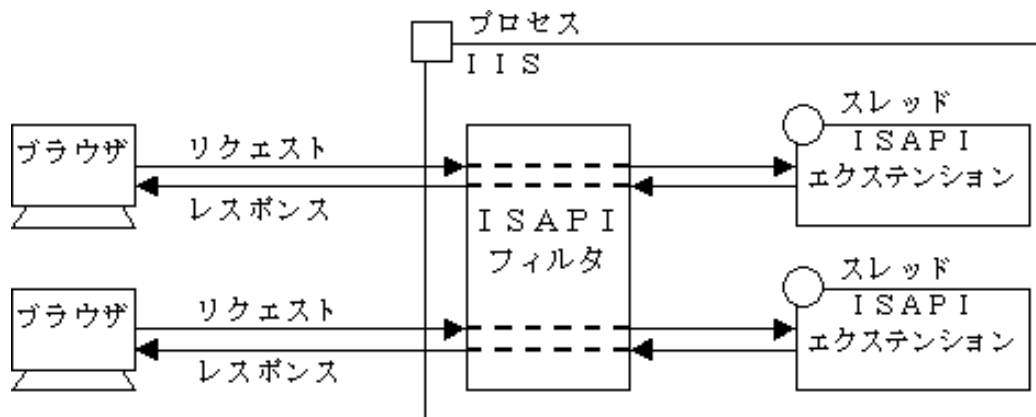
ISAPI(Internet Server Application Programming Interface)とは、IISを拡張するための枠組みとそこで利用できるインタフェースです。ISAPIを使用したアプリケーションであるInternet Server Application(以降、ISAと略します)はIISに組み込まれてスレッドとして動作するため、CGIに比べて高速に動作します。また、メモリなどの資源消費も少なくてすみます。ただし、ISAはスレッドとして動作するため、アプリケーションはマルチスレッドで動作可能でなければなりません。また、資源の解放を忘れると、IISが終了するまでその資源を使用できなくなる可能性があるため、注意しなければなりません。ISAPIで作成できるアプリケーションの形態は次の2つがあります。

ISAPIエクステンション

個々のWebアプリケーションを実装する単位です。リクエストごとに使用するアプリケーションを変えることが可能です。

ISAPIフィルタ

Webサイト全体に共通して呼び出されるWebアプリケーションです。ISAPIエクステンションの前後に実行され、アプリケーション全体に共通する処理を必要とする場合に使用します。



1.2.2 ISAの構成

ISAPIエクステンションの構成に関して紹介します。ただし、COBOL ISAPIサブルーチンではISAPIフィルタはサポートしていません。ISAPIフィルタについては書籍やIISのマニュアルを参照してください。

ISAPIエクステンションは、次の3つの関数をエクスポートしたDLLでなければなりません。なお、いずれの関数もマルチスレッドとして動作します。

GetExtensionVersion

アプリケーションDLLがロードされたときに一度だけ呼び出される関数です。この関数が終了するまで、次のHttpExtensionProcは実行されません。

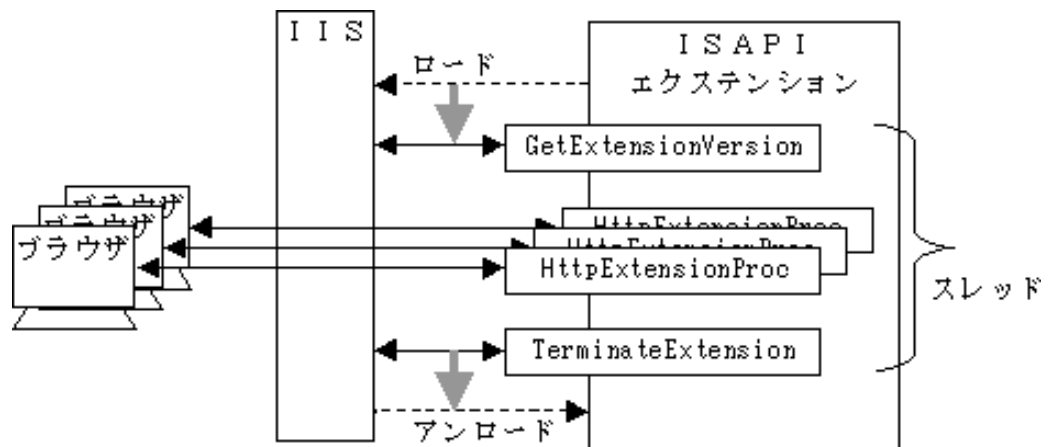
HttpExtensionProc

リクエストごとに呼び出されるアプリケーション本体です。WWWブラウザから送信されたデータを受け取り、HTML文書を送信するための機能がIISへのインタフェースとして提供されています。

TerminateExtension

アプリケーションDLLがアンロードされるときに呼び出される関数です。

図1-1 各エクスポート関数の呼び出され方



1.2.3 COBOL ISAPI サブルーチンとは

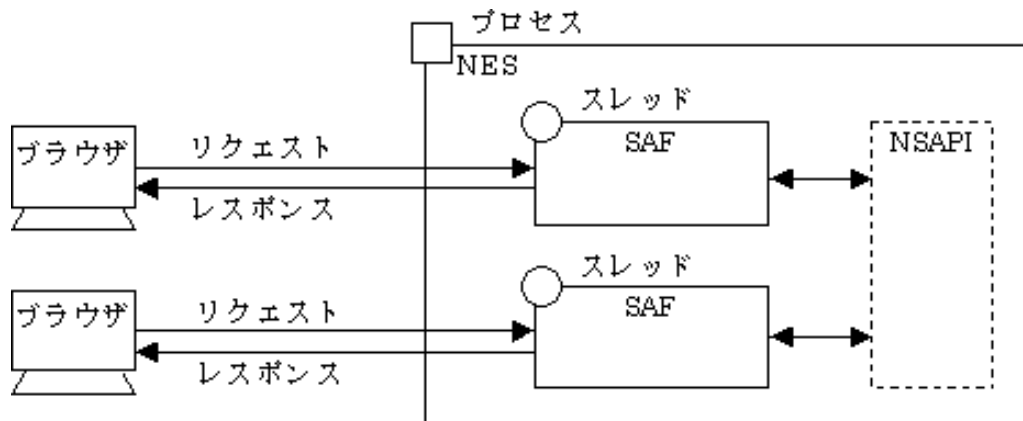
COBOL ISAPI サブルーチン(以降、ISAPI サブルーチンと略します)は、ISAPIを使用したWebサブルーチンです。このサブルーチンを使用することで、ISAPI エクステンションをCOBOLによって簡単に作成できます。このサブルーチンを使用したWebアプリケーションはISAPIの特徴をそのまま受け継ぎます。また、ISAPI サブルーチンはCOBOL CGI サブルーチンに対して上位互換があるので、COBOL CGI サブルーチンで作成したWebアプリケーションはわずかな修正でIIS配下のWebアプリケーションに移行できます。

1.3 COBOL SAFサブルーチン

1.3.1 SAFの概要

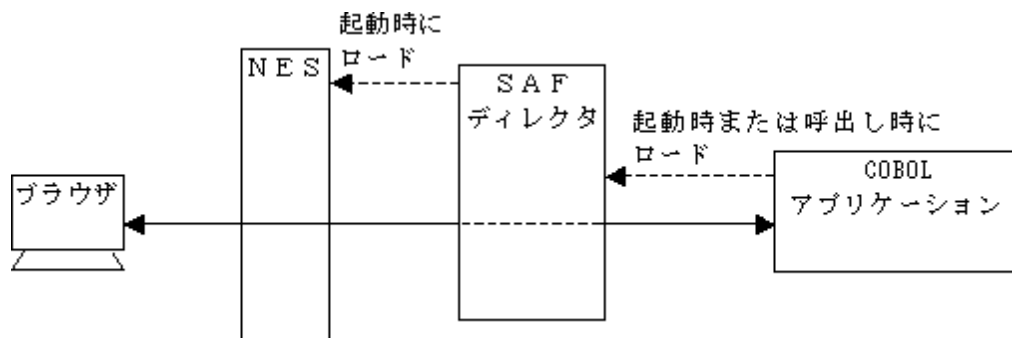
SAF(Server Application Functions)とはNSAPI(Netscape Server Application Programming Interface)を用いて作成したアプリケーションのことです。

なお、NSAPIはNESを拡張するために提供されているインタフェースです。SAFはNESに組み込まれてスレッドとして動作するため、CGIに比べて高速に動作します。また、メモリなどの資源消費も少なくすみます。ただし、SAFはスレッドとして動作するため、アプリケーションはマルチスレッドで動作可能でなければなりません。また、資源の解放を忘れると、NESが終了するまでその資源を使用できなくなることがあるので、注意しなければなりません。



1.3.2 COBOL SAFサブルーチンとは

NSAPIを使用したWebサブルーチンをCOBOL SAFサブルーチン(以下、SAFサブルーチンと略します)と呼びます。また、SAFサブルーチンを使用して作成したWebアプリケーションを実行する環境をCOBOL SAFディレクタ(以下、SAFディレクタと略します)と呼びます。SAFディレクタは、SAFサブルーチンを使用して作成したWebアプリケーションのロードおよび実行を行います。NESの起動時または初回リクエスト時にWebアプリケーションをロードし、それ以降メモリ上に常駐させるため、高速に動作します。アンロードはNES終了時まで行われません。



SAFディレクタおよび実行されるWebアプリケーションはSAFとして動作します。したがって、ここで使用するWebアプリケーションはSAFの特徴を受け継ぐことになります。また、SAFサブルーチンはCGIサブルーチンに対して上位互換があるため、CGIサブルーチンを使用して作成したWebアプリケーションをわずかな修正でNES配下のWebアプリケーションに移行できます。

第2章 CGIアプリケーションの作成と実行

2.1 必要なもの

CGIサブルーチンを使用したWebアプリケーションの作成と実行に必要なものは以下のとおりです。

WWWサーバ

Webアプリケーションを動作させるために必要です。

COBOL97 V60L10以降またはNetCOBOL V7.0L10以降

Webアプリケーションの翻訳、リンクおよび実行の際に必要です。

なお、運用時にはNetCOBOL サーバ運用パッケージが必要です。

【補足】 WWWサーバは、Webアプリケーションを動作させるために実行権を設定する必要があります。インストールや設定方法については、運用するWWWサーバのシステム管理者に相談してください。

2.2 作成するもの

CGIサブルーチンを使用してWebアプリケーションを作成するには、以下が必要です。

CGIサブルーチンを使用したCOBOLプログラム

HTML文書

呼出し用ページ(Webアプリケーション起動用)

結果出力用ページ(処理結果返却用、必要に応じて作成する)

2.2.1 CGIサブルーチンを使用したCOBOLプログラム

COBOLプログラムを作成します。

なお、ここで作成するCOBOLプログラムは、画面入出力機能以外のCOBOLの機能をすべて利用できます。

プログラム作成上の注意については、“2.3 [作成する際の注意点](#)”を参照してください。

2.2.1.1 メインプログラム

メインプログラムは、各プログラムを呼び出すためのプログラムです。メインプログラムでは、それぞれのタイミングで初期プログラム、業務プログラムおよび終了プログラムを呼び出します。メインプログラムのコーディング例を以下に示します。

```
-----*
000010*-----*
000020*      CGI メインプログラム      *
000030*-----*
000040 IDENTIFICATION  DIVISION.
000050 PROGRAM-ID.      CGI.
000060 DATA            DIVISION.
000070 PROCEDURE        DIVISION.
000080 MAIN-PROC.
000090      CALL  "INIT".                *> 初期プログラム
000100*                                *> ( ファイルのOPEN等 )
000110      CALL  "EXEC".                *> 業務プログラム
000120*
000130      CALL  "TERM".                *> 終了プログラム
000140*                                *> ( ファイルのCLOSE 等 )
000150      STOP  RUN.
-----*
```

2.2.1.2 初期プログラム

初期プログラムは、Webアプリケーションで使用する資源の初期化を行うプログラムです。たとえばファイル入出力機能を使用している場合には、ファイルのオープン処理などを実行します。初期プログラムのコーディング例を以下に示します。

```
-----*
000010*-----*
000020*  ・モジュール名      INIT      *
000030*  ・動作概要        CGI 初期プログラム      *
000040*-----*
000050 IDENTIFICATION  DIVISION.
```

```

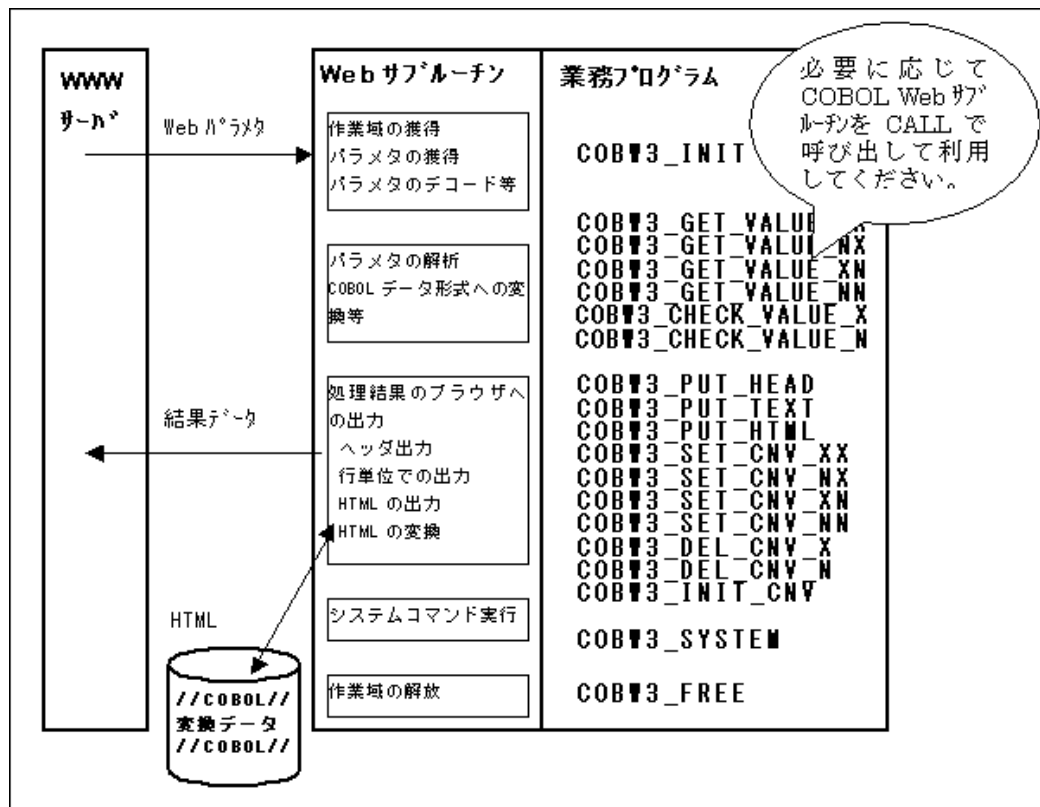
000060 PROGRAM-ID.      INIT.
000070 DATA             DIVISION.
000080 PROCEDURE         DIVISION.
000090 INIT-PROC.
000100*
000110* ファイルのOPEN等、必要に応じて初期処理を行う
                                :
000200*
000210 INIT-END.
000220*
000230      EXIT PROGRAM.

```

【補足】 初期プログラムでは、CGIサブルーチンを使用できません。CGIサブルーチンを使用した場合、動作は保証されません。

2.2.1.3 業務プログラム

業務プログラムとは、WWWブラウザからリクエストデータを受け取り、レスポンスデータ(処理結果)をWWWブラウザに出力するプログラムです。



WWWブラウザから送られてきたリクエストデータ(Webパラメタ)を受け取り、CGIサブルーチンを使って入力データを処理します。処理終了後、実行結果をWWWブラウザに出力します。業務プログラムのコーディング例を以下に示します。

```

000010*-----*
000020*   ・モジュール名      EXEC                      *
000030*   ・動作概要        CGI 業務プログラム        *
000040*-----*
000050 IDENTIFICATION  DIVISION.

```

```

000060 PROGRAM-ID      EXEC.
000070 ENVIRONMENT     DIVISION.
000080 CONFIGURATION    SECTION.
000090 INPUT-OUTPUT     SECTION.
000100 DATA             DIVISION.
000110 WORKING-STORAGE   SECTION.
000120 COPY COBW3.
000170*
000180 PROCEDURE         DIVISION.
000190*
000200 EXEC-PROC.
000210*
000220* CGIサブルーチン作業環境の設定および Webパラメタの獲得
000225 MOVE LOW-VALUE TO COBW3.
000230 CALL "COBW3_INIT" USING COBW3.
000240*
000250* 名前の取得
000260 MOVE "NAME1" TO COBW3-SEARCH-DATA.
000270 CALL "COBW3_GET_VALUE_XX" USING COBW3.
000280
000290* 結果出力用ページ中の変換データの登録
000300 MOVE "GET-NAME" TO COBW3-CNV-NAME.
000310 MOVE COBW3-GET-DATA TO COBW3-CNV-VALUE.
000320 CALL "COBW3_SET_CNV_XX" USING COBW3.
:
000760* 結果出力用ページの出力
000770 MOVE "SAMPLE.HTML" TO COBW3-HTML-FILENAME.
000780 CALL "COBW3_PUT_HTML" USING COBW3.
:
000810* CGIサブルーチン作業領域の解放
000820 CALL "COBW3_FREE" USING COBW3.
000830*
000840*
000850 EXEC-END.
000860*
000870 EXIT PROGRAM.

```

以下、CGIサブルーチンを利用する際の注意事項について説明します。

見出し部

特にありません。

環境部

特にありません。

データ部

作業場所節にCGIサブルーチンのインタフェースとなる登録集を次に示すようにCOPY文で取り込んでください。

書き方：

```

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

```

なお、この登録集(COBW3.cb1)は、COBOLをインストールしたフォルダに格納されています。

手続き部

CGIサブルーチンの環境設定およびWebパラメタの獲得のため、最初に“COBW3_INIT”を呼び出す必要があります。また、CGIサブルーチンの処理を終了するとき(手続きの最後)には、必ず“COBW3_FREE”を呼び出し、CGIサブルーチンの資源の解放をする必要があります。

その他のCGIサブルーチンについては必要に応じてCALL文で呼び出してください。

なお、Webパラメタの受け取り、参照方法および処理結果の出力方法の詳細については、“第7章 [COBOL Webサブルーチンの使い方](#)”を参照してください。

2.2.1.4 終了プログラム

終了プログラムは、Webアプリケーションの終了処理を行うプログラムです。

たとえば、ファイル入出力機能を使用している場合には、ファイルのクローズ処理などを行います。

終了プログラムのコーディング例を以下に示します。

```
-----*
000010*-----*
000020*   ・モジュール名      TERM                      *
000030*   ・動作概要        CGI  終了プログラム          *
000040*-----*
000050 IDENTIFICATION  DIVISION.
000060 PROGRAM-ID.      TERM.
000070 DATA            DIVISION.
000080 PROCEDURE        DIVISION.
000090 TERM-PROC.
000100*
000110*   ファイルのCLOSE 等、必要に応じて終了処理を行う
           :
000200*
000210 TERM-END.
000220*
000230      EXIT PROGRAM.
```

【補足】 終了プログラムでは、CGIサブルーチンを使用できません。CGIサブルーチンを使用した場合、動作は保証されません。

2.2.2 呼出し用ページ

呼出し用ページとは、Webアプリケーションを起動する時に使うHTML文書です。CGIを使用するWebアプリケーションを起動するには、<FORM>タグや<A>タグなどを使用してください。たとえば、メインプログラムをWebアプリケーションの入口として作成する実行形式ファイルの場合、次のように指定します。

例

メインプログラムCGI.cob で作成した実行形式CGI.exe を
METHOD="POST" で実行する場合

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



<A>タグを使用してWebアプリケーションを起動した場合、Webパラメタを受け渡すことはできません。

その他のHTML文書に関する内容については、“Web連携ガイド 付録A”を参照してください。

2.2.3 結果出力用ページ

結果出力用ページとは、Webアプリケーションの実行結果をWWWブラウザに返却するためのHTML文書です。Webアプリケーションから直接レスポンスデータを返却することでもできるため、必ずしも結果出力用ページを用意する必要はありません。ただし、結果出力用ページを用意することで、返却するデータとプログラムの独立性が高まり、Webアプリケーションの再翻訳なしで出力結果のレイアウト変更が可能になるなど、柔軟性、保守性を向上させることができます。

また、結果出力用ページの内容を動的に変換して出力することも可能です。

静的なHTML文書(処理の結果で出力内容が変化しないHTML文書)を出力する場合は、HTML作成ツールで作成したHTML文書をそのまま利用できます。

動的なHTML文書(処理の結果によって出力結果が変動するHTML文書)を出力する場合は、結果出力用ページに“//COBOL//”で挟んだ項目名(変換名)を指定しておくことで、動的にデータを置き換えて出力できます。結果出力用ページ中の全データを置き換えの対象とするため、結果出力用ページの文字の変更やバックのカラーの指定などのタグを置き換えることも可能です。

“7.2.2.3 [処理結果の出力](#)”の“COBW3_PUT_HTML”を参照してください。

結果出力用ページは、タイトル/本文/ヘッダなどを複数のHTML文書に分割して出力したり、プレーンテキストと組み合わせて出力することもできます。

2.3 作成する際の注意点

以下の点に注意してください。

Webアプリケーションで利用するCOBOLプログラムでは、画面操作に関する次の機能は利用しないでください。

表示ファイル(画面操作機能)

スクリーン操作機能

小入出力機能(ACCEPT文、DISPLAY文)

(ただし、環境変数、日付けおよび時間の操作機能は利用できます)

CGIサブルーチンが提供する登録集(COBW3.cbl)を取り込むCOPY文にはREPLACINGを指定しないでください。また、このCOPY文が置き換えの対象となるようなREPLACE文は指定しないでください。

データ部でデータを初期化(VALUE句など)した場合、運用環境によっては複数のリクエストにまたがった初期値の保障はされません。WWWブラウザからのリクエストのたびに初期値を保障する必要がある場合、必ず実行される手続き部でデータを初期化してください。Webアプリケーションは、WWWサーバが提供するサービス配下で動作するアプリケーションです。サービス配下で動作する場合の注意点として、“NetCOBOL 使用手引書”の“23.1.3 サービス配下で動作するプログラム”を参照してください。

2.4 実行手順

ここでは、翻訳、リンクおよび実行の基本的な方法を説明します。ここで説明していない翻訳オプションおよびリンクオプションの詳細については“NetCOBOL 使用手引書”を参照してください。

Webアプリケーションの作成および実行は、以下の手順に従ってください。

作成したプログラムを翻訳およびリンクする。

詳細については、“2.4.1 [翻訳およびリンク](#)”を参照してください。

環境を設定する。

WWWサーバの環境設定を行う。

詳細については、ご使用のWWWサーバのマニュアルを参照してください。

CGIサブルーチンの環境設定を行う。

詳細については、“2.4.2 [CGIサブルーチンの環境変数設定](#)”を参照してください。

Webアプリケーションを実行する。

WWWサーバに登録した呼出し用ページ(HTML文書)を呼び出して、起動してください。

詳細については、“2.4.3 [CGIアプリケーションの実行](#)”を参照してください。

2.4.1 翻訳およびリンク

COBOLで記述したメインプログラム、初期プログラム、業務プログラムおよび終了プログラムを以下の方法で翻訳およびリンクします。なお、プロジェクト管理機能を利用している場合、翻訳およびリンクはプロジェクトマネージャのビルドを使用してください。

COBOLソースプログラムを翻訳する

COBOLソースプログラムの翻訳は、WINCOBコマンドを実行してウィンドウ操作により行う方法と、翻訳コマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、翻訳コマンドを使用して翻訳するときの例を示します。

WINCOBコマンドを使用した翻訳や翻訳コマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

```
COBOL32 -M メインプログラム.COB
COBOL32 初期プログラム.COB
COBOL32 業務プログラム.COB
COBOL32 終了プログラム.COB
```

COBOL目的プログラムをリンクする

COBOLソースプログラムを翻訳し作成した目的プログラムのリンクは、WINLINKコマンドを実行してウィンドウ操作により行う方法と、リンクコマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、リンクコマンドを使用してリンクするときの例を示します。WINLINKコマンドを使用したリンクやリンクコマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

複数のオブジェクトファイルを1つの実行形式ファイル(.EXE)として作成(静的リンク)する場合

```
LINK メインプログラム.OBJ 初期プログラム.OBJ 業務プログラム.OBJ 終了プログラム.OBJ F3BICIMP.LIB LIBC.LIB F3BICWSR.LIB /OUT:実行形式名.EXE
```

複数のオブジェクトファイルを1つの実行形式ファイル(.EXE)と複数のダイナミックリンクライブラリ(.DLL)として作成(動的リンク)する場合

```
LINK 初期プログラム.OBJ F3BICIMP.LIB LIBC.LIB /DLL /OUT:初期プログラム.DLL
LINK 業務プログラム.OBJ F3BICIMP.LIB LIBC.LIB F3BICWSR.LIB /DLL /OUT:業務プログラム.DLL
```

```
LINK 終了プログラム.OBJ F3BICIMP.LIB LIBC.LIB /DLL /OUT:終了プログラム.DLL
LINK メインプログラム.OBJ 初期プログラム.LIB 業務プログラム.LIB 終了プロ
```

グラム.LIB F3BICIMP.LIB LIBC.LIB /OUT:実行形式名.EXE



CGIサブルーチンは下方互換がありません。このため、動作するCGIサブルーチンのバージョンレベルよりも新しいバージョンレベルの登録集(COBW3.cb1)は使用できません。また、業務プログラムで取り込んだ登録集のバージョンレベルより前のCGIサブルーチンをリンクしないでください。

2.4.2 CGIサブルーチンの環境変数設定

CGIサブルーチンを使用したWebアプリケーションを実行するためには、あらかじめ次の実行環境情報をシステム環境変数または実行用の初期化ファイル(COBOL85.CBR)に設定しておく必要があります。

システム環境変数に設定すべきもの

PATH

Webアプリケーションが使用する動的リンクライブラリ(.DLL)のパスを指定します。

システム環境変数または実行用初期化ファイルに設定するもの

@CBR_ATTACH_TOOL=TEST [起動パラメタ]

デバッグしたいプログラムからデバガを起動することを指定します。COBOLで作成したWebアプリケーションをデバガを使用してデバッグする場合には、この環境変数を設定する必要があります。

詳細については、“8.2.1 [CGIアプリケーションの動作確認](#)”を参照してください。

@MessOutFile=ファイル名

COBOLランタイムシステムが出力する実行時メッセージを格納するファイル名を指定します。これにより、メッセージボックスが画面に出力されることを抑止します。ファイル名は、絶対パスまたは相対パスで指定できます。相対パスを指定した場合、カレントフォルダからの相対パスとなります。また、既に同名のファイルが存在する場合は、そのファイルにメッセージが追加書きされます。

@WinCloseMsg=OFF

ウィンドウを閉じるときに、確認のためのメッセージを表示する(ON)か、しない(OFF)かを指定します。Webアプリケーションの実行には、確認メッセージを表示しない(OFF)を指定します。

@CBR_CGI_LOGFILE=ログファイル名

CGIサブルーチンが出力するログ情報を記録するファイルを指定します。障害時などの原因を調べるときに便利です。なお、ファイル名はサーバ上の絶対パスで指定してください。指定がない場合、ログは出力されません。

@CBR_CGI_SEVERITY=重要度

CGIサブルーチンが出力するログ情報の重要度を指定します。以下の値が指定できます。これ以外の値を指定した場合、または指定がない場合、重要度は0と見なします。

重要度	ログファイルに出力される内容
0	致命的なエラーだけ出力します。
1	0の出力内容に加えて、エラーを出力します。
2	1の出力内容に加えて、警告を出力します。
3	2の出力内容に加えて、CGIサブルーチンのトレースを出力します。



指定する数値が大きくなるに従ってアプリケーションは極端に遅くなりますので、運用時には必要なエラーだけを出力する値を使用してください。特に、障害原因の調査以外で3を指定することはおすすめしません。

他の実行環境情報および詳細については、“ NetCOBOL 使用手引書 ” を参照してください。

2.4.3 CGIアプリケーションの実行

作成したアプリケーションを仮想パスに対応する物理パスに配置し、用意した呼出し用ページを使用して起動します。

第3章 ISAPIアプリケーションの作成と実行

3.1 必要なもの

ISAPI サブルーチンを使用してISAを作成し、実行するにあたって必要なものは次のとおりです。

IIS 4.0以降

Webアプリケーションを動作させるために必要です。

COBOL97 V60L10以降またはNetCOBOL V7.0L10以降

Webアプリケーションの翻訳、リンクおよび実行の際に必要です。

なお、運用時にはNetCOBOL サーバ運用パッケージが必要です。

【補足】 WWWサーバは、Webアプリケーションを動作させるために実行権を設定する必要があります。インストールや設定方法については、運用するWWWサーバのシステム管理者に相談してください。

3.2 作成するもの

ISAPIサブルーチンを使用してISAを作成する場合は、少なくとも以下のものを作成しなければなりません。

次の3つの入口名を持つプログラムとこれらをエクスポート関数として作成したISA(DLL)

GetExtensionVersion

DLLのロード時に実行されます。初期化処理として使用できます。

HttpExtensionProc

リクエストごとに実行されるアプリケーション本体です。ISAPIサブルーチンは、この入口名を持つプログラムだけで使用できます。

TerminateExtension(必要に応じて作成します)

DLLのアンロード時に実行されます。終了処理として使用できます。

これらのプログラムの翻訳、リンクは“3.4.1 [翻訳およびリンク](#)”を参照してください。

HTML文書

呼出し用ページ(Webアプリケーション起動用)

結果出力用ページ(処理結果出力用。必要に応じて作成する)

以降、個々の内容を説明します。

3.2.1 GetExtensionVersion

ISAがIISにロードされたときに一度だけ呼び出されます。この入口名を持つプログラムの処理が終わるまで、次のHttpExtensionProcが呼び出されることはありません。したがって、作成するISA共通の初期化処理を行うことができます。

たとえば、外部データの初期化などが行えます。ただし、後ほど説明するように、終了処理が行えるタイミングが不明確であるので、終了処理を必要とする処理は行わないでください。

この入口名を持つプログラムは必ず作成しなければなりません。

また、この入口名を持つプログラムを作成する際の注意事項について説明します。

見出し部

プログラム名段落に次に示すようにプログラム名を記述してください。

```
-----
PROGRAM-ID. "GetExtensionVersion".
-----
```



注意

プログラム名および大文字/小文字を間違えないでください。間違えるとアプリケーションが動作しません。

環境部

特になし

データ部

連絡節にIISとのインタフェースとなる登録集を次に示すようにCOPY文で取り込んでください。

```
-----
LINKAGE SECTION.
COPY ISAPIINF.
-----
```

なお、この登録集(ISAPIINF.cbl)は、COBOLをインストールしたフォルダに格納されています。

手続き部

IISからの呼出しインタフェースに適合させるために次のように記述してください。

```
-----
PROCEDURE DIVISION WITH STDCALL LINKAGE USING ISAPI-INFO.
-----
```

復帰コード

この関数の終了時にIISへの復帰コードを設定する必要があります。復帰コードは次の2種類があり、通常は正常終了を設定します。

**注意**

エラー終了を設定すると、次のHttpExtensionProcが呼び出されないので注意してください。

復帰コード	意味
1	正常終了
0	エラー終了

たとえば、正常終了を設定するには次のように記述します。

```
-----
MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
-----
```

以上のことをふまえると、このプログラムのひな形は次のようになります。

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID. "GetExtensionVersion".
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
COPY ISAPIINF.
PROCEDURE DIVISION WITH STDCALL LINKAGE USING ISAPI-INFO.
*
* 必要に応じて処理を記述してください。
*
MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
-----
```

3.2.2 HttpExtensionProc

WWWブラウザからのリクエストごとに呼び出されるWebアプリケーション本体です。ISAPIサブルーチンが利用できるのはこの入口名を持つプログラムだけです。この入口名を持つプログラムは必ず作成しなければなりません。

また、この入口名を持つプログラムを作成する際の注意事項について説明します。

見出し部

プログラム名段落に次に示すようにプログラム名を記述してください。

```
-----
PROGRAM-ID. "HttpExtensionProc".
-----
```



プログラム名および大文字/小文字を間違えないでください。間違えるとアプリケーションが動作しません。

環境部

特になし

データ部

作業場所節

ISAPIサブルーチンとのインタフェースとなる登録集を次に示すようにCOPY文で取り込んでください。

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

なお、この登録集(COBW3.cb1)は、COBOLをインストールしたフォルダに格納されています。
連絡節

IISとのインタフェースとなる登録集を次に示すようにCOPY文で取り込んでください。

```
-----
LINKAGE SECTION.
COPY ISAPICTX.
-----
```

なお、この登録集(ISAPICTX.cb1)は、COBOLをインストールしたフォルダに格納されています。

手続き部

IISからの呼出しインタフェースに適合させるために次に示すように記述してください。

```
-----
PROCEDURE DIVISION WITH STDCLL LINKAGE USING ISAPI-CTX-CNT.
-----
```

また、上記の方法で取得したISAPI-CTX-CNTはISAPIサブルーチンがIISとのやりとりが必要としますので、次に示すようにそのポインタをCOBW3-CONTEXTに設定します。

```
-----
MOVE LOW-VALUE TO COBW3.
MOVE FUNCTION ADDR(ISAPI-CTX-CNT) TO COBW3-CONTEXT.
-----
```

復帰コード

この関数の終了時にIISへの復帰コードを設定する必要があります。復帰コードは次の2種類があり、通常は正常終了を設定します。また、エラー終了を設定すると、サーバは該当アプリケーションでエラーが起きたと判断し、その旨のメッセージをIISがWWWブラウザに送信します。

復帰コード	意味
-------	----

1	正常終了
4	エラー終了

たとえば、正常終了を設定するには次のように記述します。

```
-----
MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
-----
```

以上のことをふまえると、このプログラムのひな形は次のようになります。

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID. "HttpExtensionProc".
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY COBW3.
LINKAGE SECTION.
COPY ISAPICTX.
PROCEDURE DIVISION WITH STDCALL LINKAGE USING ISAPI-CTX-CNT.
*
MOVE LOW-VALUE TO COBW3.
MOVE FUNCTION ADDR(ISAPI-CTX-CNT) TO COBW3-CONTEXT.
*
* 必要に応じて処理を記述してください。
*
MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
-----
```

3.2.3 TerminateExtension

DLLがアンロードされるときにIISから一度だけ呼び出されるので、Webアプリケーション共通の終了処理を行うことができます。ただし、DLLがアンロードされるタイミングは状況に応じて異なり、プロセス終了時か強制アンロードを行うまでアンロードされないこともあります。したがって、終了処理およびGetExtensionVersionにおいて初期化処理を記述する際はこの点を十分考慮してください。なお、この入口名を持つプログラムは省略することができます。また、この入口名を持つプログラムを作成する際の注意事項について説明します。

見出し部

プログラム名段落に次に示すようにプログラム名を記述して下さい。

```
-----
PROGRAM-ID. "TerminateExtension".
-----
```



注意

プログラム名および大文字/小文字を間違えないでください。間違えるとこの関数が呼び出されません。

環境部

特になし

データ部

連絡節にIISとのインタフェースとなる登録集を次に示すようにCOPY文で取り込んでください。

```
-----
LINKAGE SECTION.
COPY ISAPIFLG.
```

なお、この登録集(ISAPIFLG.cbl)は、COBOLをインストールしたフォルダに格納されています。

手続き部

IISからの呼出しインタフェースに適合させるために次のように記述してください。

```
-----
PROCEDURE DIVISION WITH STDALL LINKAGE USING ISAPI-FLAG.
```

復帰コード

この関数の終了時にIISへの復帰コードを設定する必要があります。復帰コードは次の2種類あり、通常は正常終了を設定します。また、エラー終了を設定すると、サーバは該当アプリケーションでエラーが起きたと判断します。

復帰コード	意味
1	正常終了
0	エラー終了

たとえば、正常終了を設定するには次のように記述します。

```
-----
MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
```

また、このプログラムのひな形は次のようになります。

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID. "TerminateExtension".
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
COPY ISAPIFLG.
PROCEDURE DIVISION WITH STDALL LINKAGE USING ISAPI-FLAG.
*
* 必要に応じて処理を記述してください。
*

MOVE 1 TO PROGRAM-STATUS.
EXIT PROGRAM.
-----
```

3.2.4 呼出し用ページ

呼出し用ページとは、Webアプリケーションを起動する時に使うHTML文書です。 Webアプリケー

ションを起動するためには<FORM>タグや<A>タグなどを使用してください。

作成したダイナミックリンクライブラリISAPI.dll を
METHOD="POST" で実行する場合

```
<FORM METHOD="POST" ACTION="ISAPI.dll">
```



注意

<A>タグを使用してWebアプリケーションを起動した場合、Webパラメタを受け渡すことはできません。

その他のHTML文書に関する内容については、“Web連携ガイド 付録A”を参照してください。

3.2.5 結果出力用ページ

結果出力用ページとは、Webアプリケーションの実行結果をWWWブラウザに返却するためのHTML文書です。Webアプリケーションから直接レスポンスデータを返却することもできるため、必ずしも結果出力用ページを用意する必要はありません。ただし、結果出力用ページを用意することで、返却するデータとプログラムの独立性が高まり、Webアプリケーションの再翻訳なしで出力結果のレイアウト変更が可能になるなど、柔軟性、保守性を向上させることができます。

また、結果出力用ページの内容を動的に変換して出力することも可能です。

静的なHTML文書(処理の結果で出力内容が変化しないHTML文書)を出力する場合は、HTML作成ツールで作成したHTML文書をそのまま利用できます。

動的なHTML文書(処理の結果によって出力結果が変動するHTML文書)を出力する場合は、結果出力用ページに “//COBOL//” で挟んだ項目名(変換名)を指定しておくことで、動的にデータを置き換えて出力できます。HTML文書中の全データを置き換えの対象とするため、結果出力用ページの文字の変更やバックのカラーの指定などのタグを置き換えることも可能です。“7.2.2.3 [処理結果の出力](#)” の “COBW3_PUT_HTML” を参照してください。

結果出力用ページは、タイトル/本文/ヘッダなどを複数のHTML文書に分割して出力したり、プレーンテキストと組み合わせて出力することもできます。

3.3 ISAを作成する際の注意点

以下の点に注意してください。

Webアプリケーションで利用するCOBOLプログラムでは、画面操作に関する次の機能は利用しないでください。

表示ファイル(画面操作機能)

スクリーン操作機能

小入出力機能(ACCEPT文、DISPLAY文)

(ただし、環境変数、日付けおよび時間の操作機能は利用できます。)

ISAPIサブルーチンは入口名:HttpExtensionProcのプログラム内でだけ使用可能です。

ISAPIサブルーチンを使用する前に、必ず、パラメタで渡される集団項目(ISAPI-CTX-CNT)へのポインタを取得(FUNCTION ADDR)し、その値をCOBW3-CONTEXTに設定してください。

COBW3およびISAPI-CTX-CNTとそのポインタデータはスレッド間で共有しないでください。

IISが正常に動作しなくなる可能性があります。

ISAPI サブルーチンが提供する登録集(COBW3.cbl、ISAPICTX.cbl、ISAPIINF.cbl、ISAPIFLG.cbl)を取り込むCOPY文では、REPLACING指定を行わないでください。また、COPY文が置換えの対象となるようなREPLACE文の指定は行わないでください。

データ部でデータを初期化(VALUE句など)した場合、運用環境によっては複数のリクエストにまたがった初期値の保障はされません。WWWブラウザからのリクエストのたびに初期値を保障する必要がある場合、必ず実行される手続き部でデータを初期化してください。Webアプリケーションは、WWWサーバが提供するサービス配下で動作するアプリケーションです。サービス配下で動作する場合の注意点として、“NetCOBOL 使用手引書”の“23.1.3 サービス配下で動作するプログラム”を参照してください。

3.4 実行手順

翻訳、リンクおよび実行方法について、ここでは基本的な方法を説明します。ここで説明していない翻訳オプションおよびリンクオプションの詳細については、“NetCOBOL 使用手引書”を参照してください。

Webアプリケーションを実行するには、以下の手順に従ってください。

作成したプログラムを翻訳およびリンクする。

詳細については、“3.4.1 [翻訳およびリンク](#)”を参照してください。

環境を設定する。

IISの環境設定を行う。

詳細については、“3.4.2 [IISの設定](#)”を参照してください。

ISAPIサブルーチンの環境設定を行う。

詳細については、“3.4.3 [ISAPIサブルーチンの環境変数設定](#)”を参照してください。

Webアプリケーションを実行する。

IISに登録した呼出し用ページ(HTML文書)を呼び出して、起動してください。

詳細については、“3.4.4 [ISAPIアプリケーション\(ISA\)の実行](#)”を参照してください。

3.4.1 翻訳およびリンク

GetExtensionVersion、HttpExtensionProc、TerminateExtensionの入口名を持つプログラムを以下の方法で翻訳およびリンクします。なお、それぞれのファイルをここではVersion.cob、MainProc.cob、Terminate.cobとして説明します。なお、プロジェクト管理機能を利用している場合、翻訳およびリンクはプロジェクトマネージャのビルドを使用してください。

COBOLソースプログラムを翻訳する

COBOLソースプログラムの翻訳は、プロジェクトマネージャからWINCOBコマンドを実行してウィンドウ操作により行う方法と、翻訳コマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、翻訳コマンドを使用して翻訳するときの例を示します。

WINCOBコマンドを使用した翻訳や翻訳コマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

```
-----
COBOL32 -WC, "ALPHAL(WORD),THREAD(MULTI)" Version.cob
COBOL32 -WC, "ALPHAL(WORD),THREAD(MULTI)" MainProc.cob
COBOL32 -WC, "ALPHAL(WORD),THREAD(MULTI)" Terminate.cob
-----
```



注意

翻訳オプションALPHAL(WORD)の代りにNOALPHALも指定できます。エクスポート関数となるプログラムの入口名は大文字/小文字を区別するため、どちらかを必ず指定する必要があります。また、ISAはマルチスレッドアプリケーションであるため、翻訳オプションTHREAD(MULTI)も必ず指定してください。

COBOL目的プログラムをリンクする(DLLの作成)

COBOLソースプログラムを翻訳し作成した目的プログラムのリンクは、プロジェクトマネージャからWINLINKコマンドを実行してウィンドウ操作により行う方法と、リンクコマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、リンクコマンドを使用してリンクするときの例を示します。WINLINKコマンドを使用したリンクやリンクコマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

作成するDLLをISASMPL.dllとします。リンクを行う前に、次に示すモジュール定義ファイルを用意しておきます。ファイル名ここではISASMPL.defとして説明します。

```
LIBRARY ISASMPL
EXPORTS
    GetExtensionVersion
    HttpExtensionProc
    TerminateExtension
```

リンクは次のように行います。

```
LINK /DLL /OUT:ISASMPL.dll Version.obj MainProc.obj Terminate.obj
      F3BICBDM.obj F3BISAPI.lib F3BICIMP.lib KERNEL32.LIB LIBC.LIB
      /DEF:ISASMPL.def /ENTRY:COBDMAN
```

リンク時にF3BICBDM.objをリンクし、/ENTRY:COBDMANを指定することで、初期化ファイル(COBOL85.CBR)をカレントフォルダとは関係なく、DLLと同じフォルダに配置することができます。IISなどのサーバ配下のアプリケーションのカレントフォルダは不定であるため、初期化ファイルをDLLと同じフォルダに配置できるのは非常に便利です。なお、詳細については、“NetCOBOL 使用手引書”を参照してください。



注意

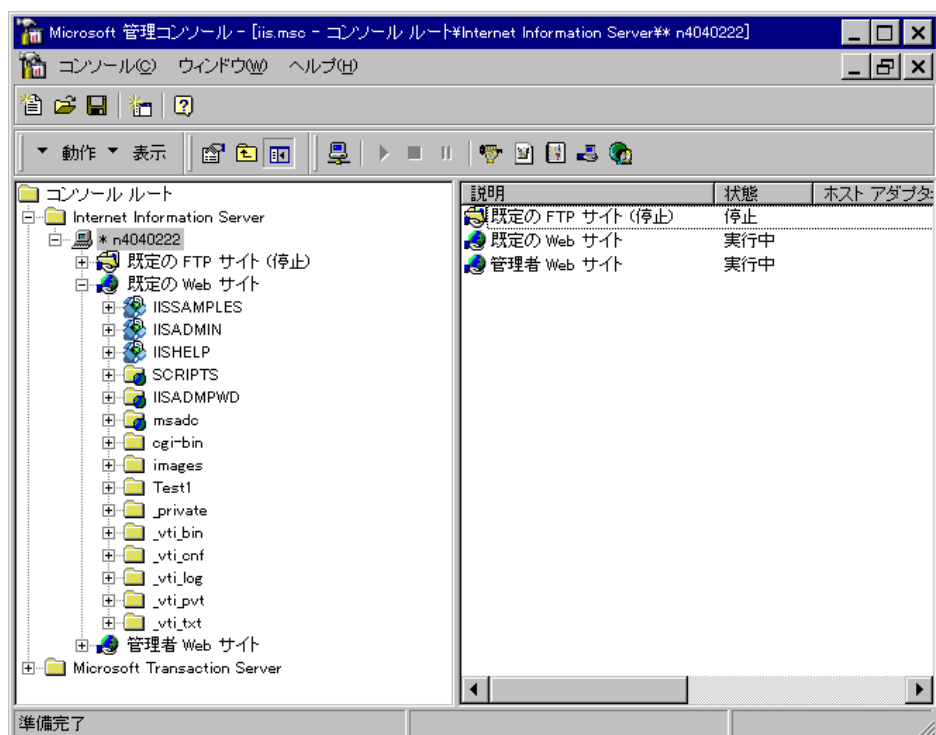
ISAPIサブルーチンは下方互換がありません。このため、動作するサブルーチンのバージョンレベルよりも新しいバージョンレベルの登録集(COBW3.cb1)は使用できません。また、Webアプリケーションで取り込んだ登録集のバージョンレベルより前のISAPIサブルーチンを結合してはなりません。

3.4.2 IISの設定

IISの設定は、呼出し用ページなどのHTML文書の参照やWebアプリケーションを実行するために必要な仮想ディレクトリの登録を行うだけです。次の手順に従ってください。ただし、より高度な設定や設定についての詳細については書籍やIISのオンラインマニュアルを参照してください。なお、IISの設定にはインターネットサービスマネージャを使用しますが、インターネットサービスマネージャにはMMC(Microsoft Management Console)版とHTML版とがあります。以下の説明はIIS4.0でMMC版を使用した場合の例です。

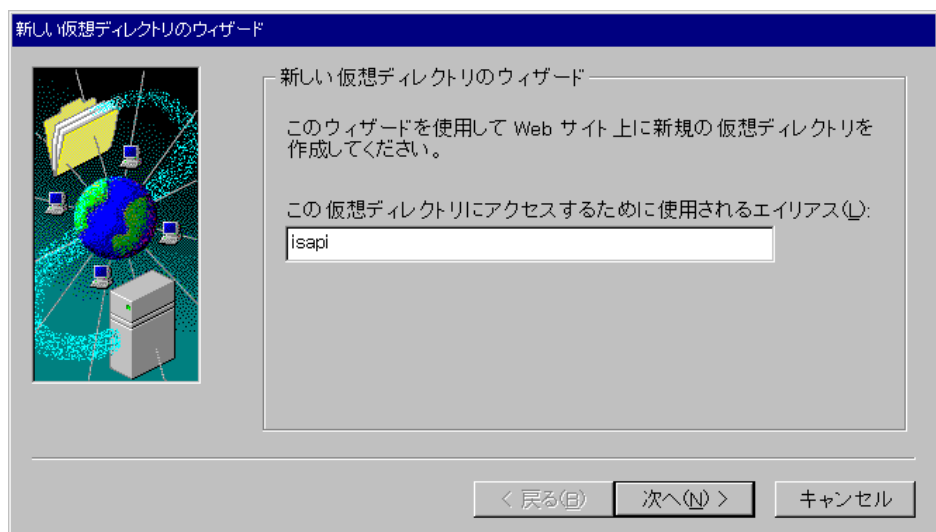
インターネットサービスマネージャを起動します。

次の画面が現れます。



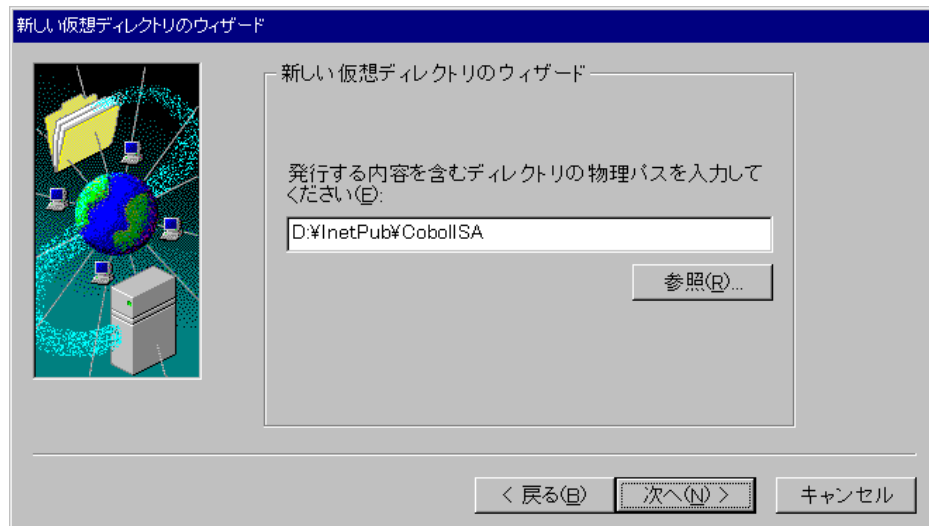
仮想ディレクトリの追加

既定のWebサイトを選択し、プルダウンメニューの[新規作成]の中の[仮想ディレクトリ]を選択すると、次の画面が表示されるので、作成したい仮想ディレクトリ名を入力してください。ここでは、“isapi”としています。



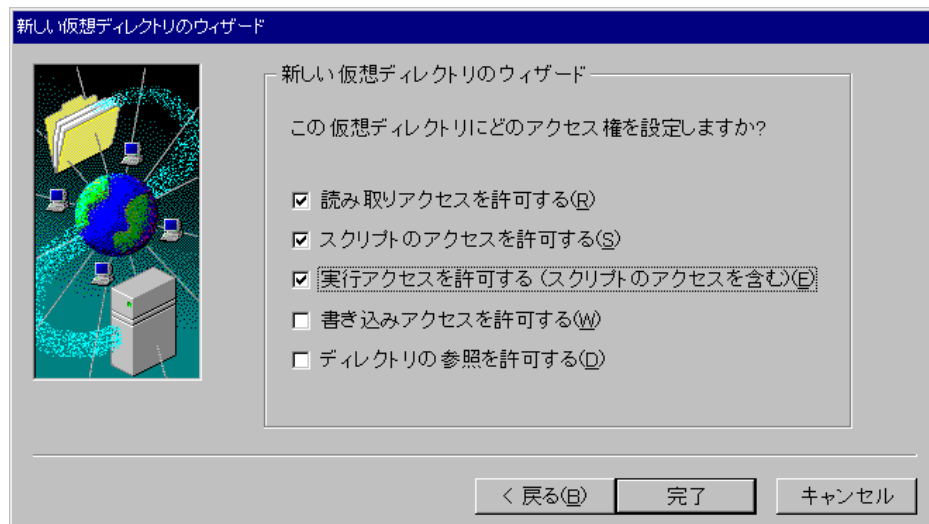
物理パスの登録

仮想ディレクトリを入力後、“次へ”を押下します。すると、物理パスの設定画面が表示されるので、呼出し用ページなどのHTML文書やCOBOLで作成したWebアプリケーションを格納する物理パスを指定してください。ここでは、“D:\inetpub\cobol\isa”としています。

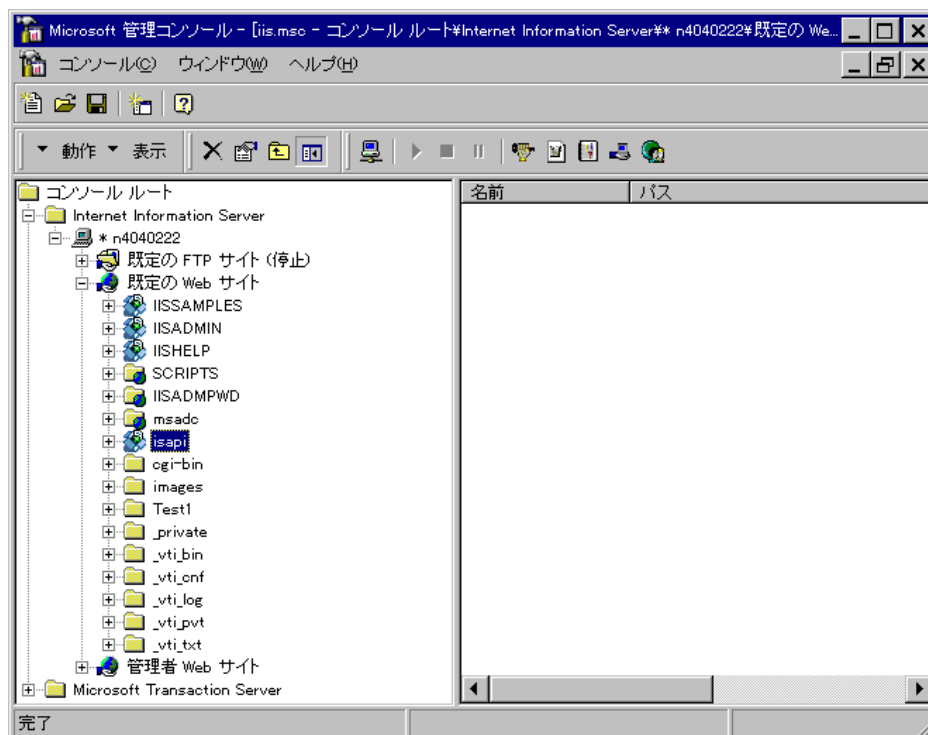


仮想ディレクトリに対する設定

物理パスを入力後、“次へ”を押下します。すると、仮想ディレクトリに対する設定を入力する画面が表示されるので、“実行アクセスを許可する”をチェックしてください。



チェック後、“完了”を押下すると仮想ディレクトリの作成は終了です。次に示すようにMMCの画面中の“既定のWebサイト”の中に“isapi”が作成されます。



【補足】 分離プロセスについて

仮想ディレクトリのプロパティには分離プロセスの設定に関する項目があります。分離プロセスの指定を行うと、その仮想ディレクトリに属するアプリケーションはIIS本体とは異なるプロセス空間で実行されます。ただし、分離プロセスを多用しすぎると性能が低下する可能性があるため、注意してください。なお、IIS5.0ではプールの指定が設けられました。このプールの指定を行うと、IIS本体とは異なる同一プロセスですべての仮想ディレクトリ内のアプリケーションが実行されるため、アプリケーションで障害などが発生してもIIS本体が停止することはありません。これらを踏まえて、分離プロセスの設定は以下の基準に従うことをおすすめします。

テスト時は、分離プロセスまたはプールの指定を行う。

運用時は、IISプロセス(分離プロセスの指定なし)またはプールの指定を行う。

[注意]

Webアプリケーションの動作が不定になる場合があるため、以下の操作は行わないでください。

各仮想ディレクトリの設定で、" ISAPIアプリケーションをキャッシュする " を無効にする。

IISプロセスでWebアプリケーションを実行した場合に、仮想ディレクトリのプロパティでアンロードを行う。

3.4.3 ISAPIサブルーチンの環境変数設定

ISAPIサブルーチンを使用したWebアプリケーションを実行するためには、あらかじめ次の実行環境情報を環境変数または実行用の初期化ファイル(COBOL85.CBR)に設定しておく必要があります。なお、業務ごとに異なる実行用の初期化ファイルを使用する場合は、各業務アプリケーションを別々のフォルダに配置し、IIS(MMC)を使用して業務ごとに異なる仮想ディレクトリを割り当て、分離プロセスとして構成する必要があります。さらに、各アプリケーションにはF3B1CBDM.objをリンクし、アプリケーションが格納されているフォルダの実行用の初期化ファイルを参照できるようにしておいてください。

システム環境変数に設定すべきもの

PATH

Webアプリケーションが使用する動的リンクライブラリ(.DLL)のパスを指定します。

システム環境変数または実行用初期化ファイルに設定するもの

@CBR_ATTACH_TOOL=TEST [起動パラメタ]

デバッグしたいプログラムからデバッガを起動することを指定します。COBOLで作成したWebアプリケーションをデバッガを使用してデバッグする場合には、この環境変数を設定する必要があります。

詳細については、“8.2.2 [ISAPIアプリケーションの動作確認](#)”を参照してください。

@MessOutFile=ファイル名

COBOLランタイムシステムにより出力される実行時メッセージを格納するファイル名を指定します。これにより、メッセージボックスが画面に出力されることを抑止します。ファイル名の指定は、絶対パスを使用してください。また、既に同名のファイルが存在する場合は、そのファイルにメッセージが追加書きされます。

@WinCloseMsg=OFF

ウィンドウを閉じるときに、確認のためのメッセージを表示する(ON)か、しない(OFF)かを指定します。Webアプリケーションの実行時は、確認メッセージを表示しない(OFF)を指定します。

@CBR_ISAPI_LOGFILE=ログファイル名

ISAPIサブルーチンが出力するログ情報を記録するファイルを指定します。障害時などの原因を調べるときに便利です。なお、この場合もファイル名はサーバ上の絶対パスで指定してください。指定がない場合、ログは出力されません。

@CBR_ISAPI_SEVERITY=重要度

ISAPIサブルーチンが出力するログ情報の重要度を指定します。以下の値が指定できます。これ以外の値を指定した場合、または指定がない場合、重要度は0とみなします。

重要度	ログファイルに出力される内容
0	致命的なエラーだけ出力します。
1	0の出力内容に加えて、エラーを出力します。
2	1の出力内容に加えて、警告を出力します。
3	2の出力内容に加えて、ISAPIサブルーチンのトレースを出力します。



注意

指定する数値が大きくなるに従ってアプリケーションは極端に遅くなりますので、運用時には必要なエラーだけを出力する値を使用してください。特に、障害原因の調査以外で3を指定することはおすすめしません。

他の実行環境情報および詳細については、“NetCOBOL 使用手引書”を参照してください。

3.4.4 ISAPIアプリケーション(ISA)の実行

作成したWebアプリケーションを仮想ディレクトリに対応する物理パスに配置し、用意した呼出し用ページを使用して起動します。

【補足】 テストなどを行っているときWebアプリケーションの入れ替えが頻繁に起きます。また、運用環境においてもWebアプリケーションの機能拡張を行った場合、入れ替えが必要になります。しかし、一度Webアプリケーションを実行すると、IISにロードされた状態のままとなり、入れ替えることができません。Webアプリケーションを入れ替える方法については、“8.2.2.4 [Webアプリケーションを入れ替える](#)”を参照してください。

第4章 SAFアプリケーションの作成と実行

4.1 必要なもの

SAFサブルーチンを使用してSAFを作成し、実行するにあたって必要なものは次のとおりです。

Netscape Enterprise Server 3.6 SP1以降またはiPlanet Web Server Enterprise Edition 4.0以降

Webアプリケーションを動作させるために必要です。

COBOL97 V60L10以降またはNetCOBOL V7.0L10以降

Webアプリケーションの翻訳、リンクおよび実行の際に必要です。

なお、運用時にはNetCOBOL サーバ運用パッケージが必要です。

【補足】 WWWサーバは、Webアプリケーションを動作させるために実行権を設定する必要があります。インストールや設定方法については、運用するWWWサーバのシステム管理者に相談してください。

4.2 作成するもの

SAFサブルーチンを使用してSAFを作成する場合は、少なくとも以下のものを作成しなければなりません。

SAFサブルーチンを使用した任意の入口名を持つCOBOLプログラムとこれをエクスポート関数として作成したDLL

これらのプログラムの翻訳、リンクは“4.4.1 [翻訳およびリンク](#)”を参照してください。
HTML文書

呼出し用ページ(Webアプリケーション起動用)

結果出力用ページ(処理結果返却用、必要に応じて作成する)

以降、個々の内容を説明します。

4.2.1 SAFサブルーチンを使用したCOBOLプログラム

COBOLプログラムの入口名にはCOBOLで指定可能な任意の名前を指定できます。なお、このCOBOLプログラムをSAFとして動作させるには、SAFサブルーチンを使用しなければなりません。

以下、SAFとして動作するCOBOLプログラムを作成する際の注意事項について説明します。

見出し部

特になし

環境部

特になし

データ部

作業場所節

SAFサブルーチンとのインタフェースとなる登録集をCOPY文で取り込んでください。

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

SAF固有の機能を使用したい場合は、続けてSAF固有サブルーチンとのインタフェースとなる登録集を取り込んでください。

```
-----
WORKING-STORAGE SECTION.
COPY COBW3.
COPY COBW3SAF.      *> SAF固有サブルーチンとのインタフェース
```

なお、これらの登録集(COBW3.cb1およびCOBW3SAF.cb1)は、COBOLをインストールしたフォルダに格納されています。

連絡節

NESとのインタフェースとなるポインタ項目を記述してください。たとえば、次に示すように記述します。

```
-----
LINKAGE SECTION.
01 SAFCTX    POINTER.
```

手続き部

NESからの呼出しインタフェースに適合させるために次に示すように記述してください。

```
-----
PROCEDURE DIVISION USING SAFCTX.
```

また、上記の方法で取得したSAFCTXはSAFサブルーチンがNESとのやりとりが必要としますので、次に示すようにCOBW3-CONTEXTに設定します。

```
-----
MOVE LOW-VALUE TO COBW3.
SET COBW3-CONTEXT TO SAFCTX.
```

SAF固有の機能を使いたい場合は、SAFCTXを次に示すようにCOBW3-SAF-CONTEXTにも設定します。

```
-----
MOVE LOW-VALUE TO COBW3SAF.
SET COBW3-SAF-CONTEXT TO SAFCTX.
```

以上のことをふまえると、このプログラムのひな形は次のようになります。

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID. "SAFAPL".
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY COBW3.
* COPY COBW3SAF.
LINKAGE SECTION.
01 SAFCTX POINTER.
PROCEDURE DIVISION USING SAFCTX.
*
MOVE LOW-VALUE TO COBW3.
* MOVE LOW-VALUE TO COBW3SAF.
SET COBW3-CONTEXT TO SAFCTX.
* SET COBW3-SAF-CONTEXT TO SAFCTX
*
* 必要に応じて処理を記述してください。
*
EXIT PROGRAM.
```

4.2.2 呼出し用ページ

通常のHTML文書を用意してください。ただし、Webアプリケーションを起動するためには<FORM>タグや<A>タグなどを使用してください。

例

作成したダイナミックリンクライブラリSAF.dll を
METHOD="POST" で実行する場合

```
<FORM METHOD="POST" ACTION="PGM.apl1">
```

**注意**

<A>タグを使用してWebアプリケーションを起動した場合、Webパラメタを受け渡すことはできません。

NES上で、拡張子apl1とSAF.dllを対応付ける設定を行います。詳細については、“4.4.2 [NESの設定](#)”を参照してください。

その他のHTML文書に関する内容については、“Web連携ガイド 付録A”を参照してください。

4.2.3 結果出力用ページ

結果出力用ページとは、Webアプリケーションの実行結果をWWWブラウザに返却するためのHTML文書です。Webアプリケーションから直接レスポンスデータを返却することもあるため、必ずしも結果出力用ページを用意する必要はありません。ただし、結果出力用ページを用意することで、返却するデータとプログラムの独立性が高まり、Webアプリケーションの再翻訳なしで出力結果のレイアウト変更が可能になるなど、柔軟性、保守性を向上させることができます。

また、結果出力用ページの内容を動的に変換して出力することも可能です。

静的なHTML文書(処理の結果で出力内容が変化しないHTML文書)を出力する場合は、HTML作成ツールで作成したHTML文書をそのまま利用できます。

動的なHTML文書(処理の結果によって出力結果が変動するHTML文書)を出力する場合は、結果出力用ページに“//COBOL//”で挟んだ項目名(変換名)を指定しておくことで、動的にデータを置き換えて出力できます。HTML文書中の全データを置き換えの対象とするため、結果出力用ページの文字の変更やバックのカラーの指定などのタグを置き換えることも可能です。詳細については、“7.2.2.3 [処理結果の出力](#)”の“COBW3_PUT_HTML”を参照してください。

結果出力用ページは、タイトル/本文/ヘッダなどを複数のHTML文書に分割して出力したり、プレーンテキストと組み合わせて出力することもできます。

4.3 SAFを作成する際の注意点

以下の点に注意してください。

Webアプリケーションで使用するCOBOLプログラムでは、画面操作に関する次の機能は利用しないでください。

表示ファイル(画面操作機能)

スクリーン操作機能

小入出力機能(ACCEPT文、DISPLAY文)

(ただし、環境変数、日付けおよび時間の操作機能は利用できます。)

SAFサブルーチンを使用する前に、必ず、パラメタで渡されるポインタ項目をCOBW3-CONTEXTに設定してください。

COBW3、COBW3SAFおよびパラメタで渡されるポインタ項目はスレッド間で共有しないでください。NESが正常に動作しなくなる可能性があります。

SAFサブルーチンが提供する登録集(COBW3.cbl、COBW3SAF.cbl)を取り込むCOPY文にはREPLACINGを指定しないでください。また、このCOPY文が置換えの対象となるようなREPLACE文は記述しないでください。

パラメタで渡されるポインタ項目を変更しないでください。変更した場合、動作は保証されません。

データ部でデータを初期化(VALUE句など)した場合、運用環境によっては複数のリクエストにまたがった初期値の保障はされません。WWWブラウザからのリクエストのたびに初期値を保障する必要がある場合、必ず実行される手続き部でデータを初期化してください。Webアプリケーションは、WWWサーバが提供するサービス配下で動作するアプリケーションです。サービス配下で動作する場合の注意点として、“NetCOBOL 使用手引書”の“23.1.3 サービス配下で動作するプログラム”を参照してください。

4.4 実行手順

翻訳、リンクおよび実行方法について、ここでは基本的な方法について説明します。ここで説明していない翻訳オプションおよびリンクオプションの詳細については、“NetCOBOL 使用手引書”を参照してください。

Webアプリケーションを実行するには、以下の手順に従ってください。

作成したプログラムを翻訳およびリンクする。

詳細については、“4.4.1 [翻訳およびリンク](#)”を参照してください。

環境を設定する。

NESの環境設定を行う。

詳細については、“4.4.2 [NESの設定](#)”を参照してください。

SAFサブルーチンの環境設定を行う。

詳細については、“4.4.3 [SAFサブルーチンの環境変数設定](#)”を参照してください。

Webアプリケーションを実行する。

NESに登録した呼出し用ページ(HTML文書)を呼び出して、起動してください。

詳細については、“4.4.4 [SAFアプリケーションの実行](#)”を参照してください。

4.4.1 翻訳およびリンク

SAFサブルーチンを使用して作成したプログラムを以下の方法で翻訳およびリンクします。ここでは、ファイル名をMainProc.cob、入口名をSAF-MAINとして説明します。なお、プロジェクト管理機能を利用している場合、翻訳およびリンクはプロジェクトマネージャのビルドを使用してください。

COBOLソースプログラムを翻訳する

COBOLソースプログラムの翻訳は、プロジェクトマネージャなどからWINCOBコマンドを実行してウィンドウ操作により行う方法と、翻訳コマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、翻訳コマンドを使用して翻訳するときの例を示します。

WINCOBコマンドを使用した翻訳や翻訳コマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

```
-----
COBOL32 -WC,"THREAD(MULTI)" MainProc.cob
-----
```



注意

SAFはマルチスレッドアプリケーションであるので、翻訳オプションTHREAD(MULTI)を必ず指定してください。また、COBOLアプリケーションの入口名および“COBW3_SAF_GET_PARM_XX”などのサブルーチンで検索するNESパラメタ名は大文字/小文字を区別するため、必要に応じて翻訳オプションのALPHAL(WORD)またはNOALPHALを指定してください。

COBOL目的プログラムをリンクする(DLLの作成)

COBOLソースプログラムを翻訳し作成した目的プログラムのリンクは、プロジェクトマネージャなどからWINLINKコマンドを実行してウィンドウ操作により行う方法と、リンクコマンドを使用してコマンドプロンプト上から指定する方法があります。ここでは、リンクコマンドを使用してリンクするときの例を示します。WINLINKコマンドを使用したリンクやリンクコマンドの詳細については、“NetCOBOL 使用手引書”を参照してください。

作成するDLLをSAFSMPL.dllとします。

リンクは次のように行います。

```

-----
LINK /DLL /OUT:SAFSMPL.dll MainProc.obj
      F3BICBDM.obj F3BINSRT.lib F3BICIMP.lib KERNEL32.LIB LIBC.LIB
      /ENTRY:COBDMAIN
-----

```

リンク時にF3BICBDM.objをリンクし、/ENTRY:COBDMAINを指定することで、初期化ファイル(COBOL85.CBR)をカレントフォルダとは関係なく、DLLと同じフォルダに配置できます。NESなどのサーバ配下のアプリケーションのカレントフォルダは不定であるため、初期化ファイルをDLLと同じフォルダに配置できるのは非常に便利です。なお、詳細については、“NetCOBOL 使用手引書”を参照してください。

4.4.2 NESの設定

Webアプリケーションを実行する上でNESに対して設定しなければならない項目は以下のとおりです。

NESにSAFディレクタを登録する。

詳細については、“4.4.2.1 [NESにSAFディレクタを登録する](#)”を参照してください。

SAFディレクタにWebアプリケーションの登録を行う。

詳細については、“4.4.2.2 [SAFディレクタにWebアプリケーションを登録する](#)”を参照してください。

MIMEタイプの設定を行う。

詳細については、“4.4.2.3 [MIMEタイプの設定](#)”を参照してください。

4.4.2.1 NESにSAFディレクタを登録する

NESでWebアプリケーションを実行できるようにするために、NESに対してSAFディレクタを登録します。SAFディレクタの登録は最初に一度だけ行えばよく、そのあと変更する必要はありません。

【補足】 obj.confファイルに対する修正はNESを再起動するまで、NESには反映されません。

SAFディレクタの登録はNESのobj.confファイルを編集することで行います。obj.confファイルは下記のフォルダにあります。iPlanet Web Serverをお使いの場合は、magnus.confファイルを編集します。

```

-----
[NESのインストールフォルダ]¥https-[サーバ名]¥config
-----

```

たとえば、NESのインストールフォルダをC:¥Netscape¥SuiteSpot、ホスト名をcobol-svrとすると次のようになります。

```

-----
C:¥Netscape¥SuiteSpot¥https-cobol-svr¥config
-----

```

obj.confファイルを適当なテキストエディタを使用して開きます。ファイルの先頭付近に“Init”で始まる行が数行あるので、その最後に次の記述を追加します。なお、ここではCOBOLインストールフォルダを“C:¥COBOL”としています。SAFディレクタはCOBOLのインストールフォルダにあります。

```

-----
Init fn="load-modules" shlib="C:/COBOL/F3BINSAF.dll"
      funcs="COBOL_Init,COBOL_Perform"
-----

```



- ・ 設定は通常、一行で記述します。複数行に分けて記述する場合は、継続行の先頭に必ず空白またはタブを記述してください。
- ・ 他のInitで始まる行は、必要がない限り編集しないでください。編集した場合、NESが動作しなくなる場合もあります。
- ・ obj.confファイル内に記述するファイルのパス名の区切りには、“¥”ではなく、“/”を使用してください。
- ・ あらかじめ、obj.confファイルをバックアップしておくことをおすすめします。

4.4.2.2 SAFディレクトリにWebアプリケーションを登録する

NESでWebアプリケーションを実行できるようにするために、SAFディレクトリに対してWebアプリケーションを登録します。

なお、SAFディレクトリに対する登録はNESを再起動するまで行われません。

NESの起動時にSAFディレクトリがWebアプリケーションをロードし、リクエストごとにロード済みのWebアプリケーションを実行します。

NES起動時のWebアプリケーションのロード指定

obj.confファイル中のSAFディレクトリの登録行の後に、次の記述を追加します。iPlanet Web Serverをお使いの場合は、magnus.confに追加します。

```
-----
Init fn="COBOL_Init" load=" dll名, 入口名[:dll名, 入口名]..."
    [env="環境設定ファイル"]
    [sev="SAFディレクトリのログ出力重要度"]
-----
```

たとえば、

Webアプリケーションの配置フォルダを "C:¥COBAPL"、

Webアプリケーションと入口名を "SAFSMPL1.DLL, SAMPLE1"、"SAFSMPL2.DLL, SAMPLE2"、

環境設定ファイルを "C:¥COBAPL¥SETUP1.BAT"、

SAFディレクトリのログ出力重要度を1

とした場合、次に示すとおりになります。

```
-----
Init fn="COBOL_Init"
    load="C:/COBAPL/SAFSMPL1.dll,SAMPLE1;C:/COBAPL/SAFSMPL2.dll,SAMPLE2"
    env="C:/COBAPL/SETUP1.BAT" sev="1"
-----
```

【補足】 環境設定ファイルの記述内容は、“4.5.1 [COBOL_Init](#)”を参照してください。

Webアプリケーションの実行指定

リクエストごとに起動するWebアプリケーションを指定します。obj.confファイル中に "Service" で始まる行が数行あるので、この中の適当な場所に次の記述を追加します。

```
-----
Service type="MIMEタイプ" fn="COBOL_Perform" app="DLL名, 入口名"
    [sev="SAFディレクトリのログ出力重要度"]
-----
```

【補足】 上記の記述は、<Object name=default>~</Object>タグで囲まれた個所に追加してください。他のObjectタグを使用する場合は、NESのマニュアルを参照してください。

また、複数のWebアプリケーションを起動できるようにしたい場合は、各Webアプリケーションごとに上記の記述を追加します。このとき、MIMEタイプは各Webアプリケーションごとに異なるものを指定してください。

たとえば、

MIMEタイプを " magnus-internal/cob-app1 " 、
 Webアプリケーションの配置フォルダを " C:¥COBAPL " 、
 Webアプリケーションと入口名を " C:¥COBAPL¥SAFSMPL1.DLL,SAMPLE1 " 、
 SAFディレクタのログ出力重要度を1
 とした場合、次に示すとおりになります。

```
-----
Service type="magnus-internal/cob-app1"
      fn="COBOL_Perform" app="C:/COBAPL/SAFSMPL1.DLL,SAMPLE1"
      sev="1"
-----
```

【補足】MIMEタイプには " magnus-internal/ " で始まる文字列を必ず記述してください。

4.4.2.3 MIMEタイプの設定

“ 4.4.2.2 [SAFディレクタにWebアプリケーションを登録する](#) ” で使用したMIMEタイプの設定を行います。MIMEタイプの設定は、NESの管理ページから行います。まず、 " Netscape Server Administration " ページを開きます。

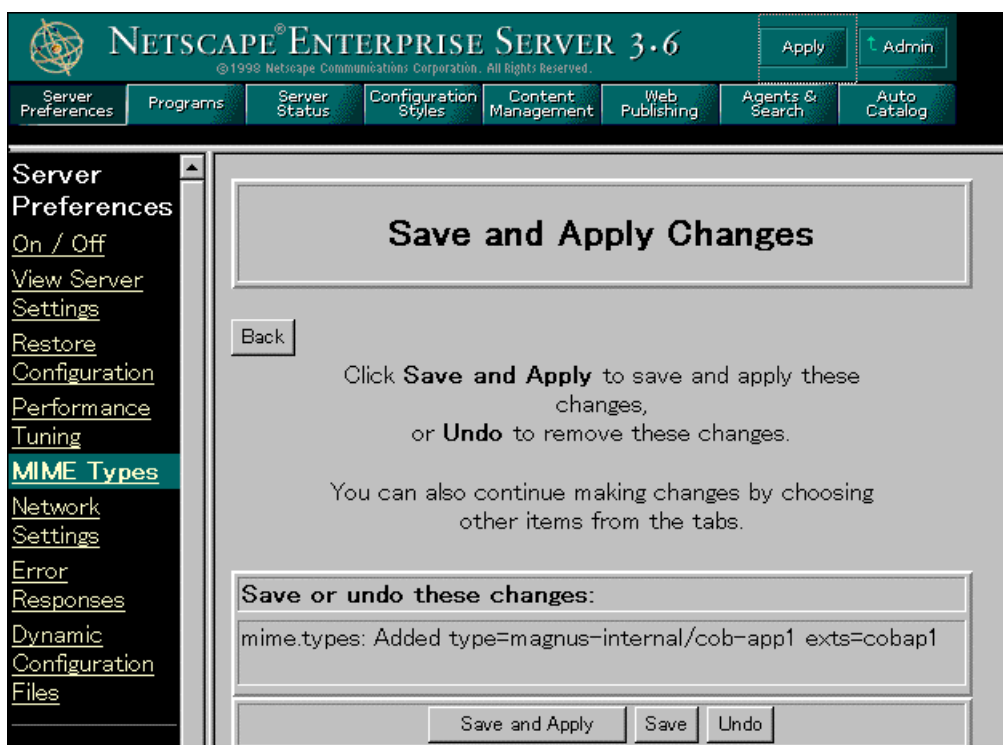


このページでWebサイトのボタン(上図 " cobol-svr " ボタン)を押下すると、 " Server Preference " ページが表示されます。左フレームの中の " MIME Types " をクリックして、 " Global Mime Types " ページを表示させます。

このページで、新しくMIMEタイプを登録します。Content-typeには“4.4.2.2 [SAFディレクトリにWebアプリケーションを登録する](#)”で使用したMIMEタイプを指定します。ここでは、“magnus-internal/cob-app1”とします。また、File Suffixには、Webアプリケーションを特定するための識別子を指定します。ここでは、“cobap1”としますが、拡張子はすでに登録されているものと重複しない範囲で、任意の文字列を登録できます。

【補足】 Categoryには必ず“type”を指定してください。

それぞれを入力後、“New Type”ボタンを押下すると、次の“Save and Apply Changes”ページが表示されるので、“Save and Apply”か“Save”ボタンを押下して、設定を保存してください。



4.4.3 SAFサブルーチンの環境変数設定

SAFサブルーチンを使用したWebアプリケーションを実行するためには、あらかじめ次の実行環境情報を環境変数またはSAFディレクトリの環境設定が実行用の初期化ファイル(COBOL85.CBR)で設定しておく必要があります。なお、実行用の初期化ファイルを使用する場合は各アプリケーションにはF3BICBDM.objをリンクし、アプリケーションが格納されているフォルダの実行用の初期化ファイルを参照できるようにしておいてください。

また、実行用の初期化ファイルはCOBOLがはじめて実行されるまで有効にならないので、SAFディレクトリの環境設定ファイルを使用することをおすすめします。環境設定ファイルの記述内容は、“4.5.1 [COBOL_Init](#)”を参照してください。



注意

NESはシステム環境変数に設定されるもののうち、NESが必要とするもの以外は環境変数ブロックから削除してしまいます。したがって、SAFサブルーチンやWebアプリケーションが使用する環境変数をシステム環境変数として登録しても、参照できません。NESが使用する環境変数についてはNESのマニュアルを参照してください。

システム環境変数、SAFディレクトリの環境設定および実行用の初期化ファイルに設定した環境変数が重複した場合、優先順位は以下のとおりになります。

実行用の初期化ファイル > SAFディレクトリの環境設定 > システム環境変数

システム環境変数またはSAFディレクトリの環境設定ファイルに設定すべきもの

PATH

Webアプリケーションが使用する動的リンクライブラリ(.DLL)のパスを指定します。

SAFディレクトリの環境設定ファイルまたは実行用初期化ファイルに設定するもの

@CBR_ATTACH_TOOL=TEST [起動パラメタ]

デバッグしたいプログラムからデバッガを起動することを指定します。COBOLで作成したWebアプリケーションをデバッガを使用してデバッグする場合には、この環境変数を設定する必要があります。

詳細については、“8.2.3 [SAFアプリケーションの動作確認](#)”を参照してください。

@MessOutFile=ファイル名

COBOLランタイムシステムにより出力される実行時メッセージを格納するファイル名を指定します。これにより、メッセージボックスが画面に出力されることを抑止します。ファイル名の指定は、絶対パスを使用してください。すでに同名のファイルが存在する場合は、そのファイルにメッセージが追加で出力されます。

@WinCloseMsg=OFF

ウィンドウを閉じるときに、確認のためのメッセージを表示する(ON)か、しない(OFF)かを指定します。Webアプリケーションの実行時は、確認メッセージを表示しない(OFF)を指定します。

@CBR_SAF_LOGFILE=ログファイル名

SAFサブルーチンが出力するログ情報を記録するファイルを指定します。障害時などの原因を調べるときに便利です。なお、この場合もファイル名はサーバ上の絶対パスで指定してください。指定がない場合、ログは出力されません。

@CBR_SAF_SEVERITY=重要度

SAFサブルーチンが出力するログ情報の重要度を指定します。以下の値が指定できます。これ以外の値を指定した場合、または指定がない場合、0が指定されたものとみなします。

0	致命的なエラーだけ出力します。
1	0の出力内容に加えて、エラーを出力します。
2	1の出力内容に加えて、警告を出力します。
3	2の出力内容に加えて、SAFサブルーチンのトレースを出力します。



注意

指定する数値が大きくなるほど、アプリケーションは極端に遅くなるため、運用時には必要なエラーだけを出力する値を使用してください。特に障害原因の究明以外で3を使用することはおすすめしません。

他の実行環境情報および詳細については、“NetCOBOL 使用手引書”を参照してください。

4.4.4 SAFアプリケーションの実行

呼出し用ページのFORMタグのACTION属性に、“4.4.2 [NESの設定](#)”で設定したWebアプリケーションが使用する拡張子を記述します。たとえば以下のとおり記述します。

```
-----
<HTML>
<BODY>
<H1>サンプル</H1>
<HR>
<FORM ACTION="test.cobap1">
:
</FORM>
</BODY>
</HTML>
-----
```

この呼出し用ページ(HTML文書)をNESのドキュメントディレクトリの適切な箇所に配置し、これを使用してWebアプリケーションを起動します。

【補足】 テストなどを行っているときWebアプリケーションの入れ替えが頻繁に起きます。また、運用環境についても、Webアプリケーションの機能拡張を行った場合などは入れ替えが必要にな

ります。しかし、一度Webアプリケーションを実行するとNESにロードされた状態のままとなるため、NESを停止する必要があります。NESの停止および再起動は、NESの管理画面から行えます。

4.5 SAFディレクタ リファレンス

4.5.1 COBOL_Init

NESが起動時にロードするWebアプリケーションの指定と環境変数の設定を行います。この処理でロードされたWebアプリケーションは、NESが終了するまでアンロードされません。

書き方:

```
-----
Init fn="COBOL_Init"
  [load="DLL名1,入口名11[,入口名12]...[;DLL名2,入口名21[,入口名22]...]..."
  [env="環境設定ファイル"] [sev="SAFディレクタのログ出力重要度"]
-----
```

呼出し時のデータ設定:

load

ロードするDLL名と実行を予定している入口名を列挙します。DLL名は絶対パスで記述してください。DLL名とそのDLLに属する入口名の区切りには、カンマを使用します。DLL名に対して少なくとも一つは入口名を記述してください。複数の入口名を記述する場合の区切りにもカンマを使用します。複数のDLL名を記述する場合は、セミコロンで区切ってください。

env

環境設定ファイルを絶対パスで指定します。環境設定ファイルは次のように記述してください。

```
-----
SET 環境変数名=環境変数値
-----
```

また、すでに定義されている環境変数を使用する場合は、次のように記述します。

```
-----
%環境変数名%
-----
```

たとえば、Webアプリケーションを "C:¥COBAP" に格納した場合の環境設定ファイルの記述例は、以下のようになります。

```
-----
set PATH=C:¥COBAP;%PATH%

rem ** For COBOL setting **
set @WinCloseMsg=OFF
set @MessOutFile=C:¥COBAP¥LOG¥cobol.txt

rem ** For COBOL SAF subroutine setting **
set @CBR_SAF_LOGFILE=C:¥COBAP¥LOG¥safap.txt
set @CBR_SAF_SEVERITY=0
-----
```

コメント行を記述するには、行の先頭にREMを記述してください。

SETおよびREMでは大文字/小文字は区別されません。

環境変数名および環境変数値は大文字/小文字を区別します。



注意

COBOL_Initを複数行記述する場合、envへの環境設定ファイルの指定は、先頭のCOBOL_Initにだけ記述するか、または、すべて同じ環境設定ファイルを複数行に記述してください。複数のCOBOL_Initに異なる環境設定ファイルを指定した場合、結果は保証されません。

sev

SAFディレクタが出力するログ情報の重要度を指定します。重要度として以下の値が指定できます。下記以外の値を指定した場合、または指定がない場合、0が指定されたものとみなします。

0	致命的なエラーだけ出力します。
1	0の出力内容に加えて、エラーを出力します。
2	1の出力内容に加えて、警告を出力します。
3	2の出力内容に加えて、SAFディレクタのトレースを出力します。

SAFディレクタが出力するログ情報は、NESのエラーログと同じ場所に記録されます。SAFディレクタのログ情報およびNESのエラーログは、NESの管理ページを使用して参照してください。



注意

sevによる重要度の指定は、各COBOL_Initの記述ごとに有効となります。また、この指定は、COBOL_Performでの重要度(env)の指定とは関連ありません。

ここで指定するログ情報の重要度は、SAFサブルーチンが出力するログ情報には影響しません。SAFサブルーチンが出力するログ情報については、“4.4.3 [SAFサブルーチンの環境変数設定](#)”を参照してください。

指定する数値が大きくなるにしたがってアプリケーションは極端に遅くなるため、運用時には必要なエラーだけを出力する値を使用してください。特に障害原因の究明以外で3を使用することはおすすめしません。



注意

NESはシステムの環境変数のうち、NESが必要とするもの以外を削除します。したがって、Webアプリケーションの実行前に設定する必要がある環境変数は、環境設定ファイルで指定してください。

4.5.2 COBOL_Perform

Webアプリケーションを実行します。WebアプリケーションをCOBOL_Initであらかじめロードしていれば、そのアプリケーションを実行します。あらかじめロードされていないときは、この呼出しでロードします。ロード後は、NESにロードされたままとなります。

書き方:

```
-----
Service  [type="MIMEタイプ"] fn="COBOL_Perform" app="DLL名,入口名"
         [sev="SAFディレクタのログ出力重要度"]
-----
```

呼出し時のデータ設定:

app

実行するCOBOLアプリケーションのDLL名と入口名を記述します。DLL名はCOBOL_Initと同様に絶対パスで記述してください。

sev

SAFディレクタが出力するログ情報の重要度を指定します。重要度として以下の値が指定できます。下記以外の値を指定した場合、または指定がない場合、0が指定されたものとみなします。

0	致命的なエラーだけ出力します。
1	0 の出力内容に加えて、エラーを出力します。
2	1 の出力内容に加えて、警告を出力します。
3	2 の出力内容に加えて、SAF ディレクタのトレースを出力します。

SAFディレクタが出力するログ情報は、NESのエラーログと同じ場所に記録されます。SAFディレクタのログ情報およびNESのエラーログは、NESの管理ページを使用して参照してください。

**注意**

sevによる重要度の指定は、各COBOL_Performの記述ごとに有効になります。また、この指定は、COBOL_Initでの重要度(env)の指定とは関連ありません。

ここで指定するログ情報の重要度は、SAFサブルーチンが出力するログ情報には影響しません。SAFサブルーチンが出力するログ情報については、“4.4.3 [SAFサブルーチンの環境変数設定](#)”を参照してください。

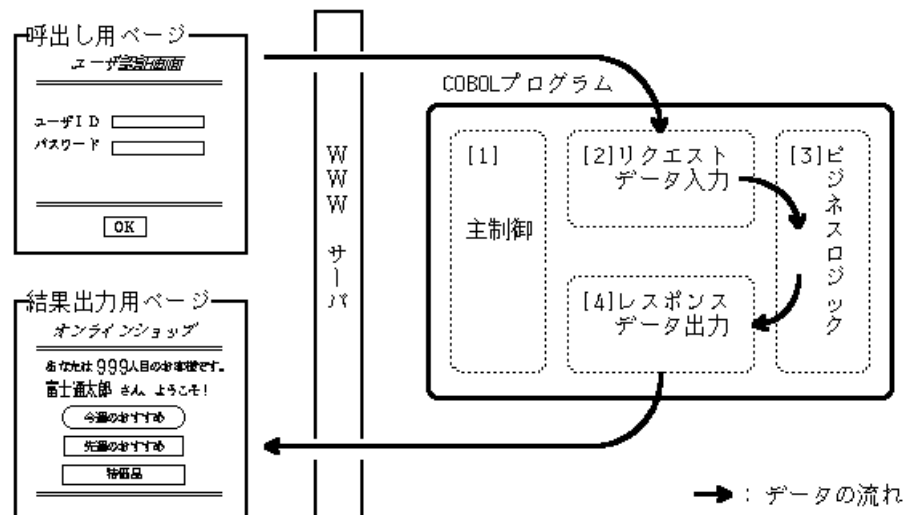
指定する数値が大きくなるにしたがってアプリケーションは極端に遅くなるため、運用時には必要なエラーだけを出力する値を使用してください。特に障害原因の究明以外で3を使用することはおすすめしません。

【補足】 Service行に指定可能な他のパラメタについては、NESのマニュアルを参照してください。

第5章 Webアプリケーション開発支援機能の利用

5.1 Webアプリケーション開発支援機能の概要

Webアプリケーションは、しばしば複数のHTML文書とそれに対応付けられたプログラムから構成される大規模なアプリケーションとなります。しかし、ほとんどの場合において次のような単純な構成のプログラムが複数集まったものと見ることができます。



このため、Webアプリケーションの開発では次の作業を繰り返すことになります。

- 呼出し用ページ(Webパラメタ)の作成
- 呼出し用ページとCOBOLプログラムとの関連付け
- セッション管理等のアプリケーションの制御ロジックの作成
- ビジネスロジックの作成
- 結果出力用ページ(変換名の指定を含む)の作成
- 結果出力用ページとCOBOLプログラムとの関連付け

Webアプリケーション開発支援機能では、上記のb、c、fの作業を軽減するために、次の3つの支援機能を提供します。

Webアプリケーションウィザード

Webアプリケーション作成のためのプロジェクトと雛形となるプログラム(上の図の[1]、[2]、[4]に該当する部分)を生成します。

また、必要に応じて呼出し用ページと結果出力用ページの雛形を生成します。

Webパラメタ取得プログラム生成

呼出し用ページを入力し、このHTML文書に含まれる入力フォームからWebパラメタを入力するプログラム(上の図の[2]に該当する部分)を生成します。

Web結果ページ出力プログラム生成

結果出力用ページのHTML文書を入力し、このHTML文書に含まれる変換名の置き換え、HTML文書出力するプログラム(上の図の[4]に該当する部分)を生成します。

これらはプロジェクトマネージャの一部として動作し、アプリケーションの開発からテスト、資産管理についても従来のCOBOL開発環境と同等の機能を提供します。

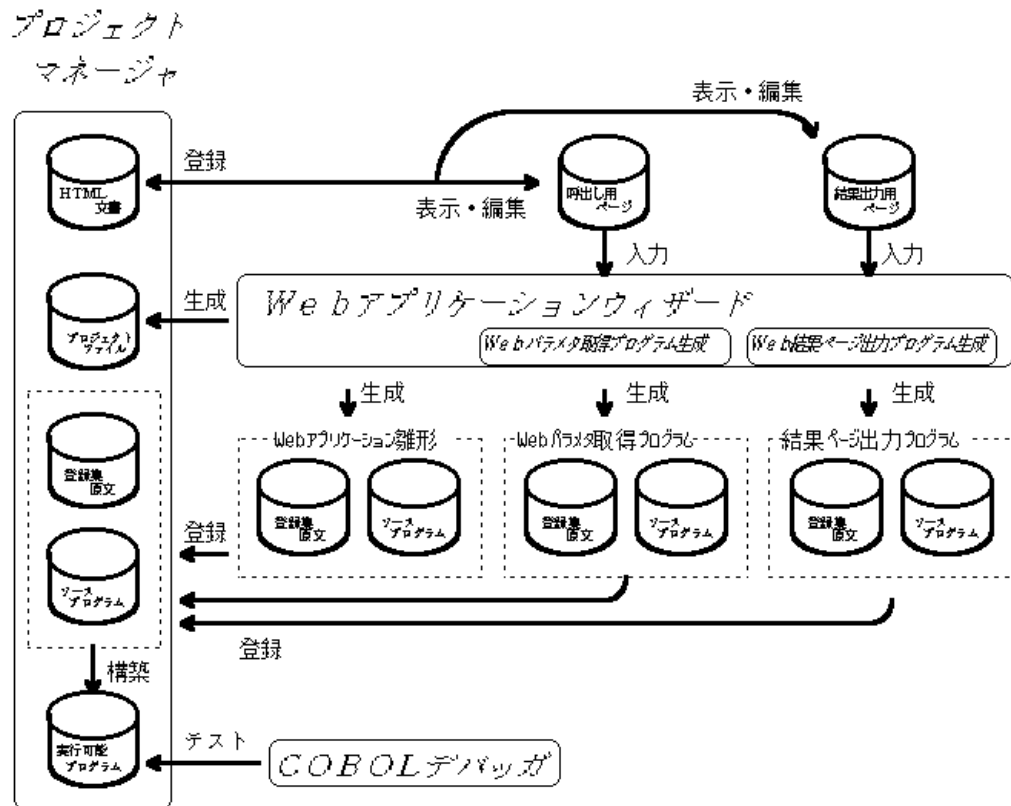


注意

HTML文書もプログラム資産の一部としてプロジェクトに登録することができます。これによりプロジェクトマネージャからHTML文書の参照、編集などの操作が可能になりますが、これらはプロジェクト管理機能の依存関係チェックの対象にはなりません。

Webアプリケーション開発支援機能の概要を“図5-1 [Webアプリケーション開発支援機能の概要](#)”に示します。

図5-1 Webアプリケーション開発支援機能の概要



Webアプリケーション開発支援機能では、次に示すインタフェースを使用するWebアプリケーションの開発が可能です。

CGI(Common Gateway Interface)

ほとんどのWWWサーバが標準でサポートするインタフェースで、実行形式プログラム(EXE)で作成します。COBOLでCGIのWebアプリケーションを作成する場合には、COBOL CGIサブルーチンを使用します。詳細については、“第2章 [CGIアプリケーションの作成と実行](#)”を参照してください。

ISAPI(Internet Server API)

Microsoft(R) Internet Information Server(IIS)などでサポートされるインタフェースで、ダイナミックリンクライブラリ(DLL)で作成します。COBOLでISAPIのWebアプリケーションを作成する場合には、COBOL ISAPIサブルーチンを使用します。詳細については、“第3章 [ISAPIアプリケーションの作成と実行](#)”を参照してください。

NSAPI(Netscape Server API)

Netscape(R) Enterprise Server(NES)などでサポートされるインタフェースで、ダイナミックリンクライブラリ(DLL)で作成します。COBOLでNSAPIのWebアプリケーションを作成する場合には、COBOL SAFサブルーチンを使用します。詳細については、“第4章 [SAFアプリケーションの作成と実行](#)”を参照してください。

WRB API(Web Request Broker API)

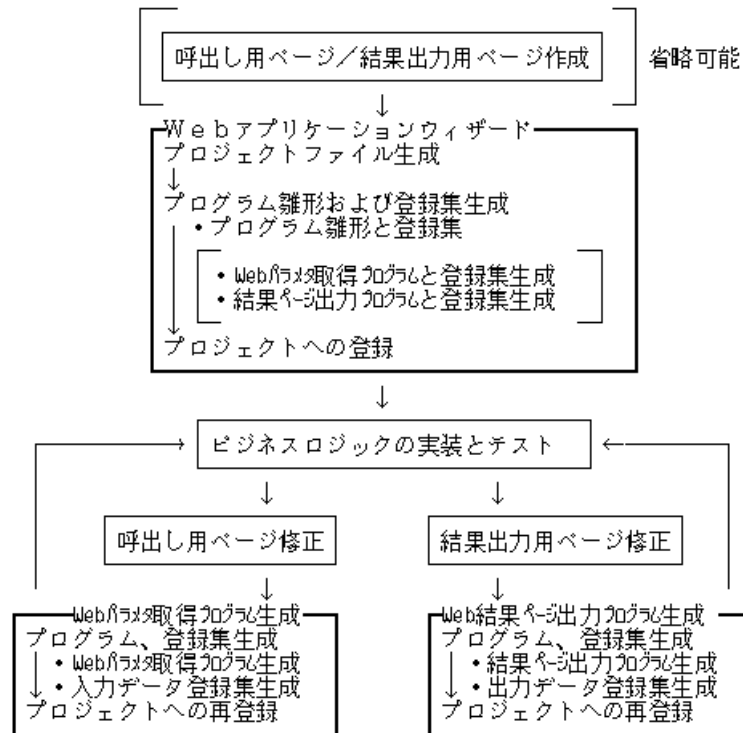
Oracle Application Serverでサポートされるインタフェースで、ダイナミックリンクライブラリ(DLL)で作成します。COBOLでWRB APIのWebアプリケーションを作成する場合には、Oracle COBOLカートリッジを使用します。詳細については、“Oracle COBOLカートリッジ・ユーザズガイド”または“Oracle COBOLカートリッジ for WindowsNT COBOLシステムベンダー固有の設定ガイド(富士通版)”を参照してください。

拡張CGI

InfoProvider Proでサポートされるインタフェースです。実行形式プログラム(EXE)で作成します。COBOLで拡張CGIのWebアプリケーションを作成する場合には、Interstage Application Serverで提供されるCOBOL Webサブルーチンを使用します。詳細については、Interstage Application Serverの“COBOL Webサブルーチン”を参照してください。

5.2 Webアプリケーション開発支援機能を使用した作業の流れ

Webアプリケーション開発支援機能を使用したアプリケーション開発は、次の手順で行います。



太線で囲んだ部分の処理が、Webアプリケーション開発支援機能が提供する機能で処理される部分です。

『ビジネスロジックの実装とテスト』の部分は通常のCOBOLアプリケーションの開発と同様の方法で行うことができます。詳細については“NetCOBOL 使用手引書”を参照してください。

5.3 Webアプリケーションウィザード

Webアプリケーションウィザードは、画面から入力された情報をもとにWebアプリケーション開発用のプロジェクトとWebアプリケーションの雛形となるプログラムを生成するツールです。

以下のような情報を設定するだけで、ビルド可能なプロジェクトと基本的なWebアプリケーションの雛形を作成することができます。

プロジェクトファイル名と格納場所

実行ファイル名

使用するAPIの種類

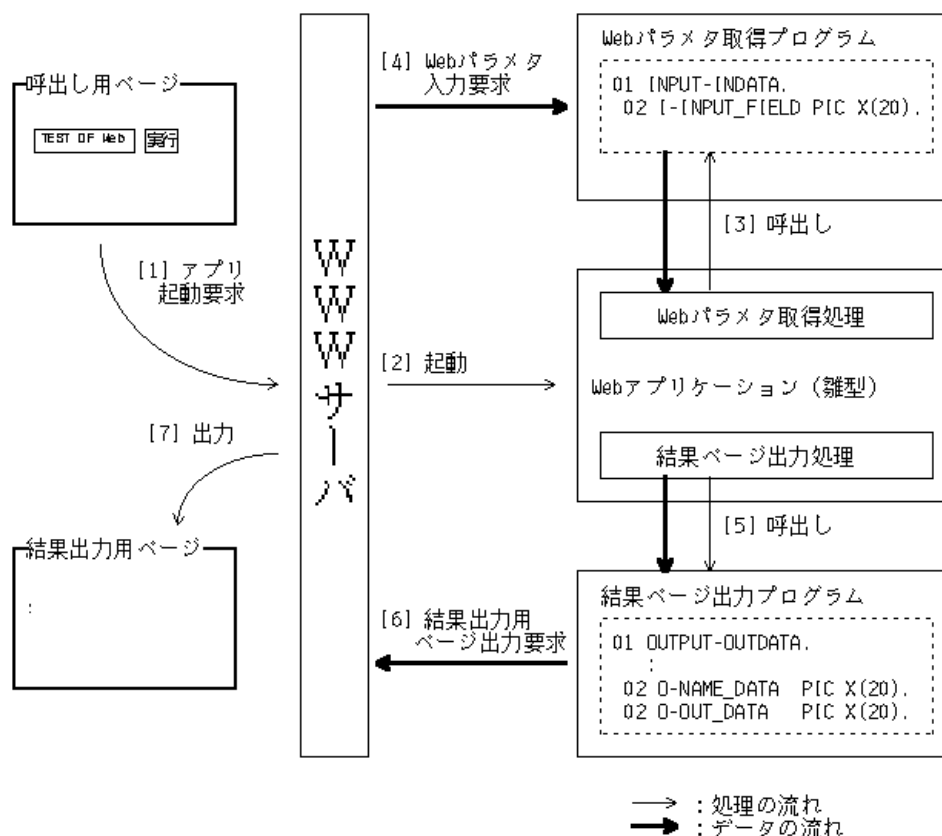
また、呼出し用ページや結果出力用ページを指定して、データの入出力を行うプログラムを生成したり、新たに呼出し用ページや結果出力用ページの雛形を生成することもできます。

5.3.1 Webアプリケーション

ここではWebアプリケーションウィザードが生成するWebアプリケーションの雛形について説明します。

Webアプリケーションの雛形

次のように動作するWebアプリケーションの雛形(呼出し用ページおよび結果出力用ページの登録で“雛形を生成する”を選択した例です。詳細については、“5.3.2.2 [Webアプリケーションウィザードの画面と操作](#)”を参照してください)を生成します。



[図の説明]

[1] 呼出し用ページの入力フォームにデータを入力し、実行ボタンを押すとWWWサーバにWebアプ

リケーションの起動要求が送られ、同時にデータが送信されます。

[2]WWWサーバによってWebアプリケーションが起動されます。

[3]Webアプリケーションは必要な初期化处理などを行った後、Webパラメタ入力用のプログラムを呼び出します。

[4]Webパラメタ入力用のプログラムはWWWサーバに要求を出して、呼出し用ページから送信されたWebパラメタを取得します。

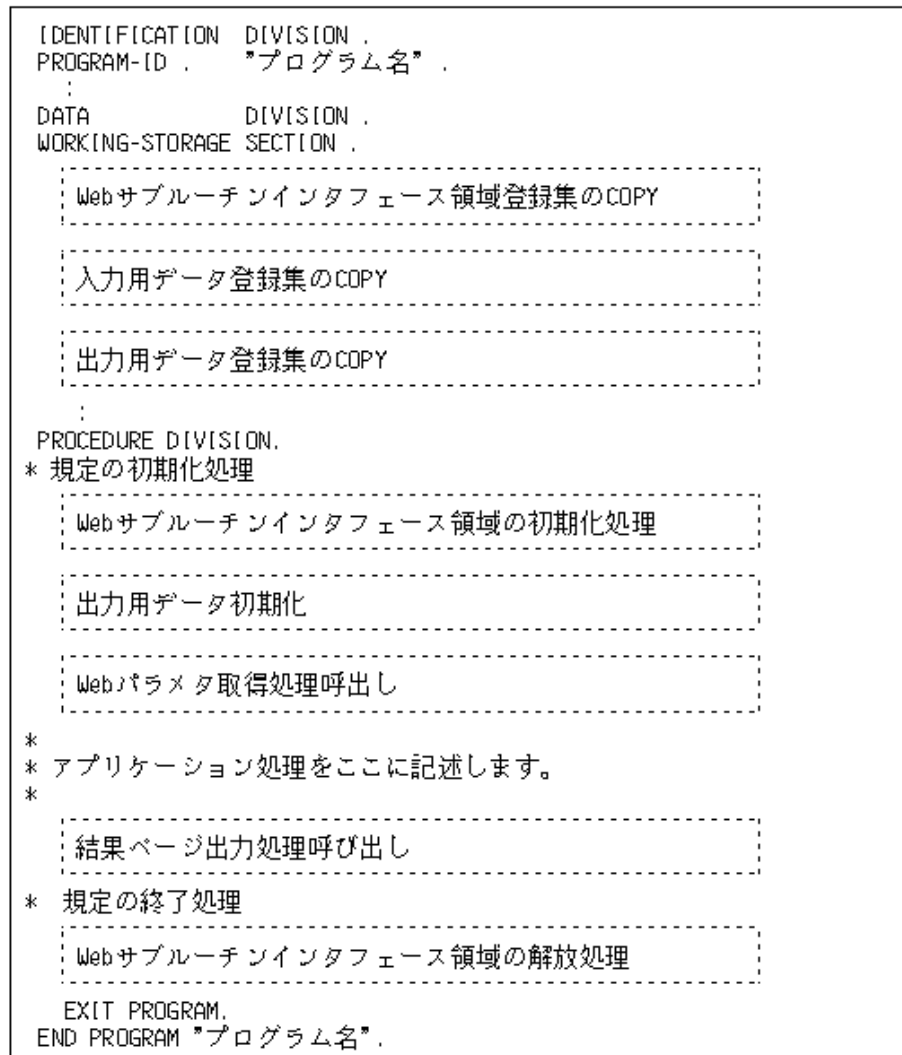
[5]Webアプリケーションは、結果ページ出力用のプログラムを呼び出します。

[6]結果ページ出力用のプログラムは結果出力用ページ内に含まれる変換名を変換文字列で置き換えた上で、結果出力用ページの出力要求をWWWサーバに送ります。

[7]WWWブラウザに結果出力用ページが表示されます。

プログラムの基本構造

Webアプリケーションの雛形は次の構造を持つプログラムです。



Webアプリケーションの雛形の修正方法

Webアプリケーションの雛形には、アプリケーション固有処理は生成されていません。このため、このアプリケーションの雛形の動作は呼出し用ページからWebパラメタを受け取りますが、何もせず結果出力用ページを表示するだけです。

このWebアプリケーションの雛形を修正して、実際のアプリケーションにするためには、次のコメントの下にアプリケーション固有の処理を追加します。

```
*
* アプリケーション処理をここに記述します。
*
```

たとえば、呼出し用ページから受け取ったデータを結果出力用ページにそのまま表示したい場合は次の記述を追加します(生成された、結果ページ出力プログラムは、結果出力用ページに含まれる変換名を変換文字列で置き換える処理を含んでいます。詳細については、“5.5.3.1 [結果出力用ページ](#)”を参照してください)。

```
*
* アプリケーション処理をここに記述します。
*
      MOVE "TEST-DATA"    TO O-NAME_DATA
      MOVE [-INPUT-FIELD  TO O-OUT_DATA
*                                     Webパラメタ取得プログラムから受け取ったデータ項目
*
```

これによりアプリケーションの動作は次のように変更されます。



なお、Webパラメタ取得プログラム、結果ページ出力プログラムおよびそれらの必要とする登録集は、Webパラメタ取得プログラム生成、Web結果ページ出力プログラム生成で生成されるものです。詳細については、“5.4 [Webパラメタ取得プログラム生成](#)”、“5.5 [結果ページ出力プログラム生成](#)”を参照してください。

5.3.2 Webアプリケーションウィザードの使用方法

Webアプリケーションウィザードの使い方には、次の3つがあります。

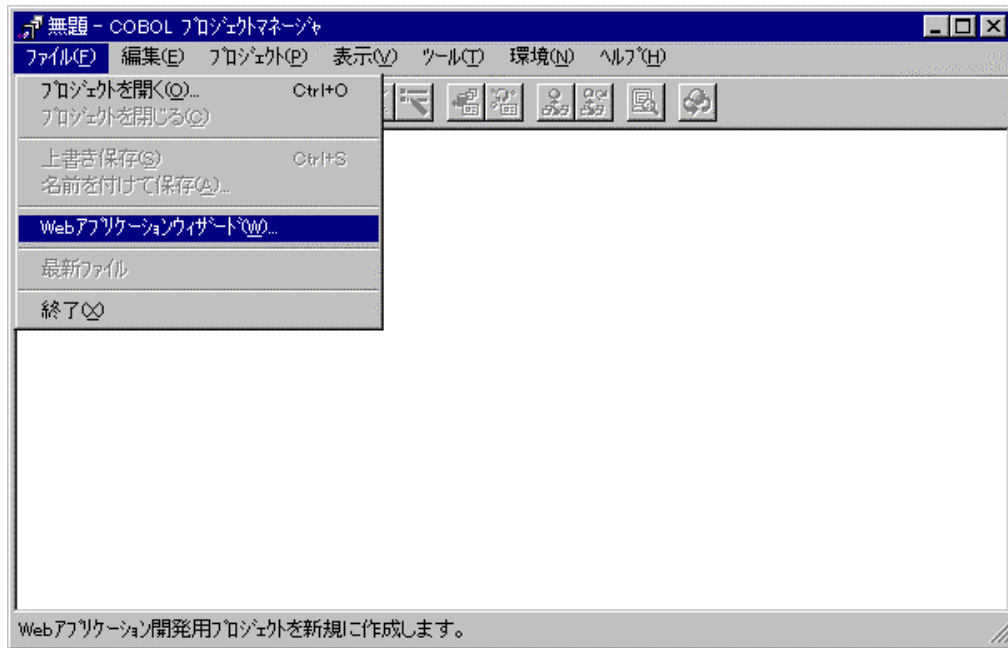
- Webアプリケーション開発用のプロジェクトとWebアプリケーションの雛形となるプログラムを生成する。
- 既存のプロジェクトに新しいターゲットファイルとして、Webアプリケーションを追加する。
- 既存のプロジェクトにすでに存在するターゲットファイルの構成要素にWebアプリケーションの雛形などを追加する。

ここでは最も一般的な使い方であるaについて説明します。その他の使い方については、“5.3.3 [Webアプリケーション雛形の構成](#)”を参照してください。

5.3.2.1 Webアプリケーションウィザードの起動

プロジェクトマネージャの[ファイル]メニューから“Webアプリケーションウィザード”を選択

します。



注意

プロジェクトマネージャが既存のプロジェクトを開いている場合、このプロジェクトは自動的に閉じられます。

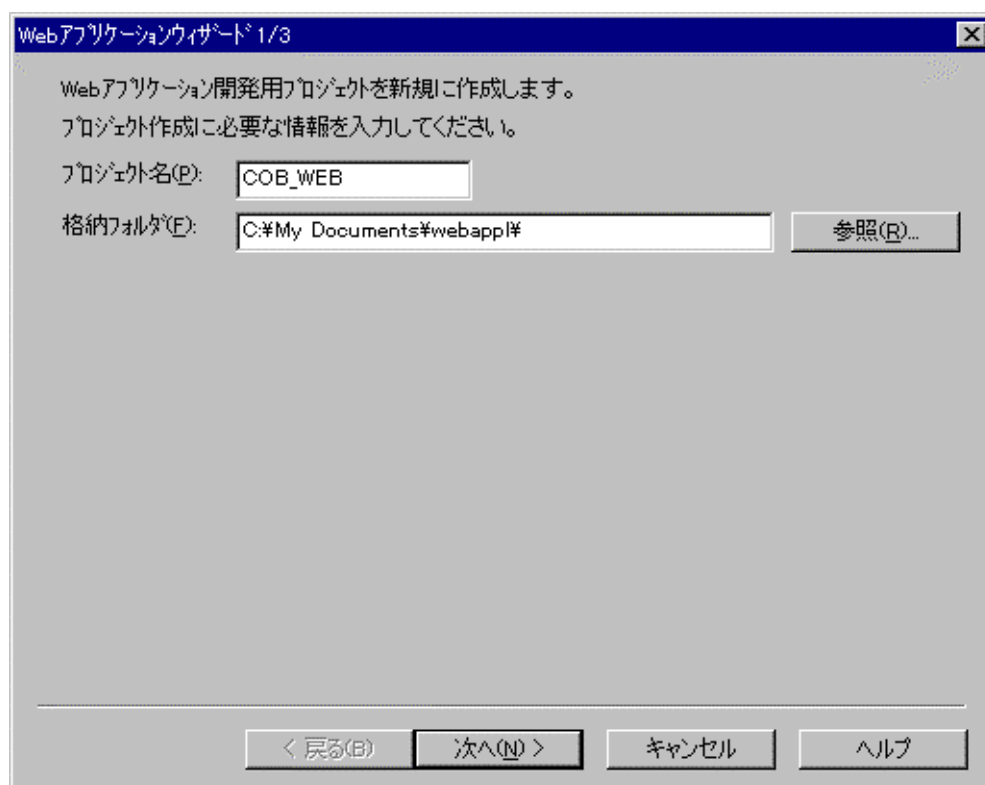
5.3.2.2 Webアプリケーションウィザードの画面と操作

Webアプリケーションウィザードは3つの画面と1つのダイアログボックスを持ちます。ここでは、順番にその画面を示し、操作を説明します。

画面1: プロジェクトの情報入力

画面1ではプロジェクト関連の情報を設定します。

“プロジェクト名”と“格納フォルダ”を設定したら、[次へ]ボタンを押してください。



[画面の説明]

エディットボックス

プロジェクト名:

生成するWebアプリケーション開発用のプロジェクトの名前を指定します。

格納フォルダ:

プロジェクトファイルおよびWebアプリケーションの雛形を構成するソースプログラム、登録集などを格納するフォルダを指定します。

ボタン

参照:

格納フォルダを選択するダイアログを開きます。

画面2: アプリケーションの情報入力

画面2では、実行ファイル名、使用するWebサブルーチンの種類など、生成するアプリケーションの雛形についての基本情報を設定します。

必要な情報を設定したら、[次へ]ボタンを押してください。



[画面の説明]

エディットボックス

実行ファイル名:

ターゲットとしてプロジェクトに追加する実行ファイル名を指定します。

インストール先フォルダ:

COBOLが提供するWebサブルーチン以外を使用する場合に、そのサブルーチンを含む製品のインストール先を指定します。

生成するアプリケーションの種類に、“WRB”または“拡張CGI”を選んだ場合に有効になります。“WRB”を選んだ場合は“COBOLカートリッジ”のインストール先のフォルダを、“拡張CGI”を選んだ場合は“Interstage COBOL Webサブルーチン”のインストール先のフォルダを指定します。

ラジオボタン

使用するAPIの種類を選択してください:

生成するWebアプリケーションの雛形で使用するWebサブルーチンのインタフェースを選択します。実行ファイル名に指定した実行形式によって選択できる種類が異なります。

EXE実行形式の場合は“CGI”、“拡張CGI”が、DLL実行形式の場合は“ISAPI”、“NSAPI”、“WRB”が選択できます。

チェックボックス

INTERSTAGE V2系互換のAPIを使用する:

拡張CGIでAPIとして、INTERSTAGE V2系互換のAPIを使用する際にチェックします。

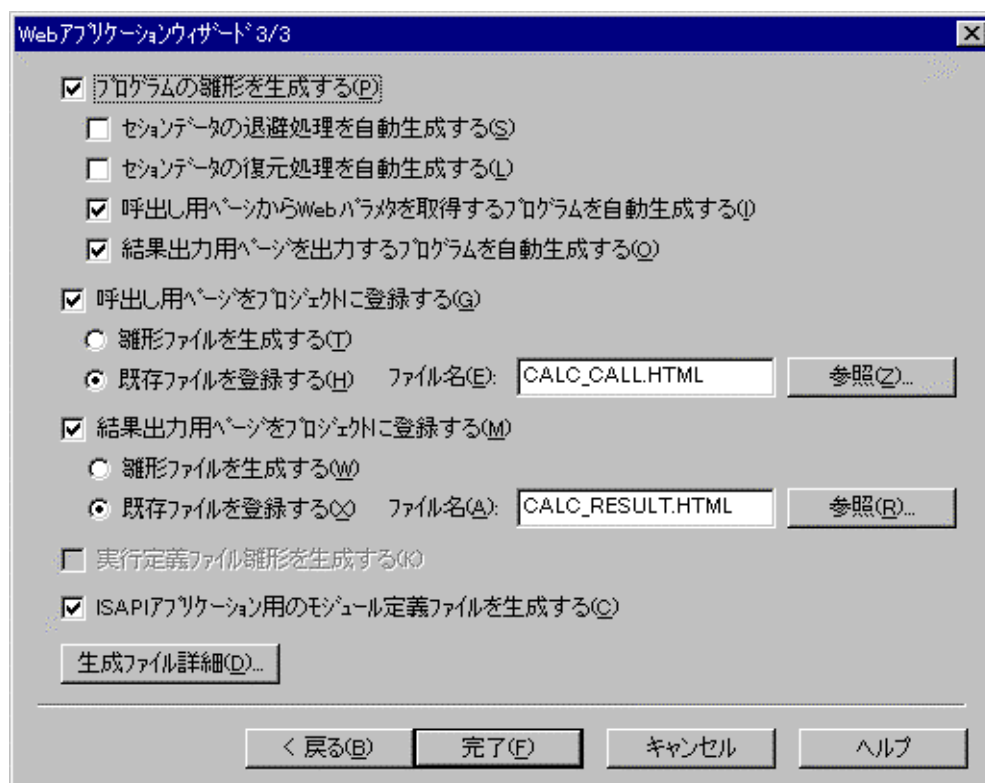
コンボボックス

Webアプリケーションのコード系を選択してください:

生成するアプリケーションの実行コード系を指定します。デフォルトはシフトJISです。使用するAPIの種類に“CGI”、“ISAPI”または“NSAPI”のどれかを選んでください。また、WindowsNT(R)4.0またはWindows(R)2000で開発する場合にのみ、Unicodeを選択することができます。

画面3:ウィザードによるファイル生成方法の指定

画面3では、ウィザードが生成するWebアプリケーションの雛形の生成方法を指定します。必要な情報を指定して、[完了]ボタンを押すとWebアプリケーションの雛形を構成する各種のプログラム資産が生成され、プロジェクトに追加されます。



[画面の説明]

チェックボックス

プログラムの雛形を生成する:

Webアプリケーションの雛形となるプログラムを生成するかどうかを指定します。デフォルトでチェックされています。この指定を変更すると不完全なプロジェクトができてしまうため、通常は変更しないでください。

セッションデータの退避処理を自動生成する:

“プログラムの雛形を生成する”が有効で、Webアプリケーションの種類に“ISAPI”、“NSAPI”または“WRB”のどれかを選択している場合にチェック可能になります。セッションデータの退避処理については“3.3 Webアプリケーション雛形の構成”を参照してください。

セッションデータの復元処理を自動生成する:

“プログラムの雛形を生成する”が有効で、Webアプリケーションの種類に“ISAPI”、“NSAPI”または“WRB”のどれかを選択している場合にチェック可能になります。セッションデータの復元処理については“3.3 Webアプリケーション雛形の構成”を参照してください。

呼出し用ページからWebパラメータを取得するプログラムを自動生成する:

呼出し用ページに含まれる入力フォームから、Webパラメータ取得プログラムおよび入力データ用登録集を生成するかどうかを指定します。“呼出し用ページをプロジェクトに登録する”が有効になっている場合にチェック可能になります。



注意

本機能の使用には“IE V5.0”以降が必要です。

結果出力用ページを出力するプログラムを自動生成する:

結果出力用ページに含まれる変換名の指定から、結果ページ出力プログラムおよび出力データ用登録集を生成するかどうかを指定します。“結果出力用ページをプロジェクトに登録する”が有効になっている場合にチェック可能になります。

呼出し用ページをプロジェクトに登録する:

呼出し用ページをプロジェクトに登録するかどうかを指定します。登録するHTML文書に既存ファイルを使用するか、ウィザードの生成する雛形ファイルを使用するかは、ラジオボタンで選択します。

結果出力用ページをプロジェクトに登録する:

結果出力用ページをプロジェクトに登録するかどうかを指定します。登録するHTML文書に既存ファイルを使用するか、ウィザードの生成する雛形ファイルを使用するかは、ラジオボタンで選択します。

実行定義ファイル雛形を生成する:

実行時に使用するWWWサーバの設定ファイルに変更が必要な場合に、定義情報を含むファイルを生成するかどうかを指定します。Webアプリケーションの種類に“NSAPI”または“拡張CGI”を選択した場合にチェック可能になります。

ISAPIアプリケーション用のモジュール定義ファイルを生成する:

ISAPI用のモジュール定義ファイルを生成するかどうかを指定します。Webアプリケーションの種類に“ISAPI”を選択した場合にのみ、チェック可能になります。

エディットボックス

ファイル名:(呼出し用ページをプロジェクトに登録する)

呼出し用ページに、“既存ファイルを登録する”を選んだ場合、そのファイル名を指定します。

ファイル名:(結果出力用ページをプロジェクトに登録する)

結果出力用ページに、“既存ファイルを登録する”を選んだ場合、そのファイル名を指定します。

ボタン

生成ファイル詳細:

[生成ファイル詳細]ダイアログを開きます。

参照:(呼出し用ページをプロジェクトに登録する)

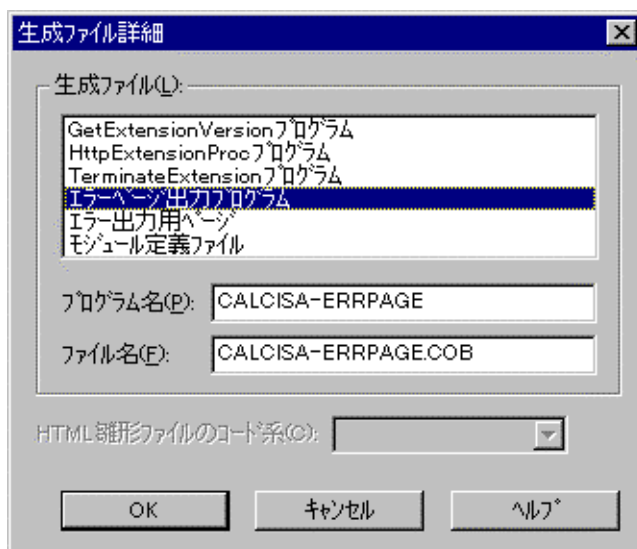
呼出し用ページに使用するファイル名を選択するダイアログを開きます。

参照:(結果出力用ページをプロジェクトに登録する)

結果出力用ページに使用するファイル名を選択するダイアログを開きます。

[生成ファイル詳細]ダイアログ

[生成ファイル詳細]ダイアログでは、ウィザードの生成するプログラム資源のファイル名などのデフォルトの指定を変更します。



[画面の説明]

リストボックス

生成ファイル:

ウィザードの生成するプログラム資源で、ファイル名などのデフォルトの指定を変更できるプログラム資源の一覧を表示します。変更したいものをこれらの中から選択します。

エディットボックス

プログラム名:

リストボックスで選択したWebアプリケーションの雛形の新しい名前を指定します。APIの種類に“ISAPI”を選択した場合は指定できません。

ファイル名:

リストボックスで選択したプログラム資産のファイル名の新しい名前を指定します。

コンボボックス

HTML雛形ファイルのコード系:

ウィザードが生成するHTML雛形ファイル(呼出し用ページ、結果出力用ページ、エラー出力ページ)の出力コード系を選択します。プログラム実行時のコード系に“Unicode”を選択した場合にだけ指定できます。

5.3.3 Webアプリケーション雛形の構成

5.3.3.1 生成するプログラム資源

Webアプリケーションウィザードは複数のプログラム資源を生成しますが、何を生成するかはウィザードの画面上での指定や使用するWebサブルーチンのインタフェースの選択に応じて異なります。

以下にWebサブルーチンの種類と生成されるプログラム資源の関係を示します。

	CGI	ISAPI	NSAPI	WRB	拡張CGI
アプリケーション雛形					
Webパラメタ取得プログラム					
結果ページ出力プログラム					
入力データ登録集					
出力データ登録集					
呼出し用ページ雛形					

結果出力用ページ雛形					
エラーページ出力プログラム					
エラー出力ページ					
モジュール定義ファイル	-		-	-	-
実行定義ファイル雛形	-	-		-	

：常に生成

：選択により生成

-：生成せず

5.3.3.2 Webアプリケーション雛形の違い

Webアプリケーション雛形のプログラム名とファイル名

Webアプリケーションウィザードによって生成されるアプリケーション雛形は基本的には、“5.3.1 [Webアプリケーション](#)”で説明した構造を持ちますが、使用するWebサブルーチンのインタフェースごとに、生成されるプログラムの名前や数が異なります。以下に各Webサブルーチン使用時に生成される名前を示します。

A P I 名	プログラム名	ファイル名	雛形
C G I	EXE名-MAIN	EXE名-MAIN.COB	
I S A P I	GetExtensionVersion	DLL名-GetExtensionVersion.COB	
	HttpExtensionProc	DLL名-HttpExtensionProc.COB	
	TerminateExtension	DLL名-TerminateExtension.COB	
N S A P I	DLL名-ENTRY	DLL名-ENTRY.COB	
W R B	DLL名-ENTRY	DLL名-ENTRY.COB	
拡張 C G I	EXE名-MAIN	EXE名-MAIN.COB	
	EXE名-EXEC1	EXE名-EXEC1.COB	

インタフェースにISAPIを使用する場合はWebアプリケーションのプログラム名は固定になりますが、他のインタフェースを使用する場合は任意の名前を使用できるため、表中のEXE名あるいはDLL名の部分にウィザードで指定した“実行ファイル名”が使用されます。

なお、表の雛形の欄に が示してあるプログラムが“5.3.1 [Webアプリケーション](#)”で説明した構造を持つプログラムです。



ウィザードで指定した“実行ファイル名”に空白またはCOBOLの利用者語に使用できない文字が含まれている場合はその文字は除いて使用します。

Webアプリケーションの起動の指定

Webアプリケーションの起動は、呼出し用ページのFORMタグ中のACTION属性で指定します。たとえば、CGIのインタフェースを使用するSAMPLE.EXEというWebアプリケーションの場合は次のようになります。

```
<FORM METHOD="POST" ACTION="SAMPLE.EXE">
```

ウィザードの画面で“雛形ファイルを生成する”を選んだ場合は使用するインタフェースに応じた適切な記述をウィザードが生成しますが、“既存ファイルを登録する”を選択した場合は、このACTION属性に指定する値を次の規則に従って修正する必要があります。

A P I 名	A C T I O N属性への指定
C G I	[URL名/] EXEファイル名
I S A P I	[URL名/] DLLファイル名

NSAPI	識別名.拡張子(*1)
WRB	[URL名/] 仮想パス/COBOL/DLL名/プログラム名(*2)
拡張CGI	[URL名/] gai-gw/識別名/ (*3)

*1 NSAPIを使用するWebアプリケーションは拡張子だけで識別されるため、識別名の部分は任意の名前でかまいません。拡張子はNESのMIME Typeの設定で追加したものを指定してください。設定の詳細は“第4章 [SAFアプリケーションの作成と実行](#)”を参照してください。

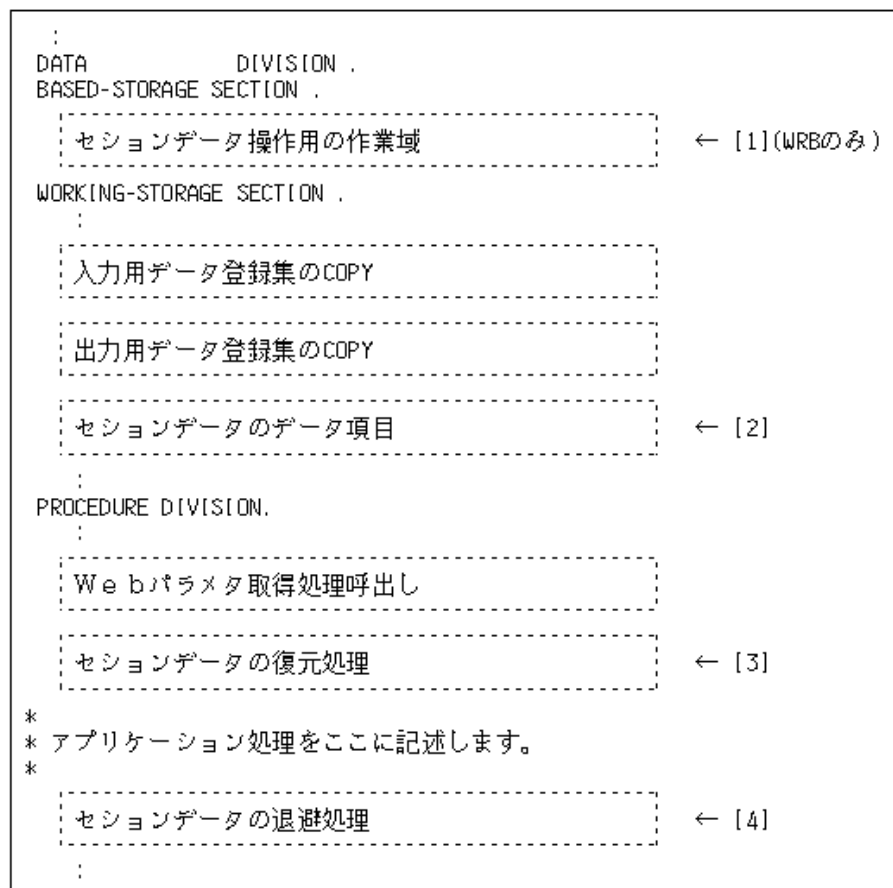
*2 プログラム名は大文字/小文字が区別されます。

*3 識別名の指定方法は、“Interstage Application Server WWWサーバ運用ガイド(InfoProvider Pro編)”を参照してください。

セッションデータ退避/復元処理

Webアプリケーションウィザードの生成するアプリケーションの雛形は“5.3.1 [Webアプリケーション](#)”で説明した構造を持ちますが、使用するAPIにISAPI、NSAPIまたはWRBを選択した場合はセッションデータの退避処理および復元処理を生成させることができます。

ウィザードで“セッションデータの退避処理を自動生成する”あるいは“セッションデータの復元処理を自動生成する”を選択した場合は次のような位置に作業用のデータ項目の雛形と処理を生成します。



なお、[2]の位置に生成されるセッションデータのデータ項目は、“-SESSION-DATA”というサフィックスを持つダミーの集団項目です。実際にセッションデータの復元/退避処理を有効にするには、対象となるデータ項目をこの集団項目に従属するデータ項目として定義する必要があります。また、[3]、[4]の位置に生成されるセッションデータの復元/退避では、セッション範囲を制御するサブルーチン呼出しを行います。セッション範囲を個別に制御する場合には、必要に応じてサブルーチン呼出し処理を修正してください。

5.3.3.3 生成するその他の資源

Webパラメタ取得プログラムと入力データ登録集

ウィザードの画面から“ 呼出し用ページからWebパラメタを取得するプログラムを自動生成する ”を選択すると生成されます。生成されるプログラムおよびファイルはWebパラメタ取得プログラム生成ツールによって生成されるものと同じです。詳細については、“ 5.4.3 [Webパラメタ取得プログラムの生成規則](#) ”を参照してください。

結果ページ出力プログラムと出力データ登録集

ウィザードの画面から“ 結果出力ページを出力するプログラムを自動生成する ”を選択すると生成されます。生成されるプログラムおよびファイルは結果ページ出力プログラム生成ツールによって生成されるものと同じです。

詳細については、“ 5.5.3 [結果ページ出力プログラムの生成規則](#) ”を参照してください。



注意

ウィザードが生成するWebパラメタ取得プログラムおよび結果ページ出力プログラムは、それぞれ[Webパラメタ取得プログラム生成]、[Web結果ページ出力プログラム生成]をデフォルト指定で実行した場合と同じ形式になります。生成内容の指定が必要な場合は、各機能を再度実行してください。

エラーページ出力プログラムとエラー出力ページ

生成したアプリケーションの雛形でエラーを検出した場合に呼び出されるプログラムとエラー出力ページです。

以下の規則に従って生成されます。

種類	プログラム名	ファイル名
エラーページ出力プログラム	EXE/DLL名-ERRPAGE	EXE/DLL名-ERRPAGE.COB
エラー出力ページ	-	EXE/DLL名-ERRPAGE.HTML

エラー出力ページのコード系は、アプリケーションの実行時のコード系がシフトJISの場合はシフトJISが、Unicodeの場合はUCS-2とUTF-8のうちからウィザードで選択したものが使われます。エラーページ出力プログラムは、エラー出力ページがアプリケーションと同じフォルダに存在するものとして処理を行います。

呼出し用ページ雛形と結果出力用ページ雛形

呼出し用ページ雛型と結果出力用ページ雛型は、ウィザードで以下を選択した場合に生成されます。

“ 呼出し用ページをプロジェクトに登録する ”と“ 雛形ファイルを生成する ”を選んだ場合

“ 結果出力用ページをプロジェクトに登録する ”と“ 雛形ファイルを生成する ”を選んだ場合

種類	ファイル名
呼出し用ページ雛形	EXE/DLL名- INPAGE .HTML
結果出力用ページ雛形	EXE/DLL名- OUTPAGE .HTML

呼出し用ページ雛形および結果出力用ページ雛形のコード系は、アプリケーションの実行時のコード系がシフトJISの場合はシフトJISが、Unicodeの場合はUCS-2とUTF-8のうちからウィザードで選択したものが使われます。

モジュール定義ファイル

ウィザードで“ ISAPIアプリケーション用のモジュール定義ファイルを生成する ”を選んだ場合に生成します。

ISAPIアプリケーションに必要なとなるモジュール定義ファイルをDLL名に“-MODULE.DEF”を付けた名前で生成します。

実行定義ファイル

APIにNSAPIまたは拡張CGIを選択し、“実行定義ファイル雛形を生成する”を選んだ場合に以下の名前で生成します。

API名	ファイル名
NSAPI	DLL名-ServerConfiguration.conf
拡張CGI	EXE名-ServerConfiguration.dat

これらはWebアプリケーションの実行時に必要なWWWサーバの設定ファイルの変更のための雛形です。具体的な修正方法については、“第4章 [SAFアプリケーションの作成と実行](#)”または“Interstage Application Server WWWサーバ運用ガイド(InfoProvider Pro編)”を参照してください。

5.3.3.4 その他のWebアプリケーションウィザードの使用法

Webアプリケーションウィザードの使い方にはここまでで説明したWebアプリケーションの雛形を含む新規プロジェクトを生成する以外に2つの使い方があります。

既存のプロジェクトにWebアプリケーションを追加する

既存の実行形式ファイルにWebアプリケーションの情報を追加する

ここではその使い方を説明します。

既存のプロジェクトにWebアプリケーションを追加する

すでに1つ以上の実行形式ファイルを含むプロジェクトに新しい実行形式ファイルとしてWebアプリケーションの雛形を追加します。対象となるプロジェクトは通常の方法で作成したもの、またはWebアプリケーションウィザードによって生成したものです。

次の手順で起動します。

1. 対象となるプロジェクトを開き、プロジェクトマネージャのツリービュー上のプロジェクトファイルを選択します。

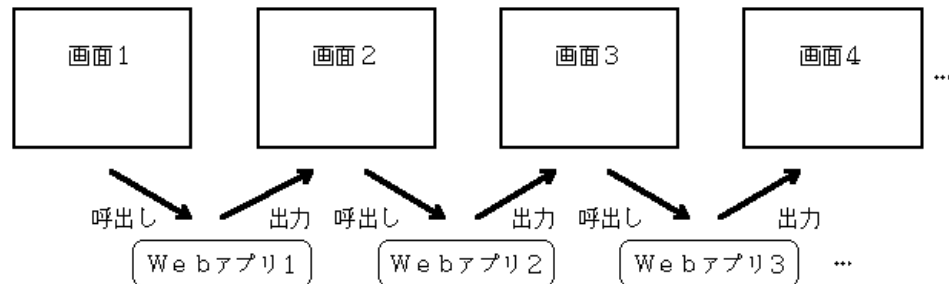


2. [編集]-[Webアプリケーション定義]メニューから“アプリケーション定義の追加”を選択します。

以降の操作は、“5.3.2.2 [Webアプリケーションウィザードの画面と操作](#)”と同じです。ただし、追加するアプリケーションの実行時のコード系は、プロジェクトにすでに存在する実行形式ファ

イルと同じものにしなければなりません。

この機能は次に示すような一連の流れを持つ複数の画面と複数のWebアプリケーションの開発を1つのプロジェクトで管理する場合にご使用ください。



既存の実行形式ファイルにWebアプリケーションの情報を追加する

プロジェクト中にすでに存在する実行形式ファイルの構成要素にWebアプリケーションの雛形をはじめとする各種ファイルを追加します。なお、対象となる実行形式ファイルはWebアプリケーションウィザードによって生成されたものであってはなりません。

次の手順で起動します。

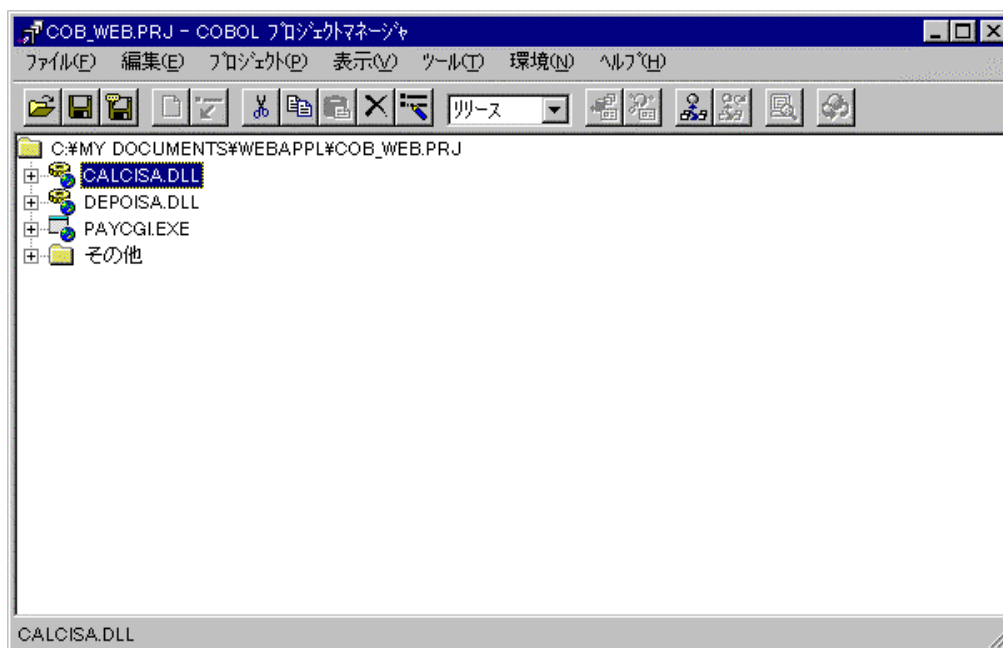
1. プロジェクトを開き、プロジェクトマネージャのツリービュー上で対象となる実行形式ファイルを選択します。



2. [編集]-[Webアプリケーション定義]メニューから“雛形と情報の登録”を選択します。以降の操作は、“5.3.2.2 Webアプリケーションウィザードの画面と操作”と同じです。ただし、対象となる実行形式ファイルの名前と実行時のコード系は変更できません。この機能は、既存のアプリケーションをそのビジネスロジックの部分のみを残し、Webアプリケーション化する場合にご使用ください。

Webアプリケーションウィザードを再実行する

Webアプリケーションウィザードによってプロジェクトに追加された、あるいはWebアプリケーション用の情報を追加された実行形式ファイルを対象にWebアプリケーションウィザードを再度実行することはできません(プロジェクト画面で異なるアイコンで表されます)。



このため、Webアプリケーションウィザードを再実行する必要がある場合には、不要な情報を一度削除する必要があります。

これは次の手順で行います。

1. プロジェクトマネージャのツリービュー上で対象となるWebアプリケーションの実行形式ファイルを選択します。



2. [編集]-[Webアプリケーション定義]メニューから“Webアプリケーションの定義解除”を選択します。

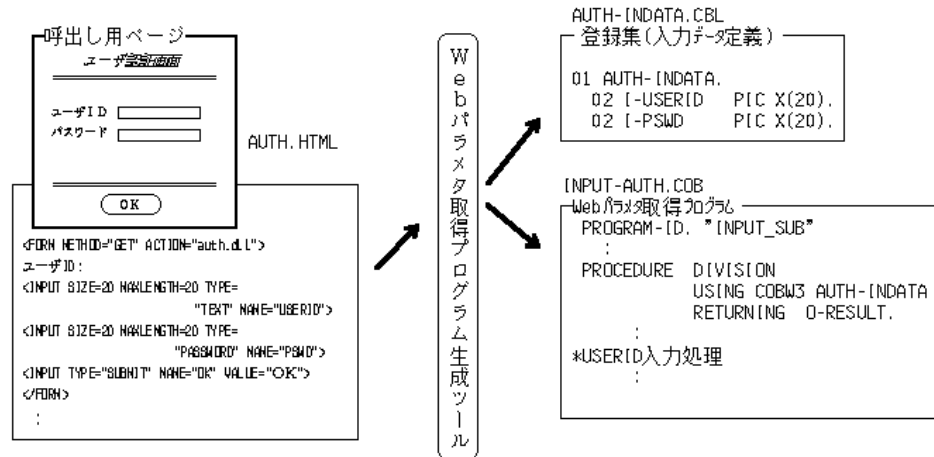


注意

この機能はプロジェクトからWebアプリケーション固有の情報を削除しますが、生成したWebアプリケーションの雛形などのファイルの削除は行いません。不要なら手動で削除してください。

5.4 Webパラメタ取得プログラム生成

Webパラメタ取得プログラム生成ツールは、呼出し用ページに含まれる入力フォームからWebパラメタを取得するプログラムおよび入力データ用の登録集を生成します。



このツールは次のような場合に使用します。

Webアプリケーションウィザードの実行時に“呼出し用ページからWebパラメタを取得するプログラムを自動生成する”を選択しないで生成したWebアプリケーションにWebパラメタ取得プログラムを追加する。

呼出し用ページを差し替えたり、修正した場合、新しい呼出し用ページに合わせてWebパラメタ取得プログラムを再生成する。



注意

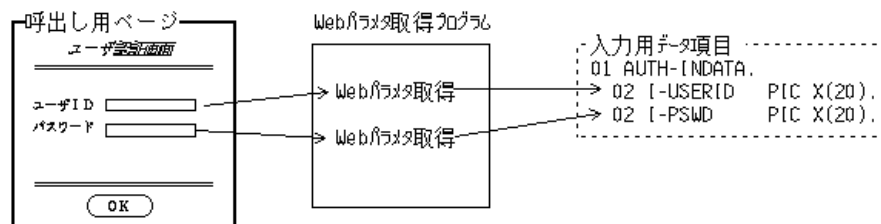
本機能の使用には“IE Version 5.0”以降が必要です。

5.4.1 Webパラメタ取得プログラム

ここではWebパラメタ取得プログラム生成ツールの生成するプログラムについて説明します。

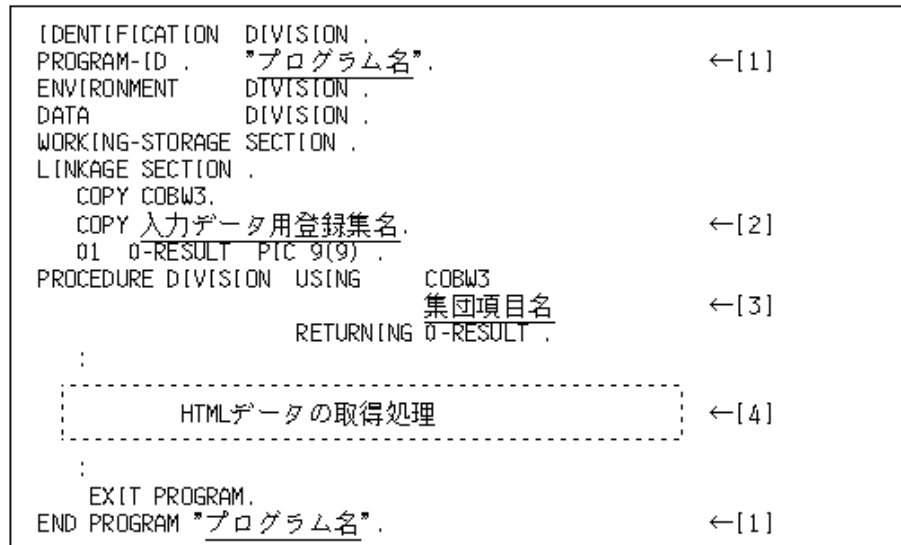
プログラムの動作概要

Webパラメタ取得プログラムには、入力フォームの各入力項目に対応するデータ項目が宣言されており、入力フォームの各入力項目から値を取り出して、対応するデータ項目に設定します。



プログラムの基本構造

Webパラメタ取得プログラムは、次のような構造を持つプログラムです。



[図の説明]

- [1]生成されたプログラム名が、プログラム名段落とプログラム終わり見出しに展開されます。
- [2]生成された入力データ用の登録集ファイル名が展開されます。
- [3]入力フォームの各入力項目に対応するデータ項目は、1つの集団項目に従属する項目として生成されます。ここにはその集団項目名が展開されます。
- [4]入力フォームの各入力項目のデータを取得し、対応するデータ項目に設定する処理が展開されます。



注意

生成されるプログラム名、データ項目名、プログラムおよび登録集のファイル名などの詳細については、“5.4.3 [Webパラメタ取得プログラムの生成規則](#)”を参照してください。

プログラムのインタフェースと使用方法

Webパラメタ取得プログラムは2つの引数と復帰値を持ち、Webアプリケーションからは次の形式で呼び出します。

CALL "Webパラメタ取得プログラム名"
USING データ名1 データ名2
RETURNING データ名3

引数および復帰値のインタフェースは次のとおりです。

データ名1はWWWサーバのAPIを使用するためのインタフェース領域です。登録集“COBW3.CBL”内に定義されているCOBW3を指定します。

データ名2は入力データを格納する領域です。生成された入力データ用登録集内に定義されている01集団項目名を指定します。

データ名3はWebパラメタ取得の成功/不成功を示す次の値を返却します。

0 :すべてのWebパラメタの取得に成功した。

0以外:何らかのエラーが発生した。



複数の入力項目がある場合、ある入力項目に対するWebパラメタの取得に失敗しても、他の項目に対するWebパラメタの取得処理は続行されます。

この呼び出しが正常に終了(復帰値の値が0)した場合は、入力フォームの各入力項目の値がデータ名2に従属する各データ項目に格納されています。

Webアプリケーションウィザードで生成したアプリケーション雛形のWebパラメタ取得プログラム部分を変更あるいは追加した場合はウィザードの生成したアプリケーションの雛形に次の修正が必要となります。

アプリケーションの雛形が取り込む入力データ用の登録集名

アプリケーションの雛形から呼び出すWebパラメタ取得プログラムの名前

Webパラメタ取得プログラム呼出しの第2引数

以下にその例を示します。例では、呼出し用ページをSAMPLE.HTMLからAUTH.HTMLに変更して、Webパラメタ取得プログラムを再度生成したものとします。

[修正前]

```
-----
:
WORKING-STORAGE SECTION.
:
COPY "SAMPLE-INDATA.CBL".
:
PROCEDURE DIVISION.
:
CALL "INPUT-SAMPLE" USING COBW3 SAMPLE-INDATA
RETURNING 0-RESULT.
-----
```

[修正後]

```
-----
:
WORKING-STORAGE SECTION.
:
COPY "AUTH-INDATA.CBL".
:
PROCEDURE DIVISION.
:
CALL "INPUT-AUTH" USING COBW3 AUTH-INDATA
RETURNING 0-RESULT.
-----
```

5.4.2 Webパラメタ取得プログラム生成ツールの使用方法

ここでは、Webパラメタ取得プログラム生成ツールの起動方法と使い方を説明します。

ツールの起動

まず、ツールの起動方法を説明します。

1. プロジェクトマネージャのツリービューから、Webパラメタ取得プログラム生成の対象とする呼出し用ページを選択します(HTML文書はプロジェクト内の“その他”のフォルダに含まれます)。
2. プロジェクトマネージャの[ツール]メニューから“Webパラメタ取得プログラム生成”を

選択します。



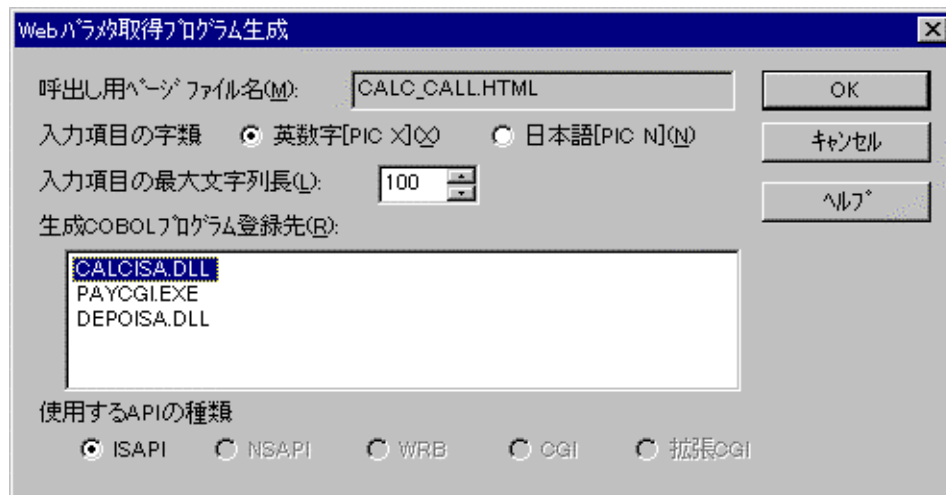
注意

このプロジェクトには1つ以上のWebアプリケーションウィザードで追加した実行形式ファイルが含まれている必要があります。

画面と操作

Webパラメータ取得プログラム生成ツールを起動すると、Webパラメータ取得プログラムの生成に必要な情報を入力するための画面が表示されます。

必要な情報を指定して、[OK]ボタンを押すとツールが実行されます。



【画面の説明】

エディットボックス

呼出し用ページ ファイル名:

ツールの起動時に指定した呼出し用ページのファイル名が表示されます。変更はできません。

入力項目の最大文字列長:

入力データを格納する領域の最大長を指定します。入力項目の字類が英数字の場合は1～1024の範囲で、日本語の場合は1～512の範囲で指定できます。デフォルトとして100が指

定されています。

リストボックス

生成COBOLプログラム登録先:

ツールの生成するWebパラメタ取得プログラムと登録集を構成要素に追加する対象となる実行形式ファイルを選択します。プロジェクト内に、Webアプリケーションウィザードにより追加した実行形式ファイルが複数登録されている場合だけ有効となります。

ラジオボタン

入力項目の字類:

入力データを格納する領域を英数字データ項目として生成するか、日本語データ項目として生成するかを指定します。デフォルトでは英数字データ項目です。

使用するAPIの種類:

生成するWebパラメタ取得プログラムで使用するAPIの種類を表示します。変更はできません。

ボタン

OK:

指定した内容を有効にして、ツールを実行します。

キャンセル:

処理を中断します。

ヘルプ:

ヘルプを表示します。

実行結果

ツールの実行の結果、Webパラメタ取得プログラムと入力データ用の登録集が生成され、指定した実行形式ファイルの構成要素に追加されます。

5.4.3 Webパラメタ取得プログラムの生成規則

ここでは、Webパラメタ取得プログラムの生成規則の詳細について説明します。

5.4.3.1 呼出し用ページ

Webパラメタ取得プログラム生成ツールは、HTML3.2準拠の入力フォームを持つ呼出し用ページを入力として受け付けます。この呼出し用ページのコード系は次のどれかである必要があります。

シフトJIS

UTF-8(Unicode)

UCS-2(Unicode)

なお、入力可能な呼出し用ページのコード系は登録先のWebアプリケーションの実行時のコード系に依存します。WebアプリケーションがシフトJISで動作する場合はシフトJISの、Unicodeで動作する場合はUTF-8またはUCS-2のHTML文書を用意してください。

Webパラメタ取得プログラム生成ツールは、次の形式のFORMタグに含まれるフォームエレメントを調べます。

```
-----
<FORM
  METHOD = "GET" | "POST"
  ACTION = "起動プログラムの指定"
  ENCTYPE= "エンコード方式"
></FORM>
-----
```

フォームエレメントにFILEタイプを使用する場合、FORMタグには必ず以下の属性を指定してください。

```
METHOD = "POST"
ENCTYPE = "multipart/form-data"
```

そして次のどれかの形式のフォームエレメントでNAME属性を持つものすべてを入力項目とみなして、対応する入力データ項目を生成します。

```
-----
<INPUT
  TYPE = TEXT | CHECKBOX | RADIO | HIDDEN | PASSWORD | SUBMIT | FILE
  NAME = 名前
>

<SELECT
  NAME = 名前
  MULTIPLE = 複数指定
> </SELECT>

<TEXTAREA
  NAME = 名前
> </TEXTAREA>
-----
```

FILEタイプは、ISAPI、NSAPI、CGIだけがサポートしています。

なお、呼出し用ページに複数のFORMタグが存在する場合は、それらに含まれるすべてのフォームエレメントが対象となります。



呼出し用ページには、上に挙げた形式以外のFORMタグやフォームエレメントを含めることができます。Webパラメタ取得プログラム生成ツールはそれらの指定を無視します。

5.4.3.2 データ入力用登録集

命名規則

入力とした呼出し用ページの名前(ファイル名から拡張子を除いた部分)とフォームエレメントのNAME属性をもとに次の規則に従って、入力データ項目や登録集の名前が決められます。

登録集ファイル名:

呼出し用ページの名前に“- INDATA.CBL”を付加した名前になります。長さが50文字を超える場合は50文字に収まるように呼出し用ページの名前の最後を切り詰めます。

集団項目名:

呼出し用ページの名前に“- INDATA”を付加した名前になります。呼出し用ページの名前が全角文字で構成されている場合、この文字列も全角文字になります。

入力データ項目名:

フォームエレメントのNAME属性で指定された入力項目名の先頭に“ I- ”を付けた名前になります。入力項目名が全角文字で構成されている場合はこの文字列も全角文字になります。なお、カウンタ項目の場合は“ C- ”が用いられます。当機能で処理可能なNAME属性文字列の上限は50文字です。

また、FILEタイプの場合、入力データ項目の従属項目として、クライアントパス名、クライアントファイル名、コンテンツタイプの3つが固定の名前で生成されます。

【注意】

集団項目名、入力データ項目名はCOBOLの利用者語の規則に従うために、次の場合は、もともになる呼出し用ページの名前やフォームエレメントのNAME属性を変更した上で名前の生成が行われます。

合計が30文字を超える場合は、もともになる名前を切り詰めます。

もとなる名前に半角文字、全角文字が混在している場合は、最初の文字と同じ種別の文字のみを有効とみなします。たとえば、“INPUT属性1”は“INPUT1”とみなされます。もとの名前に含まれる空白またはCOBOLの利用者語に使用できない文字は無視します。フォームエレメントのNAME属性に半角カナは使用できません。

入力データ項目の生成規則

生成される入力データ項目の属性は次のことから決定されます。

入力項目がどんなフォームエレメントによるものか
 フォームエレメントに含まれる属性に指定された値
 同種のフォームエレメントに同名の項目があるか

Webパラメタ取得プログラム生成ツールで指定したデータ項目の字類、最大長

以下に、各フォームエレメントに対して生成されるデータ項目の一般形式の一覧を示します。

タグ種別	同じ名前の項目がない場合	同じ名前の項目がある場合
INPUT TYPE指定なし	02 I-名前 PIC {XIN} (最大長).	02 C-名前 PIC 9(4). 02 I-名前 OCCURS 重複数 PIC {XIN} (最大長).
INPUT TYPE=TEXT		
INPUT TYPE=PASSWORD		
INPUT TYPE=CHECKBOX	02 C-名前 PIC 9(4). 02 I-名前 PIC {XIN} (最大長).	
INPUT TYPE=FILE	02 I-データ名. 03 F-CLIENT_PATH PIC X(128). 03 F-CLIENT_FILENAME PIC X(128). 03 F-CONTENT_TYPE PIC X(64). 03 F-FILE_SIZE PIC S9(18) COMP-5.	
INPUT TYPE=RADIO	02 I-名前 PIC {XIN} (最大長).	02 C-名前 PIC 9(4). 02 I-名前 OCCURS 重複数 PIC {XIN} (最大長).
INPUT TYPE=SUBMIT		
INPUT TYPE=HIDDEN		
TEXTAREA		
SELECT MULTIPLE指定なし		
SELECT MULTIPLE指定あり	02 C-名前 PIC 9(4). 02 I-名前 OCCURS OPT(ONタグ数) PIC {XIN} (最大長).	

5.4.3.3 生成するプログラム

命名規則

入力とした呼出し用ページの名前(ファイル名から拡張子を除いた部分)をもとに、次の規則に従って名前が決められます。

ファイル名:

呼出し用ページの名前の先頭に“INPUT-”を、拡張子として“.COB”を付加した名前になります。ファイル名の長さが50文字を超える場合は50文字に収まるようにHTML文書名の最後を切り詰めます。

プログラム名:

呼出し用ページの名前の先頭に“INPUT-”を付加した名前になります。

【注意】

プログラム名はCOBOLの利用者語の規則に従うために、次の場合は、元になる呼出し用ページの名前を変更した上で名前の生成が行われます。

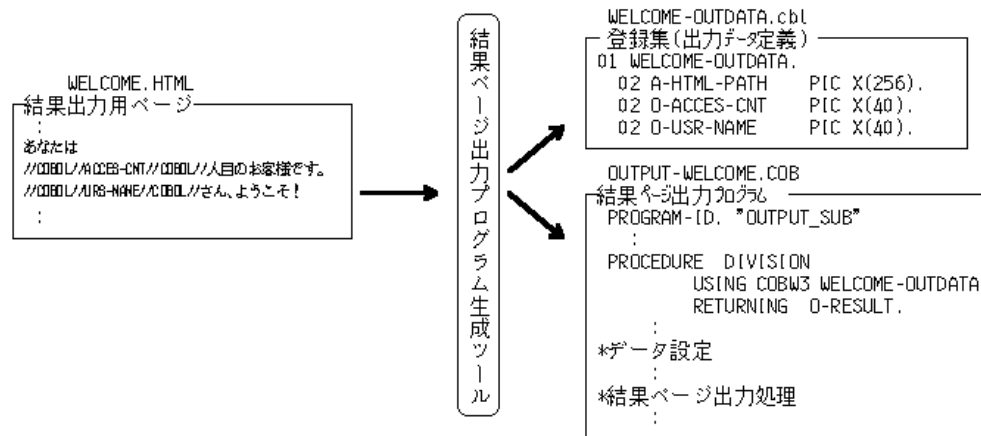
合計が30文字を超える場合は、もとなる名前を切り詰めます。

もとなる名前に半角文字、全角文字が混ざっている場合は、最初の文字と同じ種別の文

字だけが有効になります。たとえば、“INPUT属性1”は“INPUT1”とみなします。
もとの名前に含まれる空白またはCOBOLの利用者語に使用できない文字は無視します。

5.5 結果ページ出力プログラム生成

結果ページ出力プログラム生成ツールは、結果出力用ページに含まれる変換名の指定を変換文字列で置き換え、結果出力用ページを出力するプログラムを生成します。



このツールは次のような場合に使用します。

Webアプリケーションウィザードの実行時に“結果出力用ページを出力するプログラムを自動生成する”を選択しないで生成したWebアプリケーションに結果ページ出力プログラムを追加する。

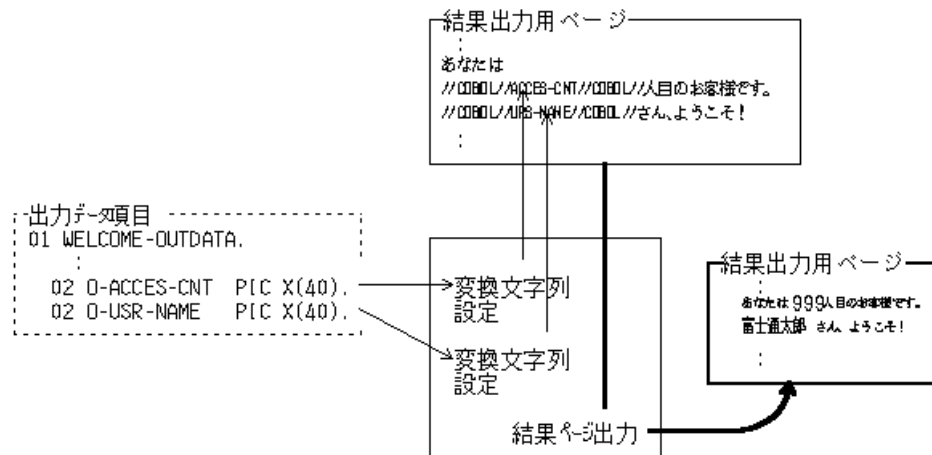
結果出力用ページを差し替えたり、修正した場合、新しい結果出力用ページに合わせて結果ページ出力プログラムを再生成する。

5.5.1 結果ページ出力プログラム

ここでは結果ページ出力プログラム生成ツールによって生成されるプログラムについて説明します。

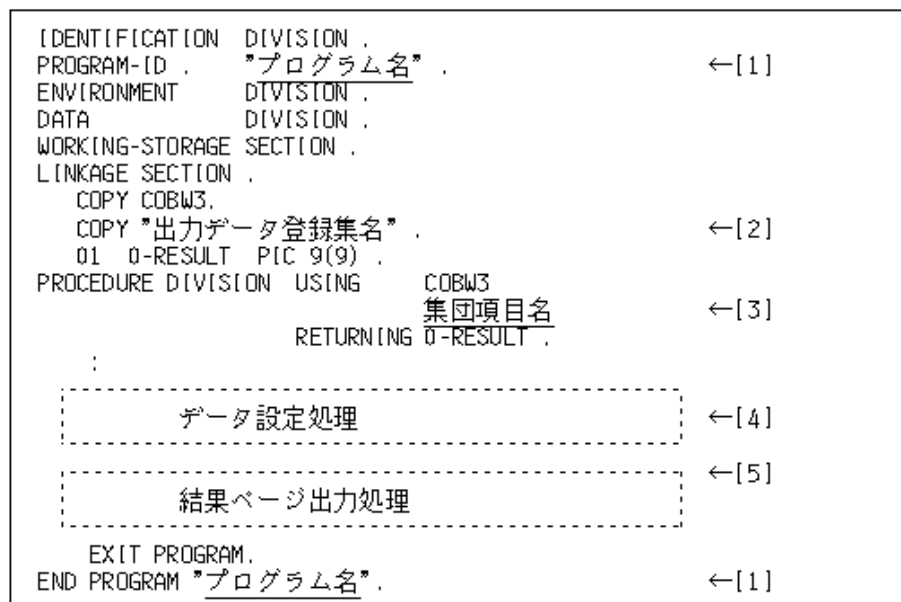
プログラムの動作概要

結果ページ出力プログラムには、結果出力用ページ内に含まれる各変換名に対応するデータ項目が宣言されており、このデータ項目に設定されている変換文字列で変換名を置き換えて、出力します。



プログラムの基本構造

結果ページ出力プログラムは次のような構造を持つプログラムです。



[図の説明]

- [1]生成されたプログラム名が、プログラム名段落とプログラム終わり見出しに展開されます。
- [2]生成された出力データ用の登録集ファイル名が展開されます。
- [3]各変換名の指定に対応するデータ項目は、1つの集団項目に従属する項目として生成されます。ここにはその集団項目名が展開されます。
- [4]変換名の指定を各出力項目の変換文字列で置き換える処理が展開されます。
- [5]結果出力用ページを出力(wwwサーバに送信)する処理が展開されます。



注意

生成されるプログラム名、データ項目名、プログラムおよび登録集のファイル名などの詳細については、“5.5.3 [結果ページ出力プログラムの生成規則](#)”を参照してください。

プログラムのインタフェースと使用方法

結果ページ出力プログラムは2つの引数と復帰値を持ち、Webアプリケーションからは次の形式で呼び出します。

```
[MOVE "結果出力用ページパス" TO A-HTML-PATH.]
CALL "結果ページ出力プログラム"
      SING データ名1 データ名2
      RETURNING データ名3
```

引数および復帰値のインタフェースは次のとおりです。

データ名1はWWWサーバのAPIを使用するためのインタフェース領域です。登録集“COBW3.CBL”内に定義されているCOBW3を指定します。

データ名2は出力データを格納する領域です。生成された出力データ用登録集内に定義されている01集団項目名を指定します。

最初の従属項目として、結果出力用ページの格納フォルダのパスを指定するデータ項目(A-HTML-PATH)を含みます。アプリケーションと異なるフォルダに結果出力用ページがある場合、このデータ項目に結果出力用ページの格納フォルダのパスを設定しておく必要があります。

データ名3は処理の成功/不成功を示す次の値を返却します。

0 :すべての処理が成功した。

0以外:何らかのエラーが発生した。

Webアプリケーションウィザードで生成したアプリケーション雛形の結果ページ出力プログラム部分を変更あるいは追加した場合はウィザードの生成したアプリケーションの雛形に次の修正が必要となります。

アプリケーションの雛形が取り込む出力データ用の登録集名

アプリケーションの雛形から呼び出す結果ページ出力プログラムの名前

結果ページ出力プログラム呼出しの第2引数

以下にその例を示します。例では、結果出力用ページをSAMPLE.HTMLからWELCOME.HTMLに変更して、結果ページ出力プログラムを再度生成したものとします。

[修正前]

```
-----
:
WORKING-STORAGE SECTION.
:
COPY "SAMPLE-OUTDATA.CBL".
:
PROCEDURE DIVISION.
:
CALL "OUTPUT-SAMPLE" USING COBW3 SAMPLE-OUTDATA
RETURNING O-RESULT.
-----
```

[修正後]

```
-----
:
WORKING-STORAGE SECTION.
:
COPY "WELCOME-OUTDATA.CBL".
:
PROCEDURE DIVISION.
```

```

:
CALL "OUTPUT-WELCOME" USING COBW3 WELCOME-OUTDATA
RETURNING 0-RESULT.

```

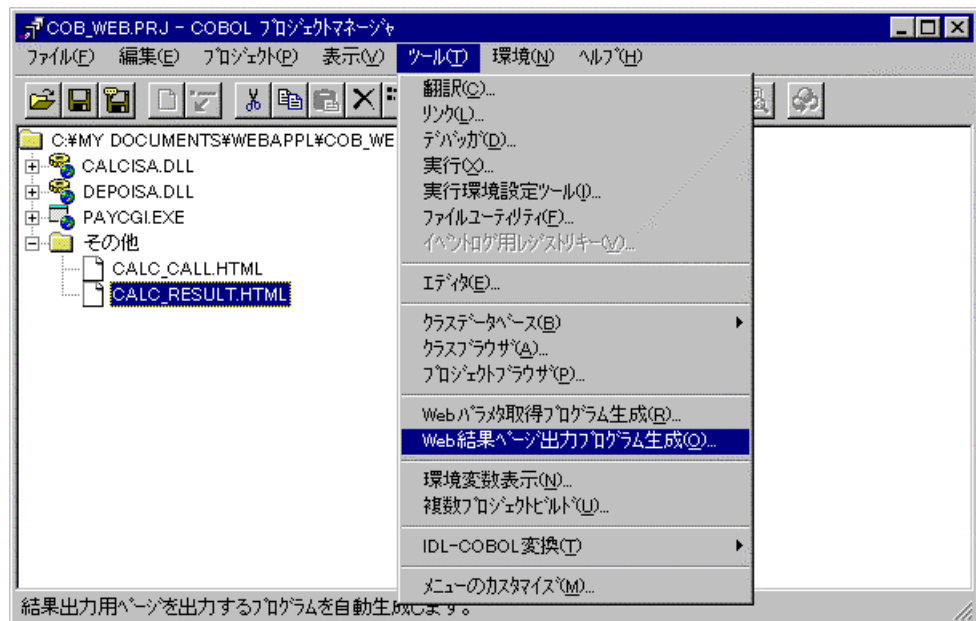
5.5.2 結果ページ出力プログラム生成ツールの使用方法

ここでは、結果ページ出力プログラム生成ツールの起動方法と使い方を説明します。

ツールの起動

まず、ツールの起動方法を説明します。

1. プロジェクトマネージャのツリービューから、結果ページ出力プログラム生成ツールの入力とする結果出力用ページを選択します(下の図のようにHTML文書はプロジェクト内の“その他”のフォルダに含まれます)。



2. プロジェクトマネージャの[ツール]メニューから“Web結果ページ出力プログラム生成”を選択します。



注意

このプロジェクトには少なくとも1つ以上のWebアプリケーションウィザードで追加した実行形式ファイルが含まれている必要があります。

画面と操作

結果ページ出力プログラム生成ツールを起動すると、結果ページ出力プログラムの生成に必要な情報を入力するための画面が表示されます。

必要な情報を指定して、[OK]ボタンを押すとツールが実行されます。

[画面の説明]

エディットボックス

結果出力用ページ ファイル名:

ツールの起動時に指定した結果出力用ページのファイル名が表示されます。変更はできません。

変換文字列の最大文字列長:

変換文字列を格納する領域の最大長を指定します。変換文字列の字類が英数字の場合は1～1024の範囲で、日本語の場合は1～512の範囲で指定できます。デフォルトとして100が指定されています。

繰返し変換データの最大設定数:

繰返し変換データを対象とする場合は生成する変換データの繰返し数の最大値を指定します。1～1000までの範囲で指定することができます。デフォルトとして30が指定されています。

チェックボックス

繰返し変換データを処理対象とする:

繰返し変換データのデータ定義および処理を生成するかどうかを指定します。APIの種類がISAPI、NSAPI、CGIの場合にのみ指定できます。デフォルトの設定では繰返し変換データを処理しません。

リストボックス

生成COBOLプログラム登録先:

ツールの生成した結果ページ出力プログラムと登録集を構成要素に追加する対象となる実行形式ファイルを選択します。プロジェクト内に、Webアプリケーションウィザードにより追加した実行形式ファイルが複数登録されている場合にのみ有効となります。

ラジオボタン

変換文字列の字類:

変換文字列を格納する領域を英数字データ項目として生成するか、日本語データ項目として生成するかを指定します。デフォルトでは英数字データ項目です。

作成するAPIの種類:

生成する結果ページ出力プログラムで使用するAPIの種類を表示します。変更はできません。

ボタン

OK:

指定した内容を有効にしてツールを実行します。

キャンセル:

処理を中断します。

ヘルプ:

ヘルプを表示します。

実行結果

ツールの実行の結果、結果ページ出力プログラムと出力データ用の登録集が生成され、指定した実行形式ファイルの構成要素に追加されます。

5.5.3 結果ページ出力プログラムの生成規則

ここでは、結果ページ出力プログラムの生成規則の詳細について説明します。

5.5.3.1 結果出力用ページ

結果ページ出力プログラム生成ツールの入力とする結果出力用ページのコード系は次のどれかである必要があります。

シフトJIS

UTF-8(Unicode)

UCS-2(Unicode)

なお、入力可能な結果出力用ページのコード系は登録先のWebアプリケーションの実行時のコード系に依存します。WebアプリケーションがシフトJISで動作する場合はシフトJISの、Unicodeで動作する場合はUTF-8またはUCS-2の結果出力用ページを用意してください。

結果ページ出力プログラム生成ツールの入力とする結果出力用ページには、ページ出力時に対応するデータ項目の値に置き換えられて表示される変換名の指定を記述することができます。変換名の指定には次の2つの書き方があります。

[書き方 1]: 単純指定

```
//COBOL//変換名//COBOL// [...]
```

[書き方 2]: 繰返し指定

```
//COBOL_REPEAT_START//
:
//COBOL_REPEAT//変換名//COBOL_REPEAT// [...]
:
//COBOL_REPEAT_END//
```

なお、書き方2の繰返し指定を持つ変換名の指定は、デフォルトでは処理の対象となりません。これは結果ページ出力プログラム生成ツールで“繰返し変換データを処理対象とする”を選択した場合にのみ処理されます。



注意

結果出力用ページのファイル名は、結果ページ出力プログラム中に文字列の形式で展開されます。このとき、ファイル名がCOBOLの文字定数の上限を超える可能性があるため、結果出力用ページのファイル名は50文字以内にしてください。

5.5.3.2 データ出力用登録集

命名規則

入力とした結果出力用ページの名前(ファイル名から拡張子を除いた部分)と変換名をもとに、次の規則に従って出力データ項目や登録集の名前が決められます。

登録集ファイル名:

結果出力用ページの名前に“-OUTDATA.CBL”を付加した名前になります。長さが50文字を超える場合は50文字に収まるように結果出力用ページの名前の最後を切り詰めます。

集団項目名:

結果出力用ページの名前に“-OUTDATA”を付加した名前になります。結果出力用ページの名前が全角文字で構成されている場合、この文字列も全角文字になります。

出力データ項目名:

変換名に“0-”を付けた名前になります。変換名が全角文字で構成されている場合、この文字列も全角文字になります。なお、繰返し指定を持つ変換名を対象とした場合は、繰返し対象の変換データを従属する集団項目“A-REPEAT_DATA_xx”と繰返し数を指定するデータ項目“A-REPEAT_COUNT_xx”(xxの部分は2けたの通番になります)が生成されます。

【注意】

集団項目名、出力データ項目名はCOBOLの利用者語の規則に従うために、次の場合は、もともになる結果出力用ページの名前や変換名を変更した上で名前の生成が行われます。

合計が30文字を超える場合は、もともになる名前を切り詰めます。

もともになる名前に半角文字、全角文字が混ざっている場合は、最初の文字と同じ種別の文字だけが有効になります。たとえば、“OUTPUT名前1”は“OUTPUT1”とみなします。

もとの名前に含まれる空白またはCOBOLの利用者語に使用できない文字は無視します。

出力データ項目の生成規則

生成される出力データ項目の属性は次のことから決定されます。

変換名の形式

結果ページ出力プログラム生成ツールで指定した項目の字類、最大長および最大繰返し数

以下に、各変換名の形式に対して生成されるデータ項目の一般形式の一覧を示します。

	変換名の指定	生成データ項目の形式
単 純 指 定	//COBOL//OUTPUT_DATA1//COBOL// //COBOL//OUTPUT_DATA2//COBOL// //COBOL//OUTPUT_DATA3//COBOL// :	02 0-OUTPUT_DATA1 PIC {X N}(指定長). 02 0-OUTPUT_DATA2 PIC {X N}(指定長). 02 0-OUTPUT_DATA3 PIC {X N}(指定長). :
繰 返 し 指 定	//COBOL_REPEAT//OUTPUT_DATA1//COBOL_REPEAT// //COBOL_REPEAT//OUTPUT_DATA2//COBOL_REPEAT// //COBOL_REPEAT//OUTPUT_DATA3//COBOL_REPEAT// :	02 A-REPEAT_COUNT_01 PIC 9(4). 02 A-REPEAT_DATA_01 OCCURS 最大繰返し数. 03 0-OUTPUT_DATA1 PIC {X N}(指定長). 03 0-OUTPUT_DATA2 PIC {X N}(指定長). 03 0-OUTPUT_DATA3 PIC {X N}(指定長). :

5.5.3.3 生成する出力プログラム

命名規則

入力とした結果出力用ページの名前(ファイル名から拡張子を除いた部分)をもとに、次の規則に従って名前が決められます。

ファイル名:

結果出力用ページの名前の先頭に“OUTPUT-”を、拡張子として“.COB”を付加した名前になります。ファイル名の長さが50文字を超える場合は50文字に収まるように結果出力用ページの名前の最後を切り詰めます。

プログラム名:

結果出力用ページの名前の先頭に“ OUTPUT- ”を付加した名前になります。

【注意】

プログラム名はCOBOLの利用者語の規則に従うため、次の場合は、もとなる結果出力用ページの名前を変更した上で名前の生成が行われます。

合計が30文字を超える場合は、もとなる名前を切り詰めます。

もとなる名前に半角文字、全角文字が混ざっている場合は、最初の文字と同じ種別の文字だけが有効になります。たとえば、“ OUTPUT名前1 ”は“ OUTPUT1 ”とみなします。

もとの名前に含まれる空白またはCOBOLの利用者語に使用できない文字は無視します。

5.6 Webアプリケーション開発のヒント

5.6.1 プロジェクトマネージャからのHTML文書の操作

プロジェクトマネージャではプロジェクトファイルの配下に“その他”というフォルダを1つだけ作成することができます。

呼出し用ページや結果出力用ページなどのHTML文書をこのフォルダに登録すると、プログラム資産の一部としてプロジェクトマネージャ上から表示や編集ができます。

HTML文書の登録

HTML文書は、プロジェクトマネージャウィンドウ上のツリーを操作することによって登録します。以下に、HTML文書の登録方法を説明します。

1. プロジェクトマネージャウィンドウ中のプロジェクトファイルの配下に“その他”フォルダが存在しない場合、フォルダを作成します。
2. “その他”フォルダを選択します。
3. [編集]メニューから“追加”を選択します。
[ファイルを開く]ダイアログが表示されます。
4. HTML文書を選択します。
フォルダの配下に選択したHTML文書が追加されます。

HTML文書の表示と編集

HTML文書は、プロジェクトマネージャウィンドウ上のツリーを操作することによって表示または編集することができます。

以下に、操作方法を説明します。

1. 表示または編集するHTML文書を選択します。
2. 表示する場合には、[プロジェクト]メニューから“表示”を選びます。
WWWブラウザが起動して、選択したHTML文書が表示されます。
3. 編集する場合には、[プロジェクト]メニューから“編集”を選びます。または、選択したHTML文書をダブルクリックします。
HTMLエディタまたはテキストエディタが起動して、選択したHTML文書が編集可能になります。



注意

HTML文書の編集時に起動されるエディタは、デフォルトではシステムでHTML文書に関連付けられているものになります。HTML文書に関連付けられているエディタが存在しない場合は通常のエディタが使用されます。またPowerGEM Plusの設定を変更することによって、システムに関連付けとは別にHTMLエディタの関連付けを行うこともできます。この設定の変更方法については、“PowerGEM Plus説明書”または“PowerGEM Plusの環境設定のヘルプ”を参照してください。

5.6.2 エディタテンプレートの活用

Webアプリケーション開発支援機能で提供されるウィザードやツールは、データの入出力の処理を生成します。しかし、Webアプリケーションで次のような処理をしたい場合、その処理を自分でWebアプリケーションの雛形に追加する必要があります。

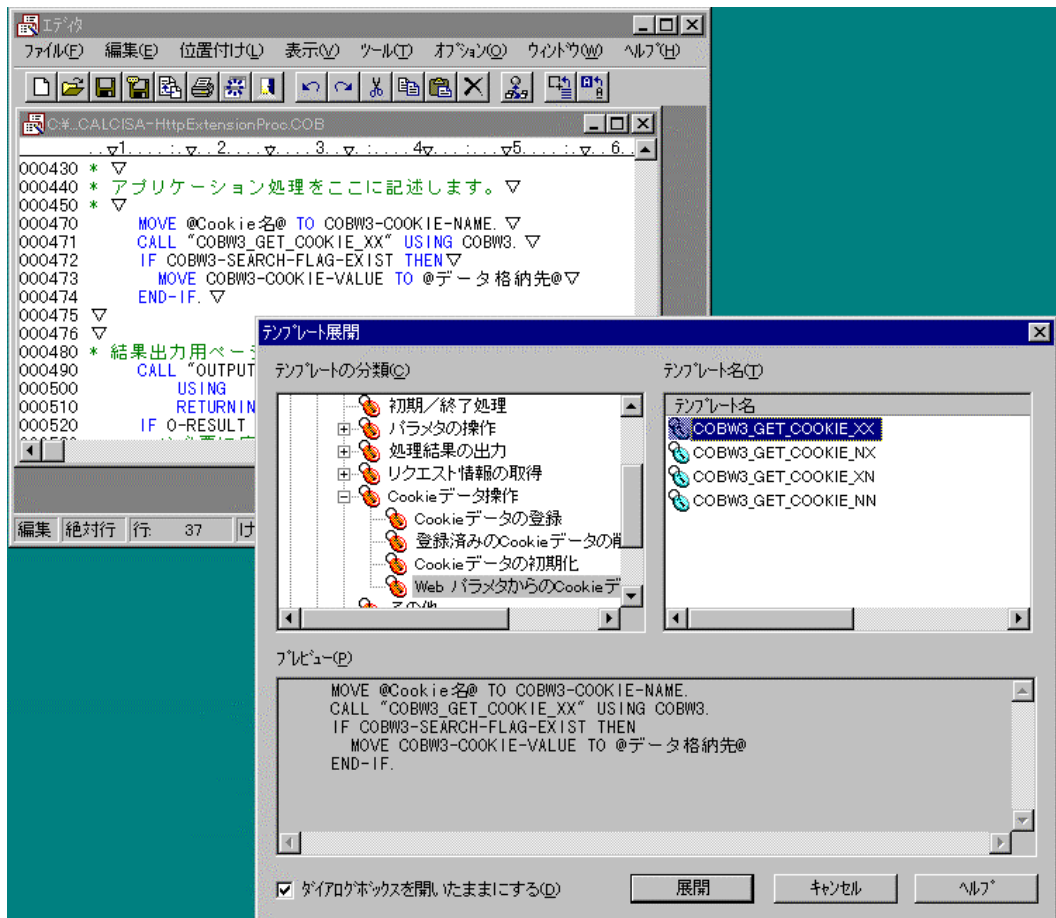
Cookieデータの処理をする

WWWサーバから固有の情報を取得する

認証情報を取得する

このような場合、エディタのテンプレート展開機能をお使いください。テンプレート展開機能は、

目的別に分類された処理の雛形をソースの指定した位置に挿入する機能です。



テンプレート展開機能の使い方を以下に示します。

1. エディタでCOBOLソースを開き、処理を挿入したい行の直前にカーソルを位置づけます。
2. エディタの[編集]メニューから“テンプレート展開”を選びます。
3. “テンプレートの分類”のツリービューから挿入する処理のカテゴリを選び、“テンプレート名”のリストボックスから挿入する処理のテンプレート名を選びます。選択したテンプレートにより実際に挿入される処理が“プレビュー”に表示されています。
4. [展開]ボタンを押すとテンプレートで用意されている処理が展開されて、ソースに挿入されます。



注意

この機能はプロジェクトマネージャの持つエディタを使用する場合だけ利用できます。

第6章 WebアプリケーションにおけるUnicode の利用

6.1 Unicodeの利用

COBOL97 V60L10以降またはNetCOBOL V7.0L10以降で提供するCOBOL Webサブルーチンは、すべてUnicodeをサポートしています。Unicodeで動作するWebアプリケーションは、文字数の増加により表現力が向上しますが、従来のシフトJISを使用したWebアプリケーションと異なる点があります。以下、Unicodeを使用したWebアプリケーションを作成、運用する上での注意事項について記述します。

なお、MeFt/WebでUnicodeを使用する場合は、“MeFt/Web 説明書”を参照してください。

はじめに、シフトJISを使用したWebアプリケーションとUnicodeを使用したWebアプリケーションの作成および運用上の違いについて下表に示します。

項番	項目	シフトJIS	Unicode
1	WWW ブラウザ	シフトJIS が表示できるもの。最近のWWW ブラウザならどれも可能	UTF-8またはUCS-2が表示でき、WebパラメータをUTF-8で送信できるWWW ブラウザ
2	WWW ブラウザからの入力コード系	シフトJIS またはEUC（注）	UTF-8（注）
3	WWWブラウザへの出力コード系	シフトJIS	UTF-8またはUCS-2
4	翻訳オプション	不要	必要
5	使用可能なサブルーチン	制限なし	一部制限あり
6	呼出し用ページに使用できるHTML文書のコード系	シフトJIS またはEUC	UTF-8またはUCS-2
7	結果出力用ページに使用できるHTML文書のコード系	シフトJIS	UTF-8、UCS-2またはシフトJIS
8	ログファイルのコード系	シフトJIS	UCS-2またはシフトJIS
9	コード系の異なるCOBOL アプリケーションの混在	基本的には不可。ただし、使用するサブルーチンやWWW サーバによっては、ある条件下で可能。	

注

COBOL WebサブルーチンがCOBOLアプリケーションのコード系に合わせて変換処理を行うため、COBOLアプリケーションで入力コード系を意識する必要はありません。

以下、個々の内容について説明します。

6.2 WWWブラウザ

従来のコード系のWebアプリケーションは、日本語化されているWWWブラウザであれば、問題なく使用することができました。しかし、UnicodeのWebアプリケーションは、UnicodeをサポートしているWWWブラウザしか使用できません。特に古いWWWブラウザはUnicodeをサポートしていないものが多いため、注意してください。また、最近のWWWブラウザでもサポートの程度に差があります。COBOL WebサブルーチンでUnicodeを使用した場合、WWWブラウザは次の条件を満たすものでなければなりません。

UTF-8またはUCS-2のHTML文書が正しく表示できる。

WWWブラウザから送信するWebパラメタをUTF-8で送信できる。

6.3 WWWブラウザからの入力コード系

シフトJISのWebアプリケーションは、WWWブラウザからの入力コード系がシフトJISかEUCかを自動判別しています。したがって、呼出し用ページのHTML文書のコード系はシフトJISでもEUCでも問題ありませんでした。しかし、Unicodeを使用したWebアプリケーションはUTF-8しか認識できません。これは、UTF-8と他のコード系を自動的に判別できないこと、および一般的なWWWブラウザがUnicodeのHTML文書を使用した時にWebパラメタのコード系をUTF-8として送信するためです。したがって、UnicodeのWebアプリケーションを起動する呼出し用ページは、Unicodeでなければなりません。

6.4 WWWブラウザへの出力コード系

従来のシフトJISのWebアプリケーションでは、WWWブラウザへの出力コード系をシフトJISとしていました。しかし、UnicodeのWebアプリケーションでは、UTF-8とUCS-2を選択できるようになっています。システム全体を通して半角英数字が多い場合はデータ量の少なくなるUTF-8、全角文字が多い場合はデータ量の少なくなるUCS-2を選択できます。このWWWブラウザへの出力コード系は以下の環境変数情報で指定します。

@CBR_WEB_OUT_CODE

WWWブラウザへの出力コード系を指定します。

値	意味
UTF8	WWW ブラウザへの出力コード系をUTF-8とします。
UCS2	WWW ブラウザへの出力コード系をUCS-2リトルエンディアンとします。



シフトJISのWebアプリケーションは出力コード系がシフトJIS固定のため、この環境変数に値を指定しても意味を持ちません。

Webアプリケーションからこの環境変数情報を操作した場合の動作は不定です。

6.5 翻訳オプション

シフトJISのWebアプリケーションには必須の翻訳オプションはありませんが、UnicodeのWebアプリケーションには翻訳オプション " RCS(UCS2) " を指定しなければなりません。

6.6 使用可能なサブルーチン

シフトJISのWebアプリケーションでは、COBOLプログラム中で使用できるWebサブルーチンに制限はありません。しかし、UnicodeのWebアプリケーションでは、下表のWebサブルーチンは使用できません。代替サブルーチンを使用してください。なお、資産の移行性を高めるためには、シフトJISのWebアプリケーションでも代替サブルーチンの使用をおすすめします。

サブルーチン名	代替サブルーチン
COBW3_NAME	COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX
COBW3_NAMEN	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
COBW3_SET_COOKIE	COBW3_SET_COOKIE_XX COBW3_SET_COOKIE_NX COBW3_SET_COOKIE_XN COBW3_SET_COOKIE_NN
COBW3_DEL_COOKIE	COBW3_DEL_COOKIE_X COBW3_DEL_COOKIE_N
COBW3_GET_COOKIE	COBW3_GET_COOKIE_XX COBW3_GET_COOKIE_NX
COBW3_GET_COOKIEEN	COBW3_GET_COOKIE_XN COBW3_GET_COOKIE_NN

サブルーチンの詳細は“第7章 [COBOL Webサブルーチンの使い方](#)”を参照してください。

6.7 呼出し用ページに使用できるHTML文書のコード系

シフトJISのWebアプリケーションでは、呼出し用ページに使用できるHTML文書のコード系はシフトJISおよびEUCです。しかし、UnicodeのWebアプリケーションでは、呼出し用ページに使用できるHTML文書のコード系はUTF-8またはUCS-2(wwwブラウザに依存します)でなければなりません。

6.8 結果出力用ページに使用できるHTML文書のコード系

シフトJISのWebアプリケーションでは、結果出力用ページに使用できるHTML文書のコード系はシフトJISだけでした。しかし、UnicodeのWebアプリケーションでは以下のコード系のHTML文書を扱えます。

- シフトJIS

- UTF-8

- UCS-2

ただし、使用するHTMLファイルのコード系とWWWブラウザへの出力コード系を統一させておく方が実行性能はよくなります。

6.9 ログファイルのコード系

Webサブルーチンが出力するログファイルのコード系は、以下により決定します。

新規にログファイルが作成される場合

実行環境で最初に動作したCOBOLアプリケーションのコード系で作成します。

すでに作成済のログファイルにログ情報が追加される場合

既存ログファイルのコード系で作成します。

6.10 コード系の異なるCOBOLアプリケーションの混在

同一プロセス内ではコード系の異なるCOBOLアプリケーションは混在できません。したがって、IISやNESでは、Webアプリケーションのコード系を混在することはできません。

ただし、IISでは分離プロセスを使用することで、同一WWWサーバ内でコード系を混在できます。また、CGIアプリケーションはそれぞれ独立したプロセスとして動作するため、同一サーバ内でコード系を混在させることができます。

COBOLアプリケーションにおけるコード系の混在に関しては“NetCOBOL 使用手引書”を参照してください。

第7章 COBOL Webサブルーチンの使い方

7.1 COBOL Webサブルーチンの機能

COBOL Webサブルーチン(以降ではWebサブルーチンといいます)では次の機能を提供しています。

7.1.1 基本機能

WWWブラウザからのリクエストに応じて業務を行うWebアプリケーションの作成を可能にします。

機能	サブルーチン名	用途
作業環境の設定および Webパラメタの獲得 (参照“7.2.2.1 作業環境の設定およびWebパラメタの獲得 ”)	COBW3_INIT	Webサブルーチン作業環境の設定およびWebパラメタの獲得
Webパラメタの操作 (参照“7.2.2.2 Webパラメタの操作 ”)	COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX COBW3_GET_VALUE_XN COBW3_GET_VALUE_NN	Webパラメタ中のNAMEの検索およびVALUEの取得
	COBW3_CHECK_VALUE_X COBW3_CHECK_VALUE_N	Webパラメタ中のVALUEの検索
処理結果の出力 (参照“7.2.2.3 処理結果の出力 ”)	COBW3_PUT_HEAD	ヘッダ出力
	COBW3_PUT_HTML	結果出力用ページの出力
	COBW3_SET_CNV_XX COBW3_SET_CNV_NX COBW3_SET_CNV_XN COBW3_SET_CNV_NN	結果出力用ページ内の変換データの登録
	COBW3_DEL_CNV_X COBW3_DEL_CNV_N	結果出力用ページ内の変換データの削除
	COBW3_INIT_CNV	結果出力用ページ内の変換データの初期化
	COBW3_SET_REPEAT_XX COBW3_SET_REPEAT_NX COBW3_SET_REPEAT_XN COBW3_SET_REPEAT_NN	結果出力用ページ内の繰返し変換データの登録
	COBW3_DEL_REPEAT_X COBW3_DEL_REPEAT_N	結果出力用ページ内の繰返し変換データの削除
	COBW3_INIT_REPEAT	結果出力用ページ内の繰返し変換データの初期化
	COBW3_PUT_TEXT	データの出力(改行コードを付加)
システムコマンドの実行 (参照“7.2.2.4 システムコマンドの実行 ”)	COBW3_SYSTEM	指定されたシステムコマンドの実行
Web実行環境の資源の解放 (参照“7.2.2.5 Web実行環境の資源の解放 ”)	COBW3_FREE	Webサブルーチンの獲得した資源を解放
リクエスト情報の取得 (参照“7.2.2.6 リクエ ”)	COBW3_RECEIVE_HEADER	HTTPヘッダの取得
	COBW3_GET_REQUEST_INFO	リクエストに関する情報の取得

スト情報の取得”)	COBW3_GET_AUTHORIZE	認証情報の取得
Cookieデータ操作 (参照 “7.2.2.7 Cookieデータ操作 ”)	COBW3_SET_COOKIE_XX	Cookieデータの登録
	COBW3_SET_COOKIE_NX	
	COBW3_SET_COOKIE_XN	
	COBW3_SET_COOKIE_NN	
	COBW3_DEL_COOKIE_X	登録済みのCookieデータの削除
	COBW3_DEL_COOKIE_N	
	COBW3_INIT_COOKIE	登録済みの全Cookieデータの初期化
	COBW3_GET_COOKIE_XX	Web パラメタからのCookieデータの取得
	COBW3_GET_COOKIE_NX	
	COBW3_GET_COOKIE_XN	
	COBW3_GET_COOKIE_NN	

【補足】

COBW3_XXXXX_XXやCOBW3_XXXXX_Nなどのサブルーチンの意味は次のとおりです。

サブルーチン名	説明
COBW3_XXXXX_XX	NAME/VALUEパラメタ共に英数字項目とみなします
COBW3_XXXXX_NX	NAMEパラメタを日本語項目、VALUEパラメタを英数字項目とみなします
COBW3_XXXXX_XN	NAMEパラメタを英数字項目、VALUEパラメタを日本語項目とみなします
COBW3_XXXXX_NN	NAME/VALUEパラメタ共に日本語項目とみなします
COBW3_XXXXX_X	入力パラメタを英数字項目とみなします
COBW3_XXXXX_N	入力パラメタを日本語項目とみなします

7.1.2 ファイルアップロード機能

ファイルアップロード機能では、クライアント側で作成したデータ(ファイル)を、サーバに転送することができます。

この機能は、以下のような場合に有効です。

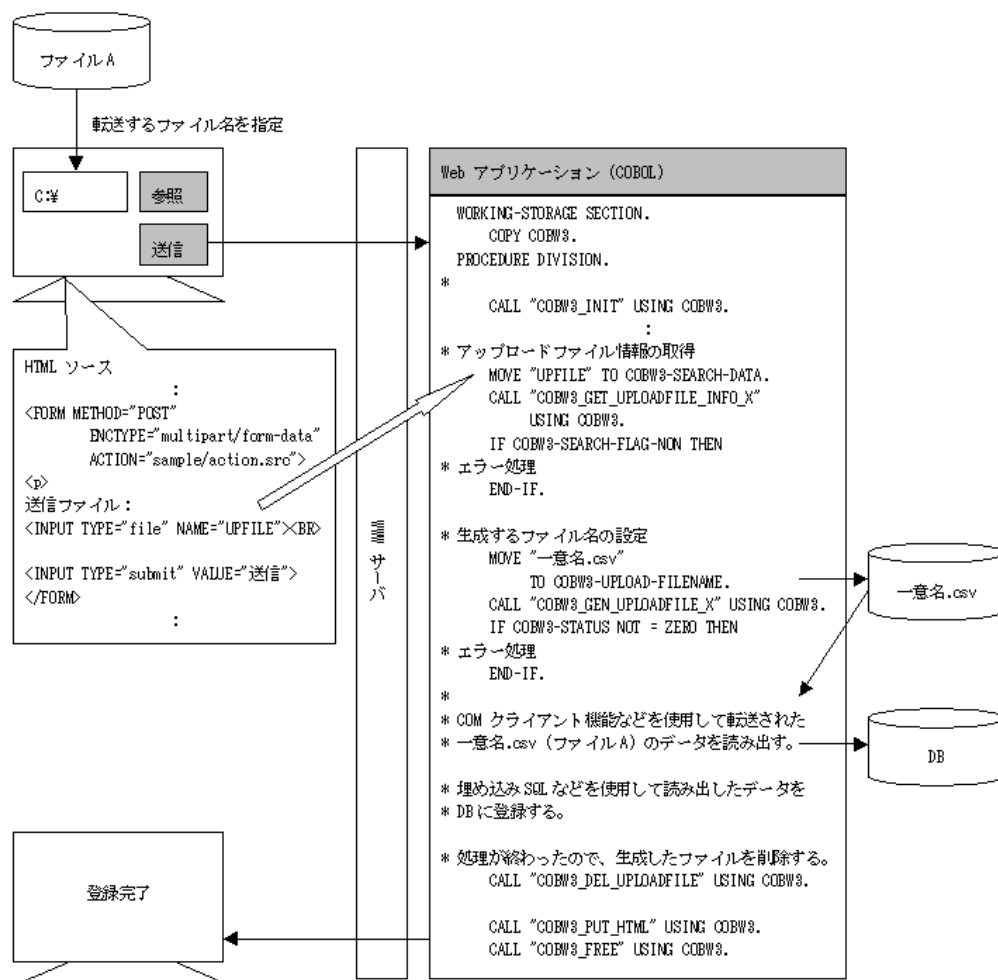
オンラインで処理するとタイムアウトになるような大量のデータエントリをする場合。

従量制課金のモバイル環境を利用する場合。

すでにクライアント側にデータエントリツールがあり、これらのツールのWeb化をするよりも、ツールが出力するデータ(ファイル)によるサーバ連携を望む場合。

従来FTPを使用していたファイル転送処理に対し、セキュリティの関係でHTTPポートしかアクセスできなくなった場合。

また、ファイルアップロード機能を使用した場合、ファイル転送とアプリケーションを関連付けることができるため、転送直後にデータのチェックおよび登録などの処理を行うことができます。ファイルアップロード機能を利用した処理の概念図を以下に示します。



Webサブルーチンでは、このような業務プログラムを簡単に作成するために、以下の機能を提供しています。

機能	サブルーチン名	用途
ファイルアップロード (参照“7.2.2.8 ファイルアップロード ”)	COBW3_GET_UPLOADFILE_INFO_X	アップロードファイルの情報の取得
	COBW3_GET_UPLOADFILE_INFO_N	
	COBW3_GEN_UPLOADFILE_X	アップロードされたファイルをサーバ上に生成
	COBW3_GEN_UPLOADFILE_N	
	COBW3_DEL_UPLOADEDFILE	アップロード済みファイルの削除

ファイルアップロードを行う場合、呼出し用ページに次の値を記述する必要があります。

タグ名	属性	値
FORM	METHOD	POST
	ENCTYPE	multipart/form-data
INPUT	TYPE	file

タグおよび属性の詳細については、“Web連携ガイド”およびHTMLについて解説している書籍やホームページなどを参照してください。

なお、上記のタグは同FORM内で他のTYPE(“file”以外)のINPUTタグと併用することができます。これらのINPUTタグに対応するデータは、ファイルアップロード機能の使用の有無にかかわらず、COBW3_GET_VALUE_XXなどのサブルーチンを使用して取得できます。



ファイルアップロードのためのINPUTタグ(TYPE="file")に指定したNAME値は、他のTYPE("file"以外)のINPUTタグでは使用できません。この場合、Webサブルーチンの動作は保証されません。

例：

```

:
<INPUT TYPE="text" NAME="FILE1">
<INPUT TYPE="file" NAME="FILE1">      指定不可
:

```

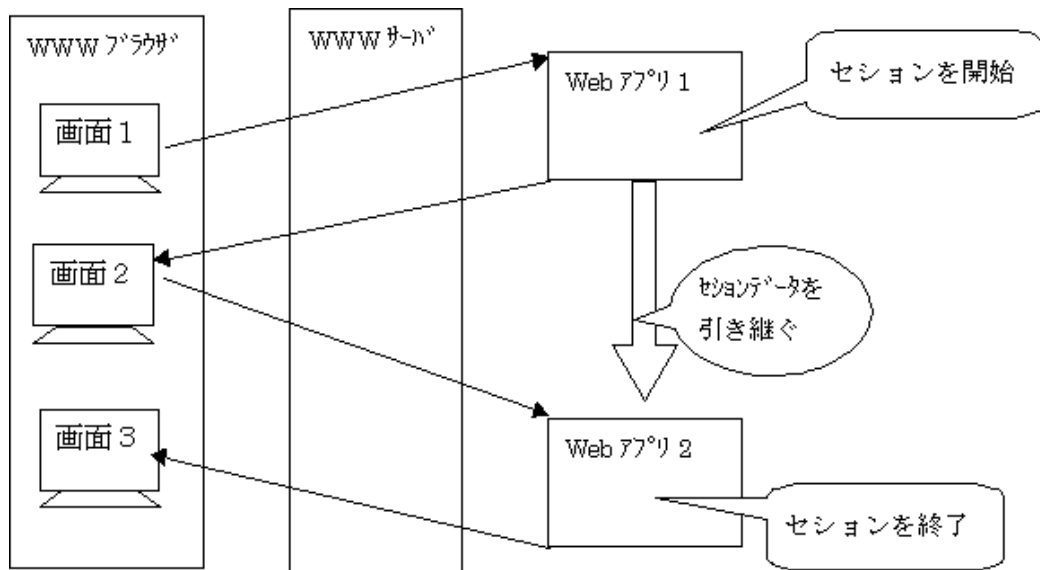
7.1.3 セッション管理機能

セッション管理機能を使用することで、特定のクライアント(WWWブラウザ)から、複数回のリクエストにまたがった業務を行うWebアプリケーションを作成できます。

機能	サブルーチン名	用途
セッション管理 (参照“7.2.2.9 セッション管理 ”)	COBW3_START_SESSION	セッションの開始
	COBW3_END_SESSION	セッションの終了
	COBW3_SET_SESSION_DATA	セッションデータの登録
	COBW3_GET_SESSION_DATA	セッションデータの取得
	COBW3_ALTER_SESSION_TIMEOUT	セッションのタイムアウト時間の変更
	COBW3_GET_SESSION_INFO	現行セッション情報の取得

【補足】 セッション管理機能は、COBOL CGIサブルーチンでは利用できません。

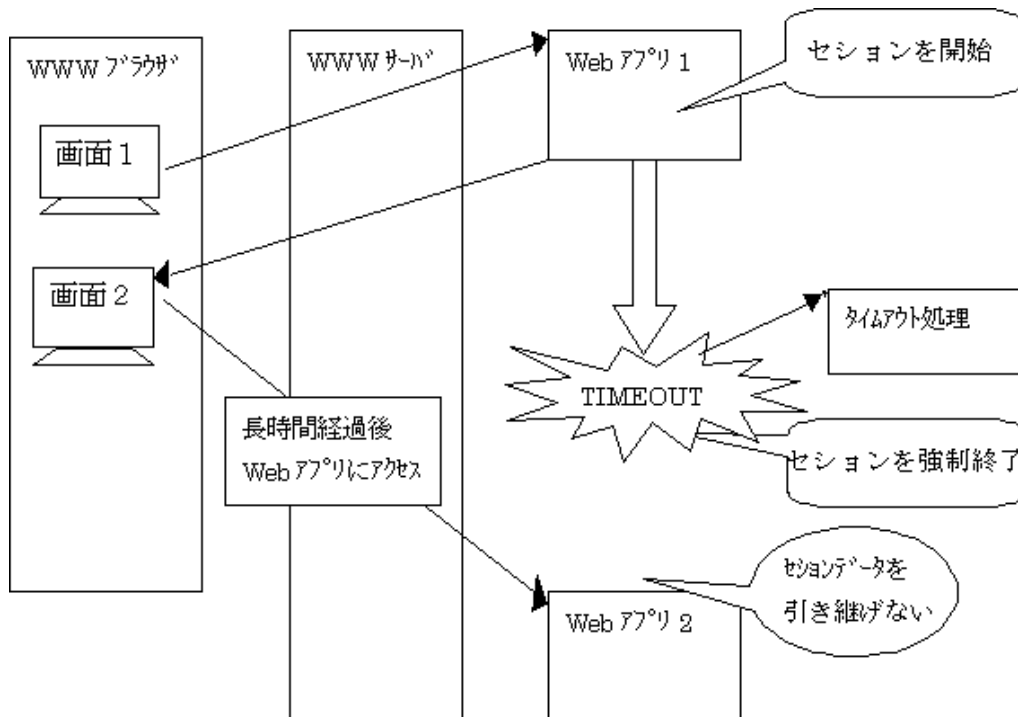
セッション管理機能を利用して、セッションを開設することにより、特定のクライアント(WWWブラウザ)との接続状態を維持することができます。また、同一セッション内では、セッションデータを利用して前のページで入力されたデータを次のページに引き継ぐことができるため、複数の画面(ページ)が遷移する業務プログラムを簡単に作成できます。



セッションの開設中に、画面操作中の長時間にわたる離席や、WWWブラウザを閉じてしまうなどの理由で業務が中断された場合、セッションのタイムアウトが発生します。

タイムアウトとは、同一セッション内でWebアプリケーションがWWWブラウザにレスポンスを返してから、WWWブラウザによって次のリクエストが発行されるまでの経過時間が規定時間に達することを言います。

タイムアウトが発生した時点で、セッション管理はセッションを強制終了します。このため、タイムアウトが発生すると、Webアプリケーションの資産はセッションが中断された状態のままになってしまうことがあります。セッション管理機能では、このような状態に陥った場合の対応手段として、タイムアウト処理の登録機構を提供しています。



注意

セッションを使用したWebアプリケーションを、SUBMITボタン (INPUTタグのTYPE属性に " SUBMIT " を指定したボタン) のダブルクリックなどの操作によって二重に起動した場合、Webアプリケーションの動作は不定となります。このため、WWWブラウザ上でJavaScriptなどを使用して二重起動を抑止してください。なお、WWWブラウザを操作する上での一般的な注意点については、“Web連携ガイド”を参照してください。

7.1.4 SAF固有機能

SAF固有機能は、NES上でSAFとしてWebアプリケーションを実行するときに、NES固有の情報の取得を可能にします。

下記のサブルーチンはSAF固有のサブルーチンです。

機能	サブルーチン名	用途
NES設定の取得 (参照 “7.2.2.10 SAF固有サブルーチン ”)	COBW3_SAF_GET_PARM_XX COBW3_SAF_GET_PARM_NX COBW3_SAF_GET_PARM_XN COBW3_SAF_GET_PARM_NN	SAFディレクトリに指定されたパラメータを取得する。
リクエスト情報の取得 (参照 “7.2.2.10 SAF固有サブルーチン ”)	COBW3_SAF_GET_BASENAME	URLに指定されたアプリケーション名を取得する。

7.2 Webサブルーチンのインタフェース

Webサブルーチンを利用する際のインタフェースについて説明します。

書き方:

Webサブルーチンを呼び出す場合の書き方について説明します。

呼出し時のデータ設定:

Webサブルーチンを呼び出す前に設定する情報(呼出しパラメタ)について説明します。

条件名が記述されているものは、SET文で値を設定してください。

条件名に下線がある場合は、その値が初期値であることを意味します。

なお、呼出し時に設定するデータの値は、Webサブルーチンを呼び出した前後で変化しません。

処理結果のデータ:

Webサブルーチンの実行により得られる処理結果について説明します。

条件名が記述されているものは、条件名条件で判定してください。

また、Webサブルーチンでの異常および警告内容はCOBW3-STATUSで、SAF固有サブルーチンでの異常および警告内容はCOBW3-SAF-STATUSでそれぞれ参照できます。

値	意味
<u>0</u>	正常終了
0 以外	エラー番号

エラー番号は、“付録B [エラー処理](#)”を参照してください。

また、ISAPIサブルーチンの実行結果はPROGRAM-STATUSで参照できます。

値	意味
<u>0</u>	正常終了
-1	異常終了



注意

各サブルーチンを呼び出す前に、そのサブルーチンの呼出し時のデータ設定で公開されたデータ以外の領域を変更するような処理が行われた場合、動作およびその領域の内容は保証されません。

また、“COBW3_INIT”を呼び出す前には、インタフェース領域のCOBW3をLOW-VALUEで初期化してください。

7.2.1 登録集

Webサブルーチンの呼出しパラメタおよび実行後の復帰パラメタのインタフェースとして登録集を提供しています。Webサブルーチンを呼び出す場合には、作業場所節に以下の登録集をCOPY文で取り込んでください。

この登録集は、COBOLをインストールしたフォルダに格納されています。

登録集名:

COBW3.cbl

書き方:

次のように指定します。

WORKING-STORAGE SECTION.

COPY COBW3.

また、SAF固有のサブルーチンを呼び出す場合には、続けてSAF固有サブルーチンとのインタフェースとなる登録集をCOPY文で取り込んでください。

この登録集は、COBOLをインストールしたフォルダに格納されています。

登録集名:

COBW3SAF.cbl

書き方:

次のように指定します。

```
WORKING-STORAGE SECTION.
```

```
COPY COBW3.
```

```
COPY COBW3SAF.      *> SAF固有サブルーチンとのインタフェース
```



注意

WebサブルーチンおよびWebサブルーチン固有の登録集を取り込むCOPY文のREPLACING指定または、このCOPY文が置き換えの対象となるようなREPLACE文の記述は行わないでください。

登録集を変更した場合、動作は保証されません。

7.2.2 Webサブルーチンの呼出し

Webサブルーチンの環境設定およびWebパラメタの獲得のため、最初に“COBW3_INIT”を呼び出す必要があります。また、Webサブルーチンの処理を終了するとき(手続きの最後)には、必ず“COBW3_FREE”を呼び出し、Webサブルーチンの資源を解放する必要があります。

その他のサブルーチンについては必要に応じてCALL文で呼び出してください。

“COBW3_INIT”を呼び出したプログラムとは異なるプログラムで、他のサブルーチンを呼び出す場合、“COBW3_INIT”を呼び出したプログラムで定義したインタフェース領域(COBW3)をプログラム間で持ち回り、その領域を使って他のサブルーチンを呼び出してください。

例

親プログラムで“COBW3_INIT”を呼び出し、子プログラムBで“COBW3_FREE”を呼び出す場合
親プログラム

```
IDENTIFICATION DIVISION.
PROGRAM-ID      "MainProc".
DATA            DIVISION.
WORKING-STORAGE SECTION.
COPY COBW3.
LINKAGE        SECTION.
01 SAFCTX      POINTER.
PROCEDURE DIVISION USING SAFCTX.
MOVE LOW-VALUE TO COBW3.
MOVE SAFCTX TO COBW3-CONTEXT.
CALL "COBW3_INIT" USING COBW3.
:
CALL "B"          USING COBW3.
*
EXIT PROGRAM.
```

 子プログラム

```

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

各サブルーチンの詳細な呼出し方法を説明します。

7.2.2.1 作業環境の設定およびWebパラメタの獲得

COBW3_INIT

Webサブルーチン作業環境の設定を行います。また、Webパラメタを獲得し、Webサブルーチンが使用する作業領域に設定します。



“COBW3_INIT”を呼び出す前にインタフェース領域のCOBW3をLOW-VALUEで初期化してください。

書き方:

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

呼出し時のデータ設定:

COBW3-CONTEXT[ISAPIサブルーチンおよびSAFサブルーチン使用の場合は必須]

IISまたはSAFディレクタとのインタフェース領域へのポインタを指定します。指定した値を変更してはなりません。

COBW3-DMODE[省略可]

Webサブルーチンのエラーメッセージの出力を指定します。

なお、COBW3-DMODEは動作コード系がUnicodeの場合には無効です。

条件名	値	意味
COBW3-DMODE-NODBG	LOW-VALUE	エラーメッセージを出力しません。
COBW3-DMODE-DBG	"1"	ブラウザにエラーメッセージを出力します。

処理結果のデータ:

なし。



“COBW3_INIT”を続けて2回以上呼び出すと、獲得したWebパラメタが初期化される場合が

あり、その後の動作は保証されません。したがって、“COBW3_INIT”の呼び出しは、WWWブラウザと接続することに他のWebサブルーチンを呼び出す前までの間に必ず一度だけ実行されるようにプログラムを作成してください。

7.2.2.2 Webパラメタの操作

COBW3_GET_VALUE_XX、COBW3_GET_VALUE_NX、COBW3_GET_VALUE_XN、COBW3_GET_VALUE_NN

“COBW3_INIT”で獲得したWebパラメタから任意の名前(NAME)の検索を行い、対応する値(VALUE)を取得します。各サブルーチンの意味は次のとおりです。

COBW3_GET_VALUE_XX

英数字文字列の名前(NAME)を検索し、対応する値(VALUE)を英数字文字列として返します。

COBW3_GET_VALUE_NX

日本語文字列の名前(NAME)を検索し、対応する値(VALUE)を英数字文字列として返します。

COBW3_GET_VALUE_XN

英数字文字列の名前(NAME)を検索し、対応する値(VALUE)を日本語文字列として返します。

COBW3_GET_VALUE_NN

日本語文字列の名前(NAME)を検索し、対応する値(VALUE)を日本語文字列として返します。

なお、取得した値(VALUE)を格納する際のパディング文字列(埋草文字)は次のとおりです。

COBW3_GET_VALUE_XX、COBW3_GET_VALUE_NXの場合：

半角空白

COBW3_GET_VALUE_XN、COBW3_GET_VALUE_NNの場合：

全角空白

書き方：

```
-----
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.
```

呼出し時のデータ設定：

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索する名前(NAME)を設定します。(名前は呼出し用ページのHTML文書のNAMEに指定した値)

COBW3_GET_VALUE_XXまたはCOBW3_GET_VALUE_XNの場合、COBW3-SEARCH-DATAに設定します。

COBW3_GET_VALUE_NXまたはCOBW3_GET_VALUE_NNの場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある名前(NAME)を検索する場合、空白を含む名前の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 1024	指定された文字列の長さで検索します。

COBW3-NUMBER[省略可]

Webパラメタに同一の名前(NAME)が複数存在する場合、検索対象の名前の出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致する名前を検索します。
-----	2 ~ 9999	指定された出現順序の名前を検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3-GET-DATA、COBW3-GET-DATA-N

検索対象の名前(NAME)に対応する値(VALUE)が設定されます。

COBW3_GET_VALUE_XXまたはCOBW3_GET_VALUE_NXの場合、COBW3-GET-DATAに設定されます。

COBW3_GET_VALUE_XNまたはCOBW3_GET_VALUE_NNの場合、COBW3-GET-DATA-Nに設定されます。

COBW3-GET-LENGTH

検索対象の名前(NAME)に対応する値(VALUE)の文字列長(バイト長)が設定されます。

COBW3_CHECK_VALUE_X、COBW3_CHECK_VALUE_N

“COBW3_INIT”で獲得したWebパラメタから任意の値(VALUE)の検索を行います。このサブルーチンは、チェックボックスのある呼出し用ページから起動された場合に、WWWブラウザでチェックされた項目を値(VALUE)で判定する場合などに使用します。各サブルーチンの意味は次のとおりです。

COBW3_CHECK_VALUE_X

英数字文字列の値(VALUE)を検索します。

COBW3_CHECK_VALUE_N

日本語文字列の値(VALUE)を検索します。

書き方:

```
-----
CALL "COBW3_CHECK_VALUE_X" USING COBW3.
-----
```

```
-----
CALL "COBW3_CHECK_VALUE_N" USING COBW3.
-----
```

呼出し時のデータ設定:

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索する値(VALUE)を設定します。

COBW3_CHECK_VALUE_Xの場合、COBW3-SEARCH-DATAに設定します。

COBW3_CHECK_VALUE_Nの場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある値(VALUE)を検索する場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。

1 ~ 1024	指定された文字列の長さで検索します。
----------	--------------------

COBW3-NUMBER[省略可]

Webパラメタに同一のVALUEが複数存在する場合、検索対象のVALUE出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致する値を検索します。
-----	2 ~ 9999	指定された出現順序の値を検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の値が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の値が存在しました。

7.2.2.3 処理結果の出力

COBW3_PUT_HEAD

指定された任意のヘッダを出力します。

結果出力用ページのHTML文書を出力する場合、Webサブルーチンがヘッダをtext/htmlとして出力するため、当サブルーチンを使用してヘッダを出力する必要はありません。HTML文書以外のファイル(プレーンテキストなど)を出力する場合または任意のヘッダをWWWサーバに出力する場合にだけ当サブルーチンを使用してください。



当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定した情報はヘッダとして無効になります。

また、COBW3-CONTENT-TYPE、COBW3-STATUS-CODEに一度省略値以外の値を設定した後、値は変えないでください。

書き方:

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

呼出し時のデータ設定:

COBW3-PUT-HEAD

ヘッダに出力する文字列を設定します。

COBW3-PUT-HEAD-LENGTH[省略可]

最後に有効な空白がある文字列をヘッダに出力する場合、空白を含む文字列の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで出力します。ただし、COBW3-PUT-HEADがすべて空白の場合、文字列長をゼロ(改行だけ)として処理します。

1 ~ 512	指定された文字列の長さで出力します。
---------	--------------------

COBW3-CONTENT-TYPE [省略可]

レスポンスデータのファイルタイプ(属性)(Content-type)を設定します。
当サブルーチンが複数回呼び出された場合、最初の指定が有効になります。

条件名	値	意味
-----	LOW-VALUE	HTML文書を出力します。
COBW3-CONTENT-TYPE-NON	HIGH-VALUE	Content-typeを出力しません。
COBW3-CONTENT-TYPE-HTML	"text/html"	HTML文書を出力します。
COBW3-CONTENT-TYPE-TEXT	"text/plain"	テキストファイルを出力します。
-----	任意の文字列	任意の文字列をヘッダのContent-typeに指定します。

COBW3-STATUS-CODE [省略可]

ステータスコード(Status-code)を設定します。
当サブルーチンが複数回呼び出された場合、最初の指定が有効になります。

条件名	値	意味
-----	LOW-VALUE	正常終了のコード "200" をStatus-codeに出力します。
COBW3-STATUS-CODE-NON	HIGH-VALUE	Status-code を出力しません。
COBW3-STATUS-CODE-200	"200"	正常終了のコード "200" をStatus-codeに出力します。
-----	任意のコード	任意のStatus-codeを出力します。



COBW3-CONTENT-TYPE-NONおよびCOBW3-STATUS-CODE-NONは、極力指定しないでください。
CGIなどでは、パッチファイルからContent-typeなどを指定することが可能なため、-NONの指定は有効です。しかし、ISAまたはSAFのようにDLLアプリケーションでは外部からの指定ができません。また、ISAおよびSAFで、-NONを指定してContent-typeやStatus-codeを指定しなかった場合の動作は不定です。

COBW3_PUT_HTML

Webアプリケーション実行後に、結果出力用ページ(HTML文書)をWWWブラウザに出力します。このとき、出力する結果出力用ページに“//COBOL//”または“//COBOL_REPEAT//”で囲まれた変換名がある場合に、事前にCOBW3_SET_CNV_XX、COBW3_SET_REPEAT_XXなどで登録された変換文字列に変換します。

書き方:

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

呼出し時のデータ設定:

COBW3-HTML-FILENAME

結果出力用ページのファイル名を設定します。

処理結果のデータ:

なし。

結果出力用ページ内の変換名の指定方法:

結果出力用ページ内の文字列で、Webアプリケーションの実行時に変換して出力したい箇所に、前後を“//COBOL//”で囲んだ文字列(変換名)を記述します。

```
... //COBOL//変換名//COBOL// ...
```



注意

以下のように“//COBOL//”は複数行にまたがって指定することはできません。

```
... //COBOL//変換名
      変換名の続き//COBOL// ...
```

変換名の指定方法によって、どのような振る舞いをするかを以下に示します。

変換名の指定方法	処理内容
This is a //COBOL//NAME//COBOL// test.	変換名"NAME"が登録されていれば、 "//COBOL//NAME//COBOL//"を登録されている変換文字列に置き換えます。
This is a //COBOL// test.	変換名の指定に誤りがあるため、エラーとなります。
This is a //COBOL test.	そのまま出力されます。
This is a //COBOL// NAME//COBOL// test.	変換名の指定に誤りがあるため、エラーとなります。
This is a //COBOL//NAME//COBOL	変換名の指定に誤りがあるため、エラーとなります。
This is a //COBOL////COBOL//	変換名の指定に誤りがあるため、エラーとなります。
This is a //cobol//NAME//cobol//.	そのまま出力されます。

使用例:

[a.htm (結果出力用ページ)]

```
-----
:
<BODY>
  <BR>
  名前: //COBOL//GET-NAME//COBOL//
  <BR>
  :
```

[COBOLプログラム]

```
-----
:
* 名前の設定
MOVE "GET-NAME" TO COBW3-CNV-NAME.
MOVE NC"富士通太郎" TO COBW3-CNV-VALUE-N.
```

```

* 結果出力用ページ内の変換データの登録
CALL "COBW3_SET_CNV_XN" USING COBW3.
:
* 結果出力用ページのファイル名の設定
MOVE "a.htm"          TO COBW3-HTML-FILENAME.
* 結果出力用ページの出力
CALL "COBW3_PUT_HTML"  USING COBW3.
:

```

[WWWブラウザ上の出力結果]

```

:
名前: 富士通太郎
:

```

結果出力用ページ内の繰返し範囲の指定方法:

結果出力用ページ内でWebアプリケーションの実行時に繰返し出力したい箇所に "`//COBOL_REPEAT_START//`"、"`//COBOL_REPEAT_END//`" を使用して繰返し範囲を記述します。繰返し回数は、変換対象になる繰返し変換文字列の登録数と同じです。

繰返し範囲開始

データを繰り返す範囲の開始位置に "`//COBOL_REPEAT_START//`" を記述します。

繰返し項目

繰返し出力を行いたいデータを記述します。データの中でWebアプリケーションの実行時に変換して出力したい箇所には、変換名を記述します。

繰返しに応じて異なる変換文字列を出力する場合は、前後を "`//COBOL_REPEAT//`" で囲んだ文字列(変換名)を記述します。繰返し時に、固定の変換文字列を出力する場合は、前後を "`//COBOL//`" で囲んだ文字列(変換名)を記述します。

繰返し範囲終了

データを繰り返す範囲の終了位置に "`//COBOL_REPEAT_END//`" を記述します。

```

//COBOL_REPEAT_START//
・ ・ //COBOL_REPEAT//変換名//COBOL_REPEAT// ・ ・
・ ・ //COBOL//変換名//COBOL// ・ ・
//COBOL_REPEAT_END//

```



"`//COBOL_REPEAT//`" で囲んだ変換名に対応する変換文字列の登録は、必ず1回以上行ってください。

繰返し項目に "`//COBOL_REPEAT//`" で囲んだ変換名が記述されない場合、繰返し回数は1回になります。

"`//COBOL//`" および "`//COBOL_REPEAT//`" は複数行にまたがって指定することはできません。

繰返し範囲の開始/終了を指示する "`//COBOL_REPEAT_START//`" または "`//COBOL_REPEAT_END//`" を含む行は、WWWブラウザに出力するときに削除されます。したがって、表示に関わるタグ等を同じ行に記述しないでください。

繰返し範囲内に記述した繰返し変換文字列は、すべて同じ登録数にしてください。登録数が異なる場合、繰返し数に対して登録データが足りない変換名が存在するため、エラーになります。

繰返し範囲開始を示す記述がされていない場合、繰返し出力は行われずに、“//COBOL//”以外の記述はデータとみなして出力されます。

繰返し範囲終了を示す記述がされていない場合、結果出力用ページの末尾まで出力後、エラーになります。

変換名の指定方法による振る舞いは、結果出力用ページ内の変換名の指定方法と同じです。

使用例:

[b.htm (結果出力用ページ)]

```

:
<TABLE>
  <TR>
    <TD>名前</TD>
    <TD>年齢</TD>
    <TD>会社</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プログラム]

```

:
PERFORM UNTIL END-FLAG="END"
* 名前の設定
  MOVE "GET-NAME" TO COBW3-CNV-NAME
  MOVE 名前 TO COBW3-CNV-VALUE-N
* 結果出力用ページ内の繰返し変換データの登録
  CALL "COBW3_SET_REPEAT_XN" USING COBW3
* 年齢の設定
  MOVE "GET-AGE" TO COBW3-CNV-NAME
  MOVE 年齢 TO COBW3-CNV-VALUE
* 結果出力用ページ内の繰返し変換データの登録
  CALL "COBW3_SET_REPEAT_XX" USING COBW3
:
END-PERFORM.
:
* 会社名の設定
  MOVE "GET-OFFICE" TO COBW3-CNV-NAME.
  MOVE NC"富士通" TO COBW3-CNV-VALUE-N.
* 結果出力用ページ内の変換データの登録
  CALL "COBW3_SET_CNV_XN" USING COBW3.
:
* 結果出力用ページのファイル名の設定

```

```

MOVE "b.htm"          TO COBW3-HTML-FILENAME.
* 結果出力用ページの出力
CALL "COBW3_PUT_HTML" USING COBW3.
      :

```

[WWWブラウザ上の出力結果]

:		
名前	年齢	会社
鈴木	35	富士通
佐藤	26	富士通
田中	23	富士通
:		

COBW3_SET_CNV_XX、COBW3_SET_CNV_NX、COBW3_SET_CNV_XN、COBW3_SET_CNV_NN

“COBW3_PUT_HTML”で出力する結果出力用ページ内に指定された“//COBOL//”で囲まれた変換名に対する変換文字列を登録します。

登録された情報はCOBW3_PUT_HTML実行時に参照され、登録された変換データに従って、結果出力用ページ内の変換が行われます。

各サブルーチンの意味は次のとおりです。

COBW3_SET_CNV_XX

英数字文字列の変換名に対応する変換文字列を英数字文字列として登録します。

COBW3_SET_CNV_NX

日本語文字列の変換名に対応する変換文字列を英数字文字列として登録します。

COBW3_SET_CNV_XN

英数字文字列の変換名に対応する変換文字列を日本語文字列として登録します。

COBW3_SET_CNV_NN

日本語文字列の変換名に対応する変換文字列を日本語文字列として登録します。



結果出力用ページ内に複数個の異なる変換名が指定されている場合は、変換名ごとに“COBW3_SET_CNV_XX”などと呼び出して変換文字列を登録してください。

書き方:

```

-----
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.
```

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

変換対象となる変換名を設定します。

COBW3_SET_CNV_XXまたはCOBW3_SET_CNV_XNの場合、COBW3-CNV-NAMEに設定します。

COBW3_SET_CNV_NXまたはCOBW3_SET_CNV_NNの場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名を指定する場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。ただし、COBW3-CNV-NAMEまたはCOBW3-CNV-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 30	指定された文字列の長さで検索します。

COBW3-CNV-VALUE、COBW3-CNV-VALUE-N

変換結果(変換文字列)を設定します。

COBW3_SET_CNV_XXまたはCOBW3_SET_CNV_NXの場合、COBW3-CNV-VALUEに設定します。

COBW3_SET_CNV_XNまたはCOBW3_SET_CNV_NNの場合、COBW3-CNV-VALUE-Nに設定します。

COBW3-CNV-VALUE-LENGTH[省略可]

最後に有効な空白がある変換文字列を指定する場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-CNV-VALUE またはCOBW3-CNV-VALUE-N がすべて空白の場合文字列長をゼロとして処理します。
1 ~ 1024	指定された文字列の長さで登録します。

COBW3-CNV-MODE[省略可]

変換種別を設定します。

条件名	値	意味
COBW3-CNV-MODE-ADDREP	LOW-VALUE	指定した変換名に対する変換文字列が登録されていない場合、変換データを追加します。指定した変換名に対する変換文字列が登録されている場合は、変換データを置き換えます。
COBW3-CNV-MODE-REPLACE	"1"	変換データの置き換えを行います。指定した変換名に対する変換文字列が登録されていない場合、エラーを出力して変換を行いません。
COBW3-CNV-MODE-ADD	"2"	変換データの追加を行います。指定した変換名に対する変換文字列が登録さ

		れていない場合、エラーを出力して追加を行いません。
--	--	---------------------------

処理結果のデータ:

なし。

COBW3_DEL_CNV_X、COBW3_DEL_CNV_N

COBW3_SET_CNV_XXなどで登録した変換データの削除を行います。各サブルーチンの意味は次のとおりです。

COBW3_DEL_CNV_X

英数字文字列の変換名に対して登録した変換データを削除します。

COBW3_DEL_CNV_N

日本語文字列の変換名に対して登録した変換データを削除します。

書き方:

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

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

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

削除対象となる変換名を設定します。

COBW3_DEL_CNV_Xの場合、COBW3-CNV-NAMEに設定します。

COBW3_DEL_CNV_Nの場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名を指定する場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。ただし、COBW3-CNV-NAMEまたはCOBW3-CNV-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1～30	指定された文字列の長さで検索します。

処理結果のデータ:

なし。

COBW3_INIT_CNV

COBW3_SET_CNV_XXなどで登録した変換データをすべて初期化します。

書き方:

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

呼出し時のデータ設定:

なし。

処理結果のデータ:

なし。

COBW3_SET_REPEAT_XX 、 COBW3_SET_REPEAT_NX 、 COBW3_SET_REPEAT_XN 、
COBW3_SET_REPEAT_NN

“ COBW3_PUT_HTML ” で出力する結果出力用ページ内に指定された “ //COBOL_REPEAT// ” で囲まれた変換名に対する繰返し変換データの登録を行います。

登録された情報はCOBW3_PUT_HTML実行時に参照され、同一の変換名に対して登録された繰返し変換文字列の登録数に応じて、結果出力用ページ内の繰返し範囲に指定された変換名が変換されます。

各サブルーチンの意味は次のとおりです。

COBW3_SET_REPEAT_XX

英数字文字列の変換名に対応する変換文字列を英数字文字列として登録します。

COBW3_SET_REPEAT_NX

日本語文字列の変換名に対応する変換文字列を英数字文字列として登録します。

COBW3_SET_REPEAT_XN

英数字文字列の変換名に対応する変換文字列を日本語文字列として登録します。

COBW3_SET_REPEAT_NN

日本語文字列の変換名に対応する変換文字列を日本語文字列として登録します。

**注意**

結果出力用ページ内に複数個の異なる変換名が指定されている場合は、変換名ごとに “ COBW3_SET_REPEAT_XX ” などと呼び出して繰返し変換文字列を登録してください。

繰返し変換時に変換される変換文字列は、 “ COBW3_SET_REPEAT_XX ” などと呼び出して登録した順序に従います。

同じ繰返し範囲に含まれる変換名に対する繰返し変換文字列は、登録数が同じになるように登録してください。

書き方:

```
-----  
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.  
-----
```

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

変換対象となる変換名を設定します。

COBW3_SET_REPEAT_XXまたはCOBW3_SET_REPEAT_XNの場合、COBW3-CNV-NAMEに設定します。

COBW3_SET_REPEAT_NXまたはCOBW3_SET_REPEAT_NNの場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
---	----

0	空白でない最後の文字までの長さで検索します。ただし、COBW3-CNV-NAMEまたはCOBW3-CNV-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 30	指定された文字列の長さで検索します。

COBW3-CNV-VALUE、COBW3-CNV-VALUE-N

変換結果(変換文字列)を設定します。

COBW3_SET_REPEAT_XXまたはCOBW3_SET_REPEAT_NXの場合、COBW3-CNV-VALUEに設定します。
COBW3_SET_REPEAT_XNまたはCOBW3_SET_REPEAT_NNの場合、COBW3-CNV-VALUE-Nに設定します。

COBW3-CNV-VALUE-LENGTH[省略可]

最後に有効な空白がある変換文字列の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白でない最後の文字までの長さで登録します。ただし、COBW3-CNV-VALUE またはCOBW3-CNV-VALUE-N がすべて空白の場合文字列長をゼロとして処理します。
1 ~ 1024	指定された文字列の長さで登録します。

【補足】 変換結果に空白を指定したい場合には、変換文字列に空白、変換文字列長にゼロを指定して、繰返し変換データを登録してください。

処理結果のデータ:

なし。

COBW3_DEL_REPEAT_X、COBW3_DEL_REPEAT_N

COBW3_SET_REPEAT_XXなどで登録した繰返し変換データの削除を行います。各サブルーチンの意味は次のとおりです。

COBW3_DEL_REPEAT_X

英数字文字列の変換名に対して登録した繰返し変換データを削除します。

COBW3_DEL_REPEAT_N

日本語文字列の変換名に対して登録した繰返し変換データを削除します。



注意

指定された変換名に対して複数の繰返し変換データが登録されている場合は、登録されているすべてのデータを削除します。

書き方:

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

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

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

削除対象となる変換名を設定します。

COBW3_DEL_REPEAT_Xの場合、COBW3-CNV-NAMEに設定します。

COBW3_DEL_REPEAT_Nの場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白でない最後の文字までの長さで検索します。ただし、COBW3-CNV-NAMEまたはCOBW3-CNV-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 30	指定された文字列の長さで検索します。

処理結果のデータ:

なし。

COBW3_INIT_REPEAT

COBW3_SET_REPEAT_XXなどで登録した繰返し変換データをすべて初期化します。

書き方:

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

呼出し時のデータ設定:

なし。

処理結果のデータ:

なし。

COBW3_PUT_TEXT

指定した文字列に改行コードを加えたデータを、クライアントのWWWブラウザに出力します。

書き方:

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

呼出し時のデータ設定:

COBW3-PUT-STRING

出力する文字列を設定します。

COBW3-PUT-STRING-LENGTH[省略可]

最後に有効な空白がある文字列を出力する場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで出力します。ただし、COBW3-PUT-STRINGがすべて空白の場合、文字列長をゼロ(改行だけ)として処理します。
1 ~ 1024	指定された文字列の長さで出力します。

処理結果のデータ:

なし。

文字列データ(改行コード含む)をWWWブラウザに出力する方法:

WWWブラウザに “ <CENTER>ご協力ありがとうございました。</CENTER> ” を出力する場合の記述例

を以下に示します。

使用例:

[COBOLプログラム]

```

:
000360* 出力する文字列のセット
000370  MOVE "<CENTER>ご協力ありがとうございました。</CENTER>"
000380                                     TO  COBW3-PUT-STRING.
000390* 文字列データの出力
000400  CALL "COBW3_PUT_TEXT"  USING COBW3.
:

```

[WWWブラウザ上の出力結果]

```

:
ご協力ありがとうございました。
:

```

7.2.2.4 システムコマンドの実行

COBW3_SYSTEM

Webアプリケーションから任意のシステムコマンドを実行します。

書き方:

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

呼出し時のデータ設定:

COBW3-SYSTEMINFO

実行するシステムコマンドおよびそのパラメタを設定します。

処理結果のデータ:

なし。



注意

第三者が不正にシステムに侵入し、システムを破壊するなどの危険があるため、WWWブラウザで入力されたコマンド文字をCOBW3_SYSTEMでそのまま実行するなど、直接クライアント(WWWブラウザ)からコマンドが実行できるような設計は行わないでください。

7.2.2.5 Web実行環境の資源の解放

COBW3_FREE

Webサブルーチンが獲得した資源を解放します。このため、当サブルーチンはWebアプリケーションの各接続の最後で必ず呼び出してください。

書き方:

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

呼出し時のデータ設定:

なし。

処理結果のデータ:

なし。

7.2.2.6 リクエスト情報の取得

COBW3_RECEIVE_HEADER

リクエストに対するHTTPヘッダを取得します。

書き方:

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

呼出し時のデータ設定:

COBW3-HEADER-NAME

取得するヘッダ名を指定します。

COBW3-HEADER-NAME-LENGTH[省略可]

最後に有効な空白があるヘッダ名を検索する場合、空白を含むヘッダ名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 64	指定された文字列の長さで検索します。

【補足】 通常、末尾に空白を持つヘッダ名はないので、0を指定してください。

処理結果のデータ:

COBW3-HEADER-VALUE

リクエストのヘッダ名に対応するヘッダ値が設定されます。

COBW3-HEADER-VALUE-LENGTH

リクエストのヘッダ名に対応するヘッダ値の文字列長(バイト長)が設定されます。

COBW3_GET_REQUEST_INFO

リクエストに関する各種情報を取得します。

書き方:

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

呼出し時のデータ設定:

COBW3-REQUEST-INFO-TYPE

取得する情報の種類を指定します。

条件名	値	意味
COBW3-URI	LOW-VALUE	リクエストURI
COBW3-URL	"1"	リクエストURL
COBW3-SERVER-TYPE	"2"	サーバの種類

(COBW3-LISTENERTYPE)		
COBW3-VIRTUALPATH	"3"	リクエストの仮想パス名
COBW3-PHYSICALPATH	"4"	リクエストの仮想パスに対応する物理パス名
COBW3-QUERYSTRING	"5"	Webパラメタ
COBW3-LANGUAGE	"6"	WWWブラウザが受け取ることのできる言語のリスト
COBW3-ENCODING	"7"	WWWブラウザが受け取ることのできるエンコーディングのリスト
COBW3-REQMIMETYPE	"8"	リクエストのMIMEタイプ
COBW3-REQUEST-METHOD	"9"	リクエストメソッド
COBW3-PATH-INFO	"A"	WWWブラウザから送信される拡張パス名
COBW3-PATH-TRANSLATED	"B"	拡張パスに対応する物理パス名
COBW3-REMOTE-ADDR	"C"	クライアントのIPアドレス
COBW3-REMOTE-HOST	"D"	クライアントのホスト名
COBW3-GATEWAY-INTERFACE	"E"	サポートCGIのリビジョン
COBW3-SERVER-NAME	"F"	サーバのホスト名
COBW3-SERVER-PORT	"G"	サーバの接続ポート番号
COBW3-SERVER-PROTOCOL	"H"	サーバの使用しているプロトコル
COBW3-SERVER-PORT-SECURE	"I"	セキュア通信が行われている場合は"1"、それ以外は"0"を返す

**注意****CGIサブルーチンの場合**

- リクエストのMIMEタイプは取得できません。
- クライアントのホスト名は、IPアドレスが返却される場合があります。
- セキュア通信情報の取得の可否は、WWWサーバによって異なります。

ISAPIサブルーチンの場合

リクエストのMIMEタイプは取得できません。

SAFサブルーチンの場合

- サーバの種類は取得できません。
- 拡張パスが指定されている場合、リクエストURI、リクエストURL、リクエストのMIMEタイプ、リクエストの仮想パス名およびリクエストの仮想パスに対応する物理パス名が正しく取得できないことがあります。

処理結果のデータ:

COBW3-REQUEST-INFO

取得したリクエスト情報が設定されます。

COBW3-REQUEST-INFO-LENGTH

取得したリクエスト情報の文字列長(バイト長)が設定されます。

COBW3_GET_AUTHORIZ

WWWブラウザからの認証情報を取得します。認証要求に対して渡されたユーザID、パスワードおよびクライアントのIPアドレスを取得します。

書き方:

```
-----
CALL "COBW3_GET_AUTHORIZ" USING COBW3.
-----
```

呼出し時のデータ設定:

なし

処理結果のデータ:

COBW3-USERID

取得したユーザIDが設定されます。

COBW3-USERID-LENGTH

ユーザIDの文字列長(バイト長)が設定されます。

COBW3-PASSWORD

取得したユーザパスワードが設定されます。

COBW3-PASSWORD-LENGTH

ユーザパスワードの文字列長(バイト長)が設定されます。

COBW3-IP-ADDRESS

取得したクライアントのIPアドレスが設定されます。

COBW3-IP-ADDRESS-LENGTH

IPアドレスの文字列長(バイト長)が設定されます。

COBW3-AUTH-TYPE

取得した認証タイプが設定されます。

COBW3-AUTH-TYPE-LENGTH

認証タイプの文字列長(バイト長)が設定されます。

**CGIサブルーチンおよびISAPIサブルーチンの場合**

基本認証ではすべての情報を取得することができ、認証タイプには“Basic”が設定されます。匿名認証の場合、IPアドレス以外の情報は取得できません。Microsoft暗号化認証では、ユーザパスワード以外の情報すべてが取得可能で、認証タイプには“NTLM”が設定されます。

SAFサブルーチンの場合

匿名認証の場合、IPアドレス以外の情報は取得できません。

基本認証ではすべての情報を取得することができ、認証タイプには“basic”が設定されます。

7.2.2.7 Cookieデータの操作

COBW3_SET_COOKIE_XX 、 COBW3_SET_COOKIE_NX 、 COBW3_SET_COOKIE_XN 、
COBW3_SET_COOKIE_NN

WWWブラウザへ送信するCookieデータの登録を行います。登録されたCookieデータはヘッダ出力時に使用されます。

各サブルーチンの意味は次のとおりです。

COBW3_SET_COOKIE_XX

英数字文字列のCookie名と、対応するCookie値を英数字文字列として登録します。

COBW3_SET_COOKIE_NX

日本語文字列のCookie名と、対応するCookie値を英数字文字列として登録します。

COBW3_SET_COOKIE_XN

英数字文字列のCookie名と、対応するCookie値を日本語文字列として登録します。

COBW3_SET_COOKIE_NN

日本語文字列のCookie名と、対応するCookie値を日本語文字列として登録します。



当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください

い。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定した情報は無効になります。

WWWブラウザへ送信されるCookieの文字コードは、COBOLの動作コード系に関係なくシフトJISになります。ただし、CookieはCOBOL以外のアプリケーションでも送受信されることおよび、主としてキー情報などを送受信するのに使用されることから、ASCIIの範囲で設定することをおすすめします。

書き方:

```
-----
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.
-----
```

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

登録するCookie名を設定します。

COBW3_SET_COOKIE_XXまたはCOBW3_SET_COOKIE_XNの場合、COBW3-COOKIE-NAMEに設定します。

COBW3_SET_COOKIE_NXまたはCOBW3_SET_COOKIE_NNの場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名を指定する場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-COOKIE-NAMEまたはCOBW3-COOKIE-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 64	指定された文字列の長さで登録します。

COBW3-COOKIE-VALUE、COBW3-COOKIE-VALUE-N

Cookie値を設定します。

COBW3_SET_COOKIE_XXまたはCOBW3_SET_COOKIE_NXの場合、COBW3-COOKIE-VALUEに設定します。

COBW3_SET_COOKIE_XNまたはCOBW3_SET_COOKIE_NNの場合、COBW3-COOKIE-VALUE-Nに設定します。

COBW3-COOKIE-VALUE-LENGTH[省略可]

最後に有効な空白があるCookie値を指定する場合、空白を含むCookie値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-COOKIE-VALUEまたはCOBW3-COOKIE-VALUE-Nがすべて空白の

	場合、文字列長をゼロとして処理します。
1 ~ 1024	指定された文字列の長さで登録します。

COBW3-COOKIE-EXPIRES[省略可]

Cookieの有効期限を指定する場合に使用します。この項目は、次の項目(年、月、日、時、分、秒)を持ちます。有効期限の指定はGMT(グリニッジ標準時刻)で行います。

COBW3-COOKIE-EXPIRES-YEAR(年)(1582 ~ 9999)

COBW3-COOKIE-EXPIRES-MONTH(月)(01 ~ 12)

COBW3-COOKIE-EXPIRES-DAY(日)(01 ~ 31、ただし、末日は月によって異なります。)

COBW3-COOKIE-EXPIRES-HOUR(時)(00 ~ 23)

COBW3-COOKIE-EXPIRES-MIN(分)(00 ~ 59)

COBW3-COOKIE-EXPIRES-SEC(秒)(00 ~ 59)



注意

有効期限が省略された場合、そのCookieデータはWWWブラウザ終了時に削除されます。また、過去の時刻を指定することで、現在WWWブラウザが持っているCookieデータを削除できます。

JST(日本標準時)とGMTの時差は9時間です。したがって、JSTから9を引いた値がGMTになります。

有効期限の指定には1582年10月15日以降の日付を使用してください。この日より前の日付を指定すると曜日の自動算出ができないため、有効期限が無効になります。

COBW3-COOKIE-DOMAIN[省略可]

Cookieが有効となるドメインを指定します。

COBW3-COOKIE-PATH[省略可]

Cookieが有効となる仮想パスを指定します。

COBW3-COOKIE-SECURE[省略可]

セキュリティを指定します。セキュリティが有効な場合、SSLを使用している場合にだけWWWブラウザからCookieデータが送信されます。

条件名	値	意味
COBW3-COOKIE-SECURE-OFF	LOW-VALUE	セキュリティを無効にします。
COBW3-COOKIE-SECURE-ON	"1"	セキュリティを有効にします。

COBW3-COOKIE-MODE[省略可]

Cookieデータの登録種別を設定します。

条件名	値	意味
COBW3-COOKIE-MODE-ADDREP	LOW-VALUE	指定したCookie名が登録されていない場合、Cookieデータを追加します。
COBW3-COOKIE-MODE-REPLACE	"1"	Cookieデータの置き換えを行います。指定したCookie名が登録されていない場合、エラーを出力して置き換えを行いません。
COBW3-COOKIE-MODE-ADD	"2"	Cookieデータの追加を行います。指定したCookie名が登録されている場合、エラーを出力して追加を行いません。

処理結果のデータ:

なし。

COBW3_DEL_COOKIE_X、COBW3_DEL_COOKIE_N

COBW3_SET_COOKIE_XXなどで登録したCookieデータを削除します。

各サブルーチンの意味は次のとおりです。

COBW3_DEL_COOKIE_X

英数字文字列のCookie名に対応するCookieデータを削除します。

COBW3_DEL_COOKIE_N

日本語文字列のCookie名に対応するCookieデータを削除します。

**注意**

当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定した情報は無効になります。

書き方:

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

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

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

削除するCookie名を設定します。

COBW3_DEL_COOKIE_Xの場合、COBW3-COOKIE-NAMEに設定します。

COBW3_DEL_COOKIE_Nの場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名を指定する場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで削除します。ただし、COBW3-COOKIE-NAMEまたはCOBW3-COOKIE-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1～64	指定された文字列の長さで削除します。

処理結果のデータ:

なし。

COBW3_INIT_COOKIE

COBW3_SET_COOKIE_XXなどで登録したCookieデータをすべて初期化します。

**注意**

当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定した情報は無効になります。

書き方:

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

呼出し時のデータ設定:

- ・ COBW3-COOKIE-INIT-MODE[省略可]
Cookieデータの初期化種別を設定します。

条件名	値	意味
COBW3-COOKIE-INIT-MODE-NORMAL	LOW-VALUE	すべてのCookieデータを初期化します。
COBW3-COOKIE-INIT-MODE-REQUEST	"1"	リクエストデータのCookieデータを初期値として初期化します。

処理結果のデータ:

なし。

COBW3_GET_COOKIE_XX 、 **COBW3_GET_COOKIE_NX** 、 **COBW3_GET_COOKIE_XN** 、
COBW3_GET_COOKIE_NN

“ COBW3_INIT ” で獲得したリクエストCookieデータから任意のCookie名の検索を行います。また、Cookie名に対応するCookie値を取得します。

各サブルーチンの意味は次のとおりです。

COBW3_GET_COOKIE_XX

英数字文字列のCookie名を検索し、対応するCookie値を英数字文字列として返します。

COBW3_GET_COOKIE_NX

日本語文字列のCookie名を検索し、対応するCookie値を英数字文字列として返します。

COBW3_GET_COOKIE_XN

英数字文字列のCookie名を検索し、対応するCookie値を日本語文字列として返します。

COBW3_GET_COOKIE_NN

日本語文字列のCookie名を検索し、対応するCookie値を日本語文字列として返します。

なお、取得したCookie値を格納する際のパディング文字列(埋草文字)は次のとおりです。

COBW3_GET_COOKIE_XX、**COBW3_GET_COOKIE_NX**の場合:

半角空白

COBW3_GET_COOKIE_XN、**COBW3_GET_COOKIE_NN**の場合:

全角空白

書き方:

```
-----
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.
-----
```

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

検索するCookie名を設定します。

COBW3_GET_COOKIE_XXまたはCOBW3_GET_COOKIE_XNの場合、COBW3-COOKIE-NAMEに設定します。

COBW3_GET_COOKIE_NXまたはCOBW3_GET_COOKIE_NNの場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名を検索する場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。ただし、COBW3-COOKIE-NAMEまたはCOBW3-COOKIE-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1～64	指定された文字列の長さで検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3-COOKIE-VALUE、COBW3-COOKIE-VALUE-N

検索対象のCookie名に対応するCookie値が設定されます。

COBW3_GET_COOKIE_XXまたはCOBW3_GET_COOKIE_NXの場合、COBW3-COOKIE-VALUEに設定されます。

COBW3_GET_COOKIE_XNまたはCOBW3_GET_COOKIE_NNの場合、COBW3-COOKIE-VALUE-Nに設定されます。

COBW3-COOKIE-VALUE-LENGTH

検索対象のCookie名に対応するCookie値の文字列長(バイト長)が設定されます。

7.2.2.8 ファイルアップロード**COBW3_GET_UPLOADFILE_INFO_X、COBW3_GET_UPLOADFILE_INFO_N**

アップロードされたファイルに関する情報を取得します。各サブルーチンの意味は次のとおりです。

COBW3_GET_UPLOADFILE_INFO_X

英数字文字列の名前(NAME)を検索し、対応するアップロードされたファイルの情報を取得します。

COBW3_GET_UPLOADFILE_INFO_N

日本語文字列の名前(NAME)を検索し、対応するアップロードされたファイルの情報を取得します。

書き方:

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

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

呼出し時のデータ設定:

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索するアップロードファイルの名前(NAME)を設定します(名前はファイル名ではなく、呼出し用ページのHTML文書のNAMEに指定した値)。

COBW3_GET_UPLOADFILE_INFO_Xの場合、COBW3-SEARCH-DATAに設定します。

COBW3_GET_UPLOADFILE_INFO_Nの場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある名前の場合、空白を含む名前の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 1024	指定された文字列の長さで検索します。

COBW3-NUMBER[省略可]

同一の名前(NAME)が複数存在する場合、検索対象の名前の出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致する名前を検索します。
-----	2 ~ 9999	指定された出現順序の名前を検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。または、ファイルのサイズが0です。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3-UPLD-CL-FILE-PATH

検索対象の名前(NAME)に対応するアップロードファイルのクライアントでのパス名が設定されます。

COBW3-UPLD-CL-FILE-PATH-LENGTH

検索対象の名前(NAME)に対応するアップロードファイルのクライアントでのパス名の文字列長(バイト長)が設定されます。

COBW3-UPLD-CL-FILE-NAME

検索対象の名前(NAME)に対応するアップロードファイルのクライアントでのファイル名が設定されます。

COBW3-UPLD-CL-FILE-NAME-LENGTH

検索対象の名前(NAME)に対応するアップロードファイルのクライアントでのファイル名の文字列長(バイト長)が設定されます。

COBW3-UPLD-CONTENT-TYPE

検索対象の名前(NAME)に対応するアップロードファイルのコンテンツタイプが文字列で設定されます。

COBW3-UPLD-CONTENT-TYPE-LENGTH

検索対象の名前(NAME)に対応するアップロードファイルのコンテンツタイプの文字列長(バイト長)が設定されます。

COBW3-UPLD-FILE-SIZE

検索対象の名前(NAME)に対応するアップロードファイルのファイルサイズ(バイト)が設定されます。



注意

クライアントがUNIXの場合、ファイル名に'¥'が含まれていると、ファイル名およびパス名を正しく受け取ることができません。

使用例:

[c.htm (呼出し用ページ)]

```

:
<FORM METHOD="POST" ENCTYPE="multipart/form-data"
  ACTION="sample/action.script">
<P>
氏名 : <INPUT TYPE="text" NAME="NAME1"><BR>
送信ファイル1 : <INPUT TYPE="file" NAME="FILE1"><BR>
<INPUT TYPE="submit" VALUE="送信">
<INPUT TYPE="reset" VALUE="取消">
</FORM>
:

```

[COBOLプログラム]

```

:
* 名前の設定
  MOVE "FILE1" TO COBW3-SEARCH-DATA.
* アップロードファイル情報の取得
  CALL "COBW3_GET_UPLOADFILE_INFO_X" USING COBW3.
:
  IF COBW3-SEARCH-FLAG-EXIST THEN
* 生成するファイル名の設定
  MOVE "d.tmp" TO COBW3-UPLOADED-FILENAME
* ファイルの生成
  CALL "COBW3_GEN_UPLOADFILE_X" USING COBW3
:
* ファイルデータの処理
:
* 生成したファイルの削除
  CALL "COBW3_DEL_UPLOADEDFILE" USING COBW3
  END-IF.
:

```

COBW3_GEN_UPLOADFILE_X、COBW3_GEN_UPLOADFILE_N

アップロードされたファイルを、指定された名前で生成します。
各サブルーチンの意味は次のとおりです。

COBW3_GEN_UPLOADFILE_X

英数字文字列の名前(NAME)を検索し、対応するアップロードされたファイルを指定された名前で生成します。

COBW3_GEN_UPLOADFILE_N

日本語文字列の名前(NAME)を検索し、対応するアップロードされたファイルを指定された名前で生成します。



注意

セキュリティ上危険なため、アップロードされたファイルを実行可能なファイルとして生成しないでください。また、システムが自動実行するようなファイルとして生成しないでください。

書き方:

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

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

呼出し時のデータ設定:

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索するアップロードファイルの名前(NAME)を設定します(名前はファイル名ではなく、呼出し用ページのHTML文書のNAMEに指定した値)。

COBW3_GEN_UPLOADFILE_Xの場合、COBW3-SEARCH-DATAに設定します。

COBW3_GEN_UPLOADFILE_Nの場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある名前の場合、空白を含む名前の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 1024	指定された文字列の長さで検索します。

COBW3-NUMBER[省略可]

同一の名前(NAME)が複数存在する場合、検索対象の名前の出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致する名前を検索します。
-----	2 ~ 9999	指定された出現順序の名前を検索します。

COBW3-UPLOADED-FILENAME

サーバ上で実際に生成するファイル名を設定します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。または、ファイルのサイズが0です。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3_DEL_UPLOADEDFILE

COBW3_GEN_UPLOADFILE_Xなどで生成したアップロード済みのファイルを削除します。

同一リクエスト(スレッド)内で、COBW3_GEN_UPLOADFILE_XまたはCOBW3_GEN_UPLOADFILE_Nを使用して生成したファイルだけを削除できます。複数のリクエスト(ページ)にまたがるファイルの生成・削除はできません。

書き方:

```
-----
CALL "COBW3_DEL_UPLOADEDFILE" USING COBW3.
-----
```

呼出し時のデータ設定:

COBW3-UPLOADED-FILENAME

削除対象となるファイル名を設定します。

処理結果のデータ:

なし。

7.2.2.9 セッション管理

【注意】COBOL CGIサブルーチンでは当機能は使用できません。

COBW3_START_SESSION

セッションを開始します。このとき、開始するセッションのセッションデータの形式およびタイムアウトの時間設定も行います。



注意

当サブルーチンは、“COBW3_PUT_TEXT”、“COBW3_PUT_HTML”および“COBW3_FREE”を使用する前に呼び出してください。これらのサブルーチンを使用した後に当サブルーチンを呼び出した場合、セッションは開始されません。

既にセッションが開始されている場合、当サブルーチンを呼び出すとエラーが通知されます。セッション開設中はセッションデータの形式を変更することはできません。変更する場合は、“COBW3_END_SESSION”を呼び出して一度セッションを終了してください。

書き方:

```
-----
CALL "COBW3_START_SESSION" USING COBW3.
-----
```

呼出し時のデータ設定:

COBW3-SESSION-DATA-TYPE

セッションデータの形式を指定します。

条件名	値	
COBW3-SESSION-DATA-GROUPITEM	"0"	セッションデータに集団項目を使用します。
COBW3-SESSION-DATA-OBJECT	"1"	セッションデータにオブジェクト参照項目を使用します。

COBW3-SESSION-TIMEOUT

セッションのタイムアウト時間(秒)を指定します。タイムアウト時間は、“COBW3_FREE”が呼び出された時点から、同一セッション内の次のスレッドで“COBW3_INIT”が呼び出された時点までの最大待ち時間です。

値	意味
1 ~ 999999	指定した時間（秒）が経過した時点でタイムアウト処理が実行されます。

処理結果のデータ:

なし。

COBW3_END_SESSION

セッションを終了します。このとき、現行セッションの開始時に設定した情報を削除します。

**注意**

当サブルーチンは、“COBW3_PUT_TEXT”、“COBW3_PUT_HTML”および“COBW3_FREE”を使用する前に呼び出してください。これらのサブルーチンを使用した後に当サブルーチンを呼び出した場合、セッションは終了できません。

既にセッションが終了されている場合、当サブルーチンを呼び出すとエラーが通知されます。

書き方:

```
-----
CALL "COBW3_END_SESSION" USING COBW3.
-----
```

呼出し時のデータ設定:

なし。

処理結果のデータ:

なし。

COBW3_SET_SESSION_DATA

現行セッションで使用するセッションデータを登録します。登録したセッションデータは、同一セッション内で共有します。

**注意**

当サブルーチンは、セッション開設中に呼び出してください。セッションの開始前およびセッションの終了後に当サブルーチンを呼び出した場合、セッションデータは登録できません。セッション開設中はセッションデータの形式を変更することはできません。“COBW3_START_SESSION”の呼出し時に設定したセッションデータの形式に合わせたセッションデータを登録してください。

書き方:

```
-----
CALL "COBW3_SET_SESSION_DATA" USING COBW3 セッションデータ.
-----
```

呼出し時のデータ設定:

COBW3-SESSION-DATA-SIZE

セッションデータ長(バイト長)を指定します。セッションを開始した時、セッションデータの形式にCOBW3-SESSION-DATA-GROUPITEMが設定されている場合だけ有効です。

値	意味
1 ~ 999999999	指定されたセッションデータの長さで登録します。

セッションデータ

登録するセッションデータを指定します。セッションを開始した時のセッションデータの形式に合わせて、集団項目またはオブジェクト参照項目を指定します。



セッションデータに集団項目を指定する場合、従属する基本項目にオブジェクト参照項目およびポインタデータ項目を含む集団項目は指定できません。

セッションデータにオブジェクト参照項目を指定する場合、オブジェクト参照項目が指すオブジェクトのクラスは、COBW3-SESSION-ADAPTERクラスを継承していなければいけません。セッションデータにオブジェクト参照項目を指定した場合、そのスレッドが終了する前に必ず、オブジェクト参照項目にNULLオブジェクトを代入してください。NULLオブジェクトを代入しない場合、セッションを終了した後、どこからも参照されないオブジェクトがメモリ上に残ることがあり、メモリ不足につながります。

セッションデータを登録すると、集団項目の場合はデータの内容が保持されます。オブジェクト参照項目の場合はオブジェクトが保持されます。

登録するセッションデータがセッション開始時に設定したセッションデータの形式と異なる場合、Webアプリケーションの動作は保証できません。

処理結果のデータ:

なし。

COBW3_GET_SESSION_DATA

現行セッションに登録されているセッションデータを取得します。



当サブルーチンは、セッション開設中に呼び出してください。セッションの開始前およびセッションの終了後に当サブルーチンを呼び出した場合、セッションデータは取得できません。セッション開設中はセッションデータの形式を変更することはできません。“COBW3_START_SESSION”の呼出し時に設定したセッションデータの形式に合わせたセッションデータを取得してください。

書き方:

```
-----
CALL "COBW3_GET_SESSION_DATA" USING COBW3 セッションデータ.
-----
```

呼出し時のデータ設定:**COBW3-SESSION-DATA-SIZE**

取得したセッションデータを格納する集団項目長(バイト長)を指定します。セッションを開始した時、セッションデータの形式にCOBW3-SESSION-DATA-GROUPITEMが設定されている場合だけ有効です。

値	意味
1 ~ 999999999	指定されたセッションデータの長さで取得します。セッションに登録されているセッションデータと異なる場合は、エラーになります。

セッションデータ

取得するセッションデータを格納するデータ項目を指定します。セッションを開始した時のセ

セッションデータの形式に合わせて、集団項目またはオブジェクト参照項目を指定します。



セッションデータにオブジェクト参照項目を登録している場合、当サブルーチンの呼び出しパラメタに指定するオブジェクト参照項目は、NULLオブジェクトで初期化されていなければなりません。NULLオブジェクトで初期化されていない場合、セッションデータは取得できません。なお、セッションデータにオブジェクト参照項目を指定する場合は、他の選択指定のないUSAGE OBJECT REFERENCE句が記述されたオブジェクト参照項目を指定してください。取得するセッションデータがセッション開始時に設定したセッションデータの形式と異なる場合、Webアプリケーションの動作は保証できません。

セッションデータに集団項目を指定する場合、従属する基本項目の構成を登録されているセッションデータの構成と一致させるために、登録集の利用をおすすめします。

処理結果のデータ:

なし。

COBW3_ALTER_SESSION_TIMEOUT

現行セッションのタイムアウト時間(秒)を変更します。



当サブルーチンは、セッション開設中に呼び出してください。セッションの開始前およびセッションの終了後に当サブルーチンを呼び出した場合、セッションのタイムアウト時間は変更できません。

書き方:

```
-----
CALL "COBW3_ALTER_SESSION_TIMEOUT" USING COBW3.
-----
```

呼出し時のデータ設定:

COBW3-SESSION-TIMEOUT

セッションのタイムアウト時間(秒)を指定します。タイムアウト時間の計測は、“COBW3_FREE”が呼び出された時点から、同一セッション内の次のスレッドで“COBW3_INIT”が呼び出された時点までの最大待ち時間です。

値	意味
1 ~ 999999	指定した時間(秒)が経過した時点でタイムアウト処理が実行されます。

処理結果のデータ:

なし。

COBW3_GET_SESSION_INFO

現行セッションの情報を取得します。

書き方:

```
-----
CALL "COBW3_GET_SESSION_INFO" USING COBW3.
-----
```

呼出し時のデータ設定:

なし。

処理結果のデータ:

COBW3-SESSION-TIMEOUT

現行セッションのタイムアウト時間(秒)が設定されます。

COBW3-SESSION-ID

現行セッションのセッションIDを表わす数字が設定されます。

COBW3-SESSION-DATA-SIZE

現行セッションのセッションデータ長(バイト長)が設定されます。セッションを開始した時、セッションデータの形式にCOBW3-SESSION-DATA-GROUPITEMが設定されている場合だけ取得できます。セッションデータの形式にCOBW3-SESSION-DATA-OBJECTが設定されている場合は、ゼロが設定されます。

COBW3-SESSION-DATA-TYPE

現行セッションのセッションデータの形式が設定されます。

条件名	値	
COBW3-SESSION-DATA-GROUPITEM	"0"	セッションデータに集団項目を使用しています。
COBW3-SESSION-DATA-OBJECT	"1"	セッションデータにオブジェクト参照項目を使用しています。

COBW3-SESSION-STATUS

現行セッションの状態が設定されます。

条件名	値	
COBW3-SESSION-STATUS-NON	"0"	セッションは開始されていないか、終了しています。
COBW3-SESSION-STATUS-STARTED	"1"	セッションは開始されています。

7.2.2.10 SAF固有サブルーチン

COBW3_SAF_GET_PARM_XX 、 COBW3_SAF_GET_PARM_NX 、 COBW3_SAF_GET_PARM_XN 、
COBW3_SAF_GET_PARM_NN

SAFディレクトリのWebアプリケーションの実行指定時に設定されるNESパラメタ情報から任意のNESパラメタ名の検索を行います。また、NESパラメタ名に対応するパラメタ値を取得します。各サブルーチンの意味は次のとおりです。

COBW3_SAF_GET_PARM_XX

英数字文字列のNESパラメタ名を検索し、対応するNESパラメタ値を英数字文字列として返します。

COBW3_SAF_GET_PARM_NX

日本語文字列のNESパラメタ名を検索し、対応するNESパラメタ値を英数字文字列として返します。

COBW3_SAF_GET_PARM_XN

英数字文字列のNESパラメタ名を検索し、対応するNESパラメタ値を日本語文字列として返します。

COBW3_SAF_GET_PARM_NN

日本語文字列のNESパラメタ名を検索し、対応するNESパラメタ値を日本語文字列として返します。

なお、取得したNESパラメタ値を格納する際のパディング文字列(埋草文字)は次のとおりです。

COBW3_SAF_GET_PARM_XX、COBW3_SAF_GET_PARM_NXの場合：

半角空白

COBW3_SAF_GET_PARM_XN、COBW3_SAF_GET_PARM_NNの場合：

全角空白

【補足】 NESの設定で指定するNESパラメタ名は大文字小文字を区別するため、必要に応じてWebアプリケーションの翻訳時に翻訳オプションのALPHAL(WORD)かNOALPHALを指定してください。



注意

NESパラメタに日本語(2バイト文字)を指定した場合、指定したパラメタの検索に失敗することがあります。このため、NESパラメタには日本語を使用しないでください。使用する場合は、実際に運用する前に、必ずその文字が正しく検索および取得できることを確認してください。

書き方：

```
-----
CALL "COBW3_SAF_GET_PARM_XX" USING COBW3SAF.
-----
```

```
-----
CALL "COBW3_SAF_GET_PARM_NX" USING COBW3SAF.
-----
```

```
-----
CALL "COBW3_SAF_GET_PARM_XN" USING COBW3SAF.
-----
```

```
-----
CALL "COBW3_SAF_GET_PARM_NN" USING COBW3SAF.
-----
```

呼出し時のデータ設定：

COBW3-SAF-PARM-NAME、COBW3-SAF-PARM-NAME-N

検索するNESパラメタ名を設定します。

COBW3_SAF_GET_PARM_XXまたはCOBW3_SAF_GET_PARM_XNの場合、COBW3-SAF-PARM-NAMEに設定します。

COBW3_SAF_GET_PARM_NXまたはCOBW3_SAF_GET_PARM_NNの場合、COBW3-SAF-PARM-NAME-Nに設定します。

COBW3-SAF-PARM-NAME-LENGTH[省略可]

最後に有効な空白があるNESパラメタ名を検索する場合、空白を含むNESパラメタ名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。ただし、COBW3-SAF-PARM-NAMEまたはCOBW3-SAF-PARM-NAME-Nがすべて空白の場合、文字列長をゼロとして処理します。
1～64	指定された文字列の長さで検索します。

処理結果のデータ：

COBW3-SAF-SEARCH-FLAG

条件名	値	意味
COBW3-SAF-PARM-FLAG-NON	"0"	検索対象の名前が存在しません。

COBW3-SAF-PARM-FLAG-EXIST	"1"	検索対象の名前が存在しました。
---------------------------	-----	-----------------

COBW3-SAF-PARM-VALUE、COBW3-SAF-PARM-VALUE-N

検索対象のNESパラメタ名に対応するNESパラメタ値が設定されます。

COBW3_SAF_GET_PARM_XXまたはCOBW3_SAF_GET_PARM_NXの場合、COBW3-SAF-PARM-VALUEに設定されます。

COBW3_SAF_GET_PARM_XNまたはCOBW3_SAF_GET_PARM_NNの場合、COBW3-SAF-PARM-VALUE-Nに設定されます。

COBW3-SAF-PARM-LENGTH

検索対象のNESパラメタ名に対応するNESパラメタ値の文字列長(バイト長)が設定されます。

COBW3_SAF_GET_BASENAME

URLに指定されるアプリケーション名を取得します。

【補足】 NESではアプリケーションをMIMEタイプ(拡張子)で識別するため、アプリケーション名を入力データの一部として処理の切替えなどに使用できます。アプリケーション名は任意ですが、"a" ~ "z"、"A" ~ "Z"、"0" ~ "9"、"-(ハイフン)" および "_ (アンダースコア)" 以外の文字が含まれている場合、動作は保証されません。

書き方:

```
-----
CALL "COBW3_SAF_GET_BASENAME" USING COBW3SAF.
-----
```

呼出し時のデータ設定:

なし

処理結果のデータ:

COBW3-SAF-BASENAME

アプリケーション名が設定されます。

COBW3-SAF-BASENAME-LENGTH

アプリケーション名の文字列長(バイト長)が設定されます。

7.2.2.11 その他

以下のサブルーチンは互換を保証するために提供されています。



注意

以下のサブルーチンは、アプリケーションの動作コード系がシフトJISの場合にだけ、動作が保証されます。

COBW3_NAME、COBW3_NAMEN

"COBW3_INIT" で獲得したWebパラメタから任意の名前(NAME)を検索します。

また、名前(NAME)に対応する値(VALUE)を取得します。なお、取得した値(VALUE)を格納する際のパディング文字列(埋草文字)は次のとおりです。

COBW3_NAMEの場合:

半角空白

COBW3_NAMENの場合:

全角空白

書き方:

```
-----
```

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

```
CALL "COBW3_NAMEN" USING COBW3.
```

呼出し時のデータ設定:

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索する名前(NAME)を設定します。(名前は呼出し用ページのHTML文書のNAMEに指定した値)

英数字文字列を検索する場合、COBW3-SEARCH-DATAに設定します。

日本語文字列を検索する場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある名前(NAME)の場合、空白を含む名前の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 1024	指定された文字列の長さで検索します。

COBW3-NUMBER[省略可]

Webパラメタに同一の名前(NAME)が複数存在する場合、検索対象の名前の出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致する名前を検索します。
-----	2 ~ 9999	指定された出現順序の名前を検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3-GET-DATA、COBW3-GET-DATA-N

検索対象のNAMEに対応するVALUEが設定されます。

COBW3_NAMEの場合、COBW3-GET-DATAに設定されます。

COBW3_NAMENの場合、COBW3-GET-DATA-Nに設定されます。

COBW3-GET-LENGTH

検索対象のNAMEに対応するVALUEの文字列長(バイト長)が設定されます。

COBW3_VALUE

“COBW3_INIT”で獲得したWebパラメタから任意の値(VALUE)の検索を行います。このサブルーチンは、チェックボックスのある呼出し用ページから起動された場合に、WWWブラウザでチェックされた項目を値(VALUE)で判定する場合などに使用します。

書き方:

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


呼出し時のデータ設定:

COBW3-SEARCH-DATA、COBW3-SEARCH-DATA-N

検索するVALUEを設定します。

英数字文字列を検索する場合、COBW3-SEARCH-DATAに設定します。

日本語文字列を検索する場合、COBW3-SEARCH-DATA-Nに設定します。

COBW3-SEARCH-LENGTH[省略可]

最後に有効な空白がある値(VALUE)の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 1024	指定された文字列の長さで検索します。

COBW3-NUMBER[省略可]

Webパラメタに同一のVALUEが複数存在する場合、検索対象のVALUEの出現順序を設定します。

条件名	値	意味
COBW3-NUMBER-INIT	1	最初に一致するVALUE を検索します。
-----	2 ~ 9999	指定された出現順序のVALUE を検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象のVALUE が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象のVALUE が存在しました。

COBW3_CNV_SET

“COBW3_PUT_HTML”で出力する結果出力用ページ内に指定された変換データを登録します。

登録された情報はCOBW3_PUT_HTML実行時に参照され、登録されたデータに従って、結果出力用ページ内の変換が行われます。



結果出力用ページ内に複数個の異なる変換名が指定されている場合は、変換名ごとに“COBW3_CNV_SET”を呼び出して変換データを登録してください。

書き方:

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

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

変換対象となる変換名を設定します。

変換名が英数字の場合、COBW3-CNV-NAMEに設定します。

変換名が日本語の場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
<u>0</u>	空白以外の最後の文字までの長さで登録します。ただし、COBW3-CNV-NAMEがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 30	指定された文字列の長さで登録します。

COBW3-CNV-VALUE、COBW3-CNV-VALUE-N

変換結果(変換文字列)を設定します。

変換文字列が英数字の場合、COBW3-CNV-VALUEに設定します。

変換文字列が日本語の場合、COBW3-CNV-VALUE-Nに設定します。

COBW3-CNV-VALUE-LENGTH[省略可]

最後に有効な空白がある変換名の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
<u>0</u>	空白以外の最後の文字までの長さで登録します。ただし、COBW3-CNV-VALUE がすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 1024	指定された文字列の長さで登録します。

COBW3-CNV-MODE[省略可]

変換種別を設定します。

条件名	値	意味
<u>COBW3-CNV-MODE-ADDREP</u>	LOW-VALUE	指定した変換名が登録されていない場合、変換データを追加します。
COBW3-CNV-MODE-REPLACE	"1"	変換データの置き換えを行います。指定した変換名が登録されていない場合、エラーを出力して変換を行いません。
COBW3-CNV-MODE-ADD	"2"	変換データの追加を行います。指定した変換名が登録されている場合、エラーを出力して追加を行いません。

処理結果のデータ:

なし。

COBW3_CNV_DEL

COBW3_CNV_SETで登録した変換データの削除を行います。

書き方:

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

呼出し時のデータ設定:

COBW3-CNV-NAME、COBW3-CNV-NAME-N

削除対象となる変換名を設定します。

変換名が英数字の場合、COBW3-CNV-NAMEに設定します。

変換名が日本語の場合、COBW3-CNV-NAME-Nに設定します。

COBW3-CNV-NAME-LENGTH[省略可]

最後に有効な空白がある変換名の場合、空白を含む値の文字列長(バイト長)を設定します。

値	意味
---	----

0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-CNV-NAMEがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 30	指定された文字列の長さで登録します。

処理結果のデータ:

なし。

COBW3_CNV_INIT

COBW3_CNV_SETで登録した変換データをすべて初期化します。

書き方:

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

呼出し時のデータ設定:

なし。

処理結果のデータ:

なし。

COBW3_SET_COOKIE

ブラウザへ送信するCookieデータの登録を行います。登録されたCookieデータはヘッダ出力時に使用されます。

**注意**

当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定したCOOKIEデータは無効になります。

WWWブラウザへ送信されるCookieの文字コードは、COBOLの動作コード系に関係なくシフトJISになります。ただし、CookieはCOBOL以外のアプリケーションでも送受信されることおよび、主としてキー情報などを送受信するのに使用されることから、ASCIIの範囲で設定することをおすすめします。

書き方:

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

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

登録するCookie名を設定します。

Cookie名が英数字の場合、COBW3-COOKIE-NAMEに設定します。

Cookie名が日本語の場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名の場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-COOKIE-NAME がすべて空白の場合、文字列長をゼロとして処理します。

1 ~ 64	指定された文字列の長さで登録します。
--------	--------------------

COBW3-COOKIE-VALUE、COBW3-COOKIE-VALUE-N

Cookie値を設定します。

Cookie値が英数字の場合、COBW3-COOKIE-VALUEに設定します。

Cookie値が日本語の場合、COBW3-COOKIE-VALUE-Nに設定します。

COBW3-COOKIE-VALUE-LENGTH[省略可]

最後に有効な空白があるCookie値の場合、空白を含むCookie値の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで登録します。ただし、COBW3-COOKIE-VALUEがすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 1024	指定された文字列の長さで登録します。

COBW3-COOKIE-EXPIRES[省略可]

Cookieの有効期限を指定する場合に使用します。この項目は、次の項目(年、月、日、時、分、秒)を持ちます。有効期限の指定はGMT(グリニッジ標準時刻)で行います。

COBW3-COOKIE-EXPIRES-YEAR(年)(1582 ~ 9999)

COBW3-COOKIE-EXPIRES-MONTH(月)(01 ~ 12)

COBW3-COOKIE-EXPIRES-DAY(日)(01 ~ 31 ただし、末日は月によって異なります。)

COBW3-COOKIE-EXPIRES-HOUR(時)(00 ~ 23)

COBW3-COOKIE-EXPIRES-MIN(分)(00 ~ 59)

COBW3-COOKIE-EXPIRES-SEC(秒)(00 ~ 59)



有効期限が省略された場合、そのCookieデータはWWWブラウザ終了時に削除されます。また、過去の時刻を指定することで、現在WWWブラウザが持っているCookieデータを削除できます。

JST(日本標準時)とGMTの時差は9時間です。したがって、JSTから9を引いた値がGMTになります。

有効期限の指定には1582年10月15日以降の日付を使用してください。この日より前の日付を指定すると曜日の自動算出ができないため、有効期限が無効になります。

COBW3-COOKIE-DOMAIN[省略可]

Cookieが有効となるドメインを指定します。

COBW3-COOKIE-PATH[省略可]

Cookieが有効となる仮想ディレクトリを指定します。

COBW3-COOKIE-SECURE[省略可]

セキュリティを指定します。セキュリティが有効な場合、SSLを使用している場合にだけWWWブラウザからCookieデータが送信されます。

条件名	値	意味
COBW3-COOKIE-SECURE-OFF	LOW-VALUE	セキュリティを無効にします。
COBW3-COOKIE-SECURE-ON	"1"	セキュリティを有効にします。

COBW3-COOKIE-MODE[省略可]

Cookieデータの登録種別を設定します。

条件名	値	意味
COBW3-COOKIE-MODE-ADDREP	LOW-VALUE	指定したCookie名が登録されていない場合、Cookieデータを追加します。

COBW3-COOKIE-MODE-REPLACE	"1"	Cookieデータの置き換えを行います。 指定したCookie名が登録されていない場合、エラーを出力して置き換えを行いません。
COBW3-COOKIE-MODE-ADD	"2"	Cookieデータの追加を行います。指定したCookie名が登録されている場合、エラーを出力して追加を行いません。

処理結果のデータ:

なし。

COBW3_DEL_COOKIE

COBW3_SET_COOKIEで登録したCookieデータの削除を行います。

**注意**

当サブルーチンは、COBW3_PUT_HTMLおよびCOBW3_PUT_TEXTを使用する前に使用してください。COBW3_PUT_HTMLまたはCOBW3_PUT_TEXTを一度でも使用した後に当サブルーチンを使用した場合、設定した情報は無効になります。

書き方:

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

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

削除するCookie名を設定します。

Cookie名が英数字の場合、COBW3-COOKIE-NAMEに設定します。

Cookie名が日本語の場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名の場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで削除します。ただし、COBW3-COOKIE-NAME がすべて空白の場合、文字列長をゼロとして処理します。
1 ~ 64	指定された文字列の長さで削除します。

処理結果のデータ:

なし。

COBW3_GET_COOKIE、COBW3_GET_COOKIE_N

“COBW3_INIT”で獲得したリクエストCookieデータから任意の名前(NAME)の検索を行います。

また、名前(NAME)に対応する値(VALUE)を取得します。なお、取得した値(VALUE)を格納する際のパディング文字列(埋草文字)は次のとおりです。

COBW3_GET_COOKIEの場合:

半角空白

COBW3_GET_COOKIE_Nの場合:

全角空白

書き方:

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

```
CALL "COBW3_GET_COOKIEEN" USING COBW3.
```

呼出し時のデータ設定:

COBW3-COOKIE-NAME、COBW3-COOKIE-NAME-N

検索するCookie名を設定します。

英数字文字列を検索する場合、COBW3-COOKIE-NAMEに設定します。

日本語文字列を検索する場合、COBW3-COOKIE-NAME-Nに設定します。

COBW3-COOKIE-NAME-LENGTH[省略可]

最後に有効な空白があるCookie名の場合、空白を含むCookie名の文字列長(バイト長)を設定します。

値	意味
0	空白以外の最後の文字までの長さで検索します。
1 ~ 64	指定された文字列の長さで検索します。

処理結果のデータ:

COBW3-SEARCH-FLAG

条件名	値	意味
COBW3-SEARCH-FLAG-NON	"0"	検索対象の名前が存在しません。
COBW3-SEARCH-FLAG-EXIST	"1"	検索対象の名前が存在しました。

COBW3-COOKIE-VALUE、COBW3-COOKIE-VALUE-N

検索対象のCookie名に対応するCookie値が設定されます。

COBW3_GET_COOKIEの場合、COBW3-COOKIE-VALUEに設定されます。

COBW3_GET_COOKIEENの場合、COBW3-COOKIE-VALUE-Nに設定されます。

COBW3-COOKIE-VALUE-LENGTH

検索対象のCookie名に対応するCookie値の文字列長(バイト長)が設定されます。

7.2.3 Webサブルーチンの定量制限

Webサブルーチンで使用するデータの定量制限は以下のとおりです。

項目	値
Webパラメタの最大長 (注1)	1073741822バイト
WebパラメタのNAMEの最大文字列長	1024バイト
WebパラメタのVALUEの最大文字列長	1024バイト
COBW3-HTML-FILENAMEに指定する結果出力用ページ(ファイル)の最大レコード長 (注2)	1024バイト
変換データを置き換えた後の結果出力用ページの1行の長さ	3072バイト

注1

WebパラメタはFORMタグのENCTYPEの指定により、データの形式が異なります。WebパラメタにはNAMEおよびVALUEの他、ファイルアップロードのデータなども含まれます。また、WebパラメタはWWWサーバのあるマシンのメモリ上に獲得されるため、使用可能なメモリが少ない場合は、データのサイズが定量制限を超えていない場合でもデータを受け取れないことがあります。このような場合は、必要に応じて呼出し用ページにアップロード可能な

ファイルのサイズを表示したり、制限を超えるファイルをアップロードしないように注意書きを表示するなどの対処を行ってください。なお、Webパラメタのデータ形式の詳細については、HTMLについて解説している書籍またはホームページなどを参照してください。

注2

市販のHTMLエディタを使用して作成されたHTML文書は、1行のバイト数が1024バイトを超える可能性があります。この場合、メモ帳などのテキストエディタを使用して1024バイトを超えないように調整してください。

7.3 Webサブルーチンの提供するクラス

Webサブルーチンでは、セッション管理機能で使用するクラスを提供しています。

COBW3-SESSION-ADAPTERクラス

セッションデータとして登録されたオブジェクトを管理するためのクラスです。セッションデータに登録するオブジェクトは、当クラスを継承するクラスから生成されなければなりません。また、当クラスに定義されているSWEEP-SESSIONメソッドをOVERRIDE句を利用して上書きしたメソッド内で、タイムアウト処理を記述します。



注意

タイムアウト処理は、セッションデータにオブジェクト参照が登録されている場合だけ利用できます。

タイムアウト処理では、Webサブルーチン呼び出すことはできません。

ファクトリメソッド:

なし。

オブジェクトメソッド:

SWEEP-SESSION

セッションのタイムアウト発生時に呼び出されるメソッド。必要に応じて、ファイルのクローズやDBのコミット/ロールバック等の処理を記述するために利用します。

メソッドの引数:

なし。

メソッドの復帰情報:

なし。

【補足】

セッションのタイムアウトが発生した場合、WebサブルーチンはSWEEP-SESSIONメソッドを呼び出し、セッションデータに登録されているオブジェクト参照にNULLオブジェクトを代入します。

SWEEP-SESSIONメソッドが上書き定義されていない場合、SWEEP-SESSIONメソッドは何も処理を行いません。

COBW3-SESSION-ADAPTERクラスは、FJBASEクラスを継承するクラスです。

使用例:

[COBOLプログラム]

```
IDENTIFICATION DIVISION.
CLASS-ID. SESSDATA INHERITS COBW3-SESSION-ADAPTER.
:
IDENTIFICATION DIVISION.
OBJECT.
:
IDENTIFICATION DIVISION.
METHOD-ID. SWEEP-SESSION OVERRIDE.
DATA DIVISION.
:
PROCEDURE DIVISION.
:
```

* 必要に応じてタイムアウト処理を記述する


```
      :  
      END METHOD SWEEP-SESSION.  
      :  
      END OBJECT.  
END CLASS SESSDATA.
```

第8章 動作確認

作成したプログラムを動作確認(テスト時の確認または問題発生時の確認)をするには、以下の方法があります。

- ログ情報を参照する

- 対話型デバッガで動作確認する

- Webサブルーチンが検出したエラーを参照する

- 実行中のデータを表示形式で参照する

以降にそれぞれのデバッグ方法について説明します。

8.1 ログ情報を参照する

Webサブルーチンではログ情報をログファイルに出力するための機構を用意しています。このログ情報はアプリケーションを変更することなく、設定を変えることができるので運用時のデバッグに有効です。設定方法は、“[2.4.2 CGIサブルーチンの環境変数設定](#)” “[3.4.3 ISAPIサブルーチンの環境変数設定](#)” “[4.4.3 SAFサブルーチンの環境変数設定](#)” をそれぞれ参照してください。なお、このログ情報はWebサブルーチンのログ情報であるため、アプリケーションそのもののトレースには役立ちませんが、動作確認の目安になります。ログファイルの構成は次のとおりです。

プロセスID	スレッドID	年 - 月 - 日	時 : 分 : 秒	重要度	メッセージ
--------	--------	-----------	-----------	-----	-------

具体的には次のような内容です。

0000000188	0000000187	1999-02-02	11:48:07	02	COB-06310: COBW3: 指定された変換名は既に登録されています。既に登録されている変換情報を有効にします。
0000000188	0000000081	1999-02-02	11:48:10	01	COB-04470: COBW3: 指定された変換情報は登録されていないため、変更できませんでした。

なお、アプリケーションのトレース情報を採取する場合には、COBOLのTRACE機能を使用してください。TRACE機能の詳細については、“[NetCOBOL 使用手引書](#)” を参照してください。

8.2 対話型デバッガで動作確認する

デバッガを使用してCOBOLで作成したWebアプリケーションをデバッグするには、デバッグしたいプログラムからデバッガを起動する方法を使用します。

リモートデバッガを使用して、WWWブラウザを起動しているクライアントからWebアプリケーションをリモートでデバッグすることも出来ます。リモートデバッガの使い方については、“NetCOBOL 使用手引書”を参照してください。

8.2.1 CGIアプリケーションの動作確認

デバッガを使用してCGIアプリケーションをデバッグを開始するには、以下の手順で行います。

1. Webアプリケーションを翻訳・リンクします。
デバッグを行うための翻訳オプションおよびリンクオプションを指定して、Webアプリケーションを作成します。デバッグを行うための翻訳およびリンクの方法については“NetCOBOL 使用手引書”を参照してください。
2. 実行環境情報を設定します。
デバッグしたいプログラムからデバッガを起動するために、あらかじめ次の実行環境情報をシステムの環境変数または実行用の初期化ファイル(COBOL85.CBR)に設定してください。システムの環境変数に設定した場合は、設定後システムを再起動してください。
@CBR_ATTACH_TOOL=TEST [起動パラメタ]
アプリケーションの実行時にデバッガを起動する(TEST)ことを指定します。“TEST”に続けてデバッガ起動パラメタを指定できます。起動パラメタについては、“NetCOBOL 使用手引書”を参照してください。
3. デバッグするWebアプリケーションが実行されるコンピュータにログオンします。
4. WWWブラウザからWebアプリケーションを呼び出します。このとき、WWWブラウザはデバッグするWebアプリケーションが実行されるコンピュータでなくてもかまいません。
5. Webアプリケーションが起動されたとき、デバッガが自動的に起動されます。デバッガが起動されたら、[デバッグを開始する]ダイアログから、デバッグ情報ファイル格納フォルダと必要な情報を指定してデバッグを開始します。

以降の操作は、通常のデバッガを使用したデバッグと同じです。デバッガの使用方法については、“NetCOBOL 使用手引書”を参照してください。



注意

デバッガを使用してデバッグを行う場合は、Webアプリケーションの動作時間が通常の場合よりも長くなります。したがって、WWWブラウザによっては、デバッグ中にタイムアウトが発生することがあります。タイムアウトが設定できるWWWブラウザは、適切な値に変更してください。また、タイムアウトが設定できるWWWサーバは、適切な値に変更してください。

8.2.2 ISAPIアプリケーションの動作確認

デバッガを使用してデバッグする場合には、IISはWebアプリケーションを同じメモリ空間に読み込んで呼び出すため、デバッグするには以下の点を考慮する必要があります。

Webアプリケーションが不正な動作をした場合、Webアプリケーションが異常になるばかりでなく、同時にロードされているほかのWebアプリケーションやIIS自身も異常になる可能性があります。デバッグ時には、まず、ほかのWebアプリケーションが何も組み込まれていない状態で検証し、次にいくつかのWebアプリケーションを混在させて確認することを推奨します。

デバッガは、Webアプリケーションだけでなく、ほかのWebアプリケーションやIIS自身も

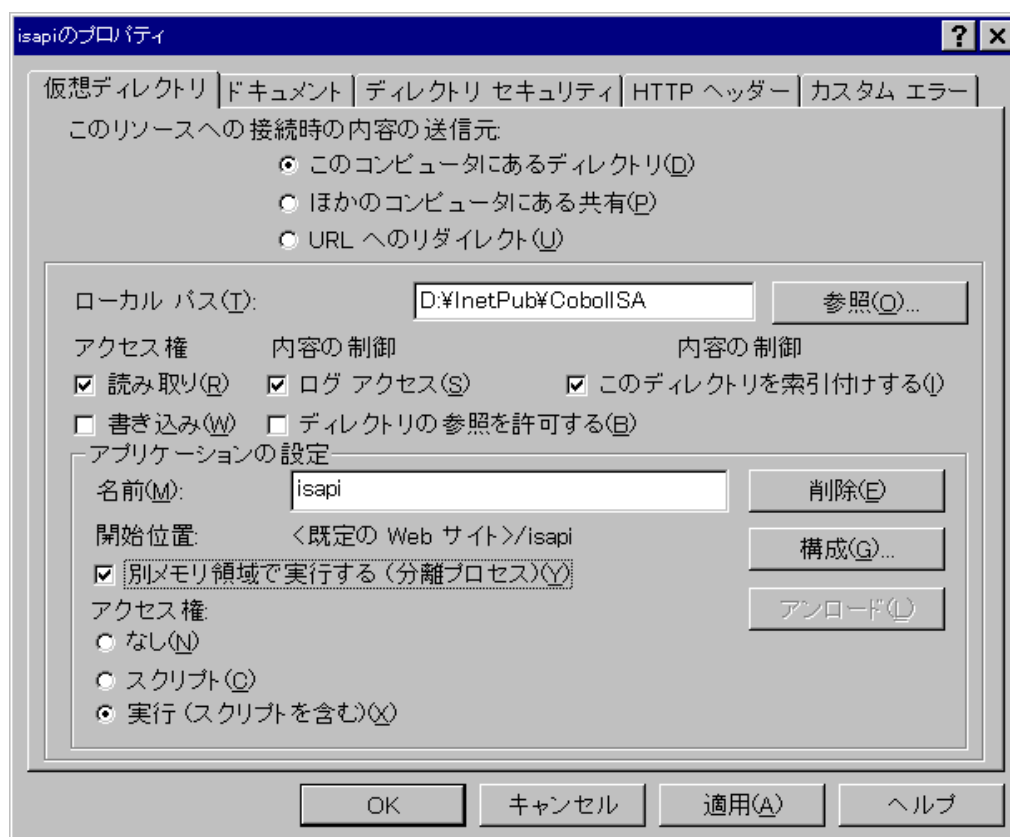
デバッグの対象としています。よって、デバッグを終了させるとほかのWebアプリケーションやIIS自身も終了するため、IISを再起動する必要があります。

IISには、WebアプリケーションをIISのメモリ空間で動作させる方法(IISプロセス)とIISとは別のメモリ空間で動作させる方法(分離プロセスまたはプール)があります。Webアプリケーションを分離プロセスまたはプールにした場合、上記の問題がなくなるため、デバッグ作業が容易になります。

IIS 4.0の場合

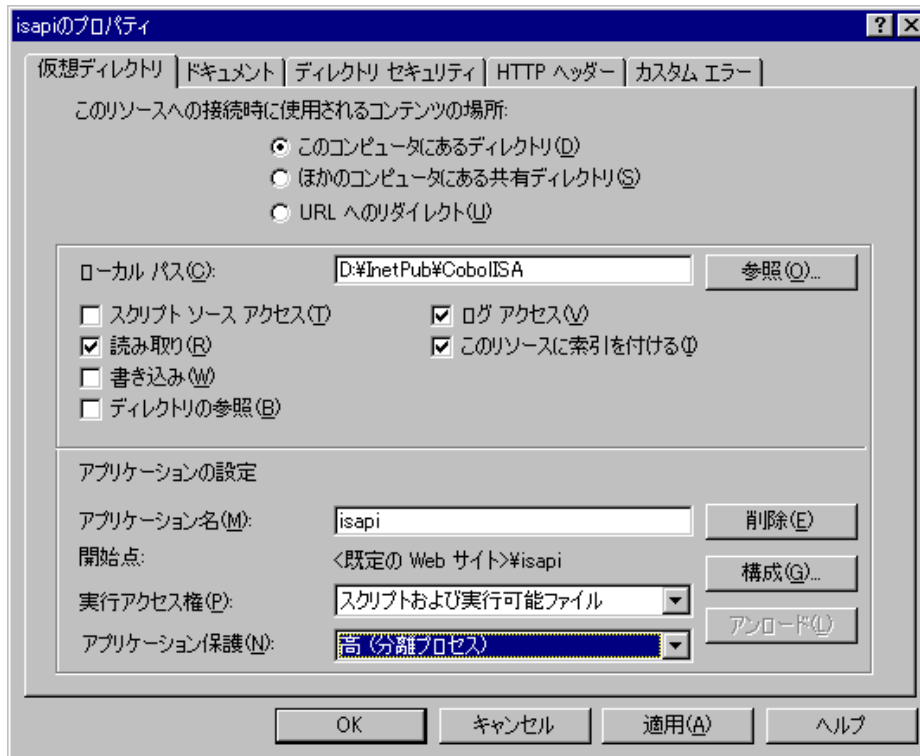
インターネットサービスマネージャを開き、Webアプリケーションが属する仮想ディレクトリの[プロパティ]を開きます。[アプリケーションの設定]の中にある[別メモリ領域で実行する(分離プロセス)]のチェックボックスをチェックしてください。この操作によって、選択された仮想ディレクトリに属するWebアプリケーションは分離プロセスとして動作します。

なお、プールはIIS 4.0ではサポートされていません。



IIS 5.0の場合

インターネットサービスマネージャを開き、Webアプリケーションが属する仮想ディレクトリの[プロパティ]を開きます。[アプリケーションの設定]の中にある[アプリケーション保護]というコンボボックスがあるので、[高(分離プロセス)]か[中(プール)]のどちらかを選択してください。この操作によって、選択された仮想ディレクトリに属するWebアプリケーションは分離プロセスまたはプールとして動作します。



注意

Webアプリケーションで発生するアプリケーションエラーは、IISが検知して処理するため、アプリケーションエラーに対するジャストインタイムデバッグ機能およびCOBOLランタイムシステムの診断機能は動作しません。

8.2.2.1 デバッガを起動する

デバッガを使用してデバッグを開始するには、以下の手順で行います。

1. Webアプリケーションを翻訳・リンクします。
デバッグを行うための翻訳オプションおよびリンクオプションを指定して、Webアプリケーションを作成します。デバッグを行うための翻訳およびリンクの方法については“NetCOBOL 使用手引書”を参照してください。
2. 実行環境情報を設定します。
デバッグしたいプログラムからデバッガを起動するために、あらかじめ次の実行環境情報を設定しておく必要があります。実行環境情報の設定方法については“3.4.3 [ISAPI サブルーチンの環境変数設定](#)”を参照してください。
@CBR_ATTACH_TOOL=TEST [起動パラメタ]
アプリケーションの実行時にデバッガを起動する(TEST)ことを指定します。
“TEST”に続けてデバッガ起動パラメタを指定できます。起動パラメタについては、“NetCOBOL 使用手引書”を参照してください。
3. デバッグするWebアプリケーションが実行されるコンピュータにログオンします。デバッグ中は、IISのサービスを停止・起動することがあるので、管理者権限でログオンしてください。
4. WWWブラウザからWebアプリケーションを起動します。このとき、WWWブラウザはデバッグするWebアプリケーションが実行されるコンピュータでなくてもかまいません。
5. Webアプリケーションが起動されたとき、デバッガが自動的に起動されます。デバッガが起動されたら、[デバッグを開始する]ダイアログから、デバッグ情報ファイル格納フォルダと必要な情報を指定してデバッグを開始します。

8.2.2.2 デバッグを行う

デバッグ操作は、通常のデバッガを使用したデバッグと同じです。デバッガの使用方法については、“NetCOBOL 使用手引書”を参照してください。ただし、デバッグを行う前に、以下の点に注意する必要があります。

デバッガは、デバッグを開始したときにWebアプリケーションの入口で実行を自動的に中断し、デバッグ操作の機会を与えます。WebアプリケーションがWWWブラウザに応答した後に次のリクエストによって実行されたときは、自動的に中断しないため、あらかじめ中断点を設定しておく必要があります。なお、次のリクエストによってロードされるWebアプリケーションで実行を中断させるには、コマンドやダイアログボックスで、ロードされるプログラムの名前を指定して中断点の設定を指示します。詳細については、“NetCOBOL 使用手引書”の“19.6.1 動的構造に対するデバッグ方法”を参照してください。

Webアプリケーションで発生するアプリケーションエラー(例外)は、IISが検知して処理するため、デバッガはアプリケーションエラーが発生したことを知るできません。デバッガでアプリケーションエラーを検知するには、デバッガの[環境設定]ダイアログの[動作]ページで、[初回のシグナル受信時に中断する]をチェックしてください。詳細については、“NetCOBOL 使用手引書”の“19.6.2 例外ハンドラで処理されるシグナルのデバッグ方法”を参照してください。

WWWブラウザに응答した後は、Webアプリケーションは次のリクエストを待っているためプログラム実行中になっています。中断点の設定などのデバッグ操作を行いたい場合は、デバッガのメニューから[実行中断]コマンドを選択し、プログラムの実行を中断させてから行ってください。

デバッグを行う際には、次の点にも注意をする必要があります。

デバッガを使用してデバッグを行う場合は、Webアプリケーションの動作時間が通常の場合よりも長くなります。したがって、WWWブラウザによっては、デバッグ中にタイムアウトが発生することがあります。タイムアウトが設定できるWWWブラウザは、適切な値に変更してください。また、IISの接続タイムアウトの値を適切な値に変更してください。

8.2.2.3 デバッガを終了する

デバッガを先に終了させる場合

Webアプリケーションが分離プロセスまたはプールの場合

デバッガを終了すると、Webアプリケーションのプロセスだけが終了します。したがって、IIS自身は終了しないため、IISを再起動する必要はありません。

WebアプリケーションがIISプロセスの場合

デバッガを終了すると、IIS自身も終了します。したがって、IISを再起動する必要があります。

Webアプリケーションを先に終了させる場合

Webアプリケーションが分離プロセスまたはプールの場合

インターネットサービスマネージャを使って、Webアプリケーションが属する仮想ディレクトリの[プロパティ]を開き、[アンロード]ボタンを押して、Webアプリケーションのプロセスを終了させます。なお、Webアプリケーションのプロセスを終了させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。

WebアプリケーションがIISプロセスの場合

IISを停止させます。

IISは、インターネットサービスマネージャを使っても完全には停止しないため、サービスを開いて、IISのすべてのサービス(IIS Admin Serviceおよびそれに依存するサービス)を停止させます。なお、IISを停止させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。

8.2.2.4 Webアプリケーションを入れ替える

Webアプリケーションの入れ替えは、以下の方法で行います。

Webアプリケーションが分離プロセスまたはプールの場合

Webアプリケーションのプロセスを終了させてからWebアプリケーションを入れ替えます。Webアプリケーションのプロセスを終了させるには、インターネットサービスマネージャを使ってWebアプリケーションが属する仮想ディレクトリの[プロパティ]を開き、[アンロード]ボタンを押します。なお、Webアプリケーションのプロセスを終了させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。この方法でWebアプリケーションを入れ替えられない場合は、WebアプリケーションがIISプロセスの場合に示す方法で行ってください。

WebアプリケーションがIISプロセスの場合

IISを停止させてからWebアプリケーションを入れ替えて、再度IISを開始します。

IISは、インターネットサービスマネージャを使っても完全には停止しないため、サービスを開いて、IISのすべてのサービス(IIS Admin Serviceおよびそれに依存するサービス)を停止させます。なお、IISを停止させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。

IISを開始するには、インターネットサービスマネージャを使って行うか、またはサービスを開いて、IISのすべてのサービス(IIS Admin Serviceおよびそれに依存するサービス)を開始します。

8.2.3 SAFアプリケーションの動作確認

デバッガを使用してデバッグする場合には、NESはWebアプリケーションを同じメモリ空間に読み込んで呼び出すため、デバッガを使用してデバッグする際には以下の点を考慮する必要があります。

Webアプリケーションが不正な動作をした場合、Webアプリケーションが異常になるばかりでなく、同時にロードされているほかのWebアプリケーションやNES自身も異常になる可能性があります。デバッグ時には、まず、ほかのWebアプリケーションが何も組み込まれていない状態で検証し、次にいくつかのWebアプリケーションを混在させて確認することを推奨します。

デバッガは、Webアプリケーションだけでなく、ほかのWebアプリケーションやNES自身もデバッグの対象としています。よって、デバッグを終了させるとほかのWebアプリケーションやNES自身も終了するため、NESを再起動する必要があります。

8.2.3.1 デバッガを起動する

デバッガを使用してデバッグを開始するには、以下の手順で行います。

1. Webアプリケーションを翻訳・リンクします。

デバッグを行うための翻訳オプションおよびリンクオプションを指定して、Webアプリケーションを作成します。デバッグを行うための翻訳およびリンクの方法については“NetCOBOL 使用手引書”を参照してください。

2. 実行環境変数を設定します。

デバッグしたいプログラムからデバッガを起動するために、あらかじめ次の実行環境情報を設定しておく必要があります。実行環境情報の設定方法については“4.4.3 [SAFサブルーチンの環境変数設定](#)”を参照してください。

@CBR_ATTACH_TOOL=TEST [起動パラメタ]

アプリケーションの実行時にデバッガを起動する(TEST)ことを指定します。“TEST”に続けてデバッグ起動パラメタを指定できます。起動パラメタについては、“NetCOBOL 使用手引書”を参照してください。

3. デバッグするWebアプリケーションが実行されるコンピュータにログオンします。
4. WWWブラウザからWebアプリケーションを起動します。このとき、WWWブラウザはデバッグ

するWebアプリケーションが実行されるコンピュータでなくてもかまいません。

5. Webアプリケーションが起動されたとき、デバグが自動的に起動されます。デバグが起動されたら、[デバグを開始する]ダイアログから、デバグ情報ファイル格納フォルダと必要な情報を指定してデバグを開始します。

8.2.3.2 デバグする

デバグ操作は、通常のデバグを使用したデバグと同じです。デバグの使用方法については、“NetCOBOL 使用手引書”を参照してください。ただし、デバグを行う前に、以下の点に注意する必要があります。

デバグは、デバグを開始したときにWebアプリケーションの入口で実行を自動的に中断し、デバグ操作の機会を与えます。WebアプリケーションがWWWブラウザに応答した後に、次のリクエストによって実行されたときは、自動的に中断しないため、あらかじめ中断点を設定しておく必要があります。

なお、次のリクエストによってロードされるWebアプリケーションで実行を中断させるには、コマンドやダイアログボックスで、ロードされるプログラムの名前を指定して中断点の設定を指示します。詳細については“NetCOBOL 使用手引書”の“19.7.1 動的構造に対するデバグ方法”を参照してください。

WWWブラウザに応答した後は、Webアプリケーションは次のリクエストを待っているためプログラム実行中になっています。中断点の設定などのデバグ操作を行いたい場合は、デバグのメニューから[実行中断]コマンドを選択し、プログラムの実行を中断させてから行ってください。

デバグを行う際には、次の点にも注意をする必要があります。

デバグを使用してデバグを行う場合は、アプリケーションの動作時間が通常の場合よりも長くなります。したがって、WWWブラウザによっては、デバグ中にタイムアウトが発生することがあります。タイムアウトが設定できるWWWブラウザは、適切な値に変更してください。また、NESのタイムアウトの値を適切な値に変更してください。

NESはダウンした場合に、自動的に再起動される仕組みを持っています。デバグを使用してデバグを行う場合に、自動再起動が不要ならば、NESの設定を変更してください。

8.2.3.3 デバグを終了する

デバグを先に終了させる場合

デバグを終了すると、NES自身も終了します。したがって、NESを再起動する必要があります。

Webアプリケーションを先に終了させる場合

NESを停止させます。NESの停止は、WWWブラウザで“Netscape Server Administration”ページから行います。なお、NESを停止させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。

8.2.3.4 Webアプリケーションを入れ替える

Webアプリケーションの入れ替えは、以下の方法で行います。

NESを停止させてから、Webアプリケーションを入れ替えて、再度NESを開始します。

NESの停止と開始は、WWWブラウザで“Netscape Server Administration”ページから行います。なお、NESを停止させる過程では、プロセスが終了するまでWebアプリケーションを実行させなければなりません。

8.3 Webサブルーチンが検出したエラーを参照する

Webサブルーチンが検出したエラーを参照する方法には、プログラム中の“COBW3_INIT”を呼び出す直前で、デバッグモードの設定を行います。

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

この設定により、Webサブルーチン内でエラーが検出された場合には、エラーメッセージがWWWブラウザに表示されます。

また、ヘッダ出力については、COBW3-CONTENT-TYPE-NON以外が設定されていると、COBW3-CONTENT-TYPE-HTMLとみなしてCOBW3_INITがContent-typeを宣言します。

[補足]

デバッグモードの設定は、プログラムを修正し、Webアプリケーションを再作成する必要があります。デバッグモードの設定により、WWWブラウザに表示されるエラーメッセージはログ情報でも参照できます。したがって、アプリケーション変更なしにエラーメッセージを参照したい場合には、ログ情報の利用をおすすめします。



注意

Webアプリケーションの動作コード系がUnicodeの場合、デバッグモードの設定は無効になります。ログ情報を参照するなどの手段によって動作確認してください。

8.4 実行中のデータを表示形式で参照する

実行中のデータを表示形式で参照するには、COBW3_PUT_TEXTを使用します。参照したいデータをCOBW3_PUT_TEXTで指定して、WWWブラウザで任意のデータを参照しながらデバッグする方法です。



注意

COBW3_PUT_TEXTによるデバッグでは、表示形式でないデータ(バイナリデータなど)は表示できません。また、翻訳/実行の繰り返し作業が頻繁に発生するため、効率が悪くなりがちです。したがって、このような場合には実行中のデータを対話型デバッガで参照する方法を使ってデバッグすることをおすすめします。

付録A Q&A集

ここでは、Webサブルーチンの使用時に起こり得る問題点などをQ&A方式でお答えします。

Q1.

使えない文字はあるの？

A1.

一般的にインターネット/イントラネットのデータ通信時には、特殊な文字の使用は制限されています。したがって、業務構築の際には以下に示すデータを取り扱わないように注意してください。

1. 外字

外字を使用すると、WWWブラウザ(クライアント)によっては正しく表示できないことがあるため、使用しないでください。

2. 半角カナ

1バイトカナ(半角カナ)や機種依存文字(丸付き数字、ローマ数字など)は使用しないでください。

インターネット上では使用する機種によりさまざまなコード系が使用されるため、1バイトカナや機種依存文字を持たない機種もあり、正しく文字を表示することができない場合があります。

3. 識別できない文字

EUCとシフトJISで文字コードが同じ値となる文字が存在します。これらの文字はなるべく使用しないでください。とくに各NAMEまたはVALUEがこれらの文字だけで構成されている場合、正しく識別されません。なお、EUCとシフトJISで重なるコードはE0A1～FCFCです。

Q2.

HTML文書のテキスト入力フィールドには、半角と全角文字が混在して入力してもいいの？

A2.

半角文字、全角文字の混在しても問題ありません。ただし、日本語項目ではなく英数字項目として扱ってください。

Q3.

WWWブラウザにエラーメッセージが表示されたのはなぜ？

A3.

以下の項目について確認してください。

WWWサーバの設定が正しいか確認してください。

obj.confで指定したWebアプリケーションを起動するためのMIMEタイプとHTML文書のFORMタグ(ACTION)に書いた拡張子に対応するMIMEタイプが等しいか確認してください。

プログラム名称とHTML文書の格納場所に間違いがないか確認してください。さらに、翻訳オプションが正しいか確認してください。

Q4.

部外者からデータを保護することは必要？

A4.

クライアントとサーバ間の通信において、部外者によってデータを参照または変更される可能性があります。このようなシステムの運用に影響する行為を防止するためにもSSL(Secure Socket Layer)などの適切なネットワークのセキュリティを施すことをおすすめします。

サーバ管理者とご相談ください。

Q5.

Webアプリケーションでファイルの入出力が正しく行われない。

A5.

ファイルに読み込みや書き込みを行う場合は、Webアプリケーションを実行するユーザIDに対して、ファイルとそのファイルが格納されているフォルダに読み込みや書き込みの権限を与えておく必要があります。

Q6.

Webアプリケーションで表示内容を動的に変える手段は？

A6.

1. COBW3_SET_CNV_XXなどを使用する

結果出力用ページの変換部分に//COBOL//変換名//COBOL//と記述しておきます。プログラムで、COBW3_SET_CNV_XXなどを用いて変換名と置き換える文字列(変換文字列)を登録した後、COBW3_PUT_HTMLで変換文字列に置き換えることにより、動的にページを変更することができます。

2. COBW3_PUT_TEXTを使用する

通常、HTML文書などの出力用ファイルに記述する内容を、COBW3_PUT_TEXTに記述できます。したがって、プログラム中で条件ごとに異なるCOBW3_PUT_TEXTを用意して、COBW3_PUT_TEXTの中でデータ名を指定することにより、動的に出力するページを変更することができます。

3. 1つのページを2つ以上のファイルに分割する

出力用のファイルは2つ以上に分割できます。

たとえば、出力するページの前半部分を記述したファイル名を指定してCOBW3_PUT_HTMLを呼び出します。ページの後半部分は条件に合わせて記述したファイルをいくつか用意して、プログラム中の条件によって出力するファイル名を変えて、再度COBW3_PUT_HTMLを呼び出すようにします。

このように、ページの後半を動的に変えることができます。また、ページを部品化することにより、同じようなファイルをいくつも用意する必要がなくなります。

さらに、1. ~ 3.の方法を組み合わせると、さまざまな場面に対応したページを作成することができます。

Q7.

Webアプリケーションを起動しても正しく動作しないのはなぜ？

A7.

Webアプリケーションの先頭でCOBW3_INITを指定しているか確認してください。

その他、COBW3_INITを指定している場合は、“第8章 [動作確認](#)”を参照して、エラー情報の収集やデバッグなどを行い、調査してください。

Q8.

WWWブラウザに次のメッセージが表示されました。どうして？

Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

または、

指定されたWebアプリケーションは、Webヘッダを一部しか返していません。

A8.

Content-typeに誤りがあると考えられます。COBW3_PUT_HEADで指定したCOBW3-CONTENT-TYPEの値を見直してください。

1. HTML文書を出力する場合

COBW3_PUT_HEADを呼び出さない、COBW3-CONTENT-TYPEの値を省略(LOW-VALUE)するまたはCOBW3-CONTENT-TYPE=HTMLを指定してCOBW3_PUT_HEADを呼び出す。

2. テキストデータを出力する場合

COBW3-CONTENT-TYPE=TEXTを指定してCOBW3_PUT_HEADを呼び出す。

Q9.

WWWブラウザにContent-typeの宣言が表示されるのは?

A9.

COBW3_INITでCOBW3-DMODE-DBGが指定されている場合、デバッグ情報としてContent-typeなどのヘッダ情報がWWWブラウザに表示されます。

COBW3-DMODEを指定しなければ、表示されません。

Q10.

Webアプリケーションを実行するとWWWブラウザに“ファイルをディスクに保存しますか”のメッセージが表示されました。どうして?

A10.

WebアプリケーションでContent-typeの宣言に誤りがある場合、このようなメッセージが出力されることがあります。

Content-typeの宣言を正しく行っているか確認してください。

Q11.

Webアプリケーション作成時にCOBOLプログラムで注意することはありますか?

A11.

Webアプリケーションでは、COBOLのほとんどの機能をそのまま使用できますが、画面操作に関する次の機能は利用することができませんので注意してください。

表示ファイル(画面操作機能)

スクリーン機能

小入出力機能(ACCEPT文、DISPLAY文)

(ただし、環境変数、日付けおよび時間の操作機能は利用できます)

なお、Webパラメタの受け取り、参照方法または処理結果の出力方法の詳細については、“第7章 [COBOL Webサブルーチンの使い方](#)”を参照してください。

Q12.

WWWサーバを起動させる時に必要な環境変数の設定はどうしたらいい?

A12.

環境変数の設定は、Windows NT(R)では、「スタート」ボタン 「設定」 「コントロールパネル」 「システム」の「環境」シートで“システム環境変数(S):”に必要な環境変数を追加し、その後システムをリブートします。Windows(R) 2000では、「スタート」ボタン 「設定」 「コントロールパネル」 「システム」の「詳細」シートの「環境変数(E)」ボタン “システム環境変数(S):”に必要な環境変数を追加し、その後システムをリブートします。

または、Webアプリケーション(CGIの場合は.exe、ISAPIまたはSAFの場合は.dll)と同じフォルダに必要な実行環境情報を記述した実行用初期化ファイル(COBOL85.CBR)を配置してください。

SAFサブルーチンを使用したWebアプリケーションに必要な環境変数は、環境設定ファイルに必要な環境変数を追加し、obj.confファイルでCOBOL_Initの設定をしている行のパラメタenvに環境設定ファイル名を指定します。その後NESを再起動します。

Q13.

COBOLデバッガは使えないの?

A13.

Web環境でも、COBOLデバッガでデバッグを行うことはできます。“8.2 [対話型デバッガで動作確認する](#)”を参照してください。

Q14.

動作確認のためにデバッガを起動しても、クライアントにデバッガの画面が表示されません。

A14.

デバッガは、通常、サーバ側の画面に表示されます。よってクライアント側には表示されません。

Q15.

WWWブラウザにWWWサーバからステータスコード(Status-code)が表示されたが意味は?

A15.

“Web連携ガイド”の“付録A Web上のCOBOLアプリケーションをはじめて作成する方のために”のHTTPの基礎を参照してください。

Q16.

タイムアウト時間を制御することはできますか?

A16.

WWWサーバの機能です。WWWサーバのマニュアルを参照してください。

Q17.

フレームを使ったHTML文書をWebアプリケーションで使うことはできますか?

A17.

フレーム機能は、Webアプリケーションで特別な設定や処理を記述する必要はありません。フレーム機能をサポートしたWWWブラウザであれば、HTMLでフレームの記述をするだけで利用できます。

Q18.

COBW3_CHECK_VALUE_Xなどで<INPUT>タグで指定したVALUEの値が正しく検索されないのは?

A18.

INPUTタグでNAMEを省略した場合、WWWブラウザによってはWebパラメタにそのINPUTタグのVALUEの値が入りません。VALUEの値が必要となるINPUTタグには必ずNAMEを指定してください。

Q19.

COBW3_GET_VALUE_XXやCOBW3_CHECK_VALUE_XなどでWebパラメタが正しく検索されないのは?

A19.

Webパラメタに含まれる検索対象の文字列の長さが定量制限を超えた場合、SAFサブルーチンが文字列を定量制限の範囲内で処理を行います。このため、期待した結果が得られないことがあります。

Q20.

絶対パスで指定したファイルがアクセスできないのは?

A20.

Webアプリケーションの実行時に使用するCOBOLファイルの格納先の指定は、使用するサーバマシンのドライブ構成に従って指定するように注意してください。

Q21.

WWWブラウザに「文書にデータが含まれません。」のエラーが表示されたのはなぜ?

A21.

Webアプリケーションから出力する結果出力用ページの<BODY>タグの前では、必ず1行改行してください。

例:

```
<HTML>
<HEAD>
...
</HEAD>

<BODY>
...
</BODY>
</HTML>
```


Q22.

COBOLデバッガは起動されたが、デバッグが開始されません。

A22.

[デバッグを開始する]ダイアログの[デバッグ情報]ページで、デバッグ情報ファイル格納フォルダが正しく設定されているか確認してください。

Q23.

サーバに「F3BIPRCT.DLLが見つかりません。」のエラーが表示されるのはなぜ?

A23.

COBOLランタイムシステムが、正しくインストールされているか確認してください。Webアプリケーションを動作させる場合は、ユーザ環境変数の指定は有効になりません。
あらかじめ、サーバマシンのシステム環境変数のPATHに、COBOLランタイムシステムのインストールフォルダを設定しておいてください。このとき、システム環境変数の設定を有効にするために、システムを再起動する必要があります。

Q24.

ネットワーク環境の資源はアクセスできないの?

A24.

Webアプリケーションの実行時には、ユーザがログインしたときのネットワークドライブの構成は有効になりません。
実行時に使用するネットワーク環境の資源は、UNC指定でアクセスしてください。詳細については、“NetCOBOL 使用手引書”の“23.1.3 サービス配下で動作するプログラム”を参照してください。

Q25.

サーバから応答がない場合は?

A25.

サーバ上でウィンドウまたはメッセージボックスの入力待ち状態になっている場合があります。COBOLプログラムをWebサーバで実行する場合、以下の環境変数情報を実行用の初期化ファイルに記述するか、もしくはシステム環境変数に設定してください。
これら環境変数情報の指定により、ウィンドウまたはメッセージボックス表示が行われなくなるため、オペレータの入力待ちの状態を回避できます。

環境変数情報	意味
@MessOutFile=ファイル名	COBOLの実行時メッセージを指定されたファイルに出力する。
@WinCloseMsg=OFF	ウィンドウを閉じるときのメッセージを表示しない。

各環境変数情報の詳細については、“NetCOBOL 使用手引書”を参照してください。
また、ウィンドウの表示を前提としている以下の機能をCOBOLプログラムから使用しないでください。

表示ファイル(画面操作機能)

スクリーン操作機能

小入出力機能(ACCEPT文、DISPLAY文)

(ただし、環境変数、日付けおよび時間の操作機能は利用できます)

詳細については、“NetCOBOL 使用手引書”の“23.1.3サービス配下で動作するプログラム”を参照してください。

このほかに、NESの設定やODBCの設定(ODBCを使用している場合)に、不備がある場合もあります。

これらについても確認してください。

Q26.

COBOLデバッガが起動されないのはなぜ？

A26.

以下の点を確認してください。

環境変数情報@CBR_ATTACH_TOOL=TESTが設定されていること

COBOL Tool Attaching Serviceが開始されていること

Q27.

Cookieデータに漢字を使ってもいいの？

A27.

Cookieデータに漢字を使用することは可能ですが、Cookieデータは他言語のアプリケーションも参照することを考慮してASCII文字を使用することをおすすめします。

Q28.

セッション管理機能を使用したWebアプリケーションを二重に起動すると、正しく動作しないことがある。

A28.

セッション管理機能を使用したWebアプリケーションは、二重に起動された場合、動作不定になります。これを回避するためには、WWWブラウザでJavaScriptなどを利用して二重起動を行わないようにしてください。WWWブラウザを操作する上での一般的な注意点については、“Web連携ガイド 付録A.5”を参照してください。

Q29.

クライアント(WWWブラウザ)側が何らかの理由で強制終了した場合、サーバ(WWWサーバ)側のWebアプリケーションの動きはどうなってしまうの？

A29.

クライアント(WWWブラウザ)が強制終了される原因として、以下の場合が考えられます。

WWWブラウザを終了してしまう。

ブラウザでタイムアウトが働いてしまう。

通常、HTTPではWWWブラウザからのリクエストによってWWWサーバ内のWebアプリケーションプロセスとの間にセッションが張られ、これに対するレスポンスの送信が完了すると、そのセッションは直ちに切断されます(一問一答型)。このため、WWWサーバで動作するWebアプリケーションは、基本的に一回のHTTPセッション内でファイルのOPEN READ/WRITE CLOSEなど、すべての処理を完結するよう構成されていなければなりません。

このようなWebアプリケーションに対して、前述のような強制終了が行われたとしても、WWWブラウザとWWWサーバ間のセッションが中断するだけで、それ自体はWebアプリケーションには直接影響を与えません。したがって、動作中のWebアプリケーションはそのコードの最後まで実行されます。ただし、HTTPのセッションが切断された後でCOBW3_PUT_HTMLやCOBW3_PUT_TEXT等によってレスポンスを送信するとエラーが通知されます。

なお、Webサブルーチンのセッション管理機能を使用して、複数のリクエストにまたがる連続した処理を行う場合には、タイムアウト時の異常終了手続きを記述して資源の回収処理を行うよう考慮が必要です。たとえば、初回リクエスト時にファイルのOPENを行い、最終リクエスト時にファイルのCLOSEを行うことを想定した場合、最終リクエストに到達する前にWWWブラウザが強制されてしまうと、ファイルが開かれたままとなり、ファイル破壊の危険性が生じます。このため、タイムアウト時の異常終了処理手続きを作成し、その中でファイルをCLOSEするような考慮が必要です。

Q30.

バッチファイルからCGIアプリケーションを起動する場合、指定したコマンドをWWWブラウザに表示させないためには？

A30.

バッチファイルの先頭に、@ECHO OFFを指定してください。

Q31.

SAFサブルーチンで正しくない値が取得されるのは?

A31.

リクエストのURLに拡張パスが指定されている場合、以下の項目については正しくない値が取得されることがあります。

COBW3_GET_REQUEST_INFOで以下の条件名が真の場合、

COBW3-URL : リクエストのURL

COBW3-URI : リクエストのURI

COBW3-MIMETYPE : リクエストのMIMEタイプ

COBW3-VIRTUALPATH : リクエストの仮想パス

COBW3-PHYSICALPATH: リクエストの仮想パスに対する物理パス

および、

COBW3_SAF_GET_BASENAMEで取得する値

Q32.

WWWサーバがNESの場合、obj.confファイルを修正する前にバックアップをとっておいた方がいい?

A32.

obj.confファイルを修正することでNESが起動できなくなる場合もあるため、バックアップをとることをおすすめします。

Q33.

ログファイルと重要度を指定しているのに、ログファイルが出力されない。

A33.

以下の項目について確認してください。

ファイル名が正しいか

サーバの実行権限とフォルダのアクセス権限に不整合がないか

ディスクがいっぱいでないか

環境変数名が正しいか

付録B エラー処理

Webサブルーチン実行時に問題が検出された場合には、Webサブルーチンがエラー処理を行います。
出力するメッセージの形式は、次のとおりです。

COB-メッセージ番号：COBW3：メッセージ本文

COB-メッセージ番号：メッセージの通し番号を表示します。

メッセージ本文：エラーの内容を表示します。

また、Webサブルーチンのログの設定を行っておくと、同様の内容がログに出力されます。
以下に、出力されるメッセージ番号およびメッセージ本文の一覧を示します。

メッセ ジ番号	メッセージ本文
00100	COBW3-CONTEXTに値が設定されていないため、処理が続行できませんでした。
01100	何らかの理由により作業領域、オブジェクトが破壊されたと考えられます。領域の使い方およびWebサブルーチンのオブジェクトを点検してください。
01500	呼出し元のCOBOLアプリケーション動作コード系ではWebサブルーチンは正しく動作しません。
01501	呼出し元のアプリケーションの動作状態に誤りがあります。Webサブルーチンは正しく動作しません。
02050	検索文字列の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
02051	検索文字列の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
03000	POST,GET以外でWebアプリケーションが呼び出されました。
03001	COBW3_INITが2回以上呼び出されました。2回目の呼び出しの前には、必ずCOBW3_FREEを呼び出してください。
03002	Webサブルーチンの作業領域を獲得できませんでした。
03003	GETで渡されたデータの長さが制限を超えました。最大長までを有効とします。
03020	POSTまたはGETで渡されたデータの獲得に失敗しました。
03021	POSTまたはGETで渡されたデータのサイズが制限を超えているため、データを獲得できませんでした。
03030	Webサブルーチンの作業領域の獲得に失敗しました。
03040	アップロードファイル情報を取得できませんでした。
03041	ヘッダ情報の取得に失敗したため、Webパラメタを処理することができません。
03042	POST以外のメソッドに対して、FORMタグのENCTYPEにmultipart/form-dataが指定されているため、Webパラメタおよびアップロードファイル情報を取得できませんでした。
03065	環境変数@CBR_WEB_OUT_CODEの設定に誤りがあります。変換コードの指定は行われなかったものとみなします。
03200	UTF-8からUCS-2への変換に失敗しました。
03201	UCS-2からUTF-8への変換に失敗しました。
03202	SJISからUCS-2への変換に失敗しました。
03203	SJISからUTF-8への変換に失敗しました。

03700	文字列として識別できない文字が指定されました。そのまま処理を行います。
03701	EUC/SJIS混在の文字列が存在しました。そのまま処理を行います。
03850	Cookieの取得に失敗しました。
03900	日本語コード変換時にコード変換エラーが発生しました。
03901	日本語コード変換時にコード変換 (iconv_close) でエラーが発生しました。
03902	日本語コード変換ライブラリがロードできません。
03903	日本語コード変換関数 (iconv) が呼び出せません。
04000	Webパラメタがありません。
04001	Webパラメタ長が0です。
04006	検索対象NAME/VALUEの出現回数値に0または負の値が指定されました。1が指定されたものとみなして処理を続行します。
04008	VALUEデータの長さが制限を超えました。最大長までを有効とします。
04400	変換名が指定されていません。変換名を確認してください。
04401	変換名の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
04402	変換名の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
04450	指定された変換情報は登録されていないため、削除できませんでした。
04470	指定された変換情報は登録されていないため、変更できませんでした。
04480	指定された変換情報の削除に失敗しました。
05000	システムコマンドの実行に失敗しました。
05001	システムコマンドが指定されていません。
05500	ファイル名が指定されていません。ファイル名を確認してください。
05501	アップロードファイル情報がありません。
05510	アップロードファイルのクライアントでのパス名の取得に失敗しました。
05511	アップロードファイルのクライアントでのパス名の文字列長が制限を超えました。最大長までを有効にします。
05520	アップロードファイルのクライアントでのファイル名の取得に失敗しました。
05521	アップロードファイルのクライアントでのファイル名の文字列長が制限を超えました。最大長までを有効にします。
05530	アップロードファイルのコンテンツタイプの取得に失敗しました。
05531	アップロードファイルのコンテンツタイプの文字列長が制限を超えました。最大長までを有効にします。
05550	指定されたファイル名はすでに存在しているため、ファイルは生成できませんでした。
05551	ファイルの生成に失敗しました。
05552	サブルーチン内部でのデータ生成に失敗しました。生成したファイルをCOBW3_DEL_UPLOADFILEで削除することはできません。
05580	指定されたファイル名は同一スレッド内で生成されたファイルではありません。ファイルの削除は行いませんでした。
05581	指定されたファイルの削除に失敗しました。
06100	指定されたHTML文書ファイルのOPENでエラーが発生しました。
06101	指定されたHTML文書ファイルのREADで入出力エラーが発生しました。
06102	指定されたHTML文書ファイルのサイズは0です。
06105	HTML文書中の変換名に対応する変換情報が存在しません。変換データを登録してください。

06106	HTML文書の出力作業領域が不足したため出力に失敗しました。変換名に対応する変換データを確認してください。
06107	HTML文書中の変換名の指定形式に誤りがあるため、出力に失敗しました。
06108	指定された変換名の文字列長が長すぎます。変換名を確認してください。
06120	HTML文書をWWWサーバへ出力するのに失敗しました。
06130	HTML文書中の一行の文字数が制限を超えました。
06140	Webアプリケーションの動作コード系がSJISの場合、扱えるHTML文書ファイルの文字コードはSJISのみです。
06300	変換名が指定されていません。変換名を確認してください。
06301	変換名の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
06302	変換名の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
06303	変換文字列が指定されていません。変換文字列を設定してください。
06304	変換文字列の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
06305	変換文字列の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
06310	指定された変換名は既に登録されています。既に登録されている変換情報を有効とします。
06320	変換情報の登録作業領域が不足したため登録に失敗しました。変換情報を初期化した後、再度登録を行ってください。
06330	変換情報の削除に失敗しました。
06502	ブラウザへのデータ出力中に予期せぬエラーが発生しました。
06503	出力する文字列長に負の値が設定されています。0が設定されたものとみなして処理を続けます。
06504	出力する文字列長に制限を超える値が指定されています。最大長が指定されたものとみなして処理を続行します。
06700	既にCOBW3_PUT_HTMLまたはCOBW3_PUT_TEXTが呼び出されています。ヘッダの出力要求は無効です。
06702	COBW3-PUT-HEAD中にコロン(:)がありません。COBW3-PUT-HEADの指定は無視されます。
06704	COBW3-PUT-HEAD-LENGTHに負の値が指定されました。0が指定されたものとみなして処理を続行します。
06706	COBW3-PUT-HEAD-LENGTHに制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
06708	ヘッダとして出力する対象が指定されていません。
06710	既に出力されている値と異なる値がCOBW3-CONTENT-TYPEに指定されました。
06714	既に出力されている値と異なる値がCOBW3-STATUS-CODEに指定されました。
06718	既に指定されている値と異なる値がCOBW3-CONTENT-LENGTHに指定されました。
06730	Content-typeの宣言で予期せぬエラーが発生しました。
06734	Status-codeの宣言で予期せぬエラーが発生しました。
06738	Content-lengthの宣言で予期せぬエラーが発生しました。
06742	ヘッダの出力に失敗しました。
06746	ヘッダの出力を完了できませんでした。
07010	ユーザ名が取得できませんでした。
07011	ユーザ名の文字列長が制限を超えました。最大長までを有効にします。

07020	パスワードが取得できませんでした。
07021	パスワードの文字列長が制限を超えました。最大長までを有効にします。
07030	IPアドレスが取得できませんでした。
07031	IPアドレスの文字列長が制限を超えました。最大長までを有効にします。
07040	認証タイプが取得できませんでした。
07041	認証タイプの文字列長が制限を超えました。最大長までを有効にします。
07500	セッションは開始されていません。
07510	既にCOBW3_PUT_HTMLまたはCOBW3_PUT_TEXTが呼び出されているため、セッションは開始できませんでした。
07520	既にCOBW3_PUT_HTMLまたはCOBW3_PUT_TEXTが呼び出されているため、セッションは終了できませんでした。
07558	タイムアウト監視が開始できませんでした。
07562	何らかの理由でタイムアウト監視に失敗しました。
07566	タイムアウト監視の中断に失敗しました。
07570	何らかの理由でタイムアウト監視の準備が出来ませんでした。セッションは開始できませんでした。
07580	タイムアウト処理が中断できませんでした。
07584	Cookieデータから取得したセッションIDが正しくありません。
07588	セッション情報の管理域の獲得に失敗しました。
07590	タイムアウト監視の実行に失敗しました。
07600	既にセッションは開始されています。
07602	タイムアウト時間に0が指定されたため、セッションは開始できませんでした。
07604	タイムアウト時間に制限を超える値が指定されました。最大時間が指定されたものとみなして処理を続行します。
07606	セッションデータタイプに無効な値が指定されたため、セッションは開始できませんでした。
07608	セッション用のCookieデータの作成に失敗したため、セッションは開始できませんでした。
07610	セッションデータの大きさに0が指定されたため、セッションデータの登録は行なえませんでした。
07614	セッションデータの大きさに制限を超える値が指定されました。最大値が指定されたものとみなして処理を続行します。
07618	セッションデータとして指定されたオブジェクトが正しくありません。
07650	指定されたセッションデータの大きさと登録されているセッションデータの大きさが異なるため、セッションデータの取得は行なえませんでした。
07660	セッションデータにnull以外の値が設定されているため、セッションデータの取得を行なえませんでした。
07670	セッションデータが登録されていません。
07680	タイムアウト時間に0が指定されたため、タイムアウト時間は変更できませんでした。
08004	COBW3-HEADER-NAME-LENGTHに負の値が指定されました。0が指定されたものとみなして処理を続行します。
08005	script-nameが取得できませんでした。
08006	COBW3-HEADER-NAME-LENGTHに制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
08008	COBW3-HEADER-NAMEが指定されていません。
08010	HTTPヘッダの取得に失敗しました。
08020	COBW3-HEADER-VALUEの長さが制限を超えました。最大長までを有効にします。

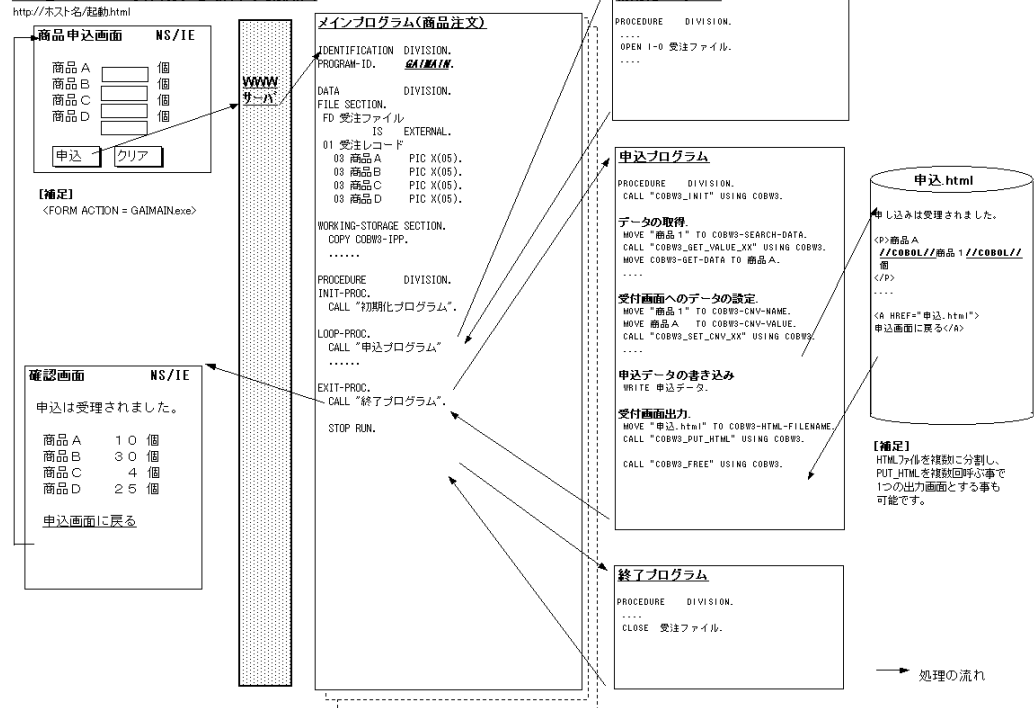
08024	COBW3-HEADER-NAMEに指定されたHTTPヘッダは見つかりませんでした。
08030	script-nameの文字列長が制限を超えました。最大長までを有効とします。
08101	キーワードが指定されていません。キーワードを設定してください。
08105	インタフェースファイル中に指定されたキーワードが見つかりませんでした。
08108	インタフェースファイル名が取得できませんでした。
08110	インタフェースファイルのOPENでエラーが発生しました。
08114	インタフェースファイルのREADでエラーが発生しました。
08118	インタフェースファイルのCLOSEでエラーが発生しました。
08130	キーワードに対応する値の長さが制限を超えました。値の最大長までを獲得します。
08150	Webサブルーチン内部でのデータ入出力でエラーが発生しました。
08500	既に COBW3_PUT_HTML または COBW3_PUT_TEXT が呼び出されています。Cookieに対する操作は無効です。
08510	Cookie名が指定されていません。Cookie名を設定してください。
08512	Cookie名の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
08514	Cookie名の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
08520	Cookie値が指定されていません。Cookie値を設定してください。
08522	Cookie値の文字列長に負の値が指定されました。0が指定されたものとみなして処理を続行します。
08524	Cookie値の文字列長に制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
08530	指定されたCookieデータは登録されていないため、変更できませんでした。
08532	指定されたCookie名は既に登録されています。既に登録されているCookieデータを有効とします。
08540	Cookieの有効期限として1582年10月15日以前の日付は無効です。有効期限は指定されなかったものとみなします。
08541	Cookieの有効期限における年の指定が正しくありません。有効期限は指定されなかったものとみなします。
08542	Cookieの有効期限における月の指定が正しくありません。有効期限は指定されなかったものとみなします。
08543	Cookieの有効期限における日の指定が正しくありません。有効期限は指定されなかったものとみなします。
08544	Cookieの有効期限における時の指定が正しくありません。有効期限は指定されなかったものとみなします。
08545	Cookieの有効期限における分の指定が正しくありません。有効期限は指定されなかったものとみなします。
08546	Cookieの有効期限における秒の指定が正しくありません。有効期限は指定されなかったものとみなします。
08550	Cookieのセキュリティに無効な値が指定されました。セキュリティは指定されなかったものとみなします。
08560	Cookieの登録種別に無効な値が指定されました。登録種別は指定されなかったものとみなします。
08570	Cookieデータの登録作業領域が不足したため登録に失敗しました。Cookie情報を初期化した後、再度登録を行ってください。
08580	Cookie情報のエンコードに失敗しました。

08610	指定されたCookieデータは登録されていないため、削除できませんでした。
08620	指定されたCookieデータの削除に失敗しました。
08710	Cookie情報の初期化種別に無効な値が指定されました。初期化種別は指定されなかったものとみなします。
08720	Cookie情報の登録作業域が不足したため、リクエストデータのCookie情報による初期化に失敗しました。
08810	Cookie値の長さが制限を超えました。最大長までを有効とします。
08820	クライアントからCookieデータが送信されませんでした。
09001	COBW3-REQUEST-INFO-TYPEに正しい値が指定されていません。
09010	リクエストの取得に失敗しました。
09020	COBW3-REQUEST-INFOの長さが制限を超えました。最大長までを有効にします。
09500	COBW3_CNV_SETはUnicode環境では使用できません。
09501	COBW3_CNV_DELはUnicode環境では使用できません。
09502	COBW3_NAMEはUnicode環境では使用できません。
09503	COBW3_NAMEN はUnicode環境では使用できません。
09504	COBW3_VALUE はUnicode環境では使用できません。
09505	COBW3_SET_COOKIEはUnicode環境では使用できません。
09506	COBW3_DEL_COOKIEはUnicode環境では使用できません。
09507	COBW3_GET_COOKIEはUnicode環境では使用できません。
09508	COBW3_GET_COOKIEEN はUnicode環境では使用できません。
20000	COBOLアプリケーションのロードに失敗しました。
20001	設定された環境設定ファイルが存在しません。
20002	実行するCOBOLアプリケーションが指定されていません。
20003	実行しようとしたCOBOLアプリケーション処理の開始アドレスの取得に失敗しました。
20004	ディレクタで原因不明のエラーが発生しました。
20005	ディレクタの環境設定処理が異常終了しました。
20006	メモリの割り当てに失敗しました。
20007	環境設定の記述に誤りがあります。
20008	環境設定のファイル名に誤りがあります。
20009	開放するCOBOLアプリケーションが指定されていません。
21000	COBW3-SAF-CONTEXTが指定されていません。
21100	COBW3-SAF-PARM-NAMEまたはCOBW3-SAF-PARM-NAMENに文字列が指定されていません。
21102	COBW3-SAF-PARM-NAME-LENGTHに負の値が設定されました。0が指定されたものとみなして処理を続行します。
21104	COBW3-SAF-PARM-NAME-LENGTHに制限を超える値が指定されました。最大長が指定されたものとみなして処理を続行します。
21112	COBW3-SAF-PARM-VALUEの文字列長が制限を超えました。最大長までを有効にします。
21210	COBW3-SAF-BASENAMEの取得に失敗しました。
21212	COBW3-SAF-BASENAMEの文字列長が制限を超えました。最大長までを有効にします。
99999	Can't open message catalog.

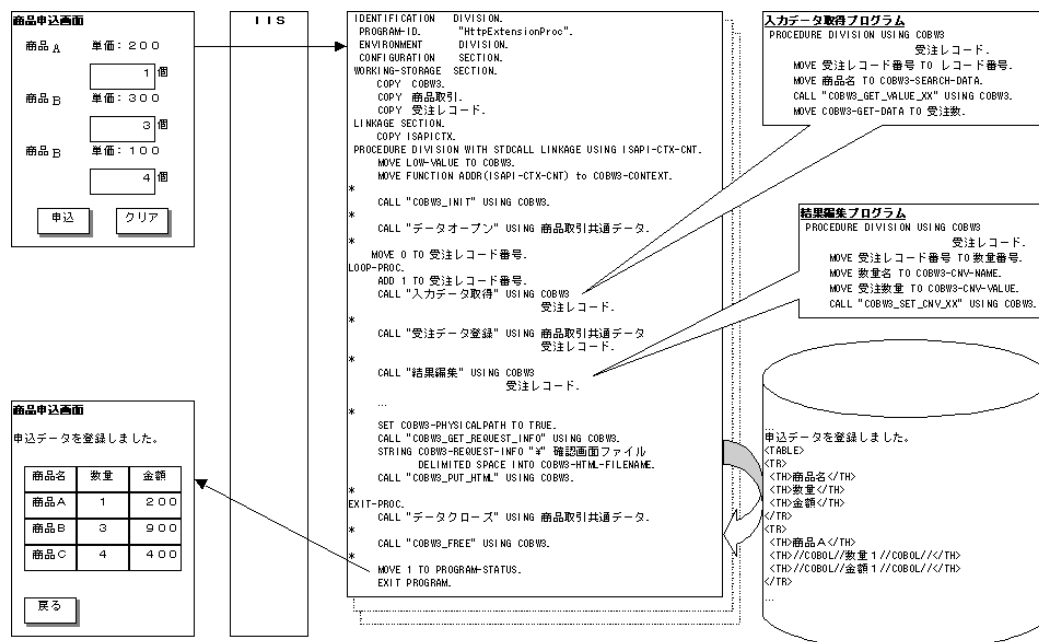
付録C WebアプリケーションをCOBOLで作成する場合の概念図

C.1 CGIアプリケーションをCOBOLで作成する場合の概念図

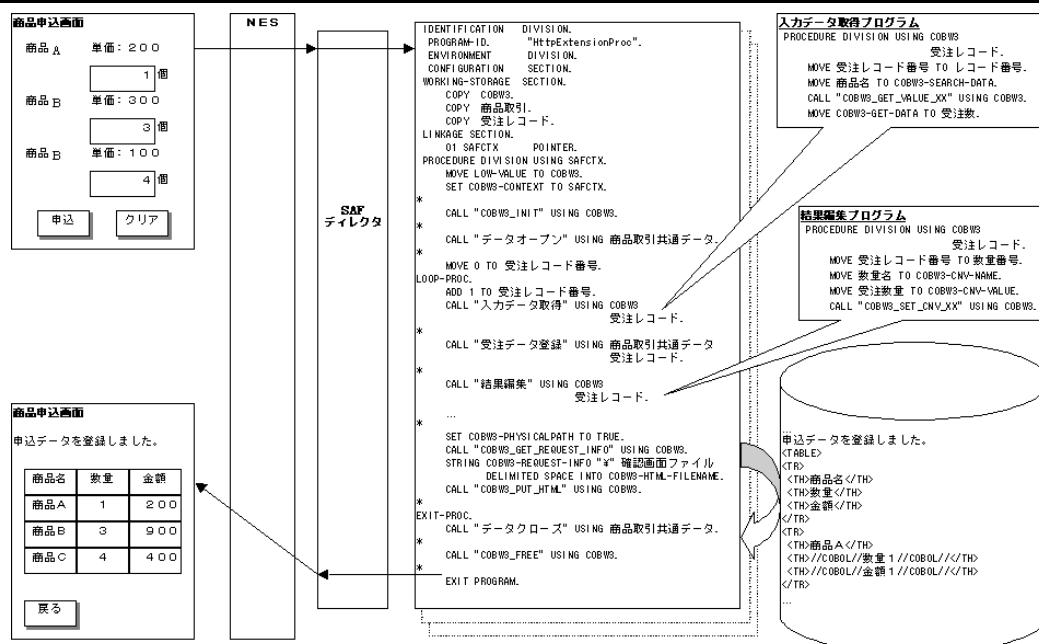
CGIをCOBOLで作成する場合の概念図



C.2 ISAをCOBOLで作成する場合の概念図



C.3 SAFをCOBOLで作成する場合の概念図



付録D CGIからISAPIへの移行の手引き

D.1 CGIからISAPIへの移行

COBOL ISAPIサブルーチンはCOBOL CGIサブルーチンに対して基本的にソースの上位互換を保証しています。このため、COBOL CGIサブルーチンで作成したWebアプリケーションからCOBOL ISAPIサブルーチンを使用したWebアプリケーションへ容易に移行できます。ただし、ISAPI固有の機能などがあるため、若干のプログラム修正や、翻訳、リンク方法の変更が必要です。以下に移行の際の注意点などについて記述します。

はじめに、COBOL CGIサブルーチンで作成したWebアプリケーション(以後、CGIアプリと呼びます)とCOBOL ISAPIサブルーチンで作成したWebアプリケーション(以後、ISAPIアプリと呼びます)の違いについて下表に記します。

項番	項目	CGI アプリ	ISAPI アプリ
1	アプリケーションの形式	主プログラム (EXE)	副プログラム (DLL)
2	実行単位	プロセス	スレッド
3	モジュール構成	自由	GetExtensionVersion HttpExtensionProc TerminateExtension をエクスポート関数として用意する それ以外は自由
4	サーバとのインタフェース領域	不要	必須
5	復帰コード	不要または自由	必須であり、値の意味が決まっている
6	環境変数の操作	任意	制限あり
7	CGI 環境変数操作	提供サブルーチンまたは、DISPLAY/ACCEPT文で操作可能	提供サブルーチンを使用することで操作可能

CGIからISAPIへ移行するには、上記相違点を修正する必要があります。個々の内容については2章以降で説明します。

また、以下のサブルーチンについては、互換のためCOBOL ISAPIサブルーチンでも提供しますが、今後、WebアプリケーションがシフトJISコード系で動作する環境でしか使用できません。ISAPIへの移行を機に代替サブルーチンへの修正をおすすめします。

サブルーチン名	代替サブルーチン
COBW3_NAME	COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX
COBW3_NAMEN	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
COBW3_SET_COOKIE	COBW3_SET_COOKIE_XX COBW3_SET_COOKIE_NX COBW3_SET_COOKIE_XN COBW3_SET_COOKIE_NN
COBW3_DEL_COOKIE	COBW3_DEL_COOKIE_X COBW3_DEL_COOKIE_N
COBW3_GET_COOKIE	COBW3_GET_COOKIE_XX COBW3_GET_COOKIE_NX
COBW3_GET_COOKIEEN	COBW3_GET_COOKIE_XN COBW3_GET_COOKIE_NN

D.2 アプリケーションの形式

CGIとISAPIとで、アプリケーションの形式が異なるため、次の項目に関する変更が必要となります。

呼出し用ページ中の起動するWebアプリケーションの記述
翻訳、リンク方法

D.2.1 呼出し用ページ

呼出し用ページに記述されているWebアプリケーションの拡張子をEXEからDLLに変更するか、場合によってはアプリケーション名そのものを変更してください。

たとえば、CGIアプリをCGI.EXE、ISAPIアプリをISAPI.DLLとすると、以下のような修正を行う必要があります。

CGI アプリの呼出し用ページ

```
<FORM ACTION="/sample/CGI.EXE">
```

```
...
```

```
</FORM>
```

ISAPI アプリ用に修正後の呼出し用ページ

```
<FORM ACTION="/sample/ISAPI.DLL">
```

```
...
```

```
</FORM>
```

D.2.2 翻訳、リンク方法について

翻訳

CGIアプリの場合、プログラムを翻訳するにあたって特に必要なオプションはなく、強いて言うなら、主プログラムには翻訳オプション "MAIN" を指定するだけでした。しかし、ISAPIアプリはDLLで作成する必要があるため、翻訳オプション "MAIN" を指定してはなりません。また、ISAPI

アプリでは、次の翻訳オプションが必須となります。翻訳オプションの詳細については、“NetCOBOL 使用手引書”を参照してください。

ALPHAL(WORD)またはNOALPHAL
THREAD(MULTI)

翻訳オプション “ALPHAL(WORD)” または “NOALPHAL” は “D.4 [モジュール構成](#)” で説明する3つの入口名を持つプログラムに対して、必ず指定してください。これは入口名の大文字/小文字を区別するためです。また、ISAPIアプリはマルチスレッドで動作するアプリケーションであるため、すべてのソースプログラムに翻訳オプションTHREAD(MULTI)を指定してください。

リンク

CGIアプリでは、オブジェクトファイルをリンクしてEXEファイルを作成します。たとえば、次のように行います。

```
-----
LINK   メインプログラム.obj  初期プログラム.obj  業務プログラム.obj
        終了プログラム.obj F3BICIMP.lib LIBC.lib F3BICWSR.lib /OUT:
        実行形式名.exe
-----
```

ISAPIアプリはDLLとして作成しなければなりません。また、CGIアプリとはリンクするインポートライブラリが異なります。たとえば、次のようになります。

```
-----
LINK   /DLL  初期プログラム.obj  業務プログラム.obj  終了プログラム.obj
        F3BICBDM.obj F3BISAPI.lib F3BICIMP.lib KERNEL32.lib LIBC.LIB
        /OUT:実行形式名.dll /DEF:モジュール定義.def /ENTRY:COBDMMAIN
-----
```

なお、モジュール定義ファイル(モジュール定義.def)には、“D.4 [モジュール構成](#)” で述べる3つのエクスポート関数を記述します。具体的には次のようになります。

<pre>LIBRARY 実行形式名 EXPORTS GetExtensionVersion HttpExtensionProc TerminateExtension</pre>

D.3 実行単位

CGIアプリからISAPIアプリへの移行は、プロセスベースのアプリケーションからスレッドベースのアプリケーションへの移行になります。このとき以下の点に注意する必要があります。

翻訳方法
共有資源へのアクセス

D.3.1 翻訳方法について

マルチスレッドオブジェクトを生成するための翻訳オプションTHREAD(MULTI)を指定する必要があります。

D.3.2 共有資源へのアクセス

CGIでは、CGIアプリ自身のプロセス内で1つの実行スレッド(シングルスレッド)として動作するため、同一プロセス内の複数のスレッド間で共有される資源は存在しませんでした。

ISAPIでは、IIS自身のプロセス内で複数のスレッド(マルチスレッド)としてISAPIアプリが動作するため、複数のスレッド間で資源を共有できます。

ISAPIアプリで、スレッド間共有資源を使用する場合、共有資源に対する競合状態が発生しないよう同期制御を行う必要があります。共有資源に対する同期制御は、その資源によってCOBOLランタイムシステムが自動的に制御するものと、ISAPIアプリ自身がCOBOLが提供する " スレッド同期制御サブルーチン " を使用して制御するものがあります。

詳細は、" NetCOBOL 使用手引書 " の " 第24章 マルチスレッド " の以下の節を参照してください。

" 24.4 スレッド間の資源の共有 "

" 24.9 スレッド同期制御サブルーチン "

D.4 モジュール構成

CGIアプリのモジュール構成は、主プログラムを必要とする以外は特に制限がありません。しかし、ISAPIアプリは次の3つのモジュールを定義しておく必要があります。

GetExtensionVersion

HttpExtensionProc

TerminateExtension

この中で、CGIアプリの主プログラムに対応するものがHttpExtensionProcです。

GetExtensionVersion、TerminateExtensionが対応するものではありません。GetExtensionVersionは初期化処理、TerminateExtensionは終了処理のための関数です。

GetExtensionVersion、HttpExtensionProc、TerminateExtensionの詳細については、" 3.2.1 [GetExtensionVersion](#) "、" 3.2.2 [HttpExtensionProc](#) " および " 3.2.3 [TerminateExtension](#) " を参照してください。また、主プログラムのプログラム名をHttpExtensionProcにする必要があります。たとえば、次のように修正してください。

CGI アプリの内容

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID.      MAINPROG.
ENVIRONMENT      DIVISION.
-----
```

ISAPI アプリ用に修正した内容

```
-----
IDENTIFICATION DIVISION.
PROGRAM-ID.      "HttpExtensionProc".
ENVIRONMENT      DIVISION.
-----
```

D.5 サーバとのインタフェース領域

CGIアプリではWWWサーバとのインタフェース(環境変数は除く)は特に意識する必要はありませんでした。しかし、ISAPIアプリのHttpExtensionProcは関数としてIISから呼ばれ、IISとのデータのやりとりを行うためのインタフェース(パラメタ)領域を受け取ります。このため、CGIアプリからISAPIアプリに移行する際は、連絡節と手続き部を修正する必要があります。また、この

インタフェース領域をCOBOL ISAPIサブルーチンで使用するため、COBOL ISAPIサブルーチンとのインタフェースであるCOBW3にそのアドレスを設定する必要があります。たとえば、次のように修正してください。

CGI アプリの内容

```
-----
WORKING-STORAGE SECTION.
    COPY COBW3.
PROCEDURE DIVISION.
    MOVE LOW-VALUE TO COBW3.
-----
```

ISAPI アプリ用に修正した内容

```
-----
WORKING-STORAGE SECTION.
    COPY COBW3.
LINKAGE SECTION.
    COPY ISAPICTX.
PROCEDURE DIVISION WITH STDCLL USING ISAPI-CTX-CNT.
    MOVE LOW-VALUE TO COBW3.
    MOVE FUNCTION ADDR(ISAPI-CTX-CNT) TO COBW3-CONTEXT.
-----
```

D.6 復帰コード

CGIアプリの主プログラムでは、復帰コードの使用は任意です。しかし、“[D.5 サーバとのインタフェース領域](#)”でも述べたように、ISAPIアプリのHttpExtensionProcは関数として呼び出され、IISはこの関数からの戻り値を期待しています。このため、ISAPIアプリでは、PROGRAM-STATUSに必ず復帰コードを設定する必要があります。たとえば、次のように修正してください。

CGI アプリの内容

```
-----
END-PROC.
STOP RUN.
-----
```

ISAPI アプリ用に修正した内容

```
-----
END-PROC.
    MOVE 1 TO PROGRAM-STATUS.
    EXIT PROGRAM.
-----
```

設定できる復帰コードは次の2種類です。通常は正常終了を設定します。また、エラー終了を設定すると、サーバは該当アプリケーションでエラーが起きたと判断し、その旨のメッセージをWWWブラウザに送信します。

復帰コード	意味
1	正常終了
4	エラー終了

D.7 環境変数操作

環境変数ブロックは、プロセスごとに1つ割り当てられるため、プロセスとして動作するCGIアプリでは、Webアプリケーションから環境変数を操作してもシステム上、何の問題ありませんでした。しかし、ISAPIアプリはIIS自身のプロセスで複数のスレッドとして動作するため、環境変数の内容を変更するとIISやほかのアプリケーションに影響を及ぼしかねません。また、環境変数の内容を参照する場合も、内容が不変であることは保証されないため、アプリケーションの動作が意図したものと異なる場合があります。したがって、CGIアプリで環境変数操作をしている場合は、環境変数を使用しないように変更する必要があります。

また、アプリケーションが呼び出されたときのカレントフォルダも固定である保証はないので、カレントフォルダを暗に使用する機能は注意が必要です。たとえば、カレントフォルダからの相対パス指定によるファイル操作が該当します。これに代わる手段として、次の方法があります。

絶対パスでファイルを指定する。

CGI アプリの内容

```
-----
MOVE "RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

ISAPI アプリ用に修正した内容

```
-----
MOVE "C:\%COBOL%\ISAPI\%RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

仮想ディレクトリに対応する物理パスを利用してファイル位置を指定する。

仮想ディレクトリに対応する物理パスを取得するには、COBW3_GET_REQUEST_INFOを使用します。

CGI アプリの内容

```
-----
MOVE "RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

ISAPI アプリ用に修正した内容

```
-----
SET COBW3-PHYSICALPATH TO TRUE.
CALL "COBW3_GET_REQUEST_INFO" USING COBW3.
MOVE COBW3-REQUEST-INFO TO COBW3-HTML-FILENAME.
MOVE "%RESPONSE.HTM" TO
      COBW3-HTML-FILENAME(COBW3-REQUEST-INFO-LENGTH + 1:13).
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

これに関連した内容として、実行用初期化ファイル(COBOL85.CBR)内のファイルの指定方法があります。CGIアプリでは絶対パス、相対パスの双方が指定可能ですが、ISAPIアプリでは相対パスは使用できません。また、代替手段もないため、絶対パスを指定するように実行用初期化ファイルを修正してください。

CGI アプリのCOBOL85.CBR

```
-----
[プログラム名]
  アクセス名=SEQFILE.DAT
-----
```

ISAPI アプリ用に修正したCOBOL85.CBR

```
-----
[プログラム名]
  アクセス名=C:¥COBOL¥ISAPI¥SEQFILE.DAT
-----
```

D.8 CGI環境変数の操作

CGIアプリでは、COBOLの環境変数操作機能を使用して” CGI環境変数 ”を参照できますが、ISAPIアプリでは参照できません。このため、以下の3つのサブルーチンを使用します。

```
COBW3_RECEIVE_HEADER
HTTPヘッダを取得します。
COBW3_GET_REQUEST_INFO
リクエストに関する各種情報を取得します。
COBW3_GET_AUTHORIZE
認証情報を取得します。
```

また、CGIアプリでは、COBOL環境変数操作機能を使用してCookieデータを設定/参照できますが、ISAPIアプリでは、COBOL環境変数操作機能によってCookieデータを扱うことができません。このため、以下に示すCookieデータを扱うための専用サブルーチンを使用します。

```
COBW3_SET_COOKIE_XXなど
Cookieデータを登録します。
COBW3_DEL_COOKIE_Xなど
登録されたCookieデータを削除します。
COBW3_INIT_COOKIE
クライアントに送信するCookieデータを初期化します。
COBW3_GET_COOKIE_XXなど
リクエストに含まれるCookieデータを取得します。
```

たとえば、HTTPヘッダから情報を取得する処理は次のように修正します。

CGI アプリの内容

```
-----
DISPLAY "HTTP_USER_AGENT" UPON 環境変数名.
ACCEPT ブラウザ情報 FROM 環境変数値
  ON EXCEPTION
    MOVE "Error" TO ブラウザ情報
END-ACCEPT.
-----
```

ISAPI アプリ用に修正した内容

```
-----
MOVE "User-agent" TO COBW3-HEADER-NAME.
CALL "COBW3_RECEIVE_HEADER" USING COBW3.
IF COBW3-STATUS = ZERO THEN
  MOVE COBW3-HEADER-VALUE TO ブラウザ情報
ELSE
```

```
MOVE "Error" TO ブラウザ情報
END-IF.
```

また、Cookieの操作は次のように修正します。

CGI アプリの内容

```
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY COBW3.
01 COOKIEDATA.
02 PIC X(19) VALUE "Sample+cookie+data=".
02 COOKIEVALUE PIC X(32).
PROCEDURE DIVISION.
:
DISPLAY "HTTP_COOKIE" UPON 環境変数名.
ACCEPT COOKIEVALUE FROM 環境変数値
ON EXCEPTION
MOVE "Error" TO COOKIEVALUE
END-ACCEPT.
```

ISAPI アプリ用に修正した内容

```
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY COBW3.
01 COOKIEVALUE PIC X(32).
PROCEDURE DIVISION.
:
MOVE "Sample cookie data" TO COBW3-COOKIE-NAME.
CALL "COBW3_GET_COOKIE" USING COBW3.
IF COBW3-STATUS = ZERO THEN
MOVE COBW3-COOKIE-VALUE TO COOKIEVALUE
ELSE
MOVE "Error" TO COOKIEVALUE
END-IF.
```

[補足]

環境変数操作でCookieデータを扱う場合は、ユーザがCookieデータのURLエンコードおよびURLデコードを行う必要があります。これを回避したい場合、Cookieデータには英数字だけを使用してください。

なお、半角空白をURLエンコードすると "+" になります。

付録E CGIからSAFへの移行の手引き

E.1 CGIからSAFへの移行

COBOL SAFサブルーチン(以後、SAFサブルーチンと呼びます)はCOBOL CGIサブルーチン(以後、CGIサブルーチンと呼びます)に対して基本的にソースの上位互換を保証しています。このため、CGIサブルーチンで作成したWebアプリケーションをSAFサブルーチンを使用したWebアプリケーションへ容易に移行できます。ただし、SAF固有の機能などがあるため、若干のプログラム修正や、翻訳、リンク方法の変更が必要です。以下に、移行の際の注意点などについて記述します。はじめに、CGIサブルーチンで作成したWebアプリケーション(以後、CGIアプリと呼びます)とSAFサブルーチンで作成したWebアプリケーション(以後、SAFアプリと呼びます)の違いについて下表に記します。

項番	項目	CGI アプリ	SAF アプリ
1	アプリケーションの形式	主プログラム (EXE)	副プログラム (DLL)
2	実行単位	プロセス	スレッド
3	サーバとのインタフェース領域	不要	必須
4	環境変数の操作	任意	制限あり
5	CGI 環境変数操作	提供サブルーチンまたは、DISPLAY/ACCEPT文で操作可能	提供サブルーチンを使用することで操作可能

CGIからSAFへ移行するには、上記相違点を修正する必要があります。個々の内容については、以降の節で説明します。

また、以下のサブルーチンについては、互換のためSAFサブルーチンでも提供しますが、今後、WebアプリケーションがUnicodeコード系で動作する環境では使用できません。SAFへの移行を機に代替サブルーチンへの修正をおすすめします。

サブルーチン名	代替サブルーチン
COBW3_NAME	COBW3_GET_VALUE_XX COBW3_GET_VALUE_NX
COBW3_NAMEN	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
COBW3_SET_COOKIE	COBW3_SET_COOKIE_XX COBW3_SET_COOKIE_NX COBW3_SET_COOKIE_XN COBW3_SET_COOKIE_NN
COBW3_DEL_COOKIE	COBW3_DEL_COOKIE_X COBW3_DEL_COOKIE_N

COBW3_GET_COOKIE	COBW3_GET_COOKIE_XX COBW3_GET_COOKIE_NX
COBW3_GET_COOKIEEN	COBW3_GET_COOKIE_XN COBW3_GET_COOKIE_NN

E.2 アプリケーションの形式

CGIとSAFでアプリケーションの形式が異なるため、次の項目に関する変更が必要となります。

呼出し用ページ中の起動するWebアプリケーションの記述

翻訳、リンク方法

E.2.1 呼出し用ページ

呼出し用ページに記述されているWebアプリケーションの拡張子をexeからそのアプリケーションがNES上で使用する拡張子(MIMEタイプ)に変更します。なお、あらかじめobj.confファイルの修正とMIMEタイプの登録を行っておく必要があります。これらについては“4.4.2 [NESの設定](#)”を参照してください。

たとえば、CGIアプリをCGI.EXE、SAFアプリをSAF.DLLとすると、以下のような修正を行う必要があります。

CGI アプリの呼出し用ページ

```
-----
<FORM ACTION="/sample/CGI.EXE">
...
</FORM>
-----
```

SAF アプリ用に修正した呼出し用ページ

```
-----
<FORM ACTION="/sample/foo.cobap">
...
</FORM>
-----
```

[補足]

NESではアプリケーションの識別を拡張子で行うため、アプリケーション名はDLL名と同じでなくても問題ありません。

E.2.2 翻訳、リンク方法について

翻訳

CGIアプリの場合、プログラムを翻訳するにあたって特に必要なオプションはなく、強いて言うなら、主プログラムには翻訳オプション“MAIN”を指定するだけでした。しかし、SAFアプリはDLLで作成する必要があるため、翻訳オプション“MAIN”を指定してはなりません。また、SAFアプリでは、次の翻訳オプションが必須となります。

なお、翻訳オプションの詳細については、“NetCOBOL 使用手引書”を参照してください。

THREAD(MULTI)

SAFアプリはマルチスレッドで動作するアプリケーションのため、すべてのソースプログラムに翻訳オプションTHREAD(MULTI)を指定してください。

リンク

CGIアプリでは、オブジェクトファイルをリンクしてEXEファイルを作成します。たとえば、次のように行います。

```
-----
LINK   メインプログラム.obj  初期プログラム.obj  業務プログラム.obj
        終了プログラム.obj F3BICIMP.lib LIBC.lib F3BICWSR.lib /OUT:
        実行形式名.exe
-----
```

SAFアプリはDLLとして作成しなければなりません。また、CGIアプリとはリンクするインポートライブラリが異なります。たとえば、次のようになります。

```
-----
LINK   /DLL  初期プログラム.obj  業務プログラム.obj  終了プログラム.obj
        F3BICBDM.obj F3BINSRT.lib F3BICIMP.lib KERNEL32.lib LIBC.LIB
        /OUT:実行形式名.dll /ENTRY:COBDMAN
-----
```

E.3 実行単位

CGIアプリからSAFアプリへの移行は、プロセススペースのアプリケーションからスレッドベースのアプリケーションへの移行になります。このとき以下の点に注意する必要があります。

翻訳方法
共有資源へのアクセス

E.3.1 翻訳方法について

マルチスレッドオブジェクトを生成するための翻訳オプションTHREAD(MULTI)を指定する必要があります。

E.3.2 共有資源へのアクセス

CGIでは、CGIアプリ自身のプロセス内で1つの実行スレッド(シングルスレッド)として動作するため、同一プロセス内の複数のスレッド間で共有される資源は存在しませんでした。

SAFでは、NES自身のプロセス内で複数のスレッド(マルチスレッド)としてSAFアプリが動作するため、複数のスレッド間で資源を共有できます。

SAFアプリでスレッド間共有資源を使用する場合、共有資源に対する競合状態が発生しないよう同期制御を行う必要があります。共有資源に対する同期制御は、その資源によってCOBOLランタイムシステムが自動的に制御するものと、COBOLが提供する“スレッド同期制御サブルーチン”を使用してSAFアプリ自身が制御するものがあります。

詳細は、“NetCOBOL 使用手引書”の“第24章 マルチスレッド”の以下の節を参照してください。

“24.4 スレッド間の資源の共有”
“24.9 スレッド同期制御サブルーチン”

E.4 サーバとのインタフェース領域

CGIアプリではWWWサーバとのインタフェース(環境変数は除く)は特に意識する必要はありませんでした。しかし、SAFアプリは関数としてNESによって起動されたCOBOL SAFディレクタから呼ばれ、COBOL SAFディレクタとのデータのやりとりを行うためのインタフェース(パラメタ)領域

を受け取ります。このため、CGIアプリからSAFアプリに移行する際は、連絡節と手続き部を修正する必要があります。また、このインタフェース領域をSAFサブルーチンで使用するため、SAFサブルーチンとのインタフェースであるCOBW3にそのアドレスを設定する必要があります。たとえば、次のように修正してください。

CGI アプリの内容

```
-----  
WORKING-STORAGE SECTION.  
    COPY COBW3.  
PROCEDURE DIVISION.  
    MOVE LOW-VALUE TO COBW3.  
-----
```

SAF アプリ用に修正した内容

```
-----  
WORKING-STORAGE SECTION.  
    COPY COBW3.  
*   COPY COBW3SAF.  
LINKAGE SECTION.  
    01 SAFCTX    POINTER.  
PROCEDURE DIVISION USING SAFCTX.  
    MOVE LOW-VALUE TO COBW3.  
*   MOVE LOW-VALUE TO COBW3SAF.  
    SET COBW3-CONTEXT TO SAFCTX.  
*   SET COBW3-SAF-CONTEXT TO SAFCTX.  
-----
```

[補足]

SAF固有の機能を使用する場合は、コメントを外してください。

E.5 環境変数操作

環境変数ブロックは、プロセスごとに1つ割り当てられるため、プロセスとして動作するCGIアプリでは、Webアプリケーションから環境変数を操作してもシステム上、何の問題もありませんでした。しかし、SAFアプリはNES自身のプロセスで複数のスレッドとして動作するため、環境変数の内容を変更するとNESやほかのアプリケーションに影響を及ぼしかねません。また、環境変数の内容を参照する場合も内容が不変であることは保証されないため、アプリケーションが意図したものと異なる場合があります。したがって、CGIアプリで環境変数を操作している場合は、環境変数を使用しないように変更する必要があります。

また、アプリケーションが呼び出されたときのカレントフォルダも固定である保証はないため、カレントフォルダを暗に使用する機能は注意が必要です。たとえば、カレントフォルダからの相対パス指定によるファイル操作が該当します。これに代わる手段として、次の方法があります。

絶対パスでファイルを指定する

CGI アプリの内容

```
-----  
MOVE "RESPONSE.HTM" TO COBW3-HTML-FILENAME.  
CALL "COBW3_PUT_HTML" USING COBW3.  
-----
```

SAF アプリ用に修正した内容


```
-----
MOVE "C:¥COBOL¥SAF¥RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

仮想パスに対応する物理パスを利用してファイル位置を指定する
 仮想パスに対応する物理パスを取得するには、COBW3_GET_REQUEST_INFOを使用します。

CGI アプリの内容

```
-----
MOVE "RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

SAF アプリ用に修正した内容

```
-----
SET COBW3-PHYSICALPATH TO TRUE.
CALL "COBW3_GET_REQUEST_INFO" USING COBW3.
MOVE COBW3-REQUEST-INFO TO COBW3-HTML-FILENAME.
MOVE "¥RESPONSE.HTM" TO
      COBW3-HTML-FILENAME(COBW3-REQUEST-INFO-LENGTH + 1:13).
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

SAFディレクタで指定した値を使用する

COBW3_SAF_GET_PARM_XXやCOBW3_SAF_GET_PARM_XNなどを使用することで、obj.confファイルのCOBOL_Performで指定した任意の名前と値の組を取得できます。これにWebアプリケーションが使用するパスを指定し、それを取得して結果出力用ページのファイル位置を指定します。

CGI アプリの内容

```
-----
MOVE "RESPONSE.HTM" TO COBW3-HTML-FILENAME.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

SAF アプリ用に修正した内容

```
-----
MOVE "APPPATH" TO COBW3-SAF-PARM-NAME.
CALL "COBW3_SAF_GET_PARAM_XX" USING COBW3SAF.
IF COBW3-SAF-PARM-FLAG-EXIST THEN
  MOVE COBW3-SAF-PARM-VALUE TO COBW3-HTML-FILENAME
  MOVE "¥RESPONSE.HTM" TO
        COBW3-HTML-FILENAME(COBW3-SAF-PARM-VALUE-LENGTH + 1:13)
ELSE
  MOVE "デフォルトパス名¥RESPONSE.HTM" TO COBW3-HTML-FILENAME
END-IF.
CALL "COBW3_PUT_HTML" USING COBW3.
-----
```

obj.confファイルでの SAFディレクタに対する設定

```
-----
Service method=(GET|POST) type="magnus-internal/cob-app"
sev="0" APPPATH="C:/COBAPL/APP"
-----
```

これに関連した内容として、実行用初期化ファイル(COBOL85.CBR)内のファイルの指定方法があります。CGIアプリでは絶対パス、相対パスの双方が指定可能ですが、SAFアプリでは相対パスは使用できません。また、代替手段もないため、絶対パスを指定するように実行用初期化ファイルを修正してください。

CGI アプリのCOBOL85.CBR

[プログラム名]
アクセス名=SEQFILE.DAT

SAF アプリ用に修正したCOBOL85.CBR

[プログラム名]
アクセス名=C:¥COBOL¥SAF¥SEQFILE.DAT

E.6 CGI環境変数の操作

CGIアプリでは、COBOLの環境変数操作機能を使用して“CGI環境変数”を参照できますが、SAFアプリでは参照できません。このため、以下の3つのサブルーチンを使用します。

COBW3_RECEIVE_HEADER
HTTPヘッダを取得します。
COBW3_GET_REQUEST_INFO
リクエストに関する各種情報を取得します。
COBW3_GET_AUTHORIZE
認証情報を取得します。

また、CGIアプリでは、COBOL環境変数操作機能を使用してCookieデータを設定/参照できましたが、SAFアプリでは、COBOL環境変数操作機能によってCookieデータを扱うことができません。このため、以下に示すCookieデータを扱うための専用サブルーチンを使用します。

COBW3_SET_COOKIE_XXなど
Cookieデータを登録します。
COBW3_DEL_COOKIE_Xなど
登録されたCookieデータを削除します。
COBW3_INIT_COOKIE
クライアントに送信するCookieデータを初期化します。
COBW3_GET_COOKIE_XXなど
リクエストに含まれるCookieデータを取得します。

たとえば、HTTPヘッダから情報を取得する処理は次のように修正します。

CGI アプリの内容

```
DISPLAY "HTTP_USER_AGENT" UPON 環境変数名.
ACCEPT ブラウザ情報 FROM 環境変数値
      ON EXCEPTION
      MOVE "Error" TO ブラウザ情報
END-ACCEPT.
```

SAF アプリ用に修正した内容

```

MOVE "User-agent" TO COBW3-HEADER-NAME.
CALL "COBW3_RECEIVE_HEADER" USING COBW3.
IF COBW3-STATUS = ZERO THEN
    MOVE COBW3-HEADER-VALUE TO ブラウザ情報
ELSE
    MOVE "Error" TO ブラウザ情報
END-IF.

```

また、Cookieの操作は次のように修正します。

CGI アプリの内容

```

DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY COBW3.
01 COOKIEDATA.
    02          PIC X(19) VALUE "Sample+cookie+data=".
    02 COOKIEVALUE PIC X(32).
PROCEDURE DIVISION.
    :
    DISPLAY "HTTP_COOKIE" UPON 環境変数名.
    ACCEPT COOKIEVALUE FROM 環境変数値
        ON EXCEPTION
            MOVE "Error" TO COOKIEVALUE
    END-ACCEPT.

```

SAF アプリ用に修正した内容

```

DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY COBW3.
01 COOKIEVALUE PIC X(32).
PROCEDURE DIVISION.
    :
    MOVE "Sample cookie data" TO COBW3-COOKIE-NAME.
    CALL "COBW3_GET_COOKIE" USING COBW3.
    IF COBW3-STATUS = ZERO THEN
        MOVE COBW3-COOKIE-VALUE TO COOKIEVALUE
    ELSE
        MOVE "Error" TO COOKIEVALUE
    END-IF.

```

[補足]

環境変数操作でCookieデータを扱う場合は、ユーザがCookieデータのURLエンコードおよびURLデコードを行う必要があります。これを回避したい場合、Cookieデータには英数字だけを使用してください。

なお、半角空白をURLエンコードすると "+" になります。

付録F CGIサブルーチン V1.0からの移行時の注意事項について

ここでは、COBOL97 V40L10で提供されているCGIサブルーチン1.0(以降V1.0)を使用したCOBOLアプリケーションをCGIサブルーチン2.0(以降V2.0)以降を使用したCOBOLアプリケーションに移行する方法について説明します。

【補足】

V2.0以降は、各種WWWサーバが提供するAPI(ISAPI、NSAPI(SAF)など)に対応したWebサブルーチンとほぼ同様のインタフェースで実現されています。このため、V2.0以降を使用して作成したWebアプリケーションは、他のWebサブルーチンへの移植(たとえば、CGIからISAPIなど)が容易で、移植工数やテスト工数を大幅に削減しつつ、各WWWサーバが提供するAPIがもたらす運用性能の向上などのメリットを得ることができます。

以上のことから、今後開発されるWebアプリケーション(新規/改造するもの)はV2.0以降を使用してください。また、V1.0で既に作成されたCOBOLアプリケーションもV2.0以降に移行することをおすすめします。

なお、V1.0を使用したCOBOLアプリケーションをそのまま使用する場合には、“第2章 [CGIアプリケーションの作成と実行](#)”を参照してください。

V1.0を使用したCOBOLアプリケーションを、V2.0以降を使用するCOBOLアプリケーションに移行する場合には以下の作業が必要です。

COBOLプログラムおよび関連資産の修正

COBOLプログラムの翻訳およびリンク

以降に各作業について説明します。

F.1 COBOLプログラムおよび関連資産の修正

次の手順に従ってソースプログラムを修正します。

F.2.1 登録集の変更

V1.0のインタフェースである登録集(FJCOB-CGI.cb1)をV2.0以降の登録集(COBW3.cb1)に変更します。

CGI サブルーチン 1.0	CGI サブルーチン 2.0以降
COPY FJCOB-CGI	COPY COBW3

F.2.2 サブルーチン名およびパラメタ名の変更

V2.0以降ではV1.0と同等の機能を提供していますが、各サブルーチンの名前とパラメタの名前に違いがあります。次の規則に従って変更してください。

パラメタ名の修正(FJCOB-CGI-xxxxをCOBW3-xxxxに変更する)

サブルーチン名の修正(COBW3_CGI_xxxをCOBW3_xxxに変更する)

CALL文のUSING指定の修正(FJCOB-CGIをCOBW3に変更する)

例

COBW3_CGI_NAMEの変更例

変更前

```
-----
MOVE "NAME1" TO FJCOB-CGI-SEARCH-DATA.
```

```
CALL "COBW3_CGI_NAME" USING FJCOB-CGI.
```

```
IF FJCOB-CGI-SEARCH-FLAG-EXIST
```

```
    MOVE FJCOB-CGI-GET-DATA TO データ
```

```
END-IF.
```

```
....
```

変更後

```
-----
MOVE "NAME1" TO COBW3-SEARCH-DATA.
```

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

```
IF COBW3-SEARCH-FLAG-EXIST
```

```
    MOVE COBW3-GET-DATA TO データ
```

```
END-IF.
```

```
....
```



注意

“COBW3_CGI_xxx_CNV”は“COBW3_CNV_xxx”に名前が変更されていますので注意してください。

V1.0とV2.0以降の各機能の対応は次の対応表を参照してください。

CGI サブルーチン 1.0	CGI サブルーチン 2.0以降
COBW3_CGI_INIT	COBW3_INIT
COBW3_CGI_NAME	COBW3_NAME (注)
COBW3_CGI_NAMEN	COBW3_NAMEN (注)
COBW3_CGI_VALUE	COBW3_VALUE (注)
COBW3_CGI_PUT_HTML	COBW3_PUT_HTML
COBW3_CGI_PUT_TEXT	COBW3_PUT_TEXT
COBW3_CGI_SET_CNV	COBW3_CNV_SET (注)
COBW3_CGI_SET_CNV2	
COBW3_CGI_DEL_CNV	COBW3_CNV_DEL (注)
COBW3_CGI_CNV_INIT	COBW3_CNV_INIT (注)
COBW3_CGI_SYSTEM	COBW3_SYSTEM
COBW3_CGI_FREE	COBW3_FREE

注

CGIサブルーチン3.0以降では、代替サブルーチンの利用をおすすめします。詳細については、“7.1 [COBOL Webサブルーチンの機能](#)”を参照してください。

F.2.3 サブルーチンの非互換に関する変更

V2.0以降ではV1.0と同等の機能を提供していますが、一部のサブルーチンに非互換があるため次の対応が必要です。

COBW3_CGI_INIT(呼出しパラメタの初期化)

COBW3_CGI_PUT_HTML(FJCOB-CGI-HTML-FSの削除および結果出力用ページファイルの変更)

COBW3_CGI_SET_CNV2(サブルーチンの変更)

COBW3_CGI_INIT

V1.0のインタフェースである“FJCOB-CGI”はVALUE句によってLOW-VALUEで初期化されていましたが、V2.0以降のインタフェースである“COBW3”はVALUE句が指定されていないため、“COBW3_INIT”を呼び出す前に“COBW3”にLOW-VALUEを転記し、初期化するように変更します。

変更前

```
-----
CALL "COBW3_CGI_INIT" USING FJCOB-CGI.
-----
```

変更後

```
-----
MOVE LOW-VALUE TO COBW3.
CALL "COBW3_INIT" USING COBW3.
-----
```

また、V1.0の“COBW3_CGI_INIT”には“FJCOB-CGI-CONTENT-TYPE”を指定することにより、Content-typeを宣言する機能がありましたが、V2.0以降ではヘッダを出力する“COBW3_PUT-HEAD”を使用する仕様に変更されています。このため、“COBW3_INIT”の実行後に“COBW3_PUT-HEAD”によってContent-typeを宣言するように修正してください。

変更前

```
-----
SET FJCOB-CGI-CONTENT-TYPE-HTML TO TRUE.
CALL "COBW3_CGI_INIT" USING FJCOB-CGI.
-----
```

変更後

```
-----
MOVE LOW-VALUE TO COBW3.
CALL "COBW3_INIT" USING COBW3.
SET COBW3-CONTENT-TYPE-HTML TO TRUE.
CALL "COBW3_PUT_HEAD" USING COBW3.
-----
```

【補足】 COBOL CGIサブルーチン2.0以降ではContent-typeなどのヘッダ出力機能が拡張されていますので、ヘッダ出力のタイミングを“COBW3_INIT”を呼び出した後にすることが可能です。詳細については、“COBW3_PUT-HEAD”を参照してください。

COBW3_CGI_PUT_HTML

V1.0では“COBW3_CGI_PUT_HTML”を実行した場合、“FJCOB-CGI-HTML-FS”に指定した結果出力用ページとして使用するファイルの入出力状態を返却していましたが、V2.0以降の“COBW3_PUT_HTML”では入出力状態は返却されません。COBOLプログラムで入出力状態を参照する場合には、“COBW3-STATUS”を参照してください。

“COBW3_PUT_HTML”で出力する結果出力用ページに“//FJCOB//変換名//FJCOB//”の変換データが記述されている場合、キーワードが“//FJCOB//”から“//COBOL//”に変更されていますの

で、修正します。

例

変換名 “NAME1” が記載されている場合の変更例
変更前

```
-----
          あなたの名前は、//FJCOB//NAME1//FJCOB// です。
-----
```

変更後

```
-----
          あなたの名前は、//COBOL//NAME1//COBOL// です。
-----
```

COBW3_CGI_SET_CNV2

変換名を登録するV1.0の“COBW3_CGI_SET_CNV2”を使用している場合には、V2.0以降の“COBW3_CNV_SET”に変更します。

例

変換名 “NAME1” を登録している場合の変更例
変更前

```
-----
MOVE  "NAME1"  TO  FJCOB-CGI-CNV-NAME.

MOVE  "名前"   TO  FJCOB-CGI-CNV-VALUE.

SET    FJCOB-CGI-CNV2-MODE-ADDREP TO TRUE.

CALL  "COBW3_CGI_SET_CNV2" USING FJCOB-CGI.
-----
```

変更後

```
-----
MOVE  "NAME1"  TO  COBW3-CNV-NAME.

MOVE  "名前"   TO  COBW3-CNV-VALUE.

SET    COBW3-CNV-MODE-ADDREP TO TRUE.

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

F.2 COBOLプログラムの翻訳およびリンク

ソースプログラムの修正後、プログラムの再翻訳を行います。

翻訳後はCOBOL CGIサブルーチン2.0以降をリンクします。

例

a.cobを翻訳/リンクする

```
cobol32 -M a.cob
link /out:a.exe a.obj F3BICWSR.LIB F3BICIMP.LIB LIBC.LIB
```

索引

記号

@CBR_ATTACH_TOOL ..	16, 33, 46, 157, 159, 161
@CBR_CGI_LOGFILE	16
@CBR_CGI_SEVERITY	16
@CBR_ISAPI_LOGFILE	33
@CBR_ISAPI_SEVERITY	33
@CBR_SAF_LOGFILE	47
@CBR_SAF_SEVERITY	47
@CBR_WEB_OUT_CODE	95
@MessOutFile	16, 33, 47
@WinCloseMsg	16, 33, 47

A

APIの種類	63, 77, 85
--------------	------------

C

CGI	2, 55
CGIアプリ	181, 189
CGIサブルーチン	3
COBOL_Init	49
COBOL_Perform	50
COBOL SAFサブルーチン	6
COBOL SAFディレクタ	6, 191
COBOLソースプログラムを翻訳する	28, 41
COBOLプログラム登録先	77
COBOL目的プログラムをリンクする	28, 41
COBW3_ALTER_SESSION_TIMEOUT	140
COBW3_CHECK_VALUE_N	113
COBW3_CHECK_VALUE_X	113
COBW3_CNV_DEL	146
COBW3_CNV_INIT	147
COBW3_CNV_SET	145
COBW3_DEL_CNV_N	121
COBW3_DEL_CNV_X	121
COBW3_DEL_COOKIE	149
COBW3_DEL_COOKIE_N	131
COBW3_DEL_COOKIE_X	131
COBW3_DEL_REPEAT_N	123
COBW3_DEL_REPEAT_X	123
COBW3_DEL_UPLOADEDFILE	136
COBW3_END_SESSION	138
COBW3_FREE	125
COBW3_GEN_UPLOADFILE_N	135
COBW3_GEN_UPLOADFILE_X	135
COBW3_GET_AUTHORIZE	127
COBW3_GET_COOKIE	149
COBW3_GET_COOKIE_N	149

COBW3_GET_COOKIE_NN	132
COBW3_GET_COOKIE_NX	132
COBW3_GET_COOKIE_XN	132
COBW3_GET_COOKIE_XX	132
COBW3_GET_SESSION_DATA	139
COBW3_GET_SESSION_INFO	140
COBW3_GET_UPLOADFILE_INFO_N	133
COBW3_GET_UPLOADFILE_INFO_X	133
COBW3_GET_VALUE_NN	112
COBW3_GET_VALUE_NX	112
COBW3_GET_VALUE_XN	112
COBW3_GET_VALUE_XX	112
COBW3_INIT	111
COBW3_INIT_CNV	121
COBW3_INIT_COOKIE	131
COBW3_INIT_REPEAT	124
COBW3_NAME	143
COBW3_NAMEN	143
COBW3_PUT_HEAD	114
COBW3_PUT_HTML	115
COBW3_PUT_TEXT	124
COBW3_RECEIVE_HEADER	126
COBW3_SAF_GET_BASENAME	143
COBW3_SAF_GET_PARM_NN	141
COBW3_SAF_GET_PARM_NX	141
COBW3_SAF_GET_PARM_XX	141
COBW3_SAF_GET_PARM_XN	141
COBW3-SESSION-ADAPTERクラス	152
COBW3_SET_CNV_NN	119
COBW3_SET_CNV_NX	119
COBW3_SET_CNV_XN	119
COBW3_SET_CNV_XX	119
COBW3_SET_COOKIE	147
COBW3_SET_COOKIE_NN	128
COBW3_SET_COOKIE_NX	128
COBW3_SET_COOKIE_XN	128
COBW3_SET_COOKIE_XX	128
COBW3_SET_REPEAT_NN	122
COBW3_SET_REPEAT_NX	122
COBW3_SET_REPEAT_XN	122
COBW3_SET_REPEAT_XX	122
COBW3_SET_SESSION_DATA	138
COBW3_START_SESSION	137
COBW3_SYSTEM	125
COBW3_VALUE	144
Cookieが有効となる仮想ディレクトリ	148
Cookieが有効となる仮想パス	130
Cookieが有効となるドメイン	130, 148
Cookieデータ	89

Cookieデータの削除 149
Cookieデータの操作 128
Cookieデータの登録 128, 147
Cookieデータを削除 131, 187, 194
Cookieデータを取得 187, 194
Cookieデータを初期化 187, 194
Cookieデータをすべて初期化 131
Cookieデータを登録 187, 194
Cookieの有効期限 130, 148
Cookie名の検索 132

G

GetExtensionVersion 4
GetExtensionVersion 21, 184

H

HTML雛形ファイルのコード系 66
HTML文書 2
HttpExtensionProc 4, 22, 184
HTTPヘッダ 187
HTTPヘッダ 194
HTTPヘッダを取得 126

I

IIS 4
IISの設定 29
INTERSTAGE V2系互換 63
ISA 4
ISAPI 4, 55
ISAPIアプリ 181
ISAPIエクステンション 4
ISAPIサブルーチン 5
ISAPIフィルタ 4
ISAを作成する際の注意点 27

M

MeFt/Web 92
Microsoft暗号化認証 128
MIME Type 68
MIMEタイプの設定 44

N

NESの設定 42
NESパラメタ情報 141
NSAPI 6, 55

O

obj.confファイル 42
obj.conf 42

S

SAF 6
SAFアプリ 189
SAF固有のサブルーチン 108
SAFディレクタ 49
SAFディレクタの環境設定ファイル 46
SAFディレクタを登録する 42
SAFを作成する際の注意点 40
Server Error 166
SSL 165

T

TerminateExtension 4, 24, 184

U

Unicode 63, 92
URLエンコード 188, 195

W

Webアプリケーションウィザード 54, 58, 60, 71
Webアプリケーション開発のヒント 89
Webアプリケーションの入れ替え 33, 47
Webアプリケーションの実行指定 43
Webアプリケーションの情報を追加 71
Webアプリケーションの雛形 58
Webアプリケーションのロード指定 43
Webアプリケーションを入れ替える 162
Webアプリケーションを入れ替える 161
Webアプリケーションを追加 70
Webアプリケーションを登録する 43
Webサブルーチンの機能 104
Webサブルーチンの種類 66
Webパラメタ 2
Webパラメタ取得プログラム 64, 69, 73, 77
Webパラメタ取得プログラム生成ツール 73, 75
Webパラメタの獲得 111
Webパラメタの操作 112
WRB API 55
WWWサーバ 2
WWWブラウザ 2, 93

い

インストール先フォルダ 63
インターネットサービスマネージャ 29
インタフェース領域 184, 191

え

エディタテンプレート 89
エラー出力ページ 69

エラー処理	173
エラーページ出力プログラム	69

か

外字	165
各種情報を取得	126
拡張CGI	56
格納フォルダ	62
仮想ディレクトリ	33, 186
仮想ディレクトリに対する設定	31
仮想ディレクトリの追加	30
仮想パス	193
カレントフォルダ	186, 192
環境変数	186
環境変数設定	16, 32, 46
環境変数操作機能	194
環境変数の設定	167
環境変数ブロック	192

き

基本認証	128
業務プログラム	10

く

クライアントのIPアドレス	127
クラス	152
繰返し範囲の指定方法	117
繰返し変換データ	85
繰返し変換データの最大設定数	85
繰返し変換データの削除	123
繰返し変換データの登録	122
繰返し変換データをすべて初期化	124

け

結果出力用ページ	2, 9, 13, 26, 39, 54, 65, 99
結果出力用ページ雛形	69
結果出力用ページ ファイル名	85
結果ページ出力プログラム	54, 65, 69, 81, 86
結果ページ出力プログラム生成ツール	81, 84

こ

コード系	63, 77, 86
------------	------------

さ

作業環境の設定	111
---------------	-----

し

識別できない文字	165
資源の解放	125

システム環境変数	16, 32, 46
システムコマンドの実行	125
実行結果	86
実行定義ファイル	70
実行定義ファイル雛形	65
実行ファイル名	63
実行用初期化ファイル	16, 46
シフトJIS	63
集団項目名	78, 87
終了プログラム	12
出力コード系	95
出力データ項目名	87
出力データ登録集	69
出力データ用登録集	65
使用可能なサブルーチン	97
初期プログラム	9
処理結果の出力	114
シングルスレッド	184, 191

す

ステータスコード	115
スレッド間共有資源	184, 191

せ

生成COBOLプログラム登録先	85
生成ファイル	66
生成ファイル詳細	65
セキュリティ	130
セッション管理	137
セッション管理機能で使用するクラス	152
セッションデータ	68
セッションデータの退避処理	64
セッションデータの復元処理	64
絶対パス	186, 192

た

代替サブルーチン	181, 189
タイムアウト時間	168
対話型デバグ	157

て

定量制限	150
データ出力用登録集	87
データ入力用登録集	78
データを表示形式で参照	164
データを保護	165
デバグを起動する	159, 161
デバグを終了する	160, 162
デバグする	162
デバグモード	163

デバッグを行う..... 160

と

登録集..... 109
登録集ファイル名..... 78, 87
トレース情報..... 156

な

名前(NAME)の検索..... 112
名前(NAME)を検索..... 143

に

入力項目の最大文字列長..... 76
入力項目の字類..... 77
入力コード系..... 94
入力データ項目名..... 78
入力データ登録集..... 69
入力データ用登録集..... 64
認証情報..... 89, 187, 194
認証タイプ..... 128

ね

ネットワーク環境の資源..... 169

は

パスワード..... 127
パラメタ取得プログラム..... 54
半角カナ..... 165
半角と全角文字が混在..... 165

ひ

表示内容を動的に変える..... 166

ふ

ファイルアップロード..... 133
プールの指定..... 32
フォームエレメント..... 77
復帰コード..... 185
物理パス..... 33, 186
物理パスの登録..... 30
フレーム..... 168
プログラム資源..... 66
プログラムの名前..... 67
プログラムの雛形..... 64
プロジェクトマネージャからの
HTML文書の操作..... 89
プロジェクト名..... 62
分離プロセス..... 32

へ

ヘッダ..... 114
変換データの削除..... 121, 146
変換データをすべて初期化..... 121, 147
変換データを登録..... 145
変換名の指定方法..... 116
変換文字列の最大文字列長..... 85
変換文字列の字類..... 85
変換文字列の登録..... 119

ほ

翻訳..... 15, 28, 41, 182, 190
翻訳オプション..... 96

ま

マルチスレッド..... 184, 191
マルチスレッドオブジェクト..... 183, 191

も

モジュール定義ファイル..... 29, 69, 183
モジュール定義ファイル..... 65

ゆ

ユーザID..... 127

よ

呼出しパラメタ..... 109
呼出し用ページ..... 2, 9, 12, 25, 38, 54, 65,
77, 98, 182, 190
呼出し用ページ雛形..... 69
呼出し用ページ ファイル名..... 76

り

リクエスト..... 187, 194
リクエスト情報の取得..... 126
リクエストデータ..... 10
リンク..... 15, 28, 41, 183, 191

れ

レスポンスデータ..... 10

ろ

ログ情報..... 16
ログ情報..... 33, 47, 156
ログ情報の重要度..... 16, 47
ログファイル..... 100
ログファイルの構成..... 156



このマニュアルはエコマーク認定の再生紙を使用しています。