COBOL

プログラマのための NetCOBOL入門

NetCOBOL for Windows の基本を押さえよう

NetCOBOL

はじめに

このテキストの目的

本書は、Windows®版 NetCOBOL 開発環境の操作方法と、MeFt/Web による簡単な画面帳票アプリケーションの作り方を学習するためのテキストです。

このテキストによる学習を終えた後、次に示す事項を習得することを目標としています。

- NetCOBOL 開発環境の機能、基本的な操作を理解する
- NetCOBOL を用いて、基本的な COBOL アプリケーションを作成できる
- MeFt/Web による簡単な画面、帳票印刷アプリケーションを作成できる

このテキストの対象者

このテキストは、これから MeFt/Web による Web アプリケーションの開発をされる方を対象に説明しています。また、次に示す知識をお持ちであることを前提に説明しています。

- COBOL 言語（基本文法、ファイル入出力等）を理解され、COBOL プログラミングの経験があること
- Windows OS の一般的な概念と、操作方法を理解されていること
- 簡単な HTML 文法と、WWW サーバに関する一般的な知識をお持ちであること

このテキストの構成

このテキストの構成と内容を次に示します。

◆ **第 1 章　NetCOBOL アプリケーション開発の基礎**

簡単なファイル入出力を行う COBOL アプリケーションを例に、NetCOBOL 開発環境の基本操作について説明しています。

◆ **第 2 章　画面帳票アプリケーションの開発**

NetCOBOL シリーズの MeFt、FORM による基本的な画面帳票アプリケーションの作成方法について説明しています。

◆ **第 3 章　MeFt/Web アプリケーションの構築**

画面帳票アプリケーションの MeFt/Web 化の方法、環境設定などについて説明しています。

このテキストで使用している製品

このテキストの説明では、次に示す製品を使用しています。

- NetCOBOL Standard Edition 開発パッケージ for Windows V8.0
- Microsoft® Windows® 2000 Server operating system
- Microsoft® Internet Information Services 5.0

なお、このテキストでは、NetCOBOL が「C:¥Program Files¥NetCOBOL」にインストールされているものとして説明しています。

略 称

このテキストでは、次に示す略称を使用しています。

正式名称	略称
Microsoft® Windows® 95 operating system	Windows® 95
Microsoft® Windows® 98 operating system	Windows® 98
Microsoft® Windows® Millennium Edition operating system	Windows® Me
Microsoft® Windows NT® Workstation operating system Version 4.0	Windows NT®または Windows NT® 4.0
Microsoft® Windows NT® Server Network operating system Version 4.0	Windows NT®または Windows NT® 4.0
Microsoft® Windows NT® Server, Enterprise Edition 4.0	Windows NT®、Windows NT® 4.0 または Windows NT® E.E.
Microsoft® Windows NT® Server 4.0, Terminal Server Edition	Windows NT®、Windows NT® 4.0 または Windows NT® T.S.E.
Microsoft® Windows® XP Professional operating system	Windows® XP
Microsoft® Windows® XP Home Edition operating system	Windows® XP
Microsoft® Windows® 2000 Professional operating system	Windows® 2000 または Windows®2000 Professional
Microsoft® Windows® 2000 Server operating system	Windows® 2000 または Windows®2000 Server
Microsoft® Windows® 2000 Advanced Server operating system	Windows® 2000、Windows®2000 Server または Windows®2000 Advanced Server
Microsoft® Windows Server™ 2003, Standard Edition	Windows® Server™ 2003
Microsoft® Windows Server™ 2003, Enterprise Edition	Windows® Server™ 2003
Windows® 95、Windows® 98、Windows® Me、Windows NT®、Windows® XP、Windows® 2000 および Windows® 2003 の総称	Windows®
Microsoft® Internet Explorer	Internet Explorer
Microsoft® Internet Information Server、および Microsoft® Internet Information Services	IIS

商 標

Microsoft、Windows、Windows NT、および ActiveX 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 は、Netscape Communications Corporation の米国およびその他の国における商標です。

平成 17 年 9 月 第二版

All Rights Reserved, Copyright©富士通株式会社 2005

目次

第1章 NetCOBOL アプリケーション開発の基礎……1
1.1 概要 2
1.1.1 NetCOBOL の開発環境 2
1.1.2 作成するアプリケーションについて 2
1.1.3 アプリケーション開発の流れ 3
1.2 プログラムの作成 4
1.2.1 プロジェクトの作成 5
1.2.2 ビルド 10
1.2.3 実行 16
1.2.4 プロジェクトマネージャの終了 19
1.3 プログラムのデバッグ 20
1.3.1 デバッグの準備 21
1.3.2 デバッガの起動 22
1.3.3 デバッグ操作 23
1.3.4 デバッガの終了 29

第2章 画面帳票アプリケーションの開発……31
2.1 概要 32
2.1.1 画面帳票アプリケーションの概要 32
2.1.2 作成するアプリケーションについて 33
2.1.3 アプリケーション開発の流れ 34
2.2 表示ファイルのプログラミング 35
2.3 画面帳票定義体の作成 39
2.3.1 画面定義体の作成 39
2.3.2 帳票定義体の作成 51
2.4 画面帳票定義体の確認 60
2.4.1 FORM 試験の概要 60
2.4.2 画面定義体の確認 61
2.4.3 帳票定義体の確認 67
2.5 プログラムの作成 68
2.5.1 プロジェクトの作成 68
2.5.2 ビルド 70
2.5.3 実行 70
2.5.4 プロジェクトマネージャの終了 71
2.6 プログラムのデバッグ 72
2.6.1 デバッグの準備 72
2.6.2 デバッガの起動 72
2.6.3 デバッグ操作 72
2.6.4 デバッガの終了 72

第 3 章 MeFt/Web アプリケーションの構築……73
3.1 概要 74
3.1.1 MeFt/Web の概要 74
3.1.2 構築するアプリケーションについて 75
3.1.3 構築作業の流れ 75
3.2 MeFt/Web サーバのセットアップ 76
3.2.1 MeFt/Web 動作環境の設定 76

3.2.2 権限の設定 79
3.3 MeFt/Web 環境への移行 82
3.3.1 MeFt/Web 移行時のアプリケーションの対応 82
3.3.2 アプリケーション資産の配置と環境設定 85
3.4 HTML の作成 89
3.5 リモート実行 94
3.5.1 HTML の表示 94
3.5.2 プログラムの実行 96
3.6 デバッグ 100
3.6.1 MeFt/Web アプリケーションのデバッグについて 100
3.6.2 デバッグの準備 100
3.6.3 デバッガの起動 101
3.6.4 デバッグ操作 101
3.6.5 デバッガの終了 101
3.7 通信が切断されるパターンについて 102

第 1 章 NetCOBOL アプリケーション開発の基礎

本章では、NetCOBOL が提供する開発環境の機能を説明するとともに、簡単なアプリケーションの作成を通じて開発環境の操作を説明します。

1.1 概要
1.2 プログラムの作成
1.3 プログラムのデバッグ

1.1 概要

NetCOBOL にてアプリケーション開発を行うときに使用する開発環境、および本章で作成するアプリケーションの概要について説明します。

1.1.1 NetCOBOL の開発環境

NetCOBOL では、次のような開発環境を提供しています。

名称	説明
COBOL プロジェクトマネージャ	プロジェクトの管理
エディタ	COBOL ソースファイル、登録集などの作成
コンパイラ	COBOL ソースプログラムの翻訳
リンカ	翻訳されたファイルから、実行ファイルおよび DLL の作成
COBOL 対話型デバッガ	対話形式による COBOL アプリケーションのデバッグ

◆ COBOL プロジェクトマネージャ

プロジェクトマネージャでは、COBOL アプリケーションの開発で各種操作(プログラムの編集・翻訳・リンク・実行・デバッグ)を行うための、簡易操作ビューを提供しています。プロジェクトマネージャを使ってプログラム開発を行うことにより、プログラム開発に必要な各種操作を、プロジェクトマネージャから行うことができます。

◆ エディタ

エディタでは、COBOL ソースプログラムや登録集原文などを作成します。行番号のリナンバや他のファイルの取り込みなどを行うことができます。

◆ コンパイラ

コンパイラは、COBOL ソースプログラムを翻訳し、目的プログラムを作成します。

◆ リンカ

リンカは、目的プログラムから実行可能ファイルまたはダイナミックリンクライブラリを作成します。

◆ COBOL 対話型デバッガ

COBOL 対話型デバッガは、COBOL アプリケーションをデバッグするときに使用します。対話型デバッガを使用すると、ウィンドウを使った簡単な操作でデバッグを行うことができます。なお、以降の説明では COBOL 対話型デバッガを「デバッガ」と記しています。

1.1.2 作成するアプリケーションについて

本章では、NetCOBOL に同梱されているサンプルのうち、行順ファイルと索引ファイルの操作を行っているサンプルを使用し、アプリケーションを作成します。

◆ アプリケーションの概要

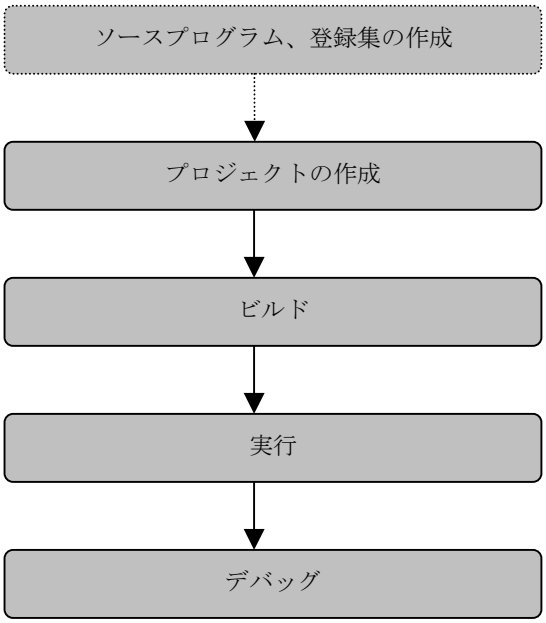
エディタを使って作成したデータファイル(行順ファイル)を読み込み、マスタファイル(索引ファイル)を作成するアプリケーションです。索引ファイルのレコードは登録集ファイルとして用意されており、COBOL の COPY 文を使って翻訳時に COBOL プログラムに取り込みます。

◆ プログラムの格納場所

説明に使用するサンプルは、NetCOBOL のインストールディレクトリの「SAMPLES」フォルダ配下の「SAMPLE02」に格納されています。

1.1.3　アプリケーション開発の流れ

本章で説明するアプリケーションの開発の流れを次に示します。
なお、ソースプログラムおよび登録集はサンプルで提供されているものを使用するため、新規に作成しません。

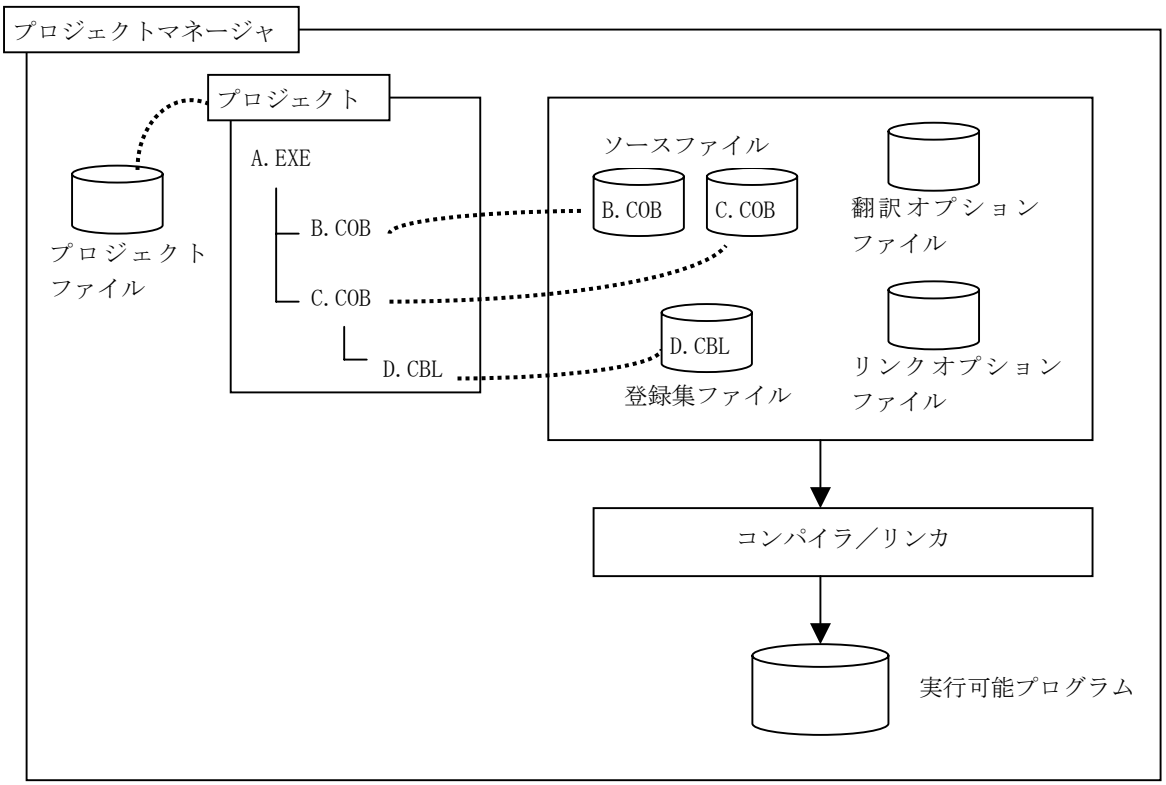


1.2 プログラムの作成

COBOL プロジェクトマネージャによる、プロジェクト管理機能を使用したプログラムの作成方法について説明します。

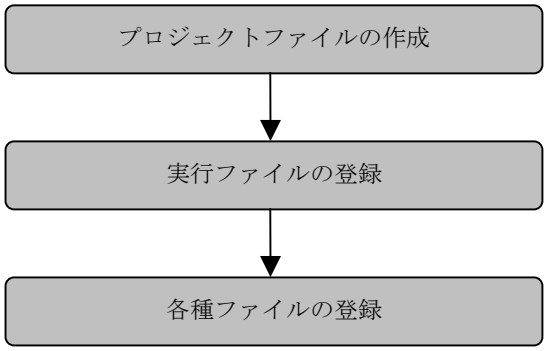
◆ プロジェクト管理機能について

プロジェクト管理機能では、1 つ以上のファイル(COBOL ソースプログラムおよび登録集)から生成されるアプリケーションをプロジェクトという形で管理することができ、COBOL プログラムの開発、保守を容易に行うことができます。実行ファイルを作成するのに必要となるソースプログラムや登録集ファイルなどの依存関係は、プロジェクトで管理されます。



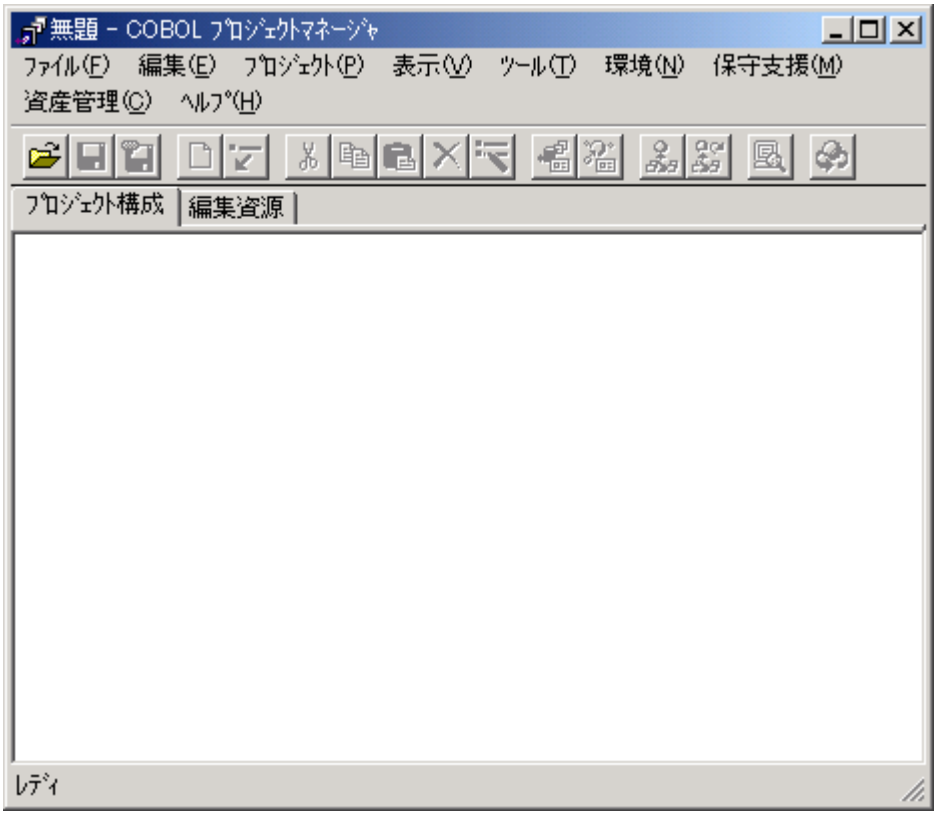
1.2.1　プロジェクトの作成

プロジェクトの作成の流れを次に示します。



◆　COBOL プロジェクトマネージャの起動

[スタート]－[プログラム]－[NetCOBOL Standard Edition 開発パッケージ for Windows V8.0]－[NetCOBOL]から[COBOL プロジェクトマネージャ]を選択して、プロジェクトマネージャを起動します。



◆ プロジェクトファイルの作成

プロジェクトファイルとは、プロジェクトで管理されるファイルの依存関係などの情報を格納するためのファイルです。

プロジェクトファイルの作成方法を次に示します。

(1)プロジェクトマネージャの[ファイル]メニューから[プロジェクトを開く]を選択します。

(2)フォルダ「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02」を選択し、「ファイルを開く」ウィンドウの「ファイル名」に、プロジェクト名として「SAMPLE2.PRJ」を入力します。入力が終わったら「開く」ボタンを選択してください。

(3)「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02¥SAMPLE2.PRJ このファイルは存在しません。作成しますか？」というダイアログボックスが表示されますので、「はい」を選択します。

(4)新規に作成されたプロジェクトファイルが、プロジェクトマネージャで表示されます。

◆ 実行ファイルの登録

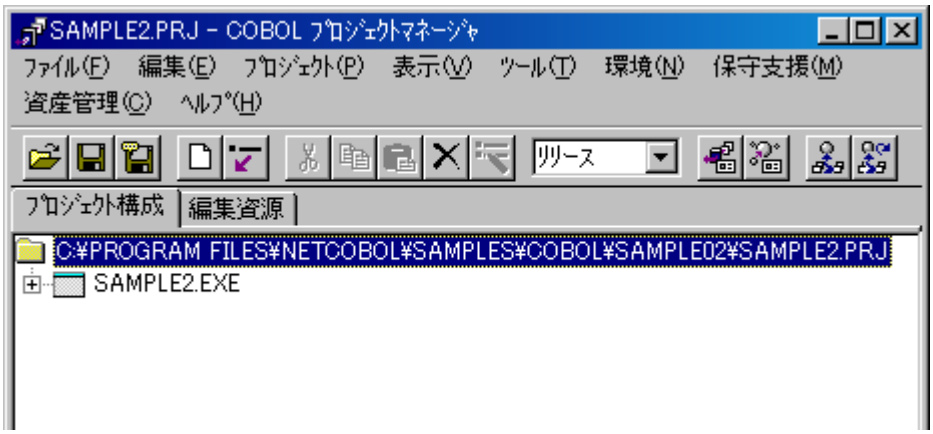
実行ファイルとは、プロジェクトで最終的に作成される実行可能ファイルまたは DLL のことです。

実行ファイルの登録方法を次に示します。

(1)プロジェクトファイルを選択し、[編集]メニューから[新規作成]を選択します。

(2)[新規ファイル.EXE]というファイルが作成され、ファイル名が変更状態になっています。ここで、実行ファイル名を直接入力することにより、実行ファイルが設定されます。

本章で作成するアプリケーションでは、実行ファイル名を「SAMPLE2.EXE」とします。



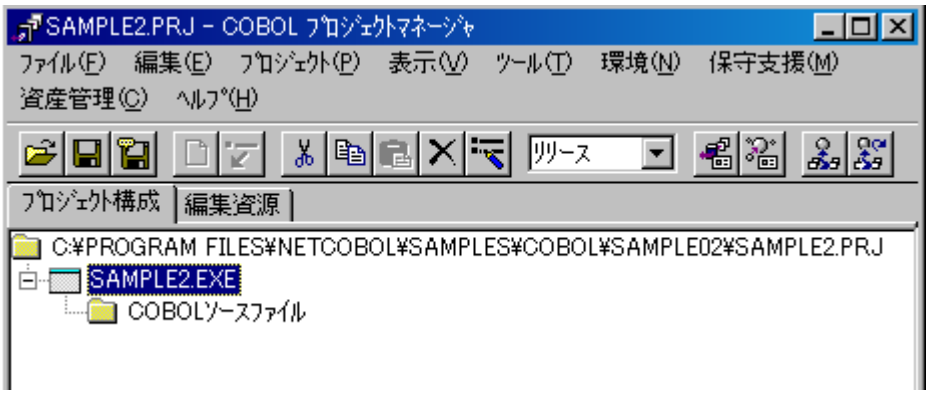
◆ 各種ファイルの登録

実行ファイルを作成するのに必要となるファイルをプロジェクトに登録します。登録するファイルとして、COBOL ソースファイルや登録集ファイル、画面帳票定義体などがあります。

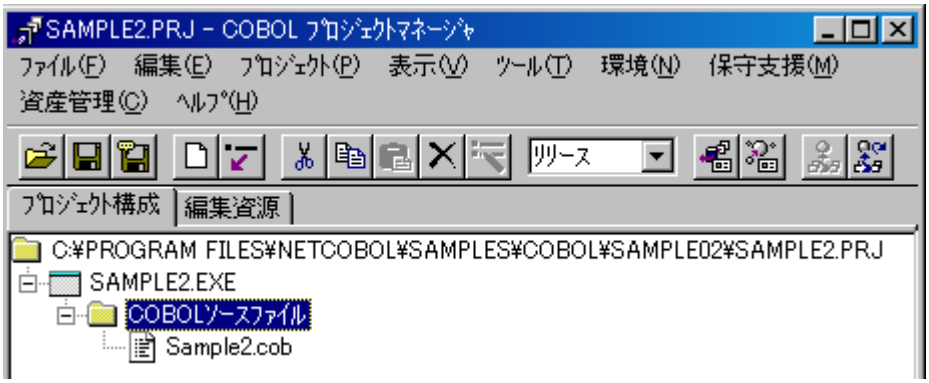
本章で作成するアプリケーションでは、COBOL ソースファイルと登録集ファイルを登録します。なお、登録するソースファイルと登録集ファイルは、サンプルプログラムで提供されているものを使用します。

COBOL ソースファイルの登録

(1)実行ファイルを選択します。
(2)プロジェクトマネージャの[編集]メニュー－[フォルダ作成]から「COBOL ソースファイル」を選択します。
実行ファイル配下に「COBOL ソースファイル」というフォルダが作成されます。



(3)フォルダ「COBOL ソースファイル」を選択します。
(4)プロジェクトマネージャの[編集]メニューから[追加]を選択します。
(5)「ファイルの参照」ダイアログが表示されるので、サンプルプログラムのフォルダにある COBOL ソースファイル「Sample2.cob」を選択します。
(6)フォルダの配下に「Sample02.cob」が追加されます



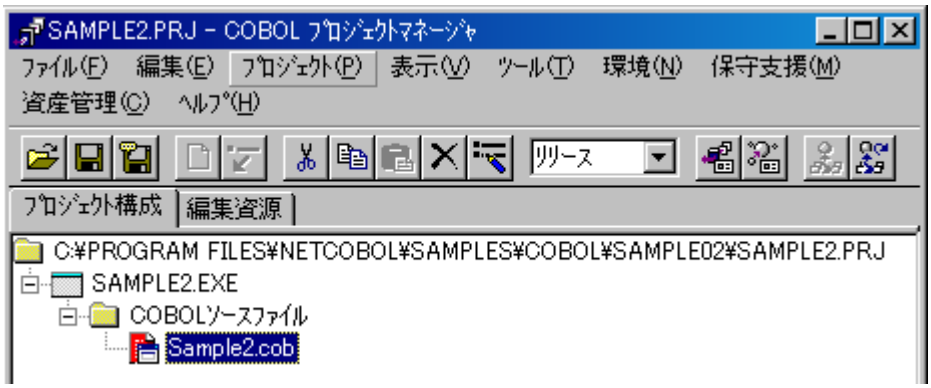
参考

COBOL ソースファイルを新規作成してプロジェクトに登録するには、次の手順で行います。
(1)プロジェクトファイルに登録されているフォルダ「COBOL ソースファイル」を選択します。
(2)プロジェクトマネージャの[編集]メニューから[新規作成]を選択します。
(3)[新規ファイル.COB]というファイルが作成され、ファイル名が変更状態になっています。ここで、COBOL ソースファイル名を直接入力することにより、COBOL ソースファイルが登録されます。
(4)プロジェクトに登録された COBOL ソースファイルをダブルクリックし、エディタでソースプログラムを作成します。

主プログラムの指定

登録した COBOL ソースファイルが最初に動作するプログラムである場合、主プログラムの指定を行います。主プログラムの指定を行う方法を次に示します。

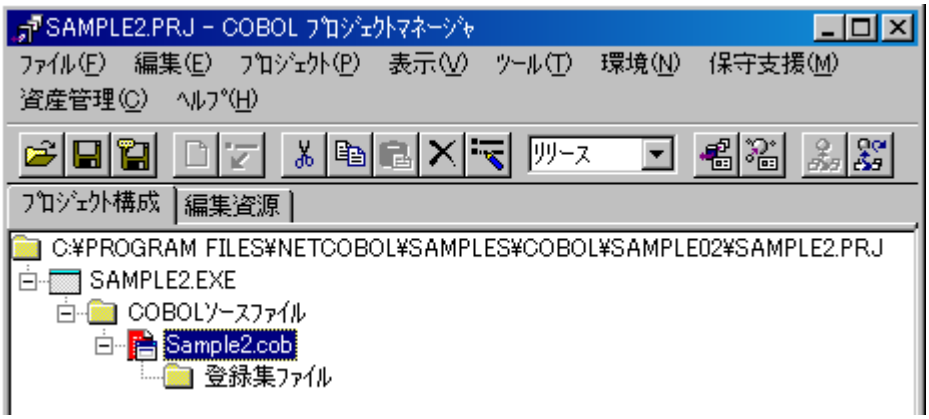
(1)登録された COBOL ソースファイルを選択します。
(2)[プロジェクト]メニュー－[オプション]－[主プログラム]から「ウィンドウ」を選択します。
(3)COBOL ソースファイルの小アイコンが変更されます。



登録集ファイルの登録

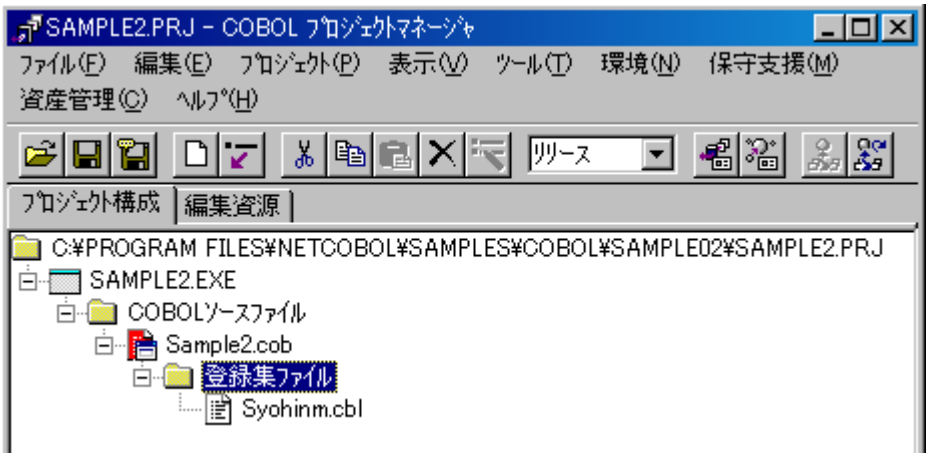
ソースファイルが利用する登録集をプロジェクトに登録します。これにより、ソースファイルとの依存関係が定義できます。また、プロジェクトマネージャから登録集ファイルを容易に参照できるようになります。

(1)COBOL ソースファイルを選択します。
(2)プロジェクトマネージャの[編集]メニュー－[フォルダ作成]から「登録集ファイル」を選択します
COBOL ソースファイル配下に「登録集ファイル」というフォルダが作成されます。



(3)フォルダ「登録集ファイル」を選択します。
(4)プロジェクトマネージャの[編集]メニューから[追加]を選択します。
(5)「ファイルの参照」ダイアログが表示されるので、サンプルプログラムのフォルダにある登録集ファイル「Syohinm.cbl」を選択します。

(6)フォルダの配下に「Syohinm.cbl」が追加されます。



> **参考**
>
> 登録集ファイルは、必要なもののみプロジェクトに登録します。全ての登録集ファイルをプロジェクトに登録する必要はありません。

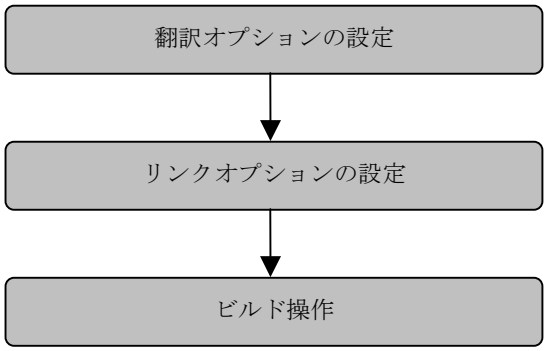
◆ プロジェクトファイルの保存

プロジェクトマネージャの[ファイル]メニューから[上書き保存]を選択して、プロジェクトファイルを保存します。

1.2.2 ビルド

プロジェクトが作成できたら、ビルドします。「ビルド」とは、1回の指示で翻訳およびリンクを行い、実行ファイルを作成することです。なお、再翻訳および再リンクを行って実行ファイルを再作成することを「リビルド」といいます。

ビルドの流れを次に示します。



◆ 翻訳オプションの設定

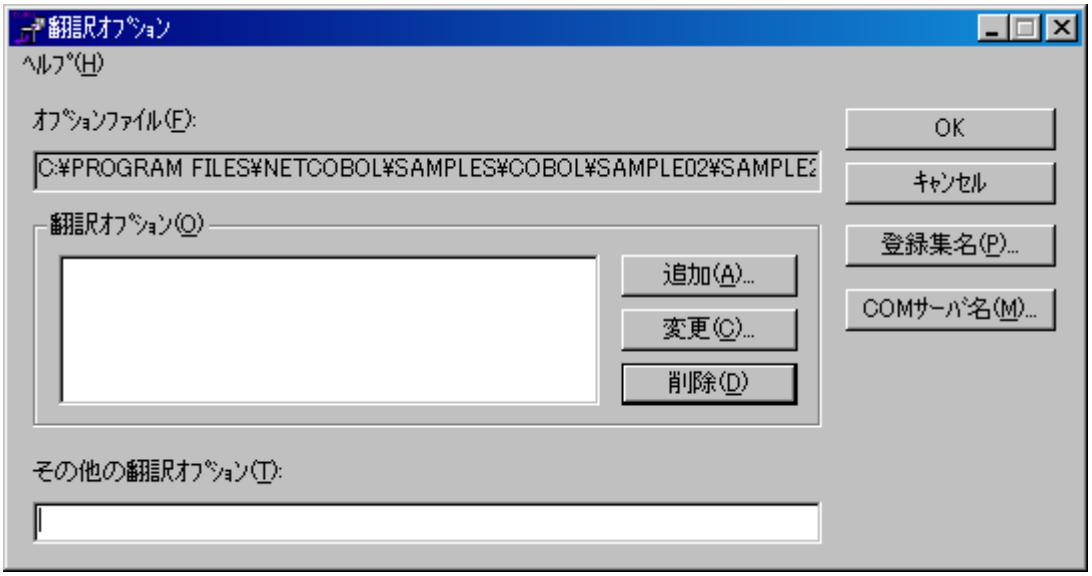
プロジェクトで管理しているソースファイルを翻訳するときに有効になる、翻訳オプションを設定します。設定した翻訳オプションは、翻訳オプションファイル(プロジェクト名.CBI)に格納されます。

ここでは、例として翻訳オプションLIB（登録集ファイルのフォルダの指定）を追加する方法を元に、翻訳オプションを設定する手順を次に示します。

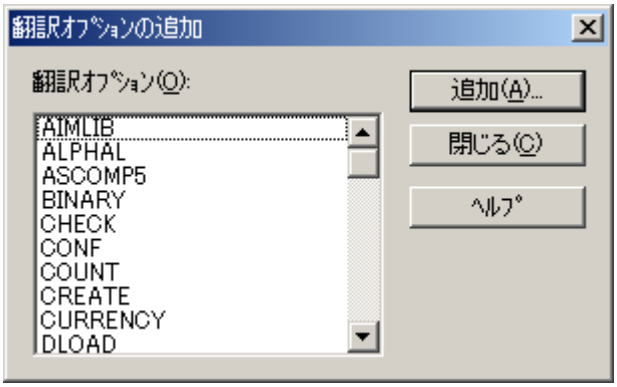
なお、設定できる翻訳オプションについては、NetCOBOL 使用手引書「A.1 翻訳オプション一覧」をご覧ください。

(1)プロジェクトを選択し、プロジェクトマネージャの[プロジェクト]メニュー－[オプション]から[翻訳オプション]を選択します。

「翻訳オプション」ダイアログが表示されます。

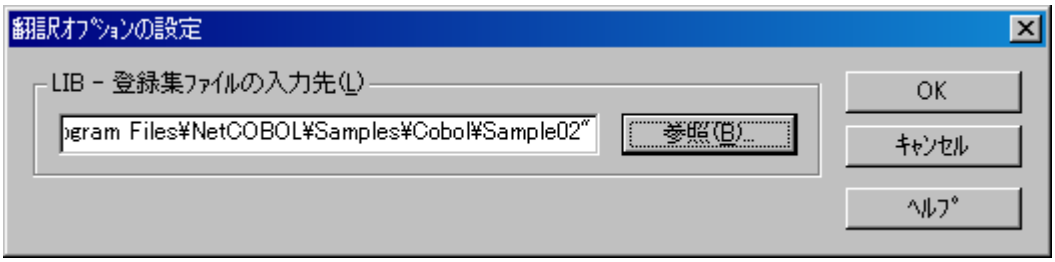


(2)「翻訳オプション」ダイアログの「追加」ボタンをクリックします。「翻訳オプションの追加」ダイアログが表示されます。



(3)翻訳オプションのリストボックスから「LIB」を選択し、「追加」ボタンをクリックします。「翻訳オプションの設定」ダイアログが表示されます。

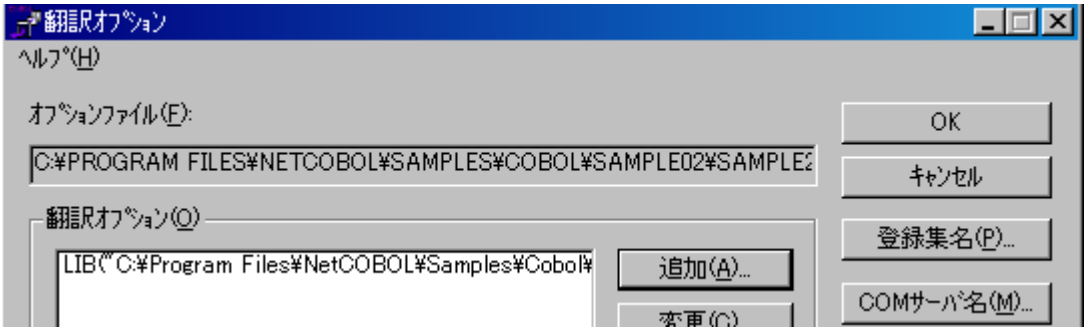
(4)登録集ファイルの入力先として、フォルダ「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02」を選択し、「OK」ボタンをクリックします。



(5)今回設定する翻訳オプションは「LIB」のみであるため、「翻訳オプションの追加」ダイアログで「閉じる」ボタンをクリックします。

設定した翻訳オプションは、「翻訳オプション」ダイアログに表示されます。

(6)「OK」ボタンをクリックします。



<u>参考</u>

翻訳オプションの設定内容を変更するには、次の手順で行います。

(1)「翻訳オプション」ダイアログで、設定内容を変更したい翻訳オプションを選択し、「変更」ボタンをクリックします。

(2)「翻訳オプションの設定」ダイアログで、変更する内容を設定します。

(3)設定が終わったら、「OK」ボタンをクリックします。

また、翻訳オプションを削除するには、次の手順で行います。

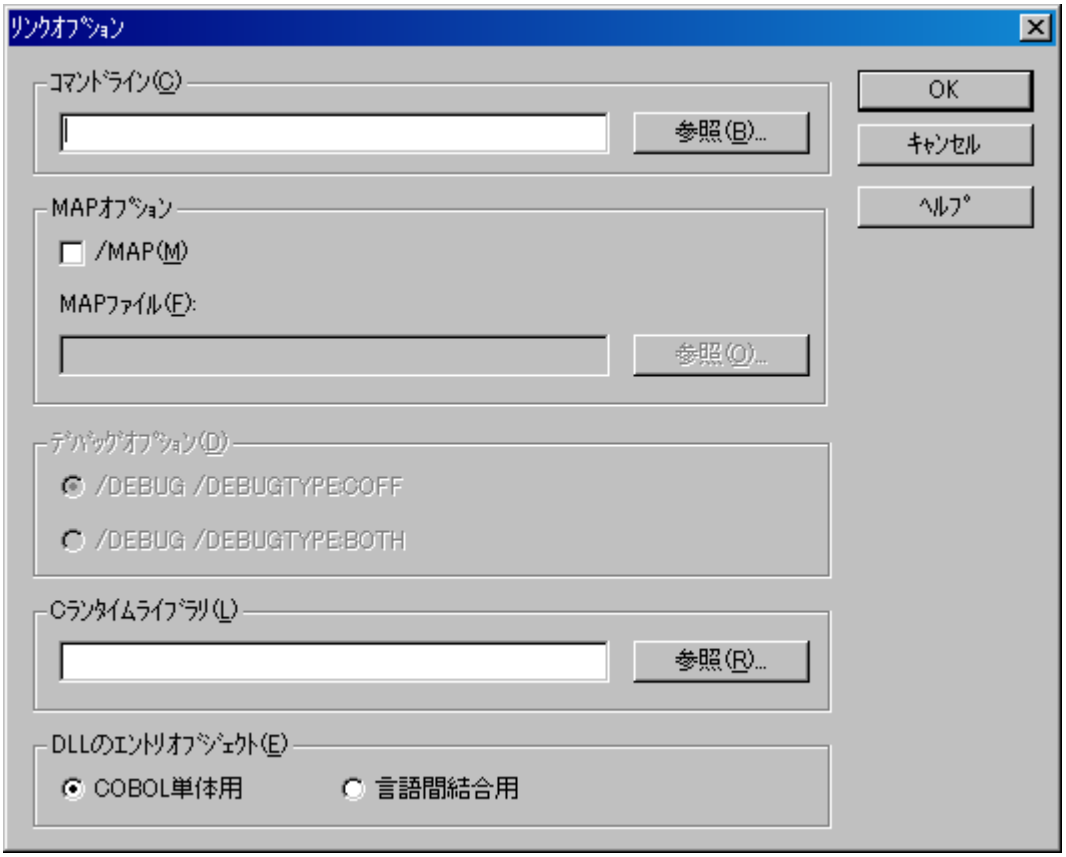
(1)翻訳オプションのリストボックスから削除したい翻訳オプションを選択し、「削除」ボタンをクリックします。翻訳オプションのリストボックスから、選択した翻訳オプションが削除されます。

◆ リンクオプションの設定

プロジェクトで管理している実行可能ファイルまたは DLL をリンクするときに有効になるリンクオプションを設定します。設定したリンクオプションの情報は、リンクオプションファイル(プロジェクト名.LNI)に格納されます。

リンクオプションは、C 言語で作成されたライブラリを結合したいときなどに設定しますが、本章で作成するアプリケーションでは、リンクオプションの設定は不要です。
ここでは、リンクオプションの設定画面の表示方法について説明します。

(1)プロジェクトを選択し、プロジェクトマネージャの[プロジェクト]メニュー－[オプション]から[リンクオプション]を選択すると、「リンクオプション」ダイアログが表示されます。

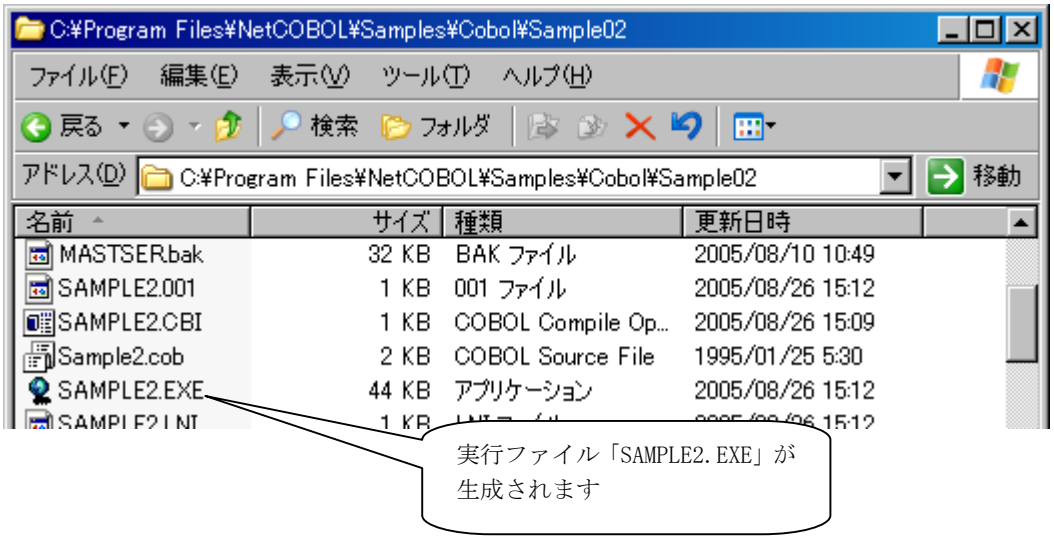


(2)必要に応じて各項目の設定を行い、「OK」ボタンをクリックします。

◆ ビルド操作

プロジェクトを選択し、プロジェクトマネージャの[プロジェクト]メニューから[ビルド]を選択します。
「ビルダ」ウィンドウが開きます。

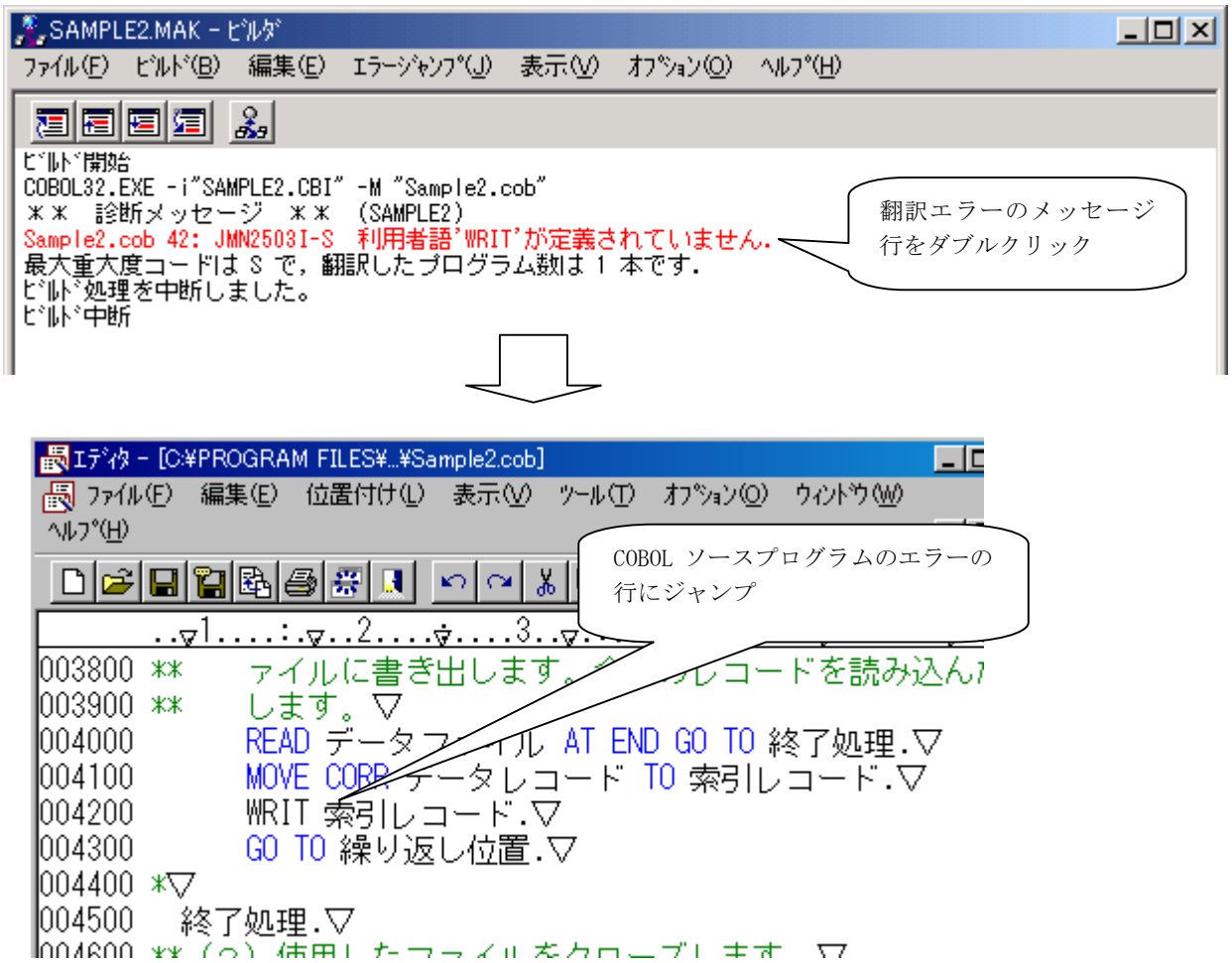
「ビルダ」ウィンドウは、ビルドの内容を表示します。エラーがなければビルドが終了し、プロジェクトファイルに登録した実行ファイルが生成されます。



翻訳エラーの修正

ビルドで翻訳エラーが検出された場合、翻訳終了時に「ビルダ」ウィンドウに翻訳メッセージが表示されます。

ビルダのエラージャンプ機能では、ビルダウィンドウに表示された診断メッセージをクリックすると、エディタが自動的に開き、翻訳エラーが検出された COBOL ソースプログラムの行にジャンプすることができます。ビルダのエラージャンプ機能を使うと、翻訳エラーの発生した文の検出を簡単に行うことができ、プログラムの修正作業が効率よく行えます。



1.2.3 実行

ビルドされた COBOL プログラムを実行します。

◆ 実行環境情報の設定

実行環境情報と初期化ファイルについて

COBOL プログラムを実行するには、実行環境情報の設定を行う必要があります。

NetCOBOL では、COBOL プログラムを実行するために割り当てる資源や情報のことを実行環境情報といいます。本章では、ファイルの入出力を行うプログラムを作成しましたので、入出力するファイルを実行環境情報として指定します。

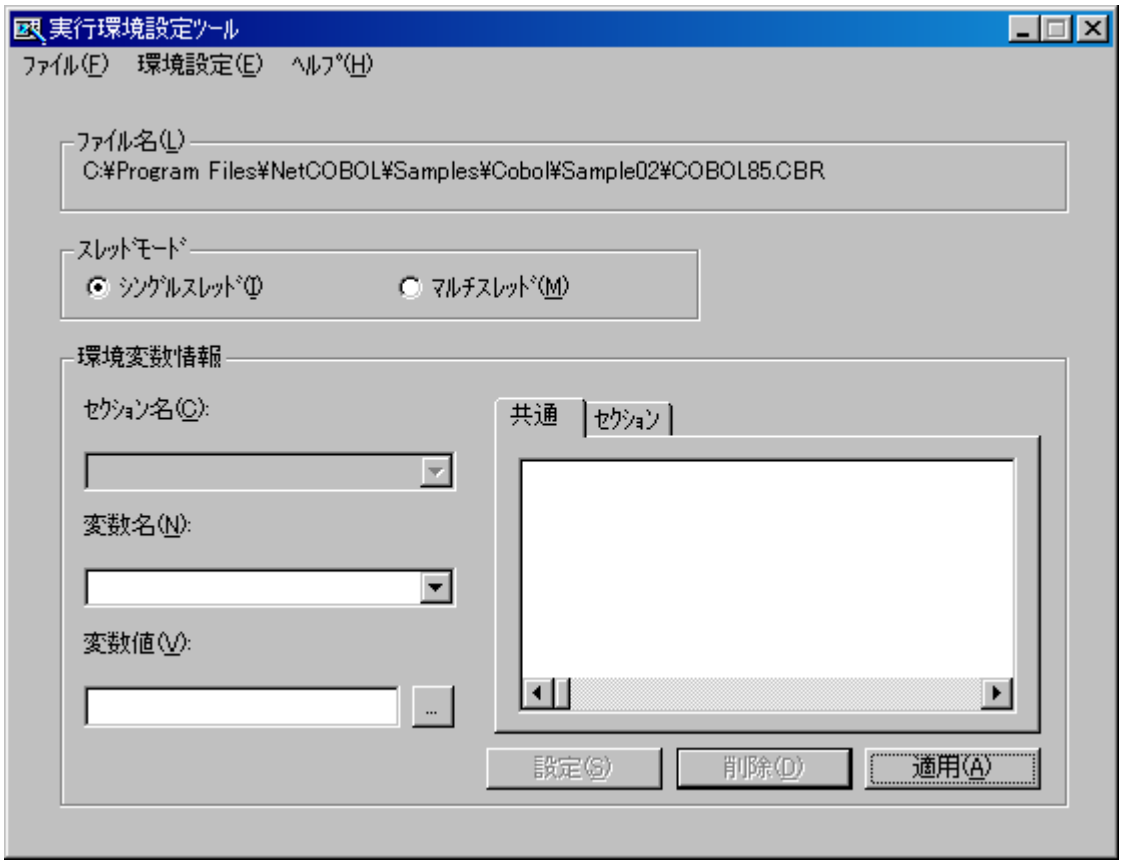
実行環境情報は、実行用の初期化ファイルに格納します。COBOL プログラムは、実行時に実行用の初期化ファイルから情報を取り出して実行します。通常、実行可能プログラム（EXE）が格納されているフォルダの“COBOL85.CBR”を実行用の初期化ファイルとして扱います。

実行環境設定ツールによる実行用の初期化ファイルの作成

NetCOBOL では、実行用の初期化ファイルの内容を編集し、実行環境情報を設定するツールとして「実行環境設定ツール」があります。ここでは、実行環境設定ツールを使用して実行用の初期化ファイルを作成する方法を以下に示します。

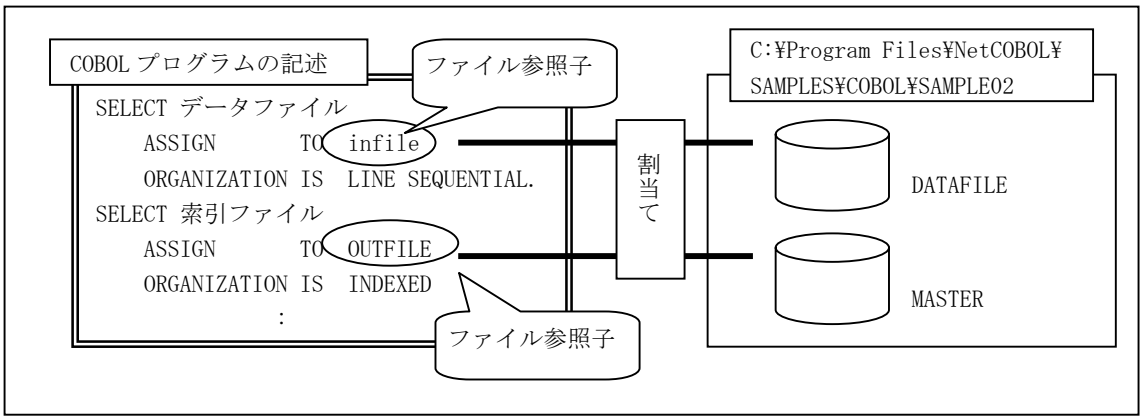
(1)COBOL プロジェクトマネージャで実行ファイルを選択します。

(2)[プロジェクト]メニューから[実行環境設定]を選択します。実行環境設定ツールが起動し、画面に「ファイル名」として「C:¥Program Files¥NetCOBOL¥SAMPLES¥Cobol¥SAMPLE02¥COBOL85.CBR」が表示されます。



ファイル識別名とファイルの関連付け

実行環境設定として、COBOL プログラムとファイルの実体との関連付けを行います。関連付けとして、本章で作成するアプリケーションでは、COBOL プログラムの ASSIGN 句に定義されたファイル参照子に実際のファイルを割り当てます。



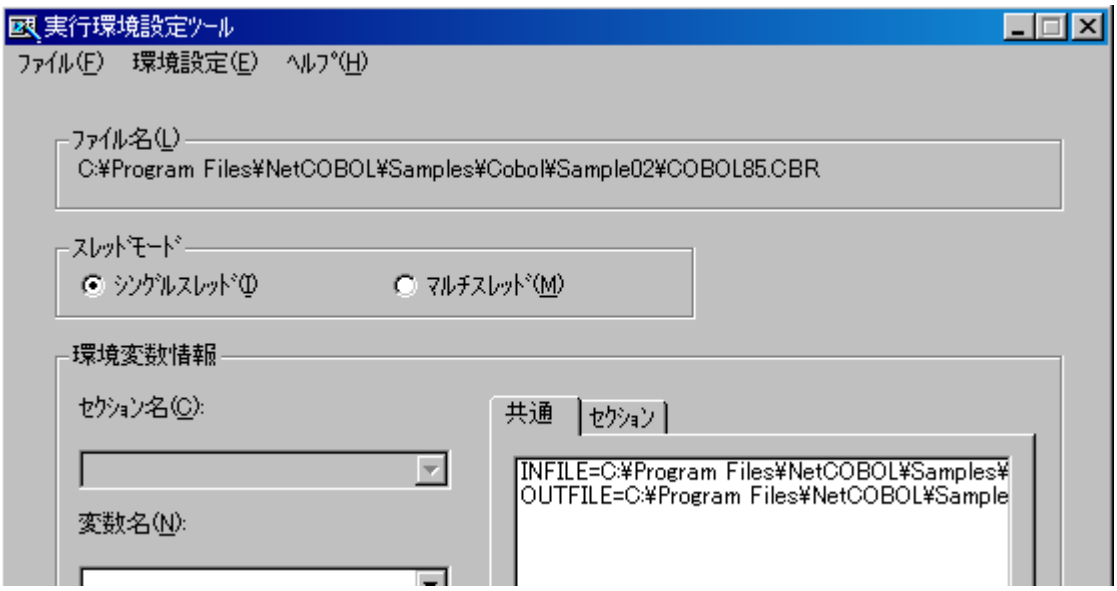
「変数値」でファイルを指定する際、「…」ボタンで表示される「ファイルの指定」ウィンドウから指定することができます。

実行環境設定ツールでは、「変数名」にファイル参照子を指定し、「変数値」には実際のファイルを指定します。本章で作成するプログラムでは、次に示す指定を行います。

変数名	変数値
INFILE	C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02¥DATAFILE
OUTFILE	C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02¥MASTER

※MASTER はフォルダ「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02」に存在しませんので、「ファイルの指定」ウィンドウの「ファイル名」に直接入力し、「開く」ボタンを選択します。

変数名と変数値を設定したら、「設定ボタン」を選択します。



実行環境設定ツールの「適用」ボタンを押して、フォルダ「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02」に「COBOL85.CBR」があることを確認します。

実行環境情報の設定が終了したら、「ファイル」メニューから「終了」を選択し、実行環境設定ツールを終了します。

参考

実行環境情報の設定として、ここで紹介した実行環境設定ツールによる設定のほかに、SET コマンドやコントロールパネルのシステム（Windows® 95、Windows® 98 および Windows® Me では AUTOEXEC.BAT）で設定する方法もあります。

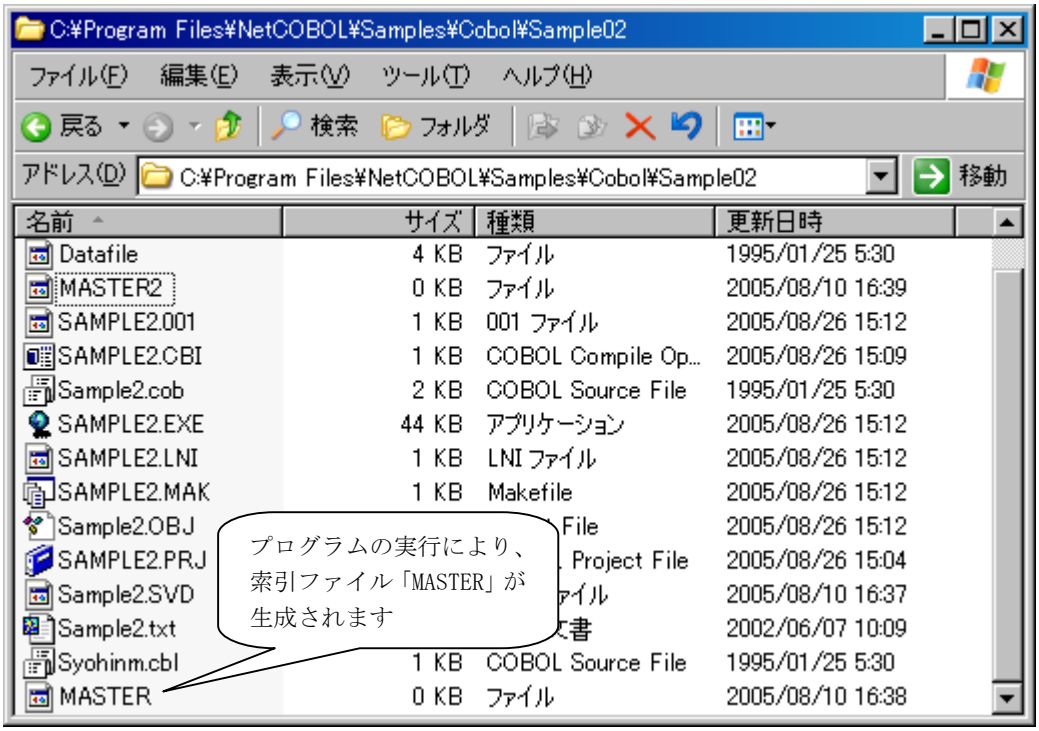
詳細は、NetCOBOL 使用手引書「5.3.2 実行環境情報の設定方法」をご覧ください。

◆ **プログラムの実行**

プログラムを実行するには、COBOL プロジェクトマネージャでプロジェクトファイルまたは実行可能ファイルを選択し、「プロジェクト」メニューから「実行」を選択します。

また、実行可能ファイルをダブルクリックして、プログラムを実行することもできます。

作成されたプログラムでは、実行の終了メッセージが画面に表示されません。実行が終了すると、索引ファイル「MASTER」が「C:¥Program Files¥NetCOBOL¥SAMPLES¥COBOL¥SAMPLE02」に作成されます。



1.2.4 プロジェクトマネージャの終了

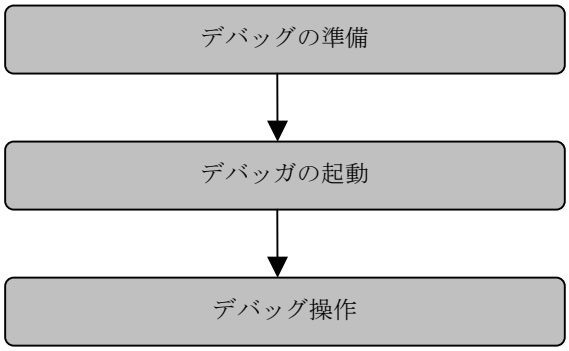
COBOL プロジェクトマネージャの[ファイル]メニューから[終了]を選択して、プロジェクトマネージャを終了します。

1.3 プログラムのデバッグ

NetCOBOL のデバッガを使用したデバッグについて説明します。

NetCOBOL のデバッガでは、実行可能プログラムをそのままデバッグの対象とし、プログラムの論理的な誤りを、プログラムを動作させながら検出することができます。
デバッグ作業は、画面に表示した COBOL ソースプログラムに対する直接的で簡単な操作で行うことができます。キーボードやマウスを使い、メニュー内のコマンドやツールバーに表示されたボタンを操作することによって行います。

デバッガを使用したデバッグの流れを次に示します。



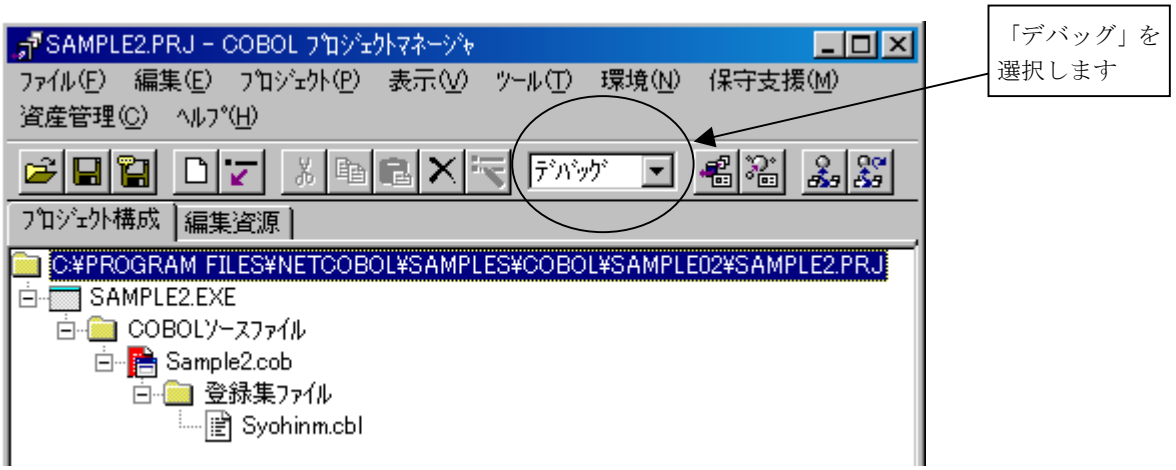
1.3.1 デバッグの準備

デバッガでは、実行可能プログラム(EXE)および実行可能プログラムから呼び出されるダイナミックリンクライブラリ(DLL)をデバッグすることができます。デバッガにデバッグの対象となるプログラムと認識させるには、デバッグモジュールおよびそのモジュールのデバッグ情報ファイルを作成する必要があります。

デバッグモジュールとは、デバッガを使用したデバッグ作業が可能な実行可能ファイルおよびダイナミックリンクライブラリのことをいいます。デバッグ情報ファイルとは、デバッグモジュールのデバッグ時に必要となるファイルであり、COBOL ソースファイル名の拡張子を SVD に置き換えたファイル名になります。

ここでは、プロジェクトマネージャを使用して、デバッグモジュールおよびデバッグ情報ファイルを作成する方法を説明します。

(1)プロジェクトマネージャで、デバッグ作業を行いたいプログラムのプロジェクトを開きます。
(2)プロジェクトマネージャの「ビルドモード」コンボボックスで、「デバッグ」を選択します。

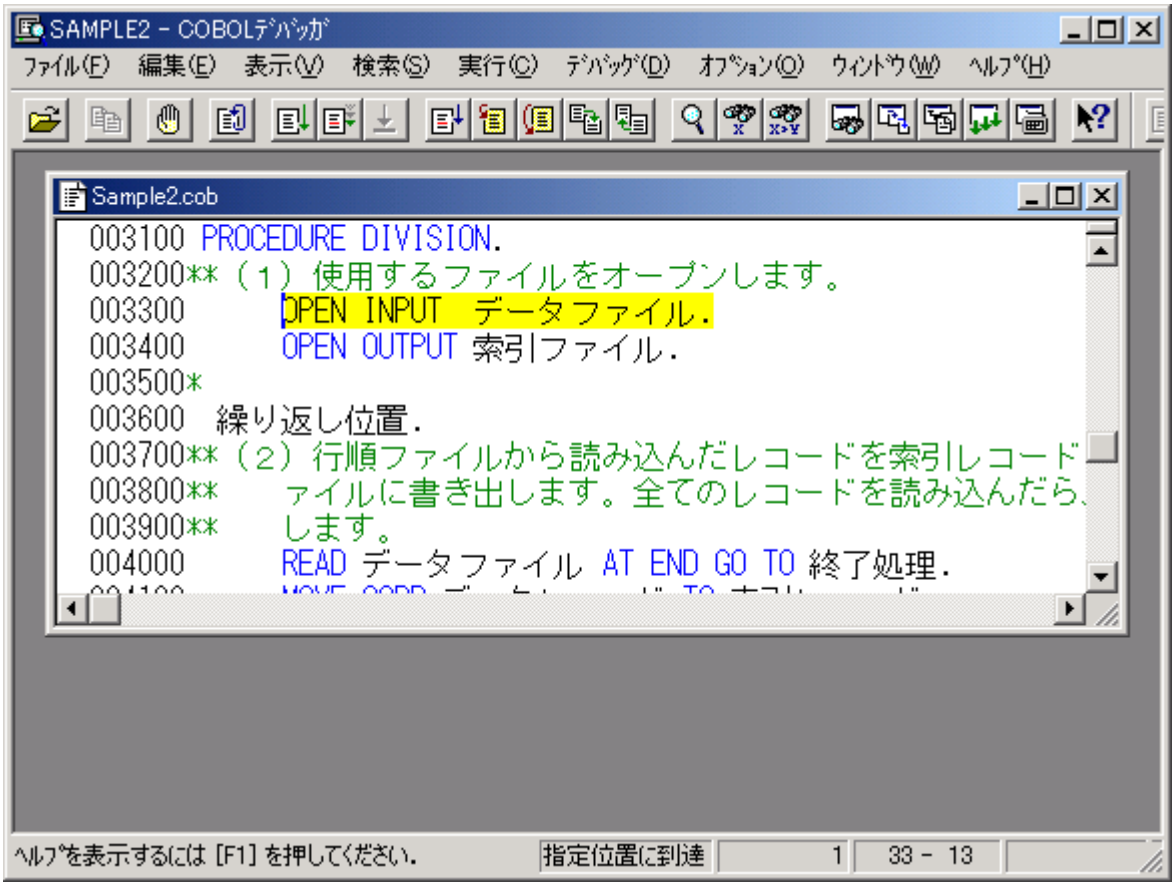


(3)プロジェクトをビルド（またはリビルド）します。
COBOL ソース名の拡張子が SVD に置き換わったデバッグ情報ファイルが作成されます。

1.3.2 デバッガの起動

デバッガは、プロジェクトマネージャから起動したり、コマンドプロンプトから起動したりすることができます。ここでは、プロジェクトマネージャから起動する方法を説明します。

(1)プロジェクトマネージャで、デバッグするプロジェクトファイルまたは実行可能ファイル（EXE）を選択します。
(2)「プロジェクト」メニューから「デバッグ」を選択します。
(3)「デバッグを開始する」ダイアログが表示されます。
(4)「デバッグを開始する」ダイアログの「アプリケーション名」にデバッグするプログラム名が表示されていることを確認し、「OK」ボタンを選択します。
(5)デバッガが起動し、デバッグ作業が行えるようになります。



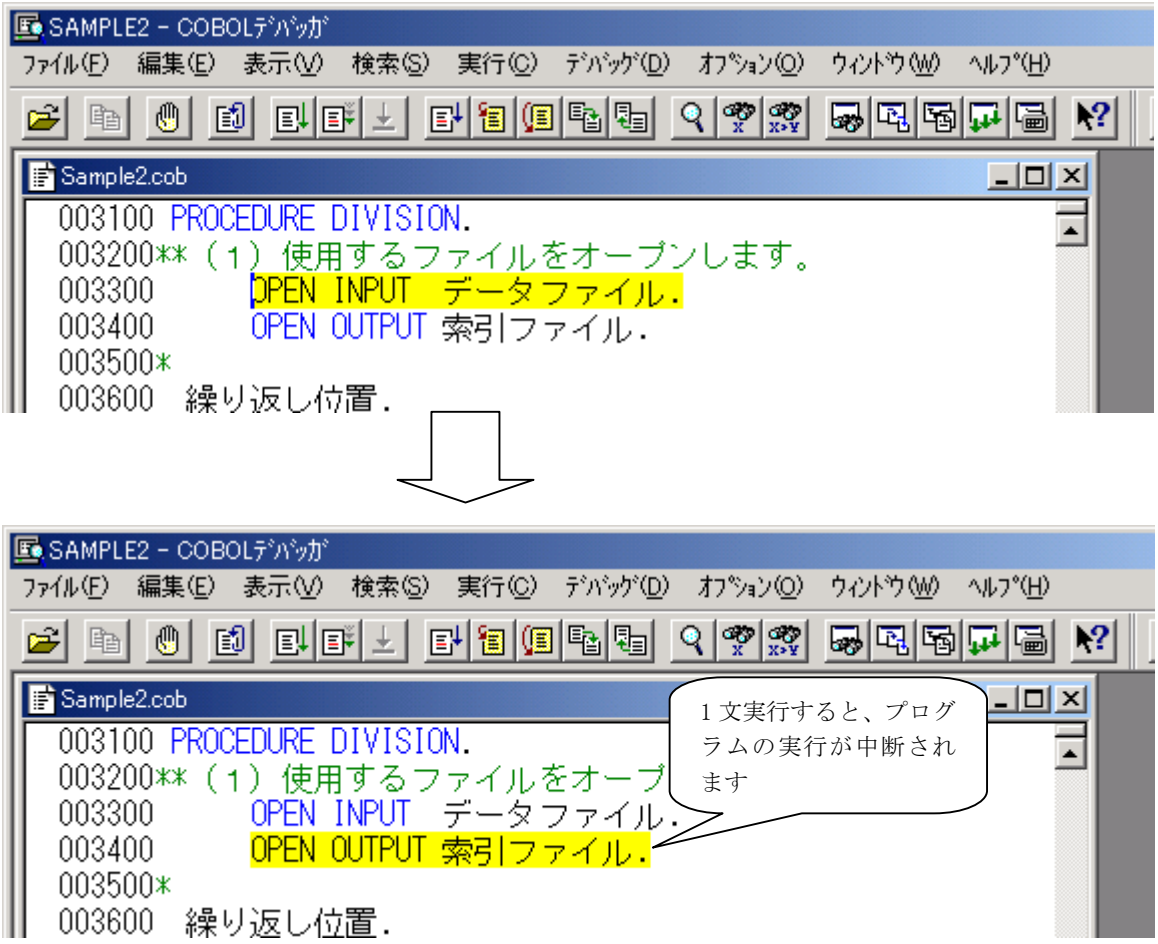
1.3.3 デバッグ操作

ここでは、次に示すようなデバッグ操作について説明します。

- 1 文だけ実行したら実行を中断する
- ある文に到達したら実行を中断する
- あるデータの値を確認する
- ある条件が成立したら実行を中断する

◆ 1 文だけ実行したら実行を中断する

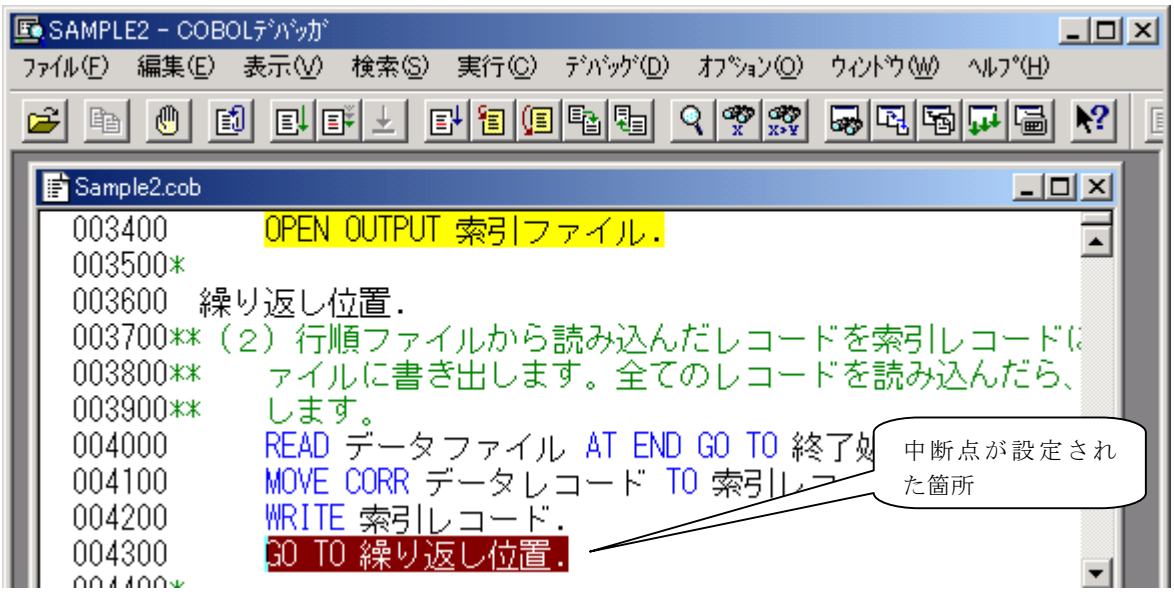
ソースプログラムの 1 文だけ実行したら実行を中断するには、デバッガの「実行」メニューから「ステップイン」を選択するか、F8 キーを押します。



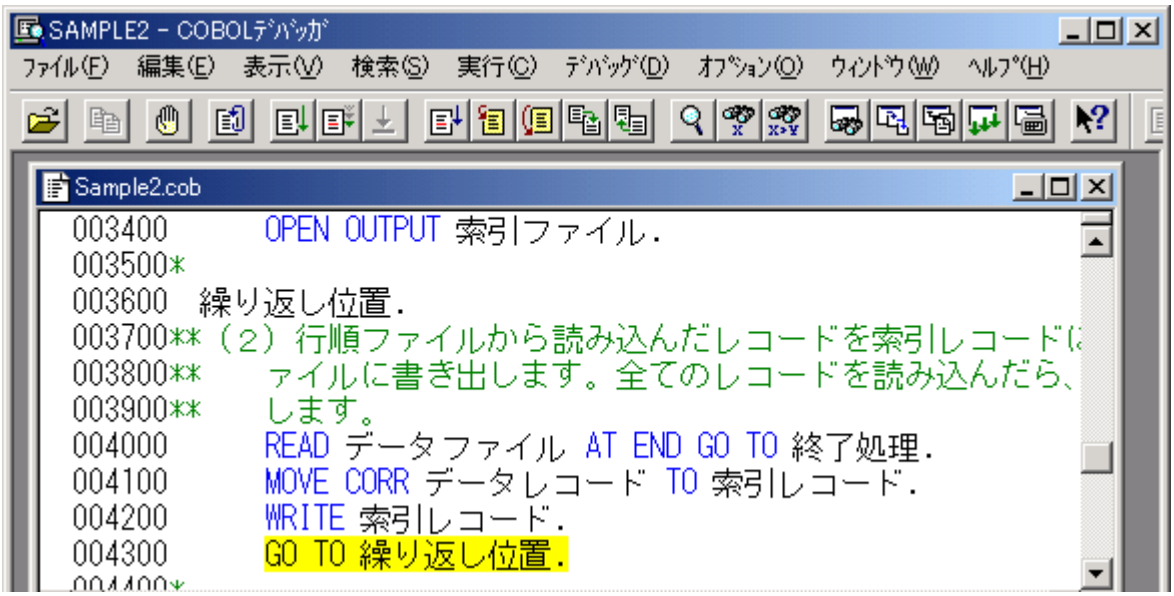
◆ ある文に達したら実行を中断する

ソースプログラム中のある文に達したら、プログラムの実行を中断してデバッグ操作を可能にするには、中断点を設定します。中断点を設定すると、中断点を設定した前の文の処理でプログラムが中断されます。
中断点を設定してのデバッグ操作の手順を説明します。

(1) 中断点を設定するには、デバッガの COBOL ソースプログラムが表示されているウィンドウで、中断点を設定する行にカーソルを置きます。
(2) マウスの右クリックのメニューから「中断点を設定」を選択します。
(3) 中断点を設定すると、行の色が茶色に変わります。



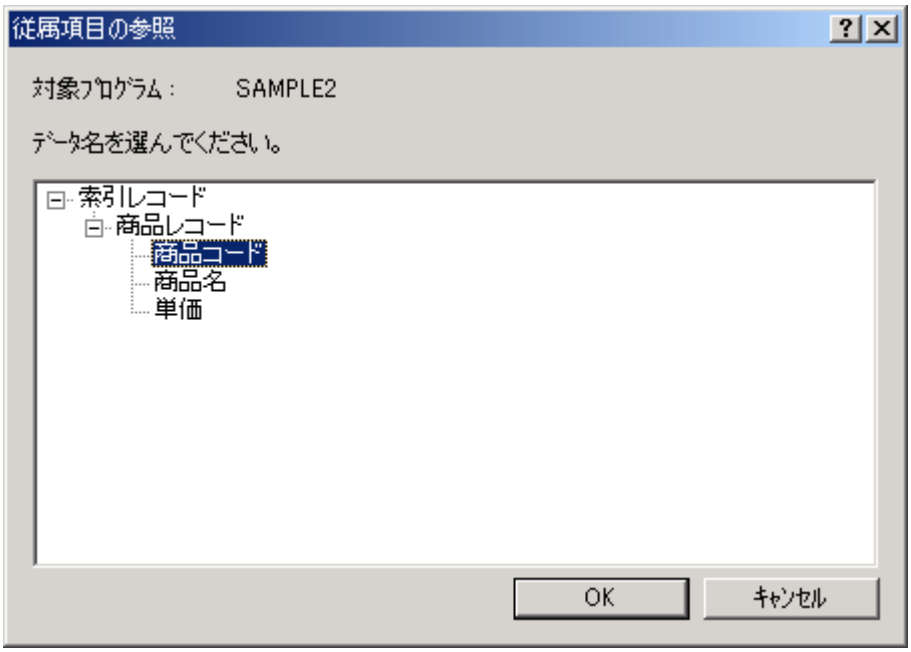
(4) デバッガの「実行」メニューから「実行」を選択してプログラムを実行します。
(5) 中断点を設定した文の前の処理でプログラムが中断され、中断点を設定した文が黄色（現在プログラムが中断している文）に変わります。



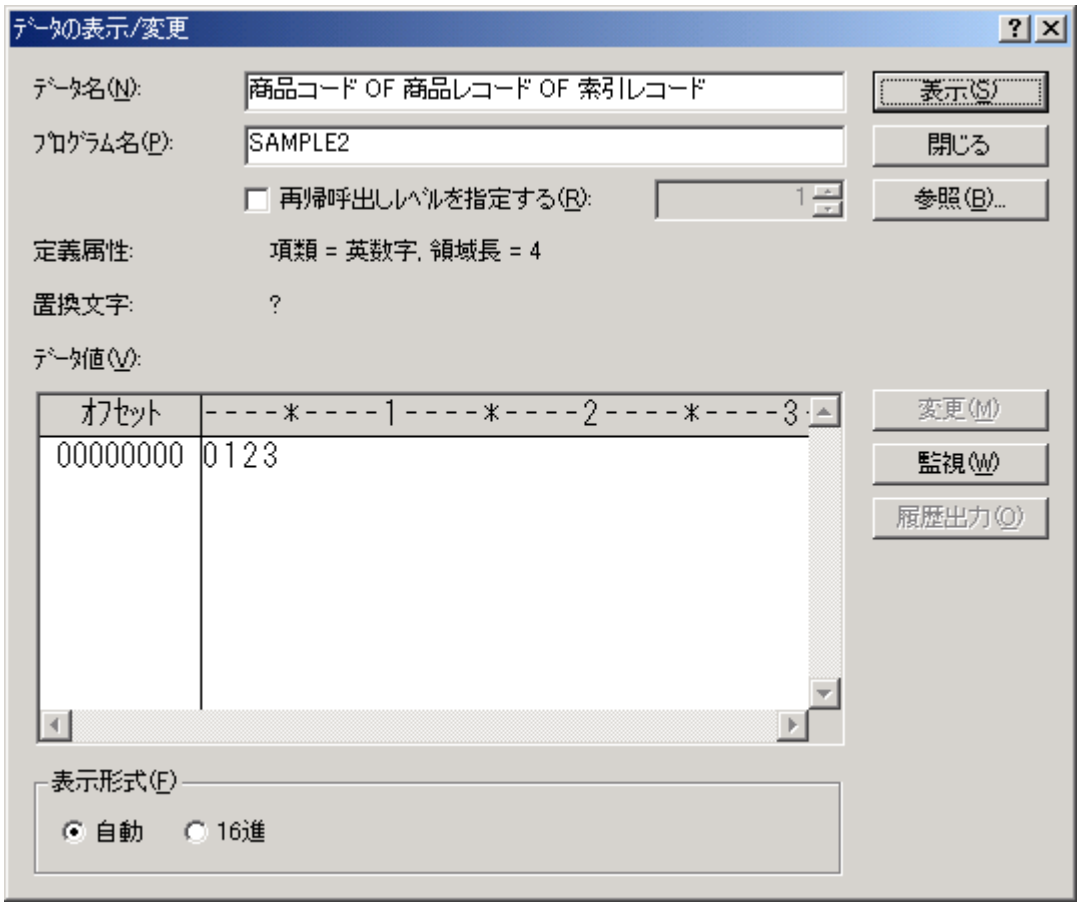
◆ あるデータの値を確認する

あるデータの現在の値を確認するには、「データの表示／変更」ウィンドウを使用します。
ここでは、データ名「商品コード OF 商品レコード OF 索引レコード」の値を確認する例を元に、データの値を確認する手順を説明します。

(1)デバッガの「デバッグ」メニューから「データの表示／変更」を選択して、「データの表示／変更」ウィンドウを表示します。
(2)「データの表示／変更」ウィンドウの「データ名」に「索引レコード」を入力します。
(3)「参照」ボタンを選択して、「従属項目の参照」ウィンドウを表示します。
(4)「索引レコード」に従属する項目がツリー状で表示されますので、「商品レコード」に属する「商品コード」を選択し、「OK」ボタンを選択します。

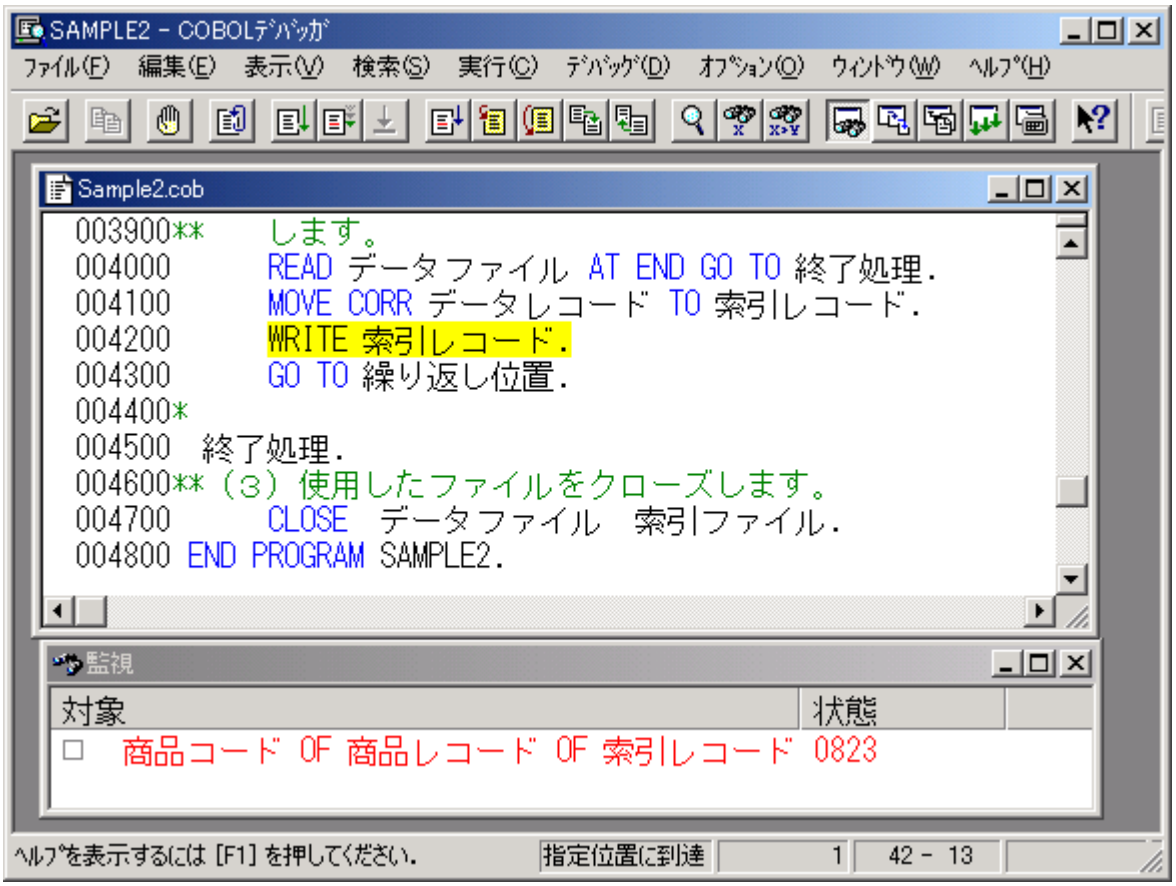


(5)「データの表示/変更」ウィンドウの「データ名」に「商品コード OF 商品レコード OF 索引レコード」が表示されます。「表示」ボタンを選択すると、「データ値」に「商品コード OF 商品レコード OF 索引レコード」の値が表示されます。



(6)プログラムを実行しながらデータの値を確認するには、「監視」ボタンを選択します。
(7)デバッガのウィンドウ内に、データ監視のウィンドウが表示されます。
(8)「閉じる」ボタンを選択して「データの表示／変更」ウィンドウを閉じます。

(9) プログラムを実行しながら、データの値が変化するのを監視します。

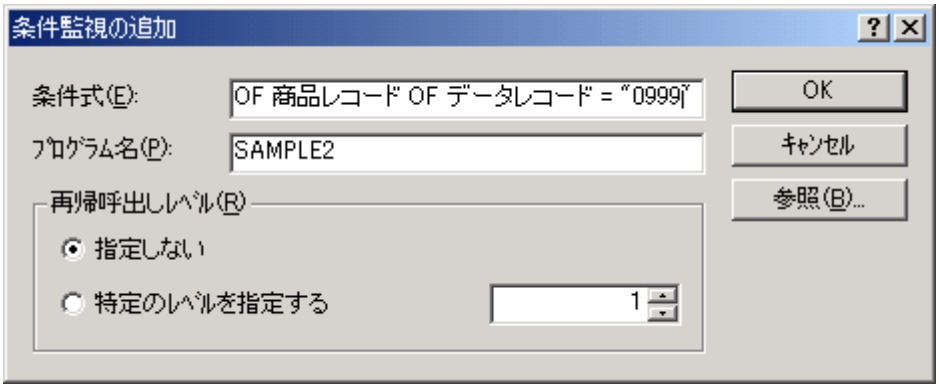


◆ ある条件が成立したら実行を中断する

例えば、「データＡの値がＢになったらデバッグ操作を行う」など、ある条件が成立したらプログラムの実行を中断してデバッグ操作を可能にするには、条件監視を行います。

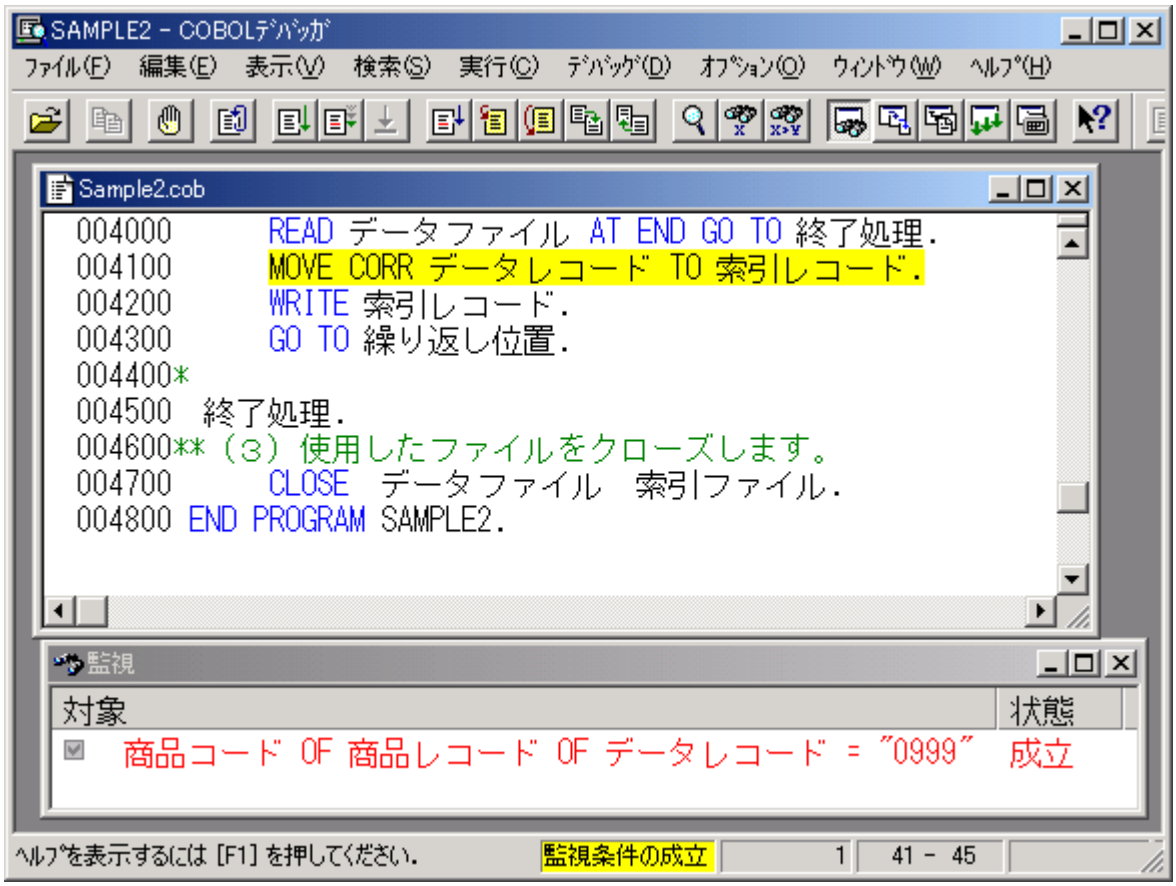
ここでは、データ名「商品コード OF 商品レコード OF データレコード」が「0999」という値になったら中断するようにする例を元に、条件監視を設定する手順を説明します。

(1)デバッガの「デバッグ」メニューから「条件監視の追加」を選択して、「条件監視の追加」ウィンドウを表示します。
(2)「条件監視の追加」ウィンドウの「条件式」に「データレコード」と入力します。
(3)「参照」ボタンを選択して、「従属項目の参照」ウィンドウを表示します。
(4)「データレコード」に従属する項目がツリー上で表示されますので、「商品レコード」に属する「商品コード」を選択し、「OK」ボタンを選択します。
「条件監視の追加」ウィンドウの「条件式」に「商品コード OF 商品レコード OF データレコード」が表示されます。
(5)「条件監視の追加」ウィンドウの「条件式」に条件式の右辺として「= “0999”」を追加し、「OK」ボタンを選択します。



(6)デバッガのウィンドウ内にデータ監視のウィンドウが表示され、設定した条件式が表示されます。
(7)デバッガの「デバッグ」メニューから「実行」を選択してプログラムを実行します。

(8) 条件式が成立した時点でプログラムの実行が中断します。



1.3.4 デバッガの終了

デバッガの「ファイル」メニューから「デバッガの終了」を選択し、デバッガを終了します。

MEMO

第 2 章 画面帳票アプリケーションの開発

本章では、画面帳票定義体を使用した画面帳票アプリケーションについて説明します。また、基本的な画面帳票アプリケーションを作成する方法について説明します。

2.1 概要
2.2 表示ファイルのプログラミング
2.3 画面帳票定義体の作成
2.4 画面帳票定義体の確認
2.5 プログラムの作成
2.6 プログラムのデバッグ

2.1 概要

NetCOBOL シリーズで作成できる画面帳票アプリケーション、および、本章で作成する画面帳票アプリケーションの概要について説明します。

2.1.1 画面帳票アプリケーションの概要

NetCOBOL シリーズでは、「画面定義体」と呼ばれる画面フォーマット、「帳票定義体」と呼ばれる帳票フォーマットおよび帳票定義体に重畳する「オーバレイ定義体」を使用することにより、COBOL によるきめ細かい画面帳票アプリケーションを作成することができます。

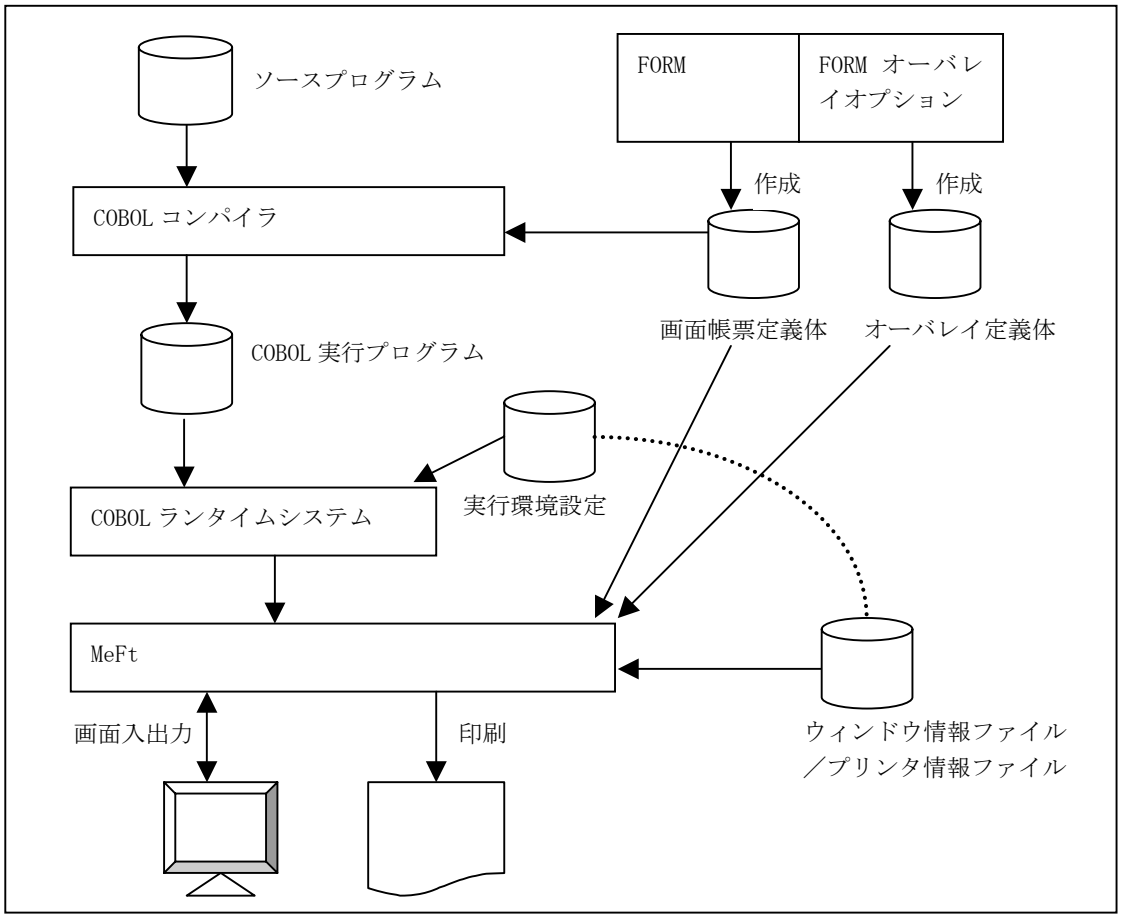
画面定義体および帳票定義体には、項目の位置、項目の種別（どのような種類のデータを扱う項目か、固定的な項目かなど）、項目の属性（文字の大きさ、色など）、罫線や網がけといった装飾などを定義します。また、オーバレイ定義体には固定的な文字や図形などを定義します。

画面定義体、帳票定義体およびオーバレイ定義体は、COBOL プログラムから独立しているため、作成や変更が容易です。なお、画面定義体と帳票定義体を総称して、「画面帳票定義体」と呼びます。

画面帳票定義体およびオーバレイ定義体を使用したプログラムの開発および実行には、NetCOBOL のほかに次のツールも使用します。これらの製品は NetCOBOL シリーズの NetCOBOL Standard および NetCOBOL Professional に含まれています。

名称	説明
FORM	画面帳票定義体を画面イメージで設計するツール
FORM オーバレイオプション	オーバレイ定義体を画面イメージで設計するツール
MeFt	画面帳票定義体およびオーバレイ定義体を元に、画面表示および帳票印刷を行うライブラリ

COBOL、FORM、FORM オーバレイオプション、MeFt の関連図を示します。



COBOL プログラムから FORM、MeFt を使用して画面入出力を行う場合、表示ファイルによるアプリケーションを作成します。表示ファイルによる画面帳票入出力では、通常のファイルを扱うのと同じように WRITE 文や READ 文を使用します。つまり、WRITE 文で画面やプリンタへの出力を行い、READ 文で画面から入力します。

プログラムは、画面およびプリンタとのデータの受渡し手段としてレコードを使用します。画面帳票定義体に定義されたデータ項目のレコードは、COBOL の COPY 文を使って、翻訳時にプログラムに取り込むことができます。そのため、画面帳票の入出力のためのレコードの定義を COBOL プログラムに記述する必要はありません。

なお、データ項目のウィンドウ内での位置や印刷位置など、ウィンドウやプリンタの制御は MeFt が行うため、COBOL プログラムでは意識する必要がありません。

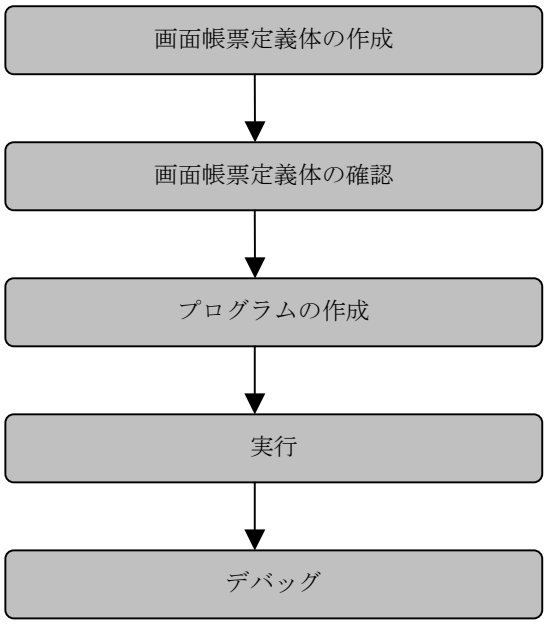
2.1.2 作成するアプリケーションについて

この章で作成するアプリケーションでは、1 つの画面定義体と 1 つの帳票定義体を使用します。画面定義体を使用してデータを入力し、入力されたデータは帳票定義体を使用して印刷します。COBOL ソースプログラムは、予め作成されているものを使用します。

なお、画面帳票定義体や COBOL ソースプログラムなどアプリケーションの資源は、C:¥EDUCATION に格納することとします。

2.1.3 アプリケーション開発の流れ

画面帳票定義体を使用した画面帳票アプリケーションの開発の流れを次に示します。



2.2 表示ファイルのプログラミング

表示ファイル機能を使って画面入出力を行うときのプログラム記述について、COBOL の各部ごとに説明します。

環境部(ENVIRONMENT DIVISION)

表示ファイルを定義します。表示ファイルは、通常のファイルを定義するときと同様に、入出力節のファイル管理段落にファイル管理記述項を記述します。

以下に、COBOL プログラムの記述例とファイル管理記述項に指定できる内容を示します。

◆ COBOL プログラムの記述例

```
ENVIRONMENT DIVISION.
  INPUT-OUTPUT SECTION.
   FILE-CONTROL.
     SELECT  ディスプレイファイル ASSIGN TO  GS-DSPFILE
       SYMBOLIC DESTINATION IS  "DSP"
       FORMAT               IS  DSP-FORMAT
       GROUP                IS  DSP-GROUP
       PROCESSING MODE      IS  DSP-MODE
       UNIT CONTROL         IS  DSP-CONTROL
       SELECTED FUNCTION    IS  DSP-ATTN
       FILE  STATUS         IS  DSP-STATUS1 DSP-STATUS2.
     SELECT  プリンタファイル  ASSIGN TO  GS-PRTFILE
       SYMBOLIC DESTINATION IS  "PRT"
       FORMAT               IS  PRT-FORMAT
       GROUP                IS  PRT-GROUP
       PROCESSING MODE      IS  PRT-MODE
       UNIT CONTROL         IS  PRT-CONTROL
       FILE  STATUS         IS  PRT-STATUS1 PRT-STATUS2.
```

表示ファイル（画面機能）の定義

表示ファイル（帳票機能）の定義

◆ ファイル管理記述項に指定できる内容

指定場所	情報の種類	指定する内容	必須／任意
SELECT 句	ファイル名	COBOL プログラム中で使用するファイル名を指定します。	必須
ASSIGN 句	ファイル参照子	“GS-ファイル識別名”の形式で指定します。このファイル識別名は、実行時に使用するウィンドウ情報ファイルまたはプリンタ情報ファイルのファイル名を設定する環境変数情報になります。	必須
SYMBOLIC DESTINATION 句	あて先種別	データの入出力のあて先を指定します。画面機能では”DSP”（または省略）、帳票機能では”PRT”を指定します。	任意（画面） 必須（帳票）
FORMAT 句	データ名	作業場所節または連絡節で、8 桁の英数字項目として定義したデータ名を指定します。このデータ名には、画面帳票定義体名を設定します。	必須
GROUP 句	データ名	作業場所節または連絡節で、8 桁の英数字項目として定義したデータ名を指定します。このデータ名には、入出力の対象となる項目群名を設定します。	必須
PROCESSING MODE 句	データ名	作業場所節または連絡節で、2 桁の英数字項目として定義したデータ名を指定します。このデータ名には、入出力の処理種別を設定します。	任意
UNIT CONTROL 句	データ名	作業場所節または連絡節で、6 桁の英数字項目として定義したデータ名を指定します。このデータ名には、制御情報を設定します。	任意

指定場所	情報の種類	指定する内容	必須／任意
SELECTED FUNCTION 句	データ名	作業場所節または連絡節で、4 桁の英数字項目として定義したデータ名を指定します。このデータ名には、READ 文完了時にアテンション情報が通知されます。画面機能で指定します。	任意
FILE STATUS 句	データ名	作業場所節または連絡節で、2 桁および 4 桁の英数字項目として定義したデータ名を指定します。このデータ名には、入出力処理の実行結果が設定されます。なお、4 桁のデータ名の領域には、実行結果の詳細情報が設定されます。	任意

※詳細は、NetCOBOL 使用手引書「8.5.4 プログラムの記述」（帳票機能の場合）および「9.2.5 プログラムの記述」（画面機能の場合）をご覧ください。

データ部(DATA DIVISION)

データ部には、表示ファイルのレコード定義およびファイル管理記述項に指定したデータの定義を記述します。表示ファイルのレコードは、XMDLIB を指定した COPY 文を使って画面帳票定義体から取り込むことができます。
以下に COBOL プログラムの記述例を示します。

```
DATA DIVISION.
FILE SECTION.
FD ディスプレイファイル.
  COPY ADDRDSP OF XMDLIB.
FD プリンタファイル.
  COPY ADDRPRT OF XMDLIB.

WORKING-STORAGE SECTION.
 01  DSP-FORMAT        PIC X(08).
 01  DSP-GROUP         PIC X(08).
 01  DSP-MODE          PIC X(02).
 01  DSP-CONTROL       PIC X(06).
 01  DSP-ATTN          PIC X(04).
 01  DSP-STATUS1       PIC X(02).
 01  DSP-STATUS2       PIC X(04).

 01  PRT-FORMAT        PIC X(08).
 01  PRT-GROUP         PIC X(08).
 01  PRT-MODE          PIC X(02).
 01  PRT-CONTROL       PIC X(06).
 01  PRT-STATUS1       PIC X(02).
 01  PRT-STATUS2       PIC X(04).
```

各表示ファイルのレコードを画面帳票定義体から COPY 文で取り込み

FILE-CONTROL で定義した表示ファイル（画面機能）の各データ項目の定義

FILE-CONTROL で定義した表示ファイル（帳票機能）の各データ項目の定義

手続き部(PROCEDURE DIVISION)

画面入出力および帳票出力の開始にはOPEN文を、終了にはCLOSE文を使用します。また、画面および帳票の入出力には、通常のファイル処理を行うときと同様に、READ文およびWRITE文を使用します。

手続き部について、画面機能と帳票機能に分けて説明します。

◆ 画面機能

COBOLプログラムの記述例

```
OPEN I-O ディスプレイファイル.
MOVE "ADDRDSP" TO DSP-FORMAT.
MOVE "@ALLF"   TO DSP-GROUP.
MOVE "  "      TO DSP-MODE.
WRITE 住所録入力画面.
MOVE "@ALLF"   TO DSP-GROUP.
MOVE "NE"      TO DSP-MODE.
READ ディスプレイファイル.
EVALUATE DSP-ATTN
    WHEN "PRT "
        :
    WHEN "END "
        :
END-EVALUATE.
CLOSE ディスプレイファイル.
```

画面機能のREAD文およびWRITE文について

画面を表示するときには表示ファイルのレコード名を指定したWRITE文を、画面からデータを読み込むときには表示ファイルを指定したREAD文を使います。

WRITE文を実行する前には、画面出力に使用する画面定義体の名前をFORMAT句に指定したデータ名に設定し、出力の対象となる画面定義体の項目群名をGROUP句に指定したデータ名に設定します。

また、画面出力の処理種別(モード)をPROCESSING MODE句に指定したデータ名に設定します。

入力の対象となる画面定義体の項目群名と入力の処理種別(モード)が出力時と異なるときは、READ文を実行する前に、画面定義体の項目群名をGROUP句に指定したデータ名に設定し、処理種別(モード)をPROCESSING MODE句に指定したデータ名に設定します。

画面定義体への入力中にファンクションキーなどが押されると、READ文が完了してCOBOLプログラムに入力結果の情報が通知されます。そのとき、SELECTED FUNCTION句に指定したデータ名にアテンション情報が通知されます。

◆ 帳票機能

COBOL プログラムの記述例

```
OPEN OUTPUT プリンタファイル.
MOVE "ADDRPRT" TO PRT-FORMAT.
MOVE "@ALLF"   TO PRT-GROUP.
MOVE "  "      TO PRT-MODE.
WRITE 住所録印刷帳票.
CLOSE プリンタファイル.
```

帳票機能の WRITE 文について

帳票を出力するときには、表示ファイルのレコード名を指定した WRITE 文を実行します。

WRITE 文を実行する前には、印刷に使用する帳票定義体の名前を FORMAT 句に指定したデータ名に設定し、印刷の対象となる帳票定義体の項目群名を GROUP 句に指定したデータ名に設定します。

また、帳票出力の処理種別(モード)を PROCESSING MODE 句に指定したデータ名に設定します。

エラー処理

表示ファイルの各命令(OPEN 文、WRITE 文、READ 文、CLOSE 文)の実行結果は、FILE STATUS 句に指定したデータ名に通知されます。FILE STATUS 句に指定したデータ名のうち 2 桁のデータ名の領域には、成功時は“00”が通知され、不成功時は“04”、“90”、“99”、“9E”のいずれかが通知されます。また、4 桁のデータ名の領域には、詳細結果として上記の 4 種類のエラーに MeFt の通知コードが付加されたものが通知されます。例えば、MeFt の通知コードが“22”であった場合、FILE STATUS 句に指定した 4 桁のデータ領域には“9022”が通知されます。

入出力文の後に、このデータ名の内容をチェックする文を記述することによって、プログラムで入出力文の結果に応じた処理手続きを実行することができます。

FILE STATUS 句の使用例を次に示します。

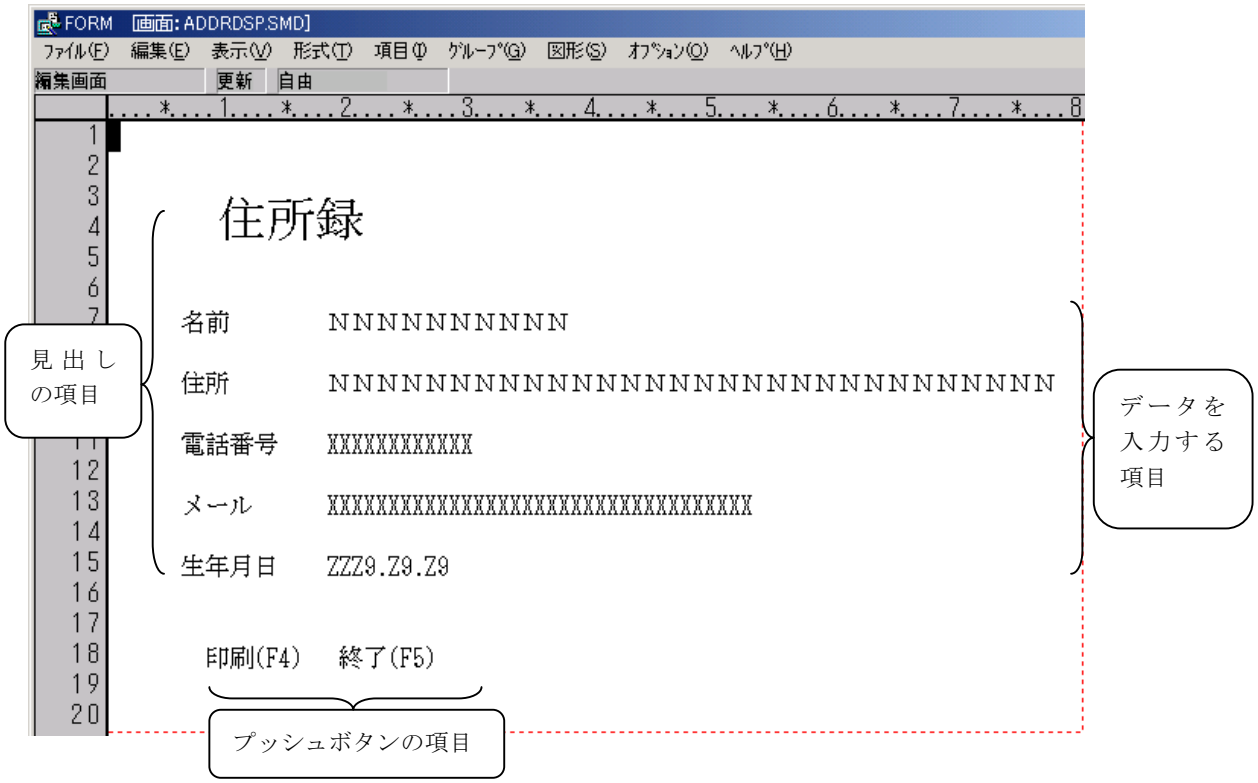
```
SELECT  ディスプレイファイル ASSIGN TO  GS-DSPFILE
    :
  FILE  STATUS         IS  DSP-STATUS1 DSP-STATUS2.
    :
WORKING-STORAGE SECTION.
01  DSP-STATUS1       PIC X(02).
01  DSP-STATUS2       PIC X(04).
    :
PROCEDURE DIVISION.
OPEN ディスプレイファイル.
    :
WRITE 住所録入力画面.
IF DSP-STATUS2 NOT = "0000" THEN
    CLOSE ディスプレイファイル
END-IF.
```

2.3 画面帳票定義体の作成

画面帳票定義体の作成について説明します。
なお、本章では基本的な操作を説明しています。操作の詳細な説明については、FORM のマニュアルをご覧ください。

2.3.1 画面定義体の作成

ここでは、次に示すような画面定義体を作成する方法を説明します。

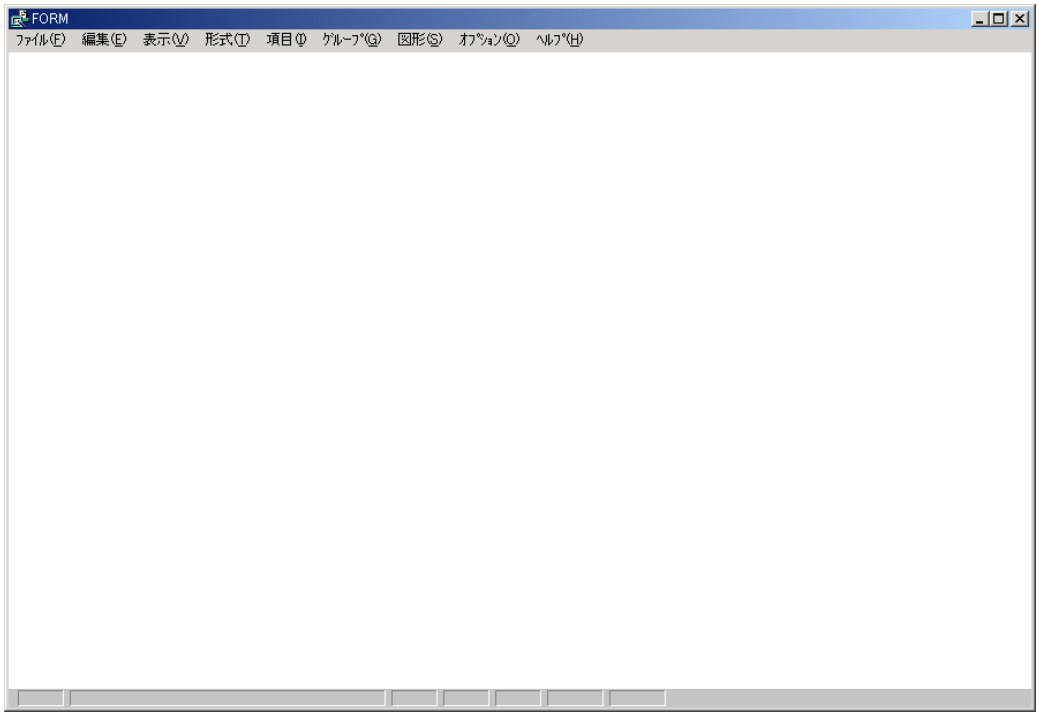


次に示す手順で画面定義体を作成していきます。

(1)画面定義体の属性設定
　画面定義体の大きさ（縦幅／横幅）など、画面定義体全体に対する属性を設定します。
(2)項目の配置と属性設定
　データを入力する項目の配置など、画面定義体のレイアウトを作成します。また、配置した項目に対する属性を設定します。
(3)アテンション情報の設定
　キーボードで押されたファンクションキーをプログラムに通知する情報（アテンション情報）を設定します。
(4)項目のボタン化
　画面定義体に配置された項目のうち、一部の項目をボタンとして使用する指定を行います。

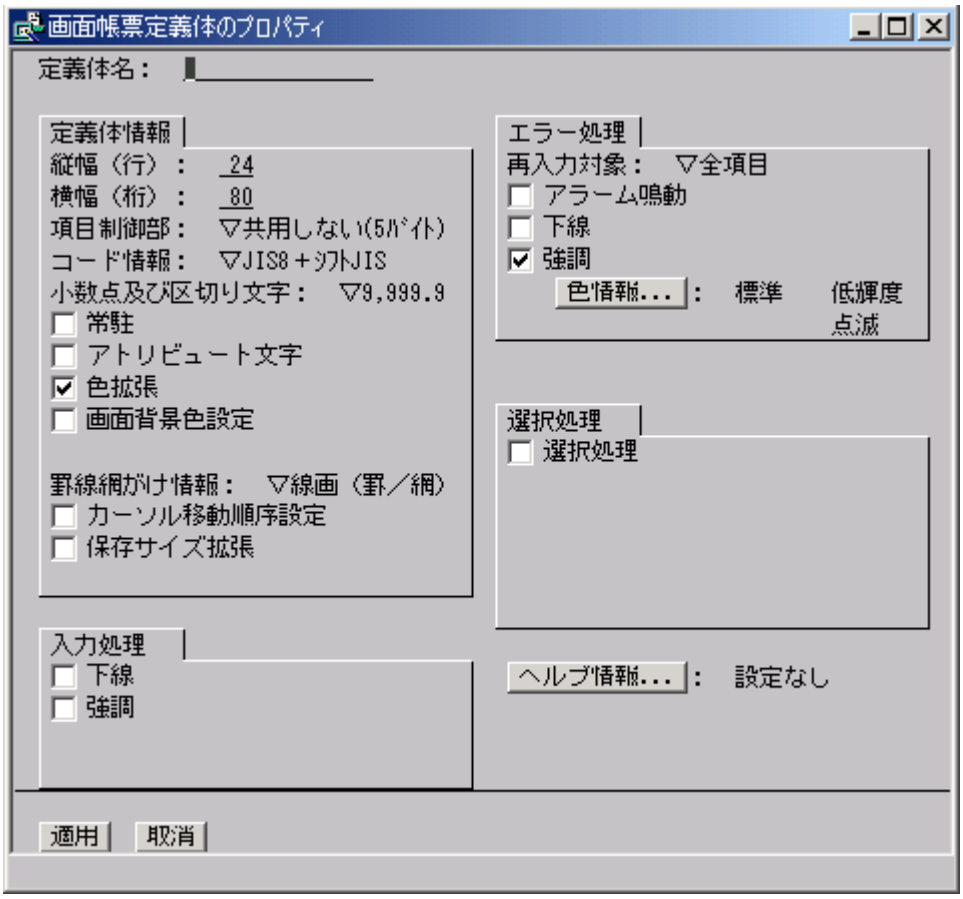
◆ FORM の起動

［スタート］ボタン－［プログラム］－［NetCOBOL］－［FORM］から［FORM］を選択して、FORM を起動します。
FORM の編集画面が表示されます。



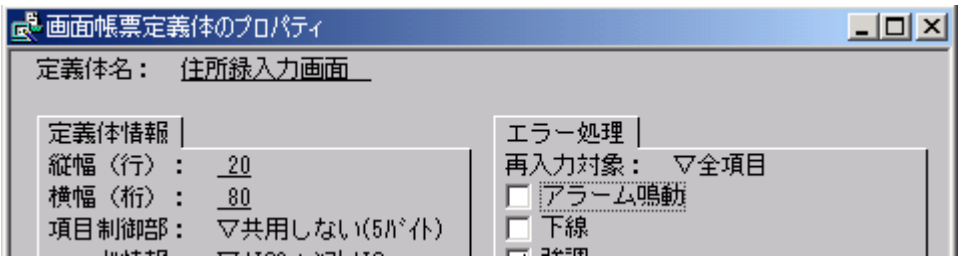
◆ 画面定義体の属性設定

(1)FORM の[ファイル]メニュー－[新規作成]から[画面定義体]を選択します。
(2)FORM の[ファイル] メニュー－[プロパティ]から[画面帳票定義体]を選択します。
(3)画面定義体のプロパティ画面が表示されます。



(4)プロパティの画面で以下に示す内容を設定します。

設定箇所	設定内容
定義体名	住所録入力画面
縦幅(行)	20
横幅(桁)	80



(5)プロパティの画面で「適用」を選択すると、編集画面に戻ります。

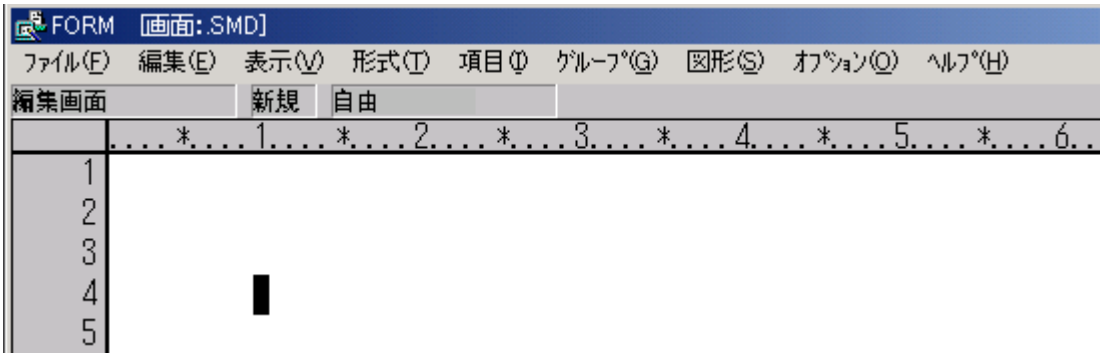
◆ 項目の配置と属性設定

項目を配置し、その属性を設定します。

項目を配置するには、FORM の「項目」メニューから配置したい項目の種類を選択し、表示されるプロパティの画面で属性を設定します。

ここでは、例として、見出しの“住所録”という文字列の項目を配置し、属性を設定する方法を説明します。

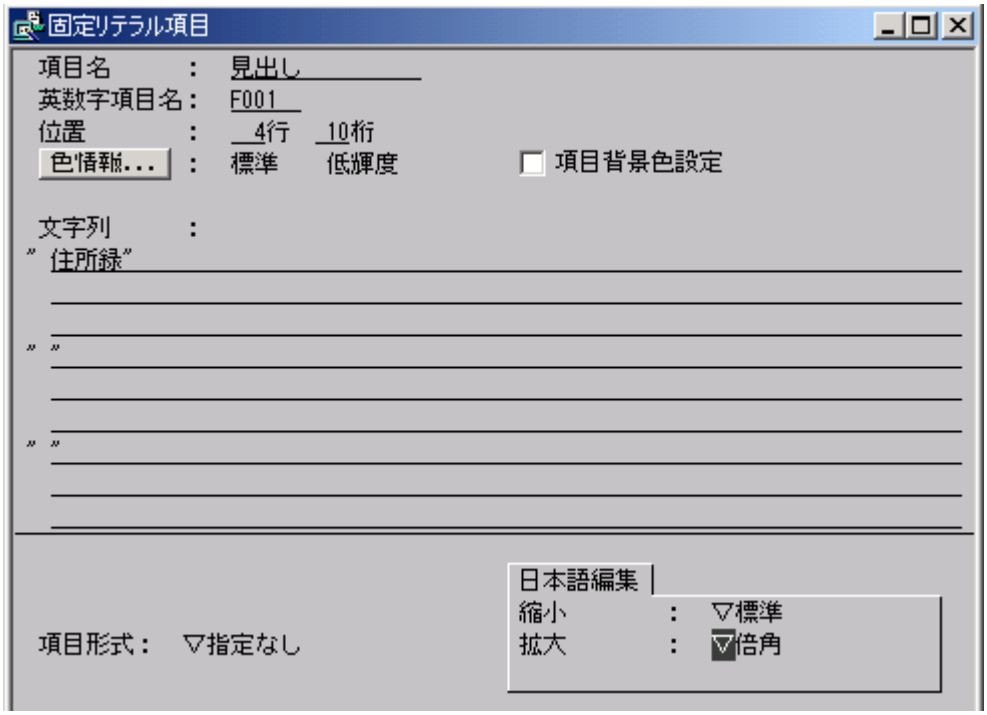
(1)編集画面で、見出しを配置したい位置にカーソルを移動します。



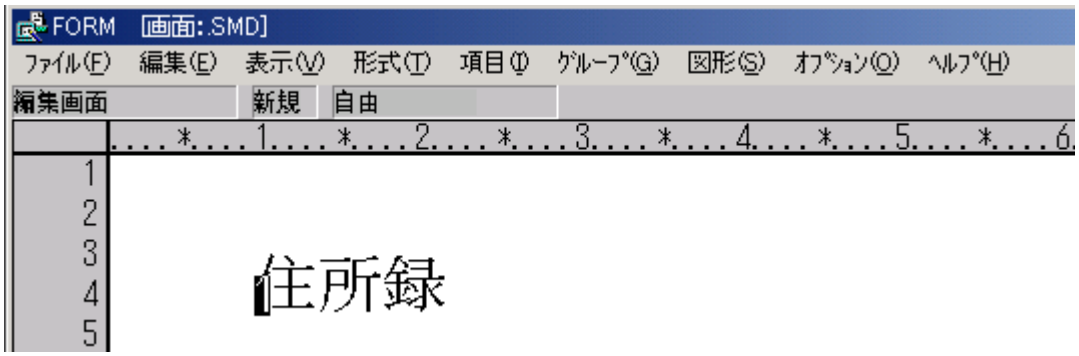
(2)見出しのように、表示する文字が固定されている項目を「固定リテラル項目」というため、FORM の[項目]メニューから[固定リテラル]を選択します。

(3)固定リテラル項目のプロパティ画面が表示されますので、次に示す内容を設定します。

設定箇所	設定内容
項目名	見出し
文字列	住所録
日本語編集の「拡大」	倍角



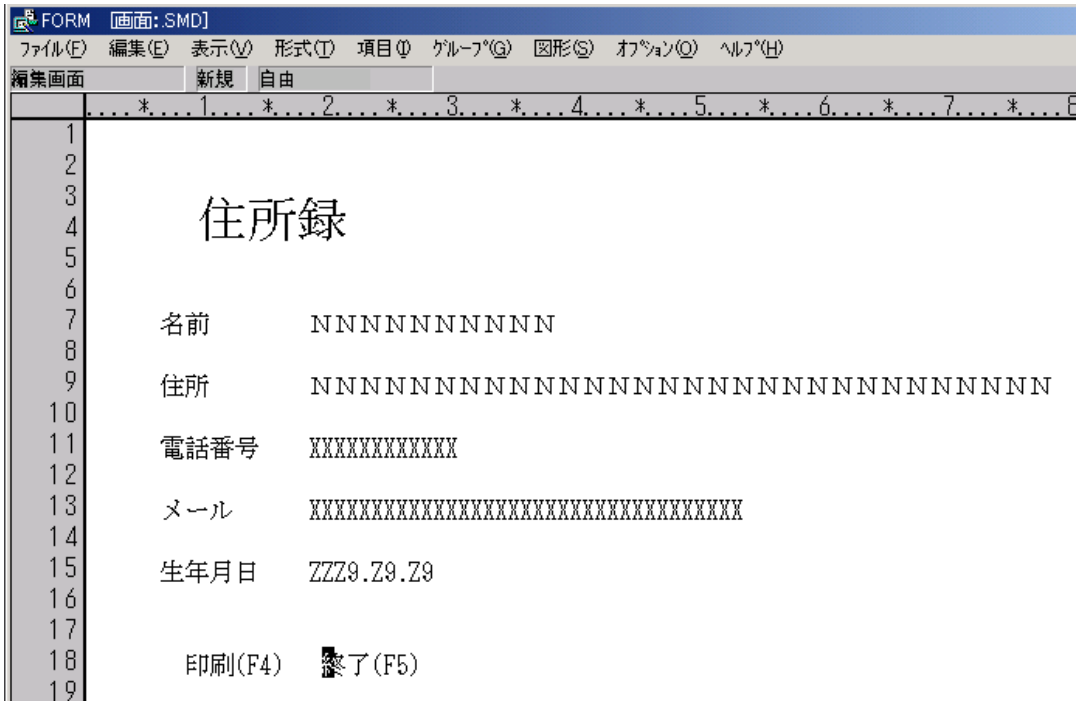
(4)「適用」ボタンを選択すると、プロパティ画面が終了し、見出しの項目が編集画面上に配置されます。



(5)その他の項目は、次に示す内容で設定します。

「項目」メニューでの項目の種類	項目名	文字列	横幅(文字数)	桁数	編集形式
固定リテラル	名前見出し	名前			
日本語	名前		10		
固定リテラル	住所見出し	住所			
日本語	住所		30		
固定リテラル	電話番号見出し	電話番号			
英数字	電話番号		12		
固定リテラル	メール見出し	メール			
英数字	メール		35		
固定リテラル	生年月日見出し	生年月日			
数字	生年月日			8	ZZZ9.Z9.Z9、全ゼロサプレスあり
固定リテラル	印刷ボタン	印刷(F4)			
固定リテラル	終了ボタン	終了(F5)			

全ての項目の配置、属性設定が終了すると、以下のような状態になります。



◆ アテンション情報の設定

アテンション情報とは

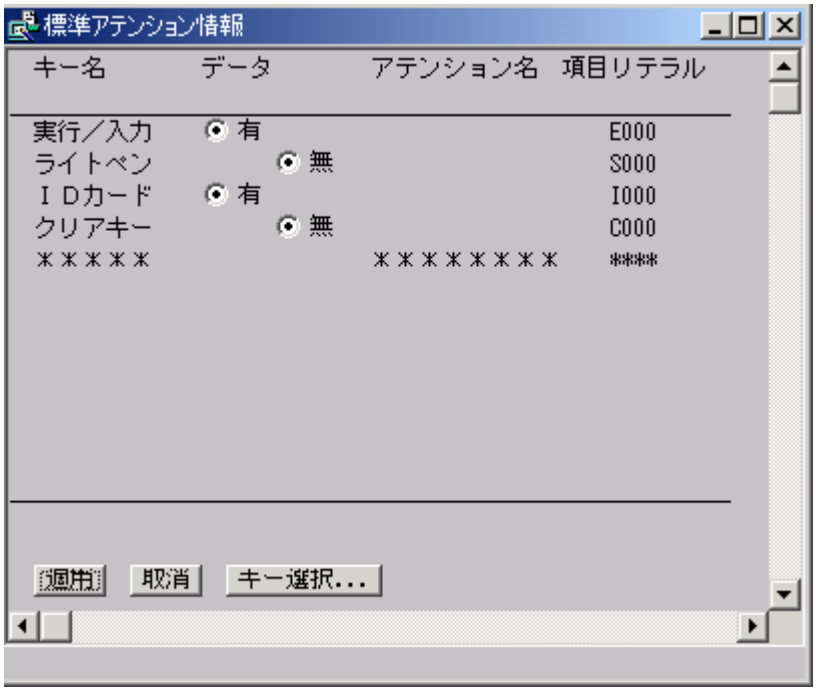
アテンション情報とは、キーボードで押されたファンクションキーをプログラムに通知する情報です。

画面定義体では、各ファンクションキーに対応するアテンション情報を設定します。プログラムでは、通知されるアテンション情報を参照すると、押されたファンクションキーを知ることができるため、アテンション情報によって処理を分けます。

利用者が設定するアテンション情報には、標準アテンション情報と拡張アテンション情報がありますが、ここでは、標準アテンション情報の設定方法を説明します。

標準アテンション情報の設定

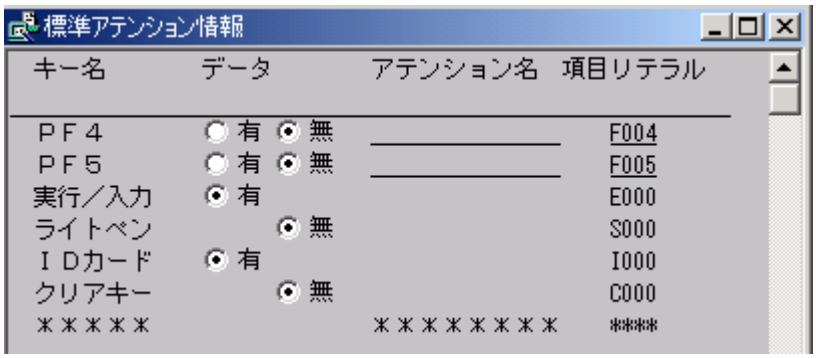
(1)FORMの[形式]メニューから[標準アテンション情報]を選択し、標準アテンション情報の設定画面を表示します。



(2)標準アテンション情報の設定画面の[キー選択]ボタンを選択し、キー選択の画面を表示します。

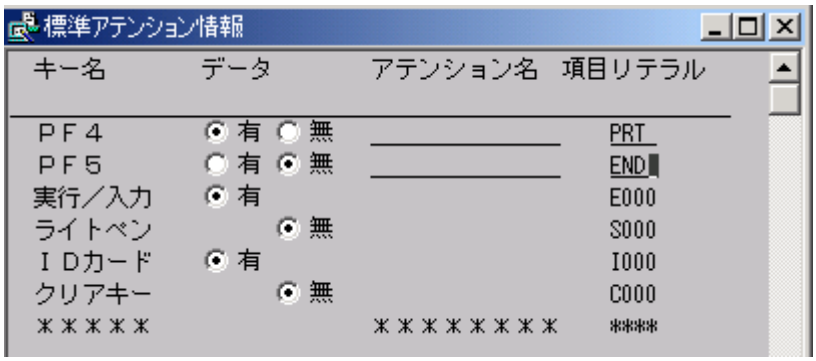
(3)キー選択の設定画面で、「PF4」と「PF5」をチェックします。

(4)「適用」ボタンを選択すると、キー選択の設定画面でチェックしたキーが、標準アテンション情報の設定画面に表示されます



(5)標準アテンション情報の設定画面では次に示す内容を設定し、「適用」ボタンを選択します。

キー名	データ	項目リテラル
ＰＦ４	有	PRT
ＰＦ５	無	END



＜データの有無について＞

ファンクションキーが押されたとき、画面に入力されたデータを有効にするか、無効にするかを指定します。

「有」の場合、入力されたデータを有効とし、押されたファンクションキーのアテンション情報と共にプログラムに通知します。

「無」の場合、押されたファンクションキーのアテンション情報だけをプログラムに通知し、入力データがあっても通知しません。

＜項目リテラルについて＞

ファンクションキーを識別する名前です。プログラムでは、READ 文のあとに通知される項目リテラルを判断し、押されたファンクションキーを判断できます。

◆ 項目のボタン化

画面定義体に定義された項目をボタン化します。なお、画面定義体では、ボタンとしてプッシュボタンのほかにチェックボックス、ラジオボタンが設定できます。
本章で作成する画面定義体では、プッシュボタンを設定します。

プッシュボタンは、次に示す手順で設定します。

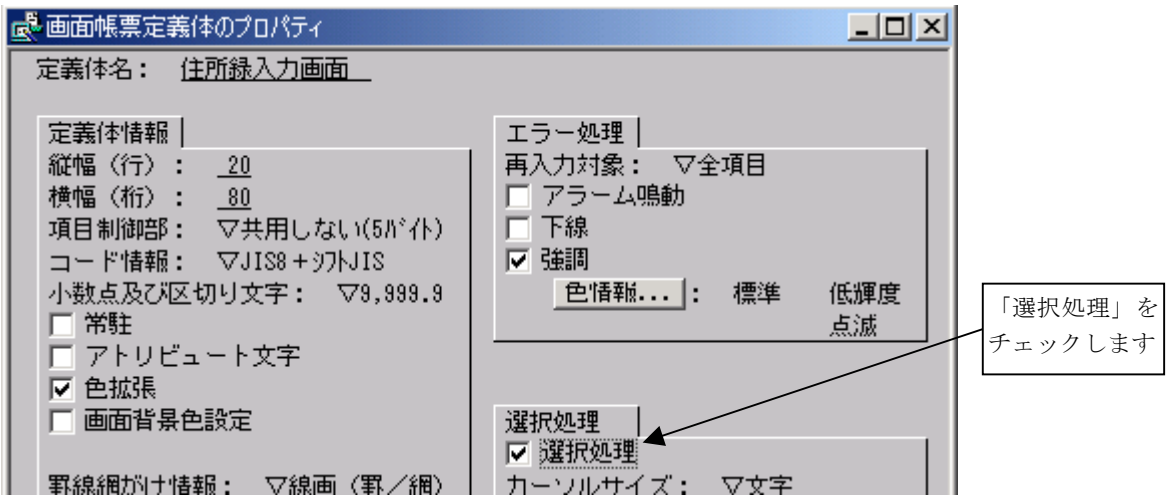
- 定義体でのボタン利用の指定
- 項目をボタンとして使用する指定
- 使用するボタンの種類や属性の指定

定義体でのボタン利用の指定

画面定義体でボタンを使用するには、画面定義体に対して選択処理を行うかどうかを設定する必要があります。
選択処理とは、画面内の項目に対する操作の通知をすることです。例えば、ボタンを押すといった項目の操作が選択処理にあたります。

選択処理を行うかどうかは、画面定義体のプロパティで指定します。

(1) FORM の[ファイル]メニュー－[プロパティ]から[画面帳票定義体]を選択し、画面定義体のプロパティ画面を表示します。
(2)表示されたプロパティ画面の「選択処理」をチェックします。



(3)「適用」ボタンを選択し、画面定義体のプロパティ画面を終了します。

項目をボタンとして使用する指定

プッシュボタンにする項目に選択属性を設定し、ボタン化する宣言をします。

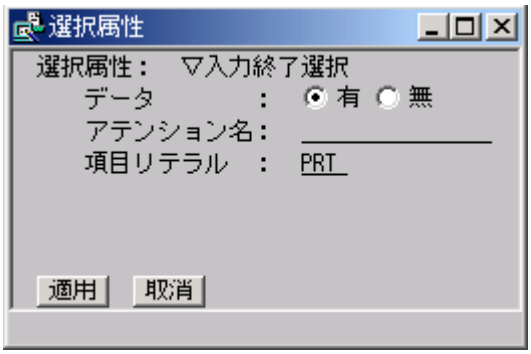
本章で作成する画面定義体では、「印刷(F4)」と「終了(F5)」という文字列の 2 つの項目をプッシュボタンにします。

(1)プッシュボタンにする項目「印刷(F4)」のプロパティ画面を表示します。

(2)表示されたプロパティ画面の「選択属性」ボタンを選択し、選択属性の設定画面を表示します。

(3)選択属性の設定画面では、そのボタンが押されたときに通知される情報を設定します。ここでは、データを「有」にし、ボタンが押されたときに通知される項目リテラルを設定するため、次に示す内容を設定します。

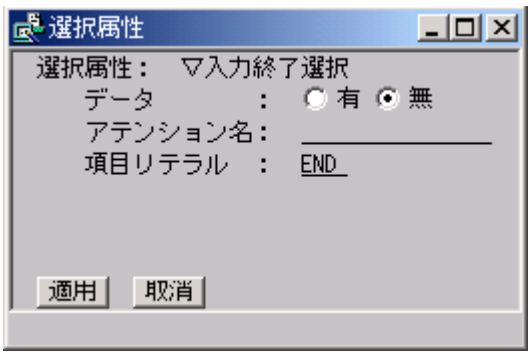
設定箇所	設定内容
選択属性	入力終了選択
データ	有
項目リテラル	PRT



(4)「適用」ボタンを選択し、選択属性の設定画面を終了します。

(5)「終了(F5)」の項目は、次に示す内容で選択属性を設定します。

設定箇所	設定内容
選択属性	入力終了選択
データ	無
項目リテラル	END

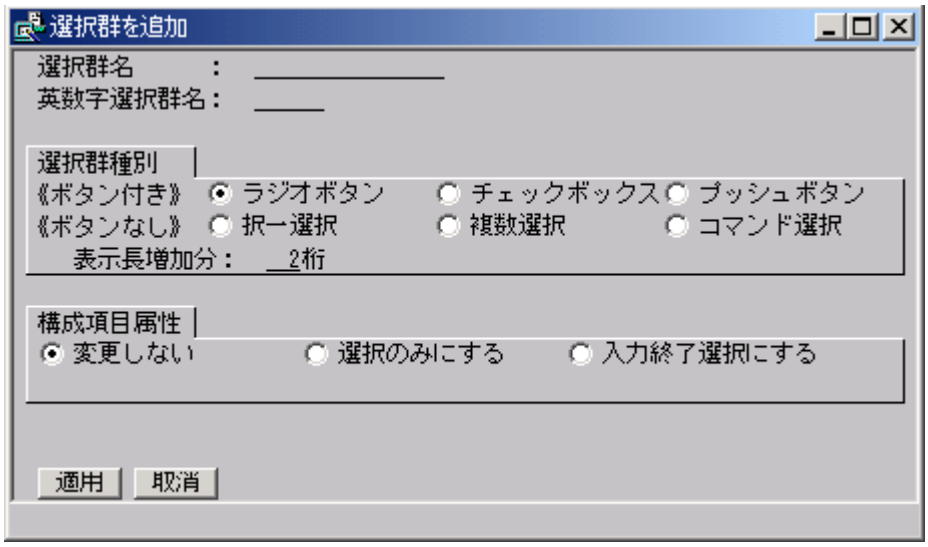


使用するボタンの種類や属性の指定

選択項目（選択属性を持った項目）は選択群を作成してグループ単位で利用します。

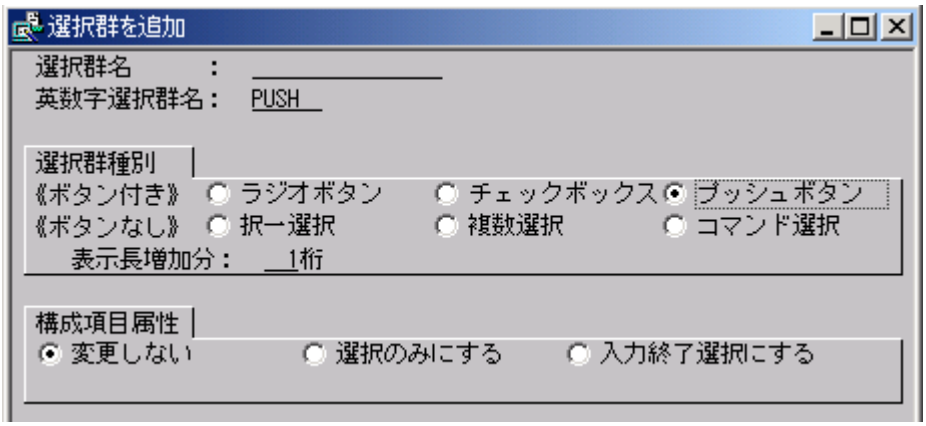
ボタンの種類や属性も、選択群に対してグループ単位で設定します。また、選択群を設定した後、選択群に属する項目を指定します。

(1-1)FORMの[グループ]メニューから[選択群を追加]を選択し、選択群の追加画面を表示します。



(1-2)選択群の追加画面では、次に示す内容を設定します。

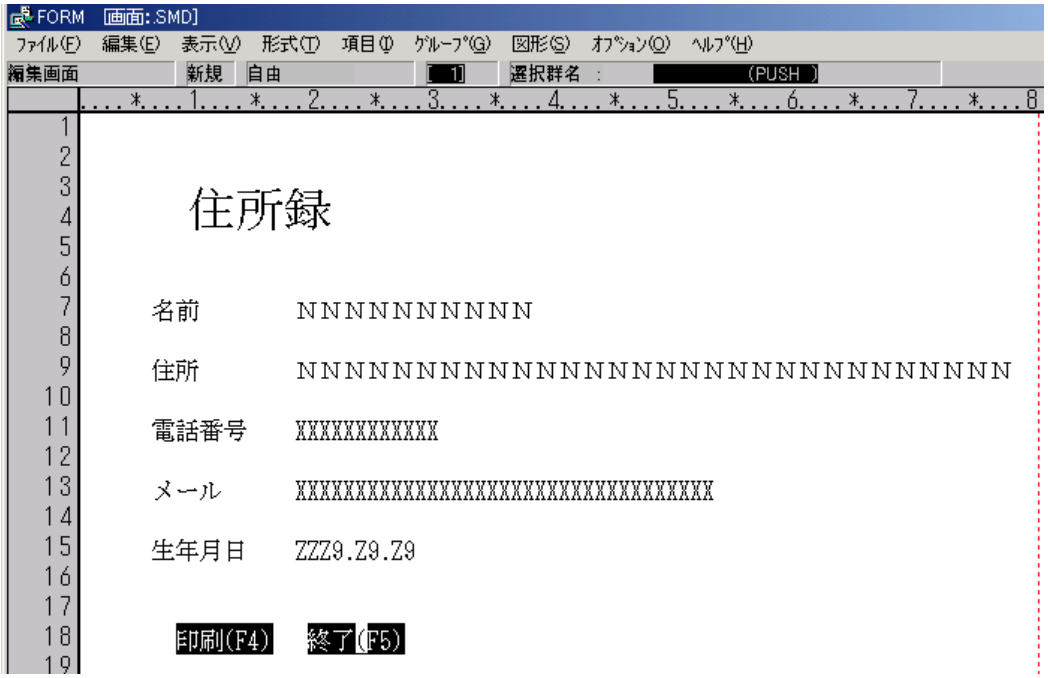
設定箇所	設定内容
英数字選択群名	PUSH
選択群種別	プッシュボタン
構成項目属性	変更しない



(1-3)「適用」ボタンを選択します。選択群の追加画面が終了し、編集画面に戻ります。

(2-1)編集画面は、選択群を構成する項目を表示する状態（構成項目表示）になっています。項目を選択群に追加するには、「印刷(F4)」の項目を選択し、マウスの右ボタンのメニューから「項目を追加」を選択します。選択群に含まれる項目は黒く反転します。

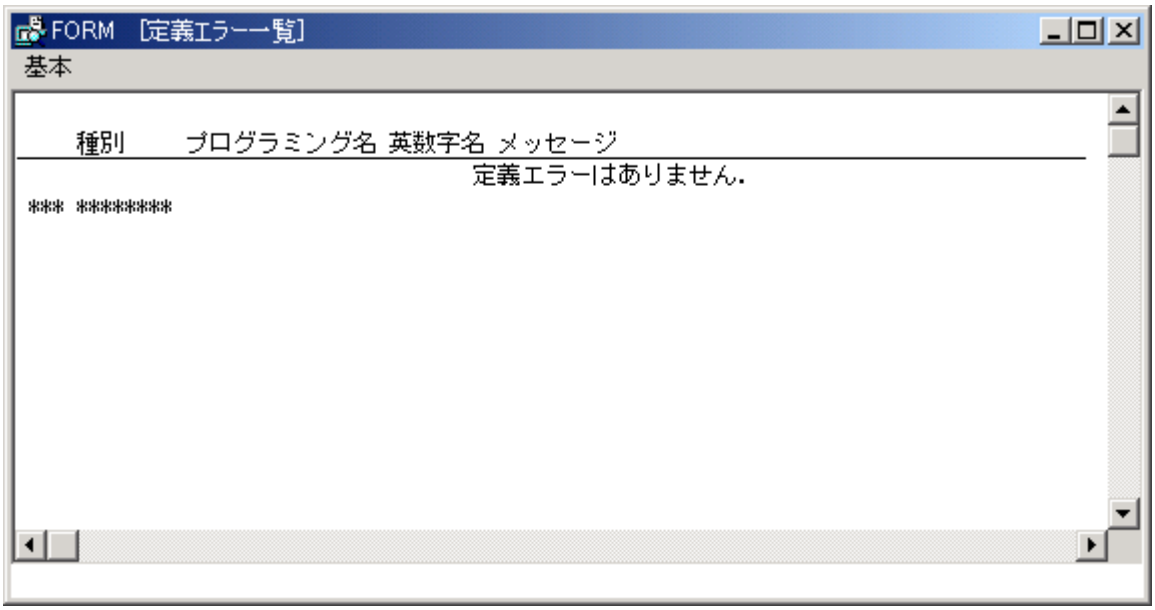
(2-2)プッシュボタンにする項目「終了(F5)」も、同じように選択群に追加します。



(2-3)FORM の[表示]メニューから[構成項目表示を解除]を選択し、選択群の構成項目の表示状態を解除します。

◆ 定義の正当性の確認

定義した内容にエラーがないか、定義エラー一覧画面で確認します。
定義エラー一覧画面は、FORM の[オプション]メニューから[定義エラー一覧]を選択することで表示されます。



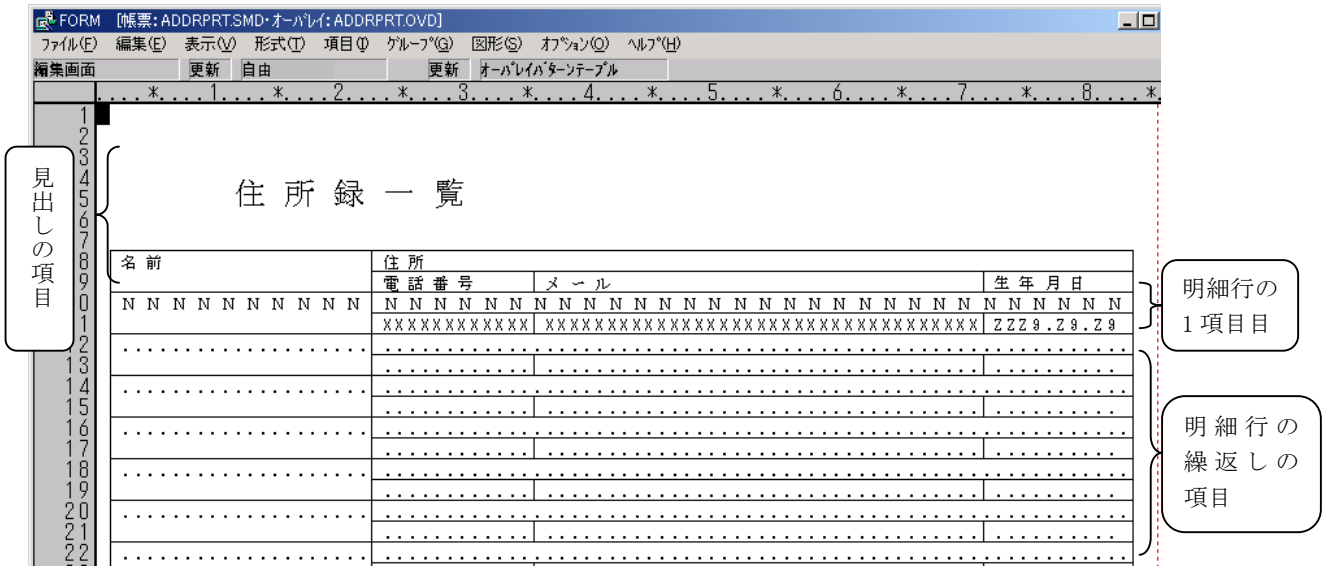
◆ 画面定義体の保存

[ファイル]メニュー－[閉じる]（あるいは[上書き保存]または[名前を付けて保存]）から[画面帳票定義体]を選択します。

「名前を付けて保存」ウィンドウで、「C:¥EDUCATION」を選択し、「ADDRDSP.SMD」という名前を付けて画面定義体を保存します。

2.3.2 帳票定義体の作成

ここでは、次に示すような帳票定義体を作成する方法を説明します。
なお、画面定義体の作成と同じ方法の事項については説明を省略しています。



次に示す手順で帳票定義体を作成していきます。

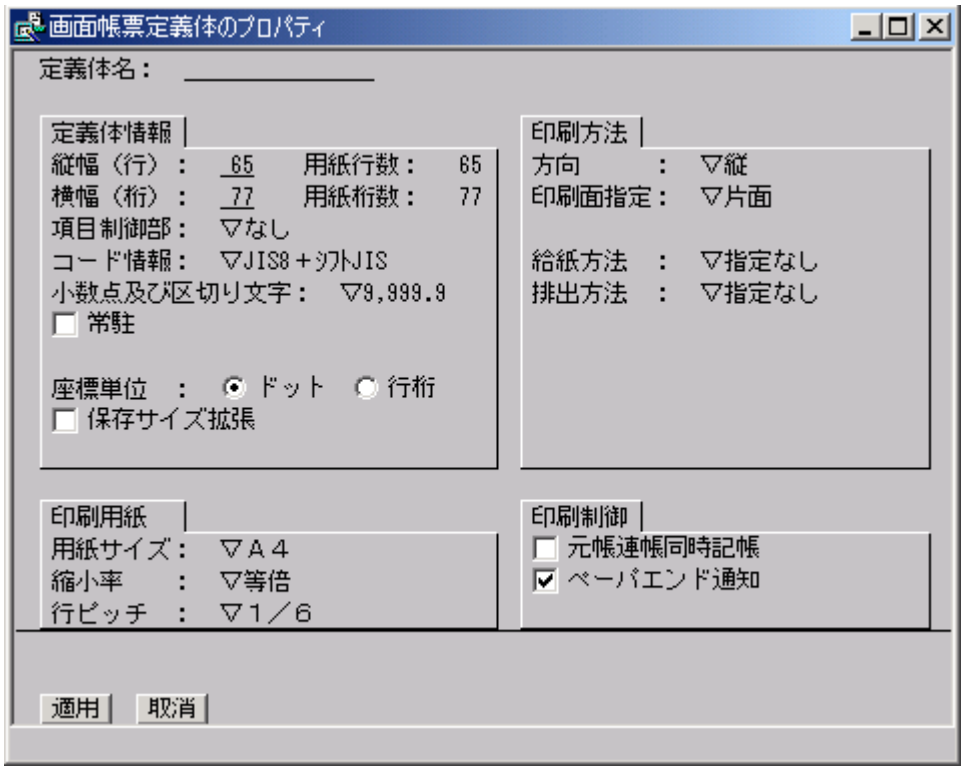
(1)帳票定義体の属性設定
　帳票定義体の大きさ（縦幅／横幅）など、帳票定義体全体に対する属性を設定します。
(2)項目の配置と属性設定
　データを出力する項目の配置など、帳票定義体のレイアウトを作成します。また、配置した項目に対する属性を設定します。
(3)繰返しの設定
　同じ属性を持つ項目を複数設定します。
(4)罫線の定義
　帳票定義体に罫線を定義します。
(5)罫線のオーバレイ定義
　帳票定義体に定義した罫線を、オーバレイ定義体の罫線にします。

◆ FORM の起動

画面定義体の作成時と同じ方法で起動します。

◆ 帳票定義体の属性設定

(1)FORM の[ファイル]メニュー－[新規作成]から[帳票定義体]を選択します。
(2)FORM の[ファイル] メニュー－[プロパティ]から[画面帳票定義体]を選択します。
(3)帳票定義体のプロパティ画面が表示されます。



(4)プロパティの画面で以下に示す内容を設定します。

設定箇所	設定内容
定義体名	住所録印刷帳票
縦幅(行)	44
横幅(桁)	85
方向	横

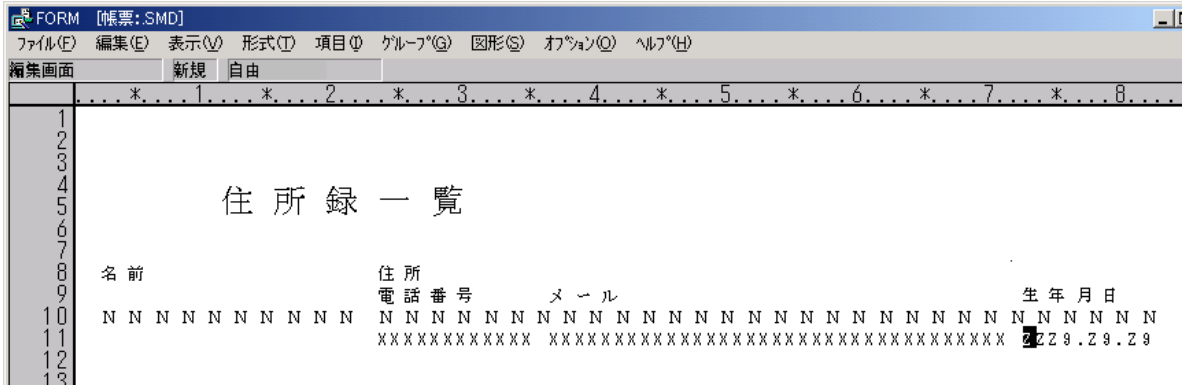
(5)プロパティの画面で「適用」を選択すると、編集画面に戻ります。

◆ 項目の配置と属性設定

項目の配置と属性設定は、画面定義体の作成と同じく FORM の[項目]メニューから行います。
各項目は次に示す内容で設定します。

「項目」メニューでの項目の種類	項目名	文字列	文字数	桁数	和文書体の拡大	編集形式
固定リテラル	見出し	住所録一覧			倍角	
固定リテラル	名前見出し	名前				
固定リテラル	住所見出し	住所				
固定リテラル	電話番号見出し	電話番号				
固定リテラル	メール見出し	メール				
固定リテラル	生年月日見出し	生年月日				
日本語	名前		10			
日本語	住所		30			
英数字	電話番号		12			
英数字	メール		35			
数字	生年月日			8		ZZZ9.Z9.Z9、全ゼロサプレスあり

ここまでの項目の配置、属性設定が終了すると、以下のような状態になります。

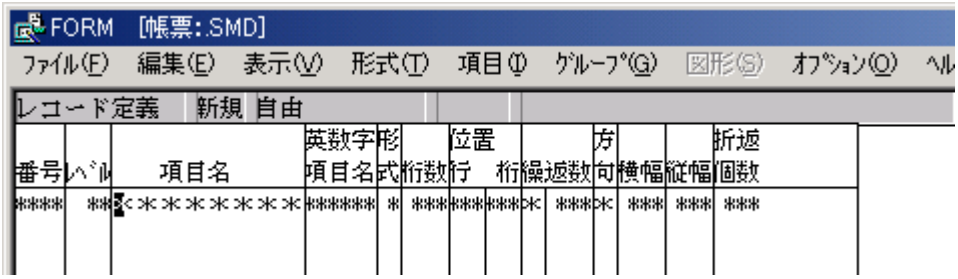


◆ 繰返しの設定

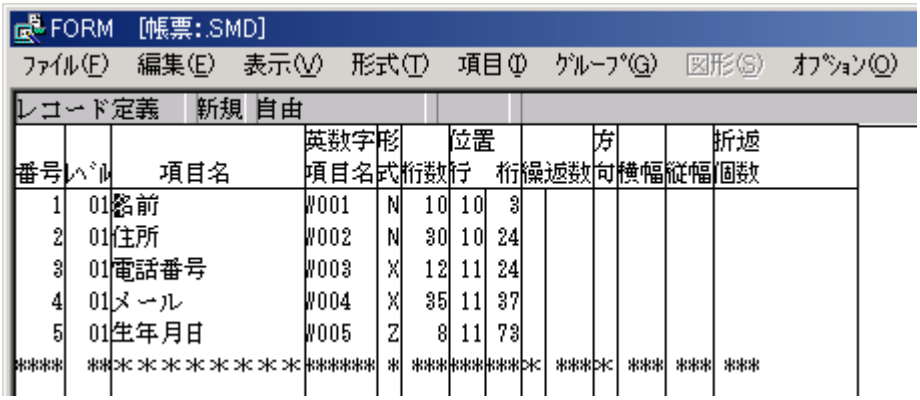
繰返しの設定とは、同じ属性を持つ項目を複数設定することをいいます。繰返しは、単一の項目にも、複数の項目から形成される集団項目にも行うことができます。本章で作成する帳票定義体では、プログラムからデータ出力を行う全ての項目を繰返し定義します。

繰返しの設定方法を説明します。なお、繰返しの設定は、帳票定義体だけではなく画面定義体でも行うことができます。

(1)FORM の[表示]メニューから[レコード定義へ]を選択し、レコード定義画面を表示します。

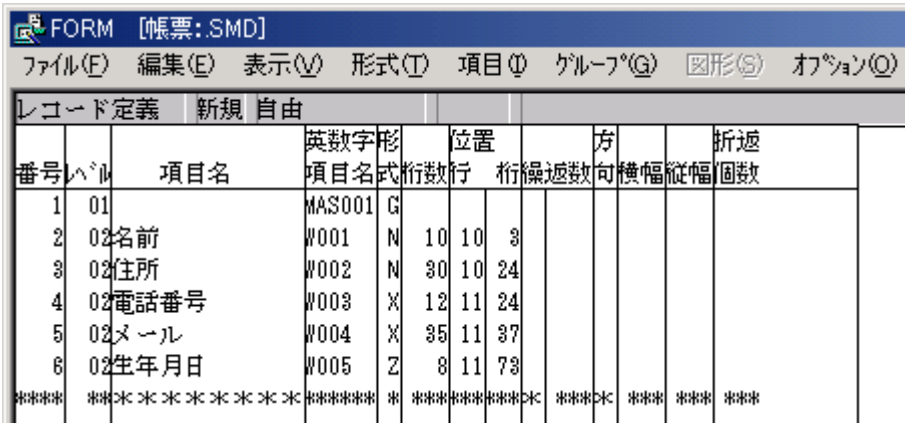


(2)[オプション]メニューから[基本レコードを生成]を選択します。帳票定義体に定義したデータ出力用の項目がレコード定義画面に表示されます。

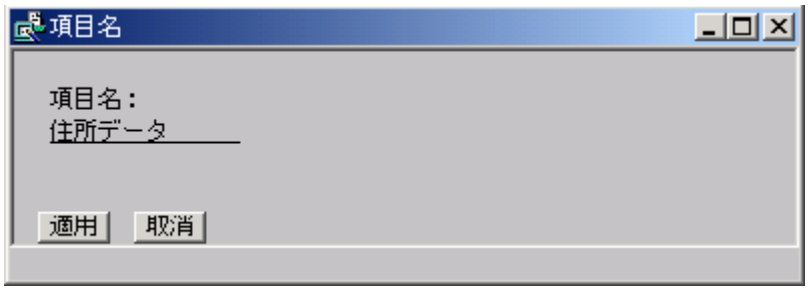


(3)[編集]メニューから[全てを選択]を選択します。全ての項目が選択されて反転します。

(4)[項目]メニューから[集団項目定義]を選択します。レコード定義画面の先頭にレベル 01 の項目が挿入され、(3)で選択した項目のレベルが 02 に変更されています。

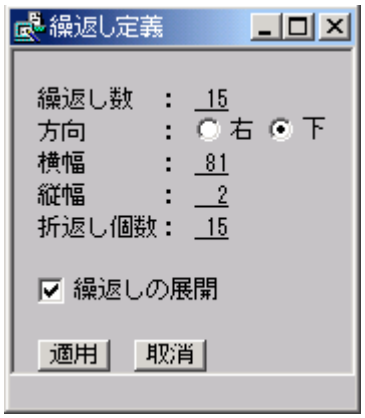


(5) レベル 01 の項目を選択し、[項目]メニューから[項目名]を選択します。
(6) 項目名の入力画面が表示されるので、項目名として「住所データ」を入力します。この名前が集団項目の名前になります。



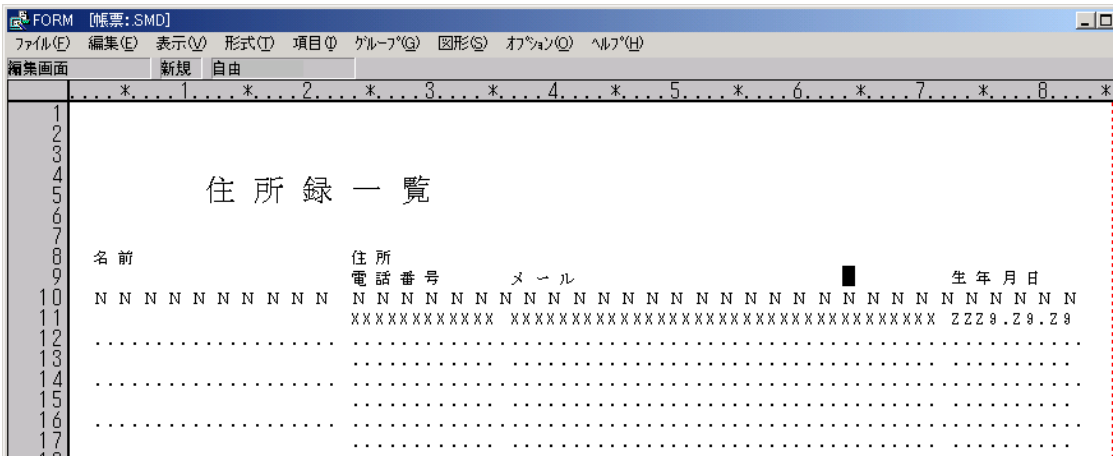
(7) 項目名の入力画面の「適用」ボタンを押し、項目名の入力画面を閉じます。レコード定義画面では、(6) で入力した集団項目の項目名が表示されています。
(8) 集団項目の項目名を選択し、[項目]メニューから[繰返し定義]を選択します。
(9) 繰返し定義の設定画面が表示されるので、次に示す内容を設定します。

設定箇所	設定内容
繰返し数	15
方向	下
折返し個数	15
繰返しの展開のチェック	チェックあり



(10) 繰返し定義の設定画面の「適用」ボタンを押し、繰返し定義の設定画面を閉じます。レコード定義画面では、(9) で入力した繰返し定義の設定が表示されています。

(11) [表示] メニューから [編集画面へ戻る] を選択し、編集画面を表示します。繰返しで設定された項目が「・・」で表示されます。


FORM [帳票:SMD]
ファイル(F) 編集(E) 表示(V) 形式(T) 項目(I) グループ(G) 図形(S) オプション(O) ヘルプ(H)
編集画面
新規
自由
住 所 録 一 覧
名 前
住 所
電 話 番 号
メ ー ル
生 年 月 日
N N N

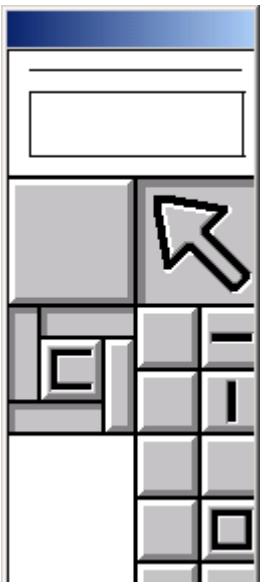
◆ 罫線の定義

罫線を定義する方法を説明します。

罫線は、帳票定義体に定義する方法とオーバレイ定義体に定義する方法があります。ここでは、帳票定義体に定義する方法を説明します。

なお、画面定義体にも同じ方法で罫線が定義できます。

(1) FORM 編集画面の左側にあるシンボルメニューで図形を選択します。



本章で作成する帳票では枠、水平線および垂直線を定義します。

枠を定義する際はシンボルメニューで「□」の図形を選択します。水平線は「—」、垂直線は「｜」のシンボルメニューの図形を選択します。

(2) 作図状態（マウスポインタがクロスポインタに変わります）になるので、作図範囲を開始位置から終了位置までドラッギングします。

全ての罫線を作図し終わると、以下のような状態になります。

◆ オーバレイ定義体の作成（罫線のオーバレイ定義）

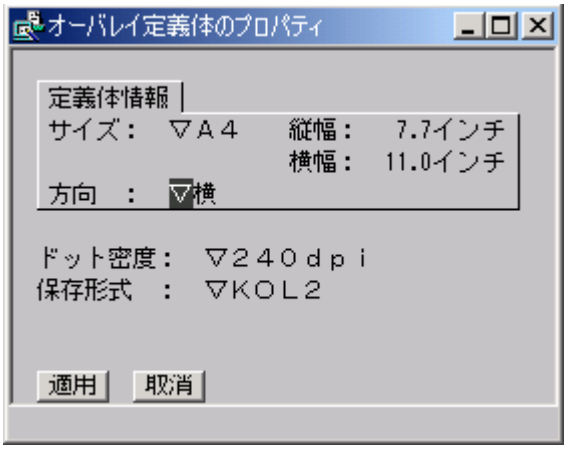
帳票定義体にオーバレイ定義体を重畳して新規作成することにより、帳票定義体に定義された罫線をオーバレイ定義体の罫線にできます。

帳票定義体にオーバレイ定義体を重畳して作成する方法を説明します。

(1) オーバレイ定義体を重畳して作成するには、[ファイル]メニュー－[新規作成]から[オーバレイ定義体]を選択します。
FORM編集画面のタイトルバーにオーバレイも表示されます。

(2) [ファイル]メニュー－[プロパティ]から[オーバレイ定義体]を選択し、オーバレイ定義体のプロパティ画面を表示します。

(3) 帳票定義体では用紙の指定をA4横にしているので、オーバレイ定義体もプロパティ画面でA4横を指定します。



(4) 「適用」ボタンを押して、オーバレイ定義体のプロパティ画面を閉じます。

次に、帳票定義体の印刷時に重畳するオーバレイ定義体名を帳票定義体に設定します。

(1) [形式]メニューから[オーバレイ名の指定]を選択します。

(2) オーバレイ名の指定画面が表示されるので、格納時に命名を予定しているオーバレイ定義体の名前「ADDRPRT.OVD」から拡張子を除いた名称を指定します。

(3) 「適用」ボタンを押して、オーバレイ名の指定画面を閉じます。

◆ 定義の正当性の確認

画面定義体の作成時と同じ方法で定義エラーの確認を行います。

◆ 帳票定義体の保存

［ファイル］メニュー－［閉じる］（あるいは［上書き保存］または［名前を付けて保存］）から［画面帳票定義体］を選択します。

「名前を付けて保存」ウィンドウで「C:¥EDUCATION」を選択し、「ADDRPRT.SMD」という名前を付けて帳票定義体を保存します。

◆ オーバレイ定義体の保存

［ファイル］メニュー－［閉じる］（あるいは［上書き保存］または［名前を付けて保存］）から［オーバレイ定義体］を選択します。

「名前を付けて保存」ウィンドウで、「C:¥EDUCATION」を選択し、「ADDRPRT.OVD」という名前を付けてオーバレイ定義体を保存します。

<u>参考</u>

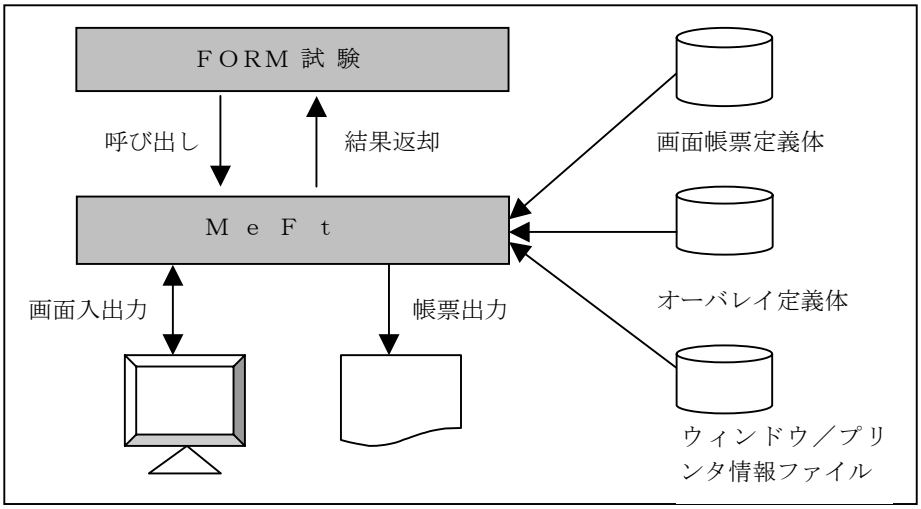
FORM では、帳票定義体を作成せずにオーバレイ定義体だけを編集することが可能です。

2.4 画面帳票定義体の確認

FORM 試験を使用して、作成した画面帳票定義体の動作や表示を確認する方法について説明します。

2.4.1 FORM 試験の概要

FORM 試験とは、プログラムを使用せずに画面帳票定義体の動作を確認するための機能です。FORM 試験では、プログラムからの実行時と同じように MeFt を使用して画面帳票定義体を動作させるため、プログラムを作成しなくても実行時の画面帳票定義体の動作や表示を確認することができます。
また、各入出力命令の復帰値も確認することができます。



2.4.2 画面定義体の確認

◆ ウィンドウ情報ファイルの作成

FORM 試験で画面定義体を確認する場合、プログラムの実行時と同じように、ウィンドウ情報ファイルが必要となります。

ウィンドウ情報ファイルとは、MeFt が参照する環境定義ファイルであり、画面定義体を表示するウィンドウの体裁などを定義します。形式はテキストであり、ファイル名、格納ディレクトリ、拡張子の有無は利用者が自由に決められます。

ウィンドウ情報ファイルには、ウィンドウの大きさ、位置、フォント名など多くの定義を行うことができますが、ここでは、次の 2 点の指定を行います。なお、ウィンドウ情報ファイルで定義できる内容については、MeFt 説明書「8.2 ウィンドウ情報ファイル」をご覧ください。

```
MEDDIR C:¥EDUCATION
WDFONTHIG 16
```

※“MEDDIR”は、画面定義体の格納ディレクトリを指定するキーワードです。
“WDFONTHIG”は、文字サイズをピクセル単位で指定するキーワードです。

本章では、ウィンドウ情報ファイルを画面帳票定義体と同じディレクトリ「C:¥EDUCATION」に「DSP.ENV」として作成することとします。

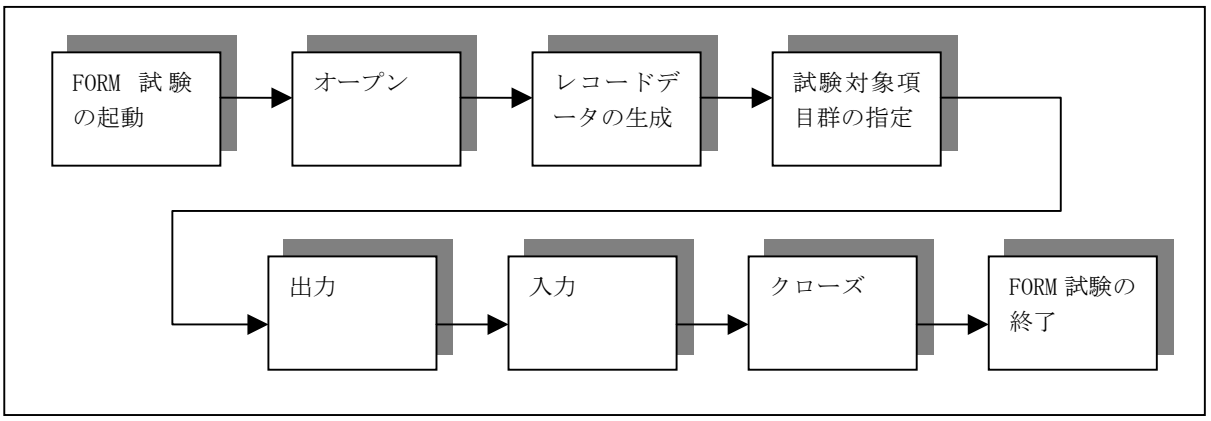
ここでは、FORM 試験で使用するためウィンドウ情報ファイルを作成しますが、作成したウィンドウ情報ファイルは、プログラムの実行時にも必要となります。

参考

プログラムの実行時、ウィンドウ情報ファイルと COBOL プログラムとの関連付けは、COBOL の実行環境情報の設定で行います。

◆ FORM 試験による画面定義体の確認の流れ

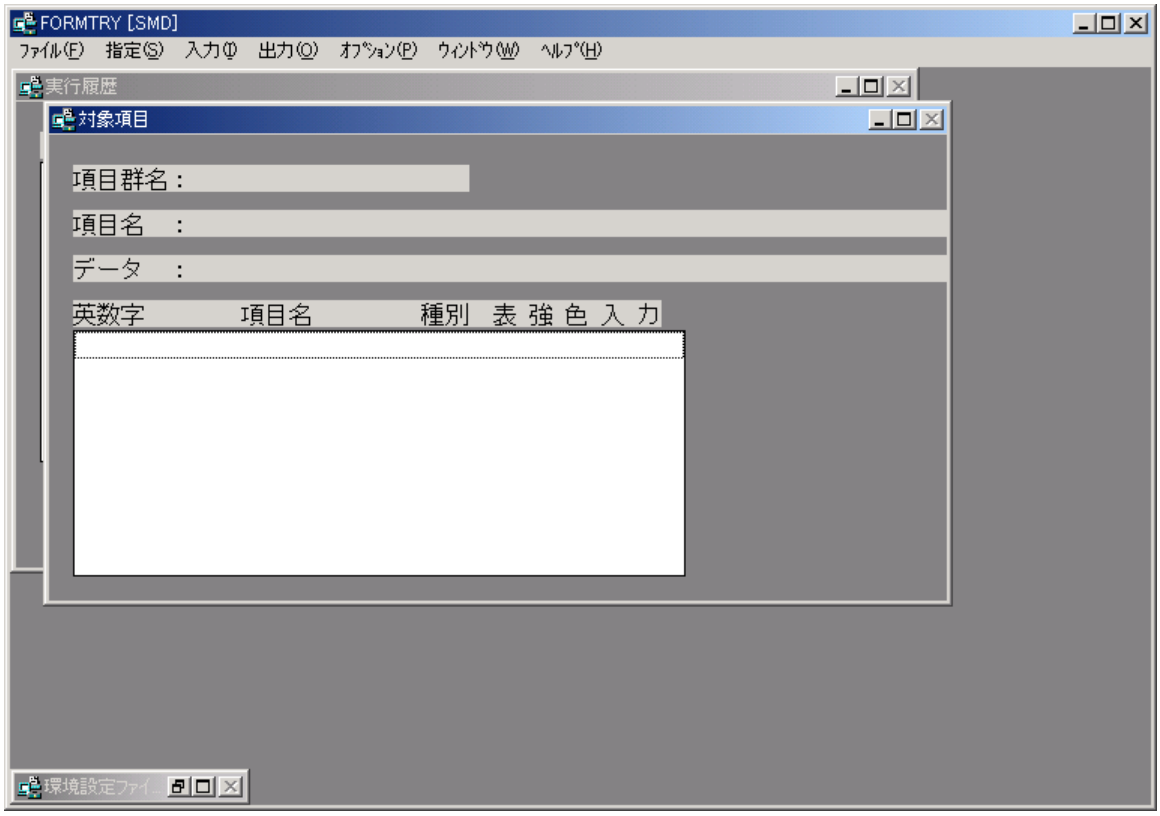
FORM 試験による画面定義体の確認の流れを次に示します。



◆ 画面定義体の確認

FORM 試験の起動

［スタート］－［プログラム］－［NetCOBOL］－［FORM］から［FORM 試験］を選択して、FORM 試験を起動します。

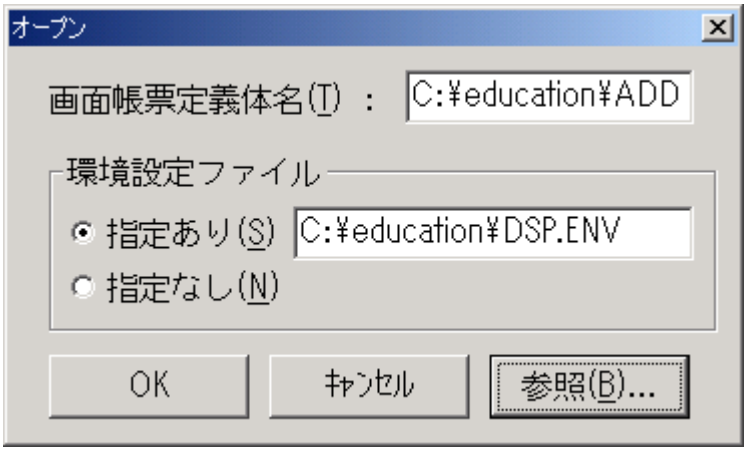


オープン

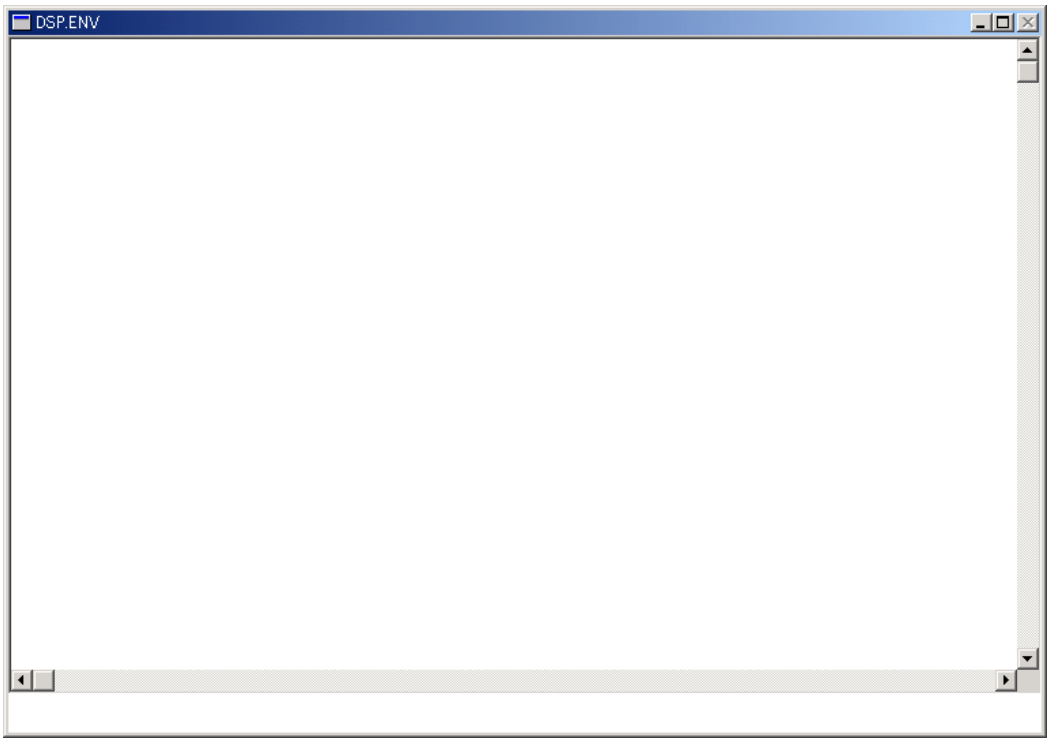
(1)FORM 試験の［ファイル］メニューから［オープン］を選択します。

(2)画面帳票定義体名と環境設定ファイルを指定する画面が表示されるので、「参照」ボタンにより画面帳票定義体とウィンドウ情報ファイルを指定して「OK」ボタンを押します。

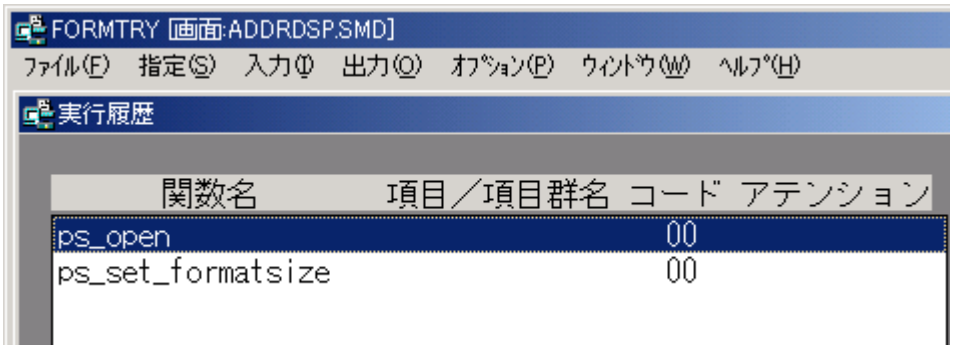
なお、「環境設定ファイル」とは、ウィンドウ情報ファイルおよびプリンタ情報ファイルのことです。



(3)オープンの試験が成功すると、ウィンドウが表示されます。



(4)FORM 試験の画面では、行った処理のステータスが「実行履歴」に表示されます。「コード」が“00”は成功です。



レコードデータの生成

画面定義体を出力する際に出力項目に設定する試験データを自動で生成し、設定することができます。データは各出力項目の形式に応じ、項目の桁数分が自動で生成されます。出力項目の形式に応じて生成されるデータは、次のとおりです。

出力項目の形式	生成されるデータ
日本語項目	N
英数字項目	X
数字項目	9

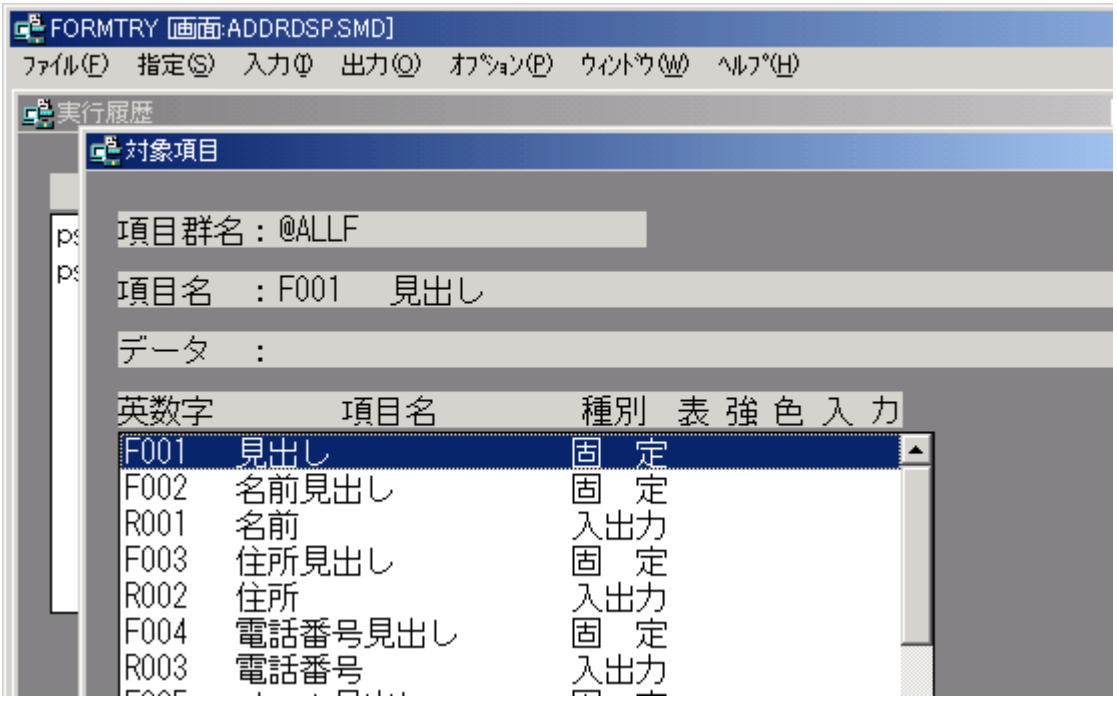
レコードデータの生成は、次の方法で行います。

(1)FORM 試験の[ファイル] メニューから[レコードデータ生成]を選択します。
(2)生成後、「レコードデータを生成しました」というメッセージボックスが表示されますので、「OK」をクリックします。

試験対象項目群の指定

入出力の対象となる項目群（一度に処理したい項目のグループ）名を設定します。ここでは、画面定義体に定義された全ての項目（全項目）を試験対象とします。

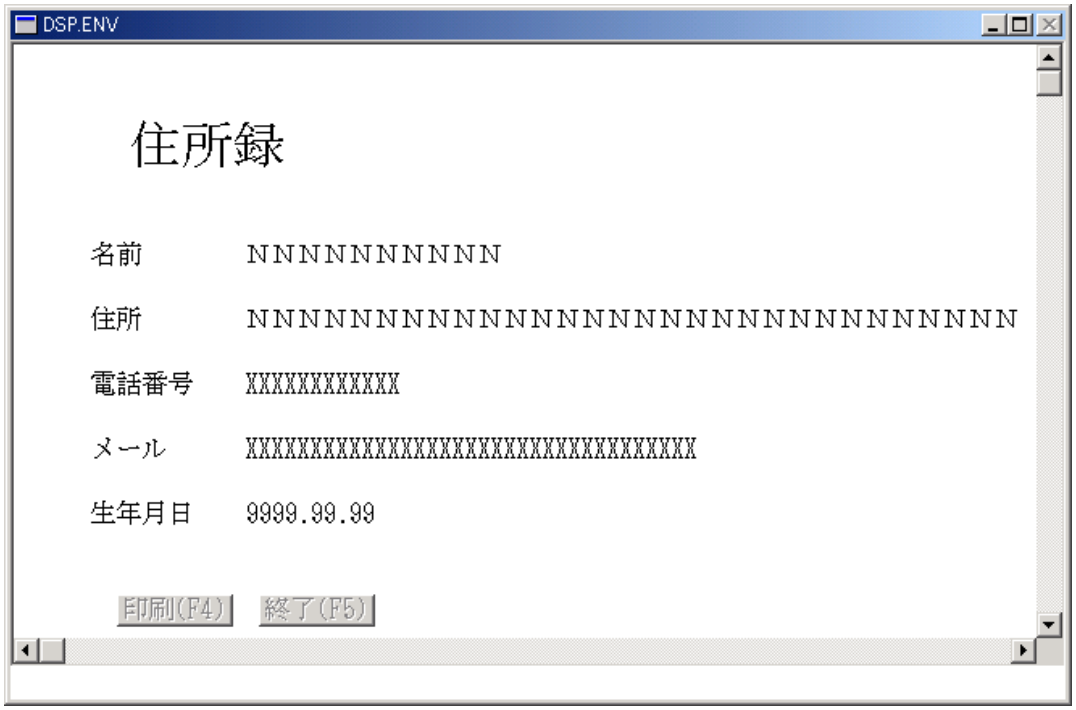
(1)FORM 試験の[指定]メニューから[全項目指定]を選択します。
(2)「データ設定、属性設定を行いますか？」というメッセージボックスが表示されますので、「いいえ」を選択します。
(3)設定後、FORM 試験の「対象項目」の画面に項目の一覧や設定されているデータなどが表示されます。



出力

(1)FORM 試験の[出力]メニューから[通常出力]を選択します。

(2)画面定義体を表示するウィンドウが前面に表示され、データがどのように出力されているか確認できます。
また、FORM 試験の画面の「実行履歴」で、試験対象の項目群名や出力処理のステータスを確認できます。



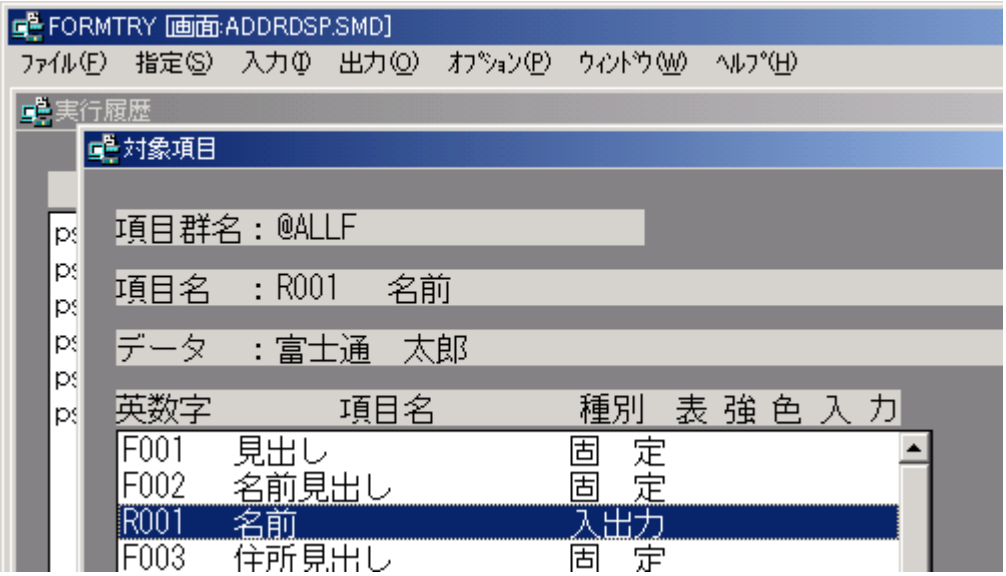
入力

(1)FORM 試験の[入力]メニューから[非消去入力]を選択します。

(2)画面にカーソルが表示され、実際にデータの入力を確認することができます。

(3)画面定義体に定義したファンクションキーまたはプッシュボタンが押されると、入力が終了します。

(4)FORM 試験の画面の「実行履歴」で入力処理のステータスやアテンション名を確認できます。また、「対象項目」で入力されたデータなどが確認できます。



クローズ

FORM 試験の[ファイル]メニューから[クローズ]を選択し、クローズします。

FORM 試験の終了

[ファイル]メニューから[終了]を選択し、FORM 試験を終了します。

2.4.3 帳票定義体の確認

◆ プリンタ情報ファイルの作成

FORM 試験で帳票定義体を確認する場合、プログラムの実行時と同じように、プリンタ情報ファイルが必要となります。

プリンタ情報ファイルとは、MeFt が参照する環境定義ファイルであり、帳票定義体を出力する際のプリンタの操作方法などを定義します。形式はテキストであり、ファイル名、格納ディレクトリ、拡張子の有無は利用者が自由に決められます。

プリンタ情報ファイルには、印刷名、出力プリンタデバイス名、フォント名など多くの定義を行うことができますが、ここでは、次に示す指定を行います。なお、プリンタ情報ファイルで定義できる内容については、MeFt 説明書「8.3 プリンタ情報ファイル」をご覧ください。

```
MEDDIR C:¥EDUCATION
OVLPDIR C:¥EDUCATION
FORMKIND C
```

※“MEDDIR”は、帳票定義体の格納ディレクトリを指定するキーワードです。
“OVLPDIR”は、オーバレイ定義体の格納ディレクトリを指定するキーワードです。
“FORMKIND”は、用紙種別として単票か連帳かを指定するキーワードで、設定値“C”は単票を意味します。

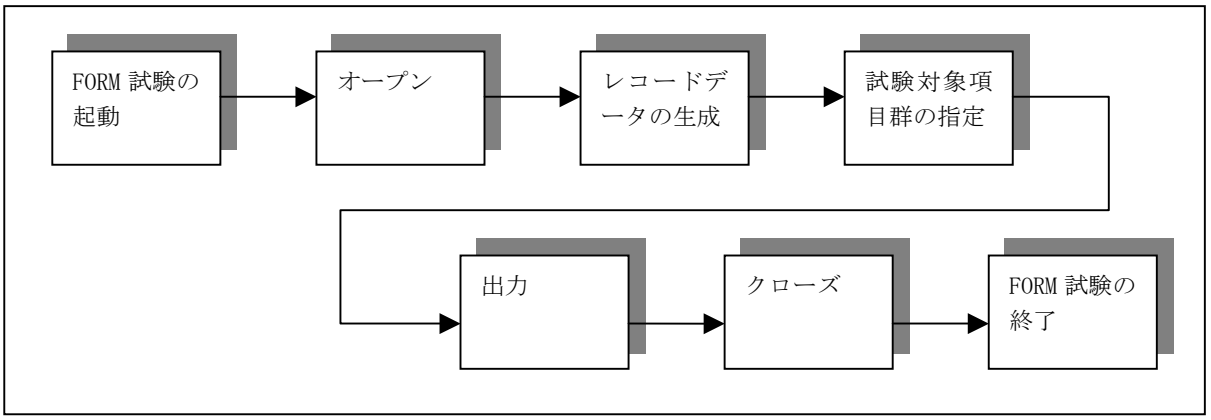
本章では、プリンタ情報ファイルを画面帳票定義体と同じディレクトリ「C:¥EDUCATION」に「PRT.ENV」として作成することとします。

ここでは、FORM 試験で使用するためプリンタ情報ファイルを作成しますが、作成したプリンタ情報ファイルは、プログラムの実行時にも必要となります。

> **参考**
> プログラムの実行時、プリンタ情報ファイルと COBOL プログラムとの関連付けは、COBOL の実行環境情報の設定で行います。

◆ FORM 試験による帳票定義体の確認の流れ

FORM 試験による帳票定義体の確認の流れを次に示します。



◆ 帳票定義体の確認

帳票定義体の確認手順は、入力がないことを除けば、画面定義体の確認と同じです。試験の結果、帳票が印刷されるので、出力結果を確認します。

2.5 プログラムの作成

COBOL プロジェクトマネージャによる、プロジェクト管理機能を使用した画面帳票アプリケーションの作成方法について説明します。

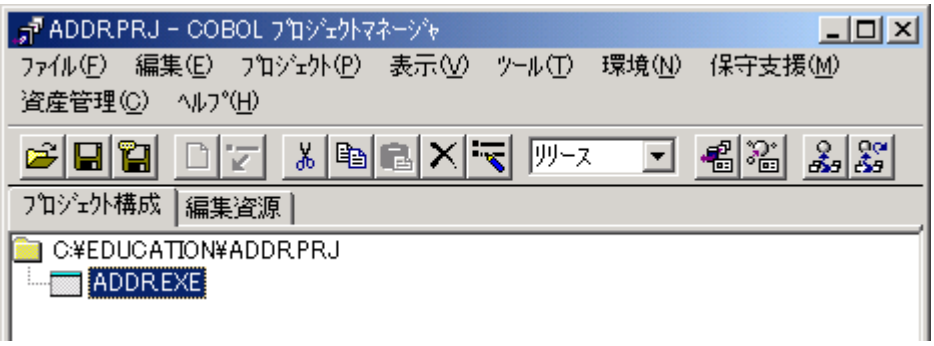
なお、画面帳票アプリケーションの作成方法は、第 1 章で作成したアプリケーションとほぼ同じです。ここでは、画面帳票アプリケーション固有の設定について説明します。

2.5.1 プロジェクトの作成

- **COBOL プロジェクトマネージャの起動**
- **プロジェクトファイルの作成**
- **実行ファイルの登録**

画面帳票アプリケーション固有の設定はありません。

本章で作成するアプリケーションでは、画面帳票定義体と同じディレクトリ「C:¥EDUCATION」にプロジェクトファイルとして「ADDR.PRJ」を作成し、実行ファイルとして「ADDR.EXE」を登録します。



◆ 各種ファイルの登録

実行ファイルを作成するのに必要となるファイルとして、ソースファイルと画面帳票定義体をプロジェクトに登録します。

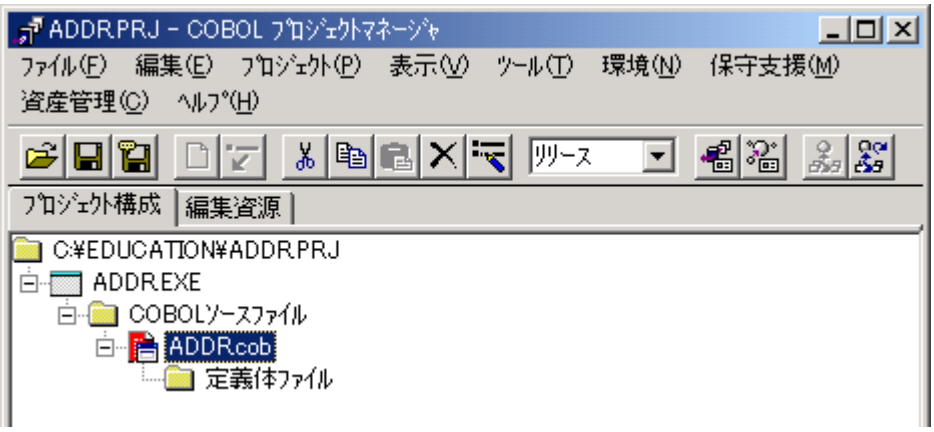
ソースファイルの登録は、画面帳票アプリケーション固有の設定はありません。「C:¥EDUCATION」にある「ADDR.COB」を選択し、プロジェクトに登録します。また、主プログラムの指定も行います。

一方、画面帳票定義体の登録は画面帳票アプリケーション固有です。

以下に、画面帳票定義体を登録する方法を説明します。

(1)COBOL ソースファイルを選択します。

(2)プロジェクトマネージャの[編集]メニュー－[フォルダ作成]から「定義体ファイル」を選択します。
COBOL ソースファイル配下に「定義体ファイル」というフォルダが作成されます。

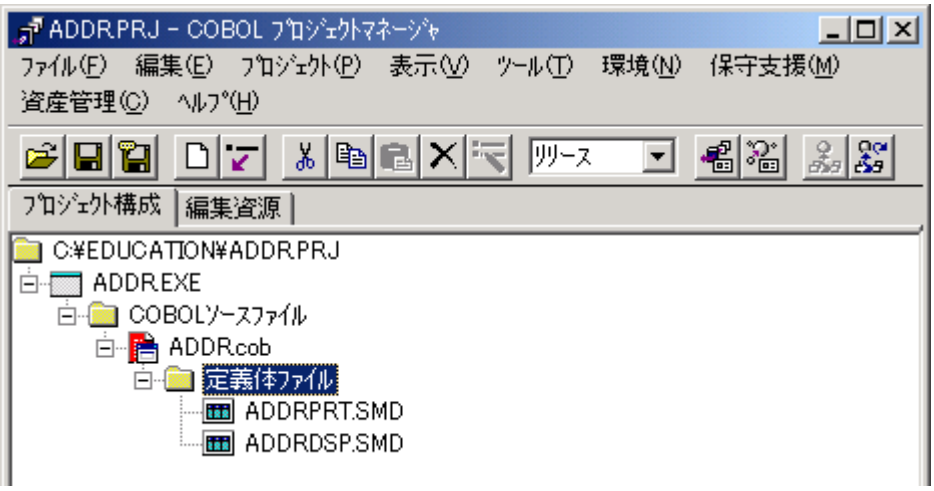


(3)フォルダ「定義体ファイル」を選択します。

(4)プロジェクトマネージャの[編集]メニューから[追加]を選択します。

(5)「ファイルの参照」ダイアログが表示されるので、「C:¥EDUCATION」にある画面定義体「ADDRDSP.SMD」および帳票定義体「ADDRPRT.SMD」を選択します。

(6)定義体ファイルのフォルダ配下に「ADDRDSP.SMD」および「ADDRPRT.SMD」が追加されます。



◆ プロジェクトファイルの保存

プロジェクトマネージャの[ファイル]メニューから[上書き保存]を選択して、プロジェクトファイルを保存します。

2.5.2 ビルド

◆ 翻訳オプションの設定

◆ リンクオプションの設定

画面帳票アプリケーション固有の設定はありません。

なお、画面帳票定義体に関する翻訳オプションとして次のようなものがあり、必要に応じて設定します。本章で作成するアプリケーションでは、設定する必要はありません。

- FORMEXT（画面帳票定義体ファイルの拡張子）
- FORMLIB（画面帳票定義体ファイルのフォルダ）

◆ ビルド操作

画面帳票アプリケーション固有の設定はありません。

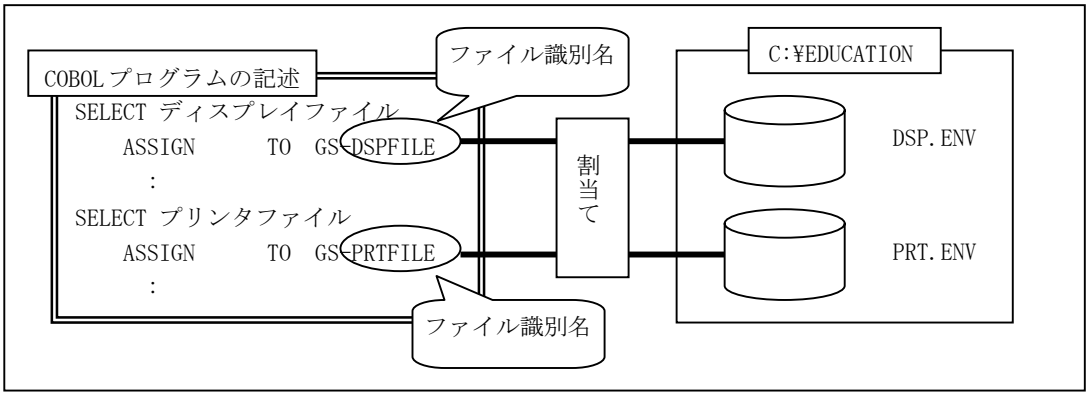
第1章で作成したアプリケーションと同じようにビルド操作を行うと、「ADDR.EXE」が生成されます。

2.5.3 実行

◆ 実行環境情報の設定

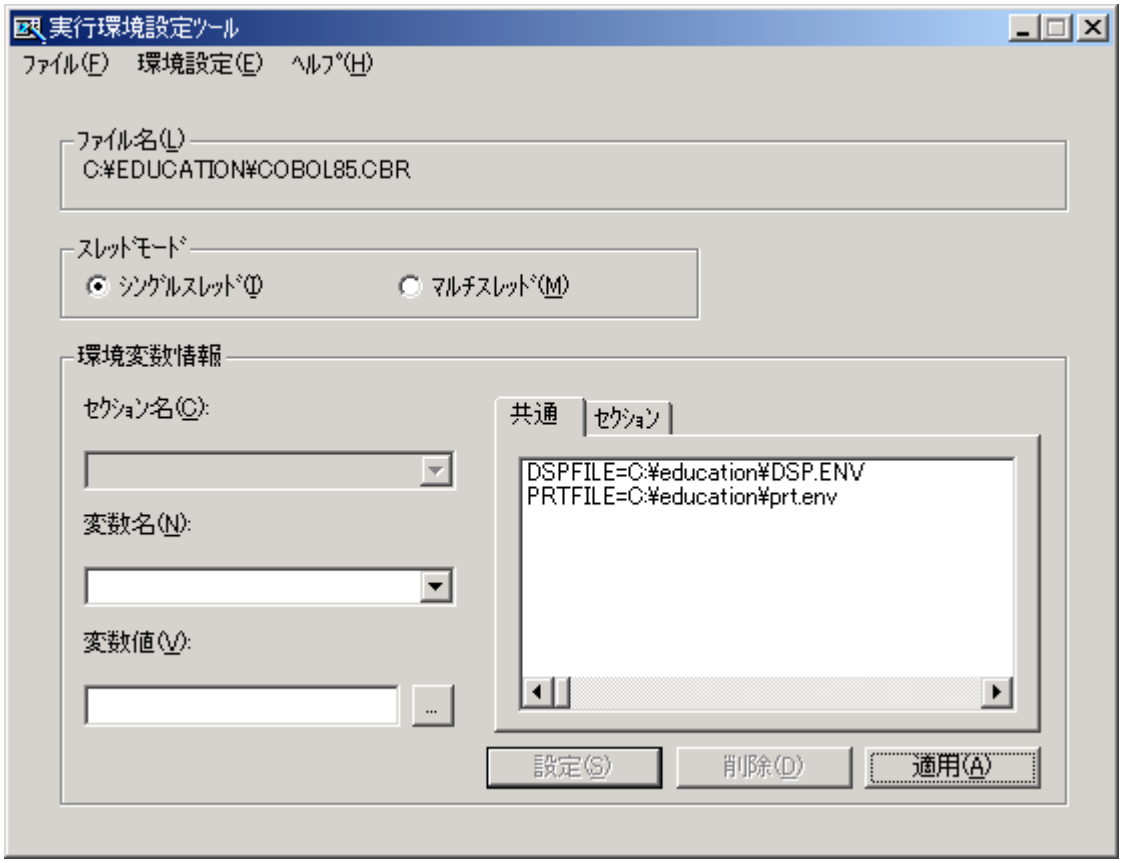
画面帳票アプリケーションを実行する場合、実行環境情報として、画面入出力用のファイルのASSIGN句で定義されたファイル識別名にウィンドウ情報ファイル名を設定し、帳票印刷用のファイルのASSIGN句で定義されたファイル識別名にプリンタ情報ファイル名を設定します。

本章のアプリケーションでは、画面入出力の表示ファイルのファイル識別名「DSPFILE」と「C:¥EDUCATION」に格納されているウィンドウ情報ファイル「DSP.ENV」を割り当て、帳票印刷の表示ファイルのファイル識別名「PRTFILE」と「C:¥EDUCATION」に格納されているプリンタ情報ファイル「PRT.ENV」を割り当てます。



実行環境設定ツールの起動や操作には、画面帳票アプリケーション固有の部分はありません。
第１章で作成したアプリケーションと同じ手順で「C:¥EDUCATION」に「COBOL85.CBR」を作成し、実行環境設定ツールで次に示す指定を行います。

変数名	変数値
DSPFILE	C:¥EDUCATION¥DSP.ENV
PRTFILE	C:¥EDUCATION¥PRT.ENV



◆ **プログラムの実行**

プログラムの実行についても、画面帳票アプリケーション固有の部分はありません。
プログラムを実行し、画面の入出力および帳票出力を確認します。

2.5.4 プロジェクトマネージャの終了

COBOL プロジェクトマネージャの[ファイル]メニューから［終了］を選択し、プロジェクトマネージャを終了します。

2.6 プログラムのデバッグ

NetCOBOL のデバッガを使用した画面帳票アプリケーションのデバッグについて説明します。

2.6.1 デバッグの準備

画面帳票アプリケーション固有の設定はありません。

第 1 章で作成したアプリケーションと同じ方法で、プロジェクトをビルド（またはリビルド）し、デバッグモジュールおよびデバッグ情報ファイルを作成します。

2.6.2 デバッガの起動

画面帳票アプリケーション固有の設定はありません。

2.6.3 デバッグ操作

画面帳票アプリケーション固有の設定はありません。

画面帳票アプリケーションでも、中断点を設定してのデバッグなどの第 1 章で説明したデバッグ操作が行えます。

2.6.4 デバッガの終了

デバッガの「ファイル」メニューから「デバッガの終了」を選択し、デバッガを終了します。

第 3 章 MeFt/Web アプリケーションの構築

本章では、画面帳票アプリケーションを Web 環境で動作させることができる MeFt/Web について説明します。また、画面帳票アプリケーションを MeFt/Web アプリケーションとして構築する方法について説明します。

3.1 概要
3.2 MeFt/Web サーバのセットアップ
3.3 MeFt/Web 環境への移行
3.4 HTML の作成
3.5 リモート実行
3.6 デバッグ
3.7 通信が切断されるパターンについて

3.1 概要

MeFt/Web の概要および本章で構築する MeFt/Web アプリケーションについて説明します。

3.1.1 MeFt/Web の概要

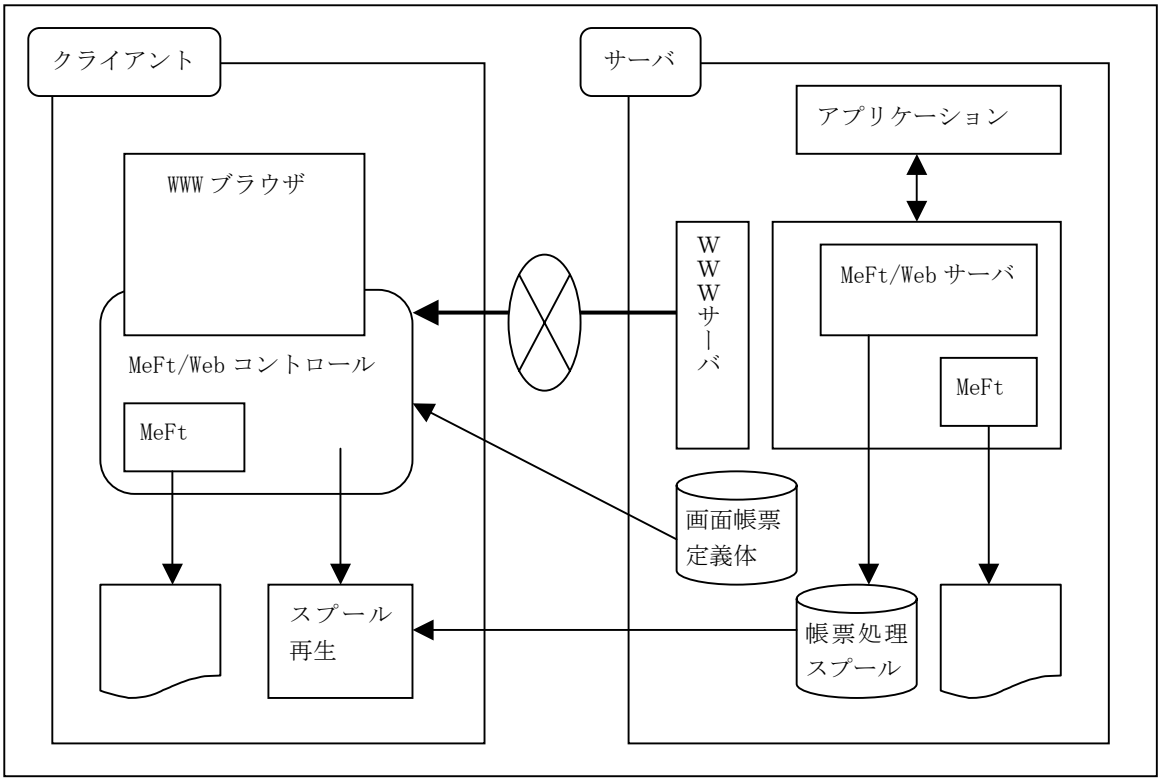
MeFt/Web とは、WWW サーバ上で動作するアプリケーションのディスプレイ装置またはプリンタ装置への入出力を、WWW ブラウザ上で行うことができる通信プログラムです。

MeFt/Web を使用することにより、画面帳票定義体の入出力を行うプログラムを Web 環境で動作させることができるようになります。

MeFt/Web は、サーバ上で動作する WWW サーバ連携プログラム（以降、MeFt/Web サーバ）と、クライアント側で動作する ActiveX コントロール（以降、MeFt/Web コントロール）から構成されています。

MeFt/Web サーバは、アプリケーションから MeFt に要求された入出力要求を WWW サーバを介して、クライアント側の MeFt/Web コントロールに渡すなどの処理を行っています。

MeFt/Web コントロールは、MeFt/Web サーバからの入出力要求を WWW ブラウザやプリンタ装置に対して行います。また、MeFt/Web コントロールは、MeFt/Web サーバとの通信処理や MeFt 機能が ActiveX コントロール化されたものであり、必要時にサーバ上からダウンロードされます。MeFt/Web コントロールからサーバ上のアプリケーションを実行し、画面帳票の入出力を WWW ブラウザで行うことを「リモート実行」といいます。



MeFt/Web には、次のような機能があります。

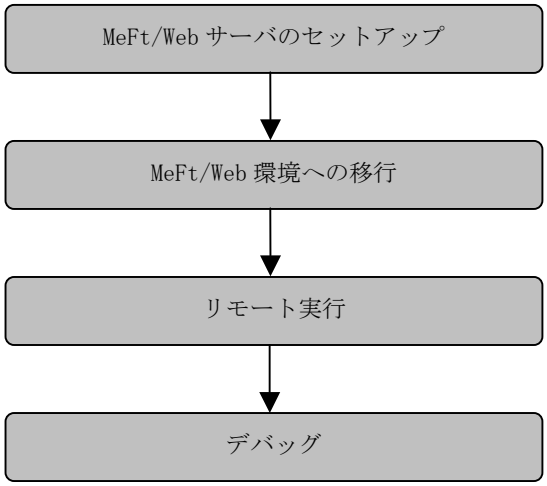
機能名		説明
画　面機能	画面処理	リモート実行した利用者プログラムからの画面入出力を WWW ブラウザ上で行います。
	ハイパーリンク	項目に URL を設定することができます。
印　刷機能	プレビューおよびクライアント印刷機能	印刷イメージを WWW ブラウザ上に表示したり、クライアントに接続されているプリンタ装置に印刷します。
	サーバ印刷機能	サーバに接続されているプリンタ装置に印刷します。
	スプール機能およびスプール再生機能	利用者プログラムからの印刷要求をサーバ上にスプールしたり、その帳票結果を WWW ブラウザ上で再生（プレビュー）します。
サービスマネージャ機能		サーバ上の利用者プログラムを起動したり、起動しているプログラムの一覧表示などを行います。

3.1.2　構築するアプリケーションについて

第２章で作成した画面帳票アプリケーションを WWW サーバに移行して、MeFt/Web アプリケーションを構築します。
なお、WWW サーバのホスト名は「SampleSvr」とし、アプリケーションの実行に必要な資産の格納ディレクトリは「C:¥ADDRESS」とします。

3.1.3　構築作業の流れ

MeFt/Web アプリケーションの構築の流れを次に示します。

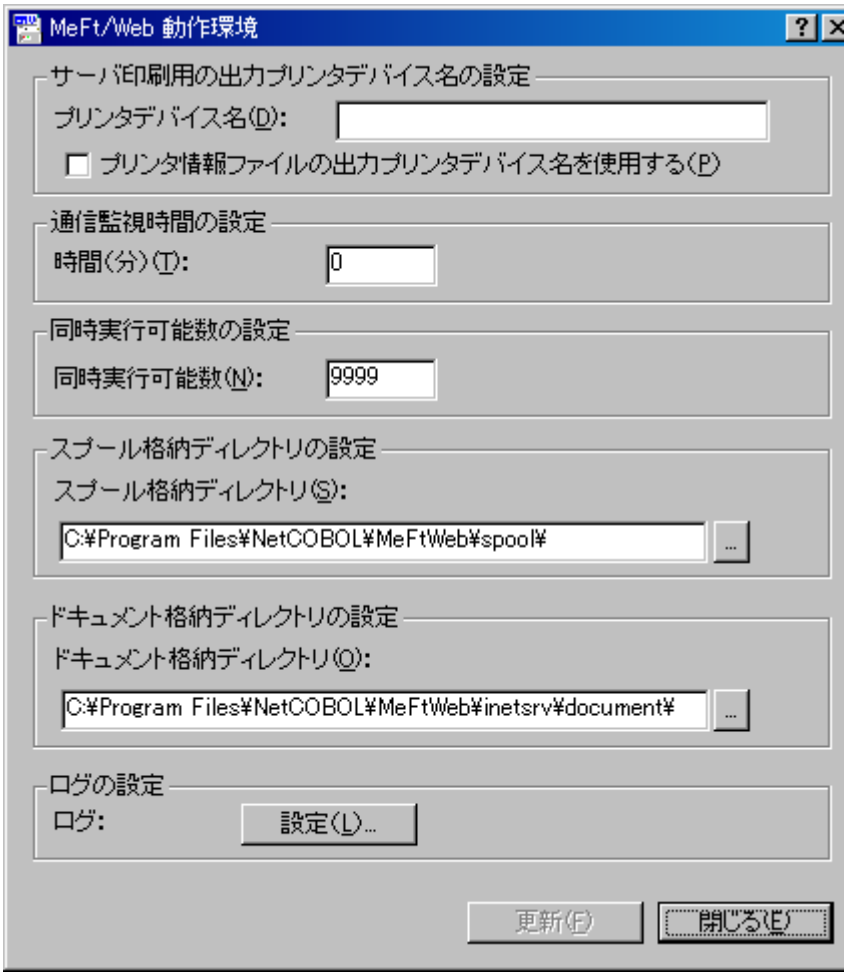


3.2 MeFt/Web サーバのセットアップ

MeFt/Web サーバのセットアップについて説明します。セットアップとして、MeFt/Web 動作環境の設定と権限の設定を行います。

3.2.1 MeFt/Web 動作環境の設定

[スタート]ボタン－[プログラム]－[NetCOBOL]－[MeFt/Web]から[MeFt/Web 動作環境]を選択して、MeFt/Web 動作環境の設定画面を表示します。



動作環境設定の各項目の概要を次に示します。詳細は、MeFt/Web 説明書「2.2 MeFt/Web の動作環境を設定する」をご覧ください。

項目名	説明
サーバ印刷用の出力プリンタデバイス名	サーバ印刷で使用するプリンタデバイス名を指定します。
プリンタ情報ファイルの出力プリンタデバイス名を使用する	MeFt/Web 動作環境とプリンタ情報ファイルの両方に出力プリンタデバイス名が指定された場合、どちらの指定のプリンタに印刷するか指定します。
通信監視時間	WWW ブラウザからの無応答を監視する時間を指定します。
同時実行可能数	アプリケーションの同時実行可能数を指定します。
スプール格納ディレクトリ	スプール機能を実行した際に印刷データを格納するディレクトリを指定します
ドキュメント格納ディレクトリ	MeFt/Web ドキュメントを格納するディレクトリを指定します。なお、MeFt/Web ドキュメントは、WWW ブラウザが Netscape Navigator である場合に使用します。
ログ	MeFt/Web サーバで採取するトレースログ環境を指定します

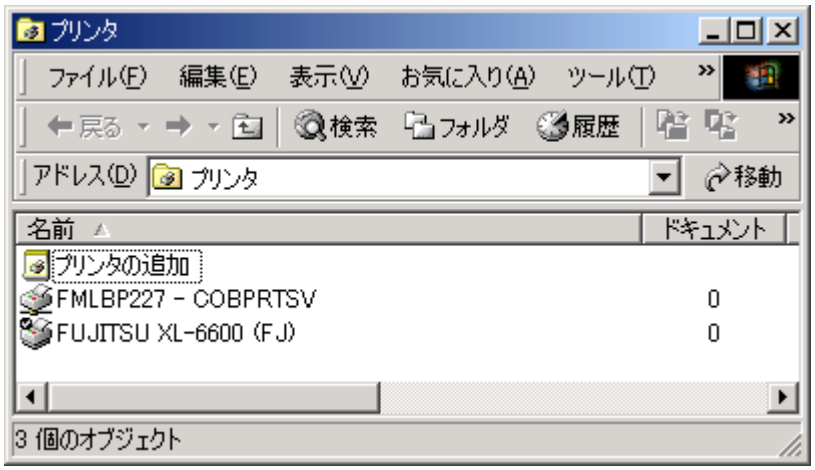
動作環境設定の各項目のうち、サーバ印刷の出力プリンタデバイス名、通信監視時間および同時実行可能数について説明します。

◆ サーバ印刷の出力プリンタデバイス名

サーバ印刷で使用するプリンタデバイス名は、MeFt/Web の動作環境とサーバ印刷用のプリンタ情報ファイルの２ヵ所で設定することができます。

両方で指定があった場合、「プリンタ情報ファイルの出力プリンタデバイス名」の指定に従います。

なお、出力プリンタデバイス名は、<コントロールパネル>ウィンドウの<プリンタ>アイコンを選択して表示される一覧を参照し、プリンタ名を""で囲んで指定します。ただし、ローカルプリンタとネットワークプリンタとで指定方法が違います。



<u>ローカルプリンタの場合</u>

プリンタの一覧で得られる名前で指定します。

上記の場合、"FUJITSU XL-6600 (FJ)"がローカルプリンタなので、この名前を指定します。

<u>ネットワークプリンタの場合</u>

"¥¥サーバ名¥プリンタ名" という形で指定します。

上記の場合、"FMLBP227 - COBPRTSV"が、COBPRTSV というサーバに接続された FMLBP227 というネットワークプリンタなので、"¥¥COBPRTSV¥FMLBP227"と指定します。

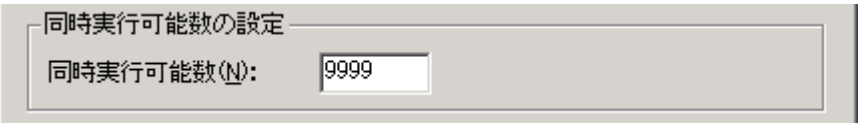
◆ 通信監視時間

ネットワークの異常やクライアントマシンの強制終了などによって WWW ブラウザからの応答をサーバのアプリケーションに返すことができなくなった場合、アプリケーションは応答待ちのままになってしまいます。この問題に対処するため、通信監視時間を設定します。

MeFt/Web サーバでは、一定の時間（通信監視時間）を超えてアプリケーションに応答することができなかった場合、MeFt の通知コードとして”N7”を通知します。COBOL アプリケーションでは、表示ファイルの各命令後に FILE STATUS 句に指定した 4 桁のデータ領域が”90N7”であるか判定することで、通信監視時間を越えて応答がなかったことを知ることができます。

◆ 同時実行可能数

MeFt/Web では、リモート実行されるアプリケーションの同時実行可能数を指定することができます。サーバマシンの性能などを考慮し、同時に実行するアプリケーションの数を設定します。



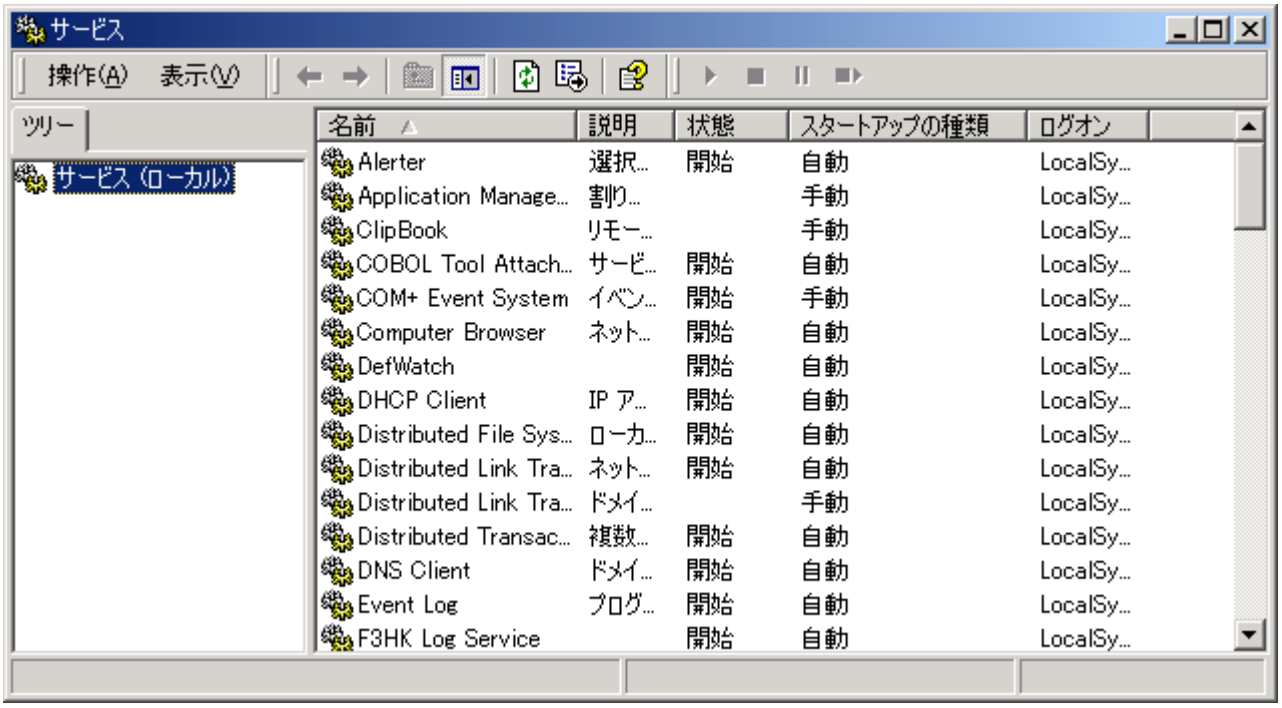
ここで指定された同時実行可能数を越えてプログラムをリモート実行しようとすると、WWW ブラウザに「P2006 プログラムを処理できませんでした。同時実行可能数を超えました。」という MeFt/Web コントロールのエラーが表示されます。

なお、同時実行可能数のチェックの対象となるのは、単一のアプリケーションではなく、リモート実行される全アプリケーションです。また、同時に実行するクライアントの台数ではなく、実行されるアプリケーションの数が対象となります。

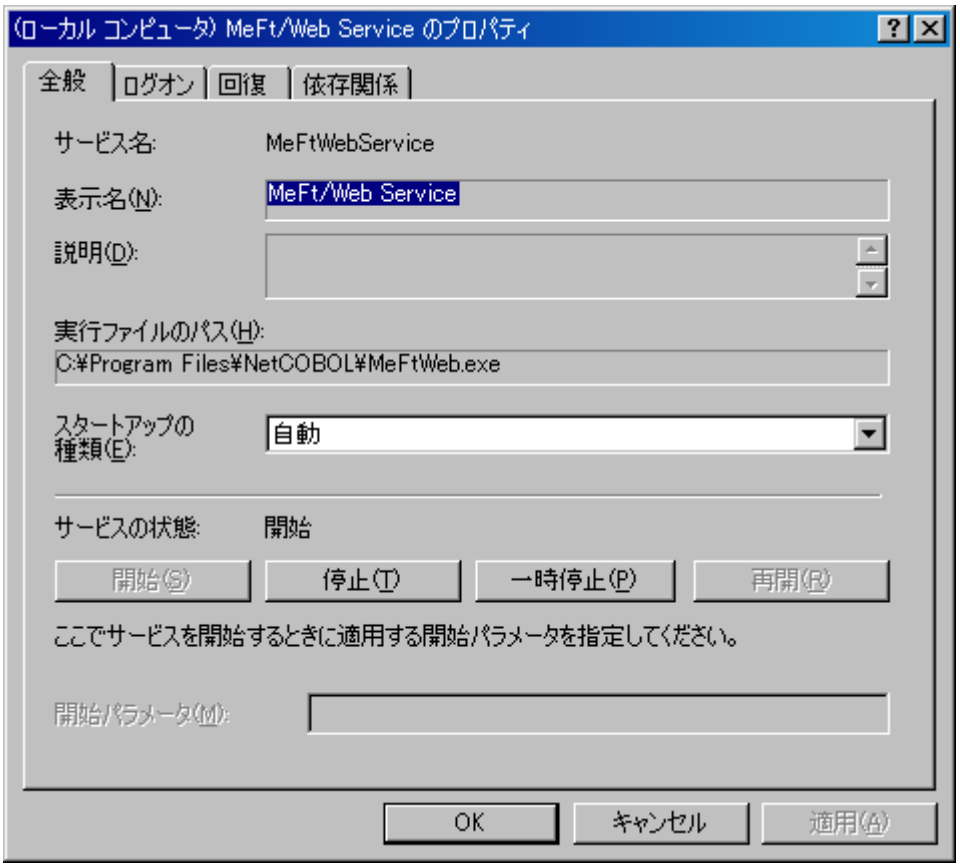
3.2.2　権限の設定

MeFt/Web で起動される COBOL プログラムの権限を設定します。COBOL プログラムが扱う資源（ディレクトリ、プリンタなど）に応じて、アカウントを設定します。
プログラムの権限を設定する方法を説明します。

(1)MeFt/Web がインストールされたマシンで、[スタート]ボタン－[プログラム]－[管理ツール]から[サービス]を選択します。
(2)サービスの画面が表示されます。



(3)サービスの画面から「MeFt/Web Service」を選択します。
「操作」メニューから「プロパティ」を選択して、MeFt/Web Service のプロパティ画面を表示します。



(4)「ログオン」タブで「アカウント」を選択し、使用するユーザアカウントとそのパスワードを設定します。次に示す例では、ユーザアカウントとして「Administrator」を設定しています。

(5)「OK」ボタンを押して、MeFt/Web Service のプロパティ画面を閉じます。

参考

本章では、Windows® 2000 での設定方法を説明しています。Windows NT®では、以下の方法で設定します。

(1) MeFt/Web がインストールされたマシンにおいて、[スタート]ボタンをクリックし、[設定]-[コントロールパネル]をクリックします。さらに、コントロールパネル画面から[サービス]を起動します。
(2) サービスの画面から「MeFt/Web Service」を選択し、スタートアップボタンをクリックします。
MeFt/Web Service の設定画面が表示されます。
(3) 「ログオン」で「アカウント」を選択し、使用するユーザアカウント名とパスワードを設定します。
(4) 「OK」ボタンを押します。

参考

MeFt/Web Service は、インストール時はシステムアカウントになっていますが、システムアカウントを指定するとプロセスを強制終了できないなどの不都合が発生するため、システムアカウント以外にします。

また、システムアカウントのままだと、イベントビューアの「アプリケーション ログ」に次に示すイベントが通知されます。

項目	内容
ソース	MeFt/Web Service
イベント ID	122
説明	ユーザレジストリのロードに失敗しました。

3.3 MeFt/Web 環境への移行

スタンドアロンで動作していた画面帳票アプリケーションを、MeFt/Web 環境へ移行する方法を説明します。

3.3.1 MeFt/Web 移行時のアプリケーションの対応

スタンドアロンの画面帳票アプリケーションを MeFt/Web 環境へ移行するにあたり、対応が必要となる点について説明します。

◆ 表示ファイル以外の画面

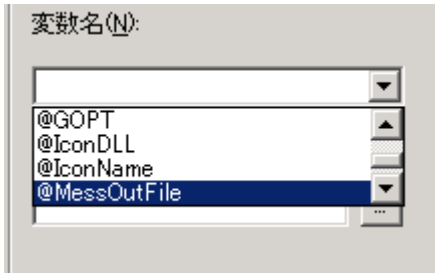
MeFt/Web でリモート実行したプログラムで表示される画面のうち、表示ファイルを使用した画面以外はサーバ上で処理されます。サーバ上で処理される画面には、次のようなものがあります。

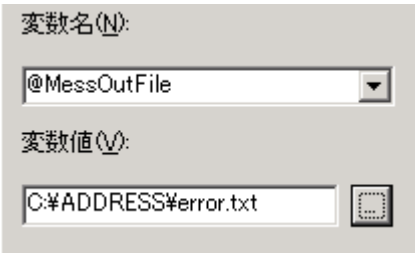
- エラーメッセージ
- コンソール画面
- 診断機能メッセージ
- ウィンドウクローズメッセージ

しかし、通常、WWW サーバを介して起動されたプログラムの画面は表示されないため、その画面への応答ができずに入力待ち状態になってしまいます。その結果、クライアントの WWW ブラウザが無応答となってしまいます。
この問題への対応について、画面ごとに説明します。

エラーメッセージ

COBOL プログラムの実行時にエラーが発生すると、エラーのメッセージボックスが表示され、「OK」ボタンが押されるのを待つ状態になります。しかし、MeFt/Web 環境ではメッセージボックスが表示されても応答できないため、必ずエラーメッセージをファイルに出力する指定をします。
エラーメッセージをファイルに出力するには、実行環境情報で、COBOL が用意する環境変数情報「@MessOutFile」に出力するファイル名を割り当てます。なお、実行環境設定ツールでは、COBOL が用意する環境変数情報を「変数名」のリストから選択することができます。
次の例では、メッセージを出力するファイルとして「C:¥ADDRESS¥ERROR.TXT」を割り当てています。





コンソール画面

コンソール画面への DISPLAY/ACCEPT はできません。コンソール画面への出力は、ファイルに出力するよう変更します。コンソール画面への出力をファイルにするには、翻訳オプション“SSOUT”で任意の環境変数情報名を指定し、実行環境情報でその環境変数情報名とファイル名を対応づけます。
以下に、翻訳オプションと実行環境情報の設定例を示します。

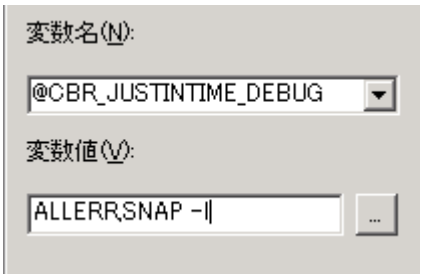
```
＜翻訳オプションの指定内容＞
    SSOUT(OUTFILE)

＜実行環境情報の指定内容＞
    OUTFILE=C:¥ADDRESS¥SSOUT.DAT
```

診断機能メッセージ

COBOL プログラムの実行時にアプリケーションエラーなどが発生すると、COBOL の診断機能によりエラーの発生箇所などを記載した診断レポートファイルが出力され、そのことがメッセージボックスに表示されます。MeFt/Web 環境では、そのメッセージを表示しないようにするか、診断機能を起動しないようにする指定を必ず行います。

この指定は、実行環境情報で実行環境変数「@CBR_JUSTINTIME_DEBUG」にて行います。診断機能のメッセージを表示させないようにするには、「@CBR_JUSTINTIME_DEBUG」の変数値を「ALLERR,SNAP -l」とし、診断機能を起動しないようにするには、変数値を「NO」とします。

次の例では、診断機能のメッセージを表示させないように「ALLERR,SNAP -l」を指定しています。



ウィンドウクローズメッセージ

ウィンドウを閉じる時の確認メッセージであるウィンドウクローズメッセージを非表示にするため、実行環境情報の実行環境変数「@WinCloseMsg=OFF」を指定する必要があります。

◆ プロセス型プログラムとスレッド型プログラム

MeFt/Web では、次の 2 種類の COBOL プログラムを起動できます。

プログラムの種類	説明
プロセス型プログラム	実行可能なモジュール形式(EXE)のプログラムです。
スレッド型プログラム	ダイナミックリンクライブラリ形式(DLL)のプログラムです。

プロセス型プログラムとスレッド型プログラムの比較

プロセス型プログラムとスレッド型プログラムの違いについて下表に示します。

項目	プロセス型プログラム	スレッド型プログラム
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）
実行単位	プロセス	スレッド
起動性能	スレッド型プログラムと比べ低速	スタートアップのオーバヘッドがないため高速。
サーバの資源消費	大	小
既存資産の活用性	ソース修正および再翻訳・再リンクとも不要。	再翻訳・再リンクが必要。場合によっては若干のソース修正が必要。
アプリ異常終了時の影響範囲	異常が発生したプログラム以外には影響が及ばない。	同じプロセスで動作する他のスレッド型プログラムも異常終了してしまう。

プロセス型プログラムをスレッド型プログラムに移行する場合の修正点

既存のプロセス型プログラムをスレッド型プログラムに移行する際、プログラム修正が必要となる部分について説明します。

- 環境変数操作

  スレッド型プログラムでは、1つのプロセスで複数のスレッドが動作するため、環境変数の内容が変更されると、他のプログラムに影響を及ぼす可能性があります。したがって、スレッド型プログラムでは環境変数操作を行うことはできません。環境変数操作を行っている場合は、環境変数操作は行わないように修正します。

- 引数の受渡し方法

  プロセス型プログラムでは、起動時に指定された引数を受け取るには、コマンド行引数の操作機能を使用します。コマンド行引数の操作機能については、NetCOBOL 使用手引書「11.2 コマンド行引数の取出し」をご覧ください。

  一方、スレッド型プログラムはC呼出し規約に従って呼び出されるため、起動時に指定された引数を受け取るには、手続き部(PROCEDURE DIVISION)の見出しの USING 指定にデータ名を記述します。したがって、引数の受渡しを行っている場合は、受け取り方法を変更します。C呼出し規約による呼出しの詳細については、NetCOBOL 使用手引書「10.3.2 C プログラムから COBOL プログラムを呼び出す方法」をご覧ください。

> 参考
>
> MeFt/Web アプリケーションでは、起動時に指定する引数は MeFt/Web コントロールの argument プロパティに指定します。argument プロパティについては、MeFt/Web 説明書「3.4.3 利用者プログラムの指定方法 (pathname/argument/environment/funcname) 」をご覧ください。

- プログラムの終了

  スレッド型プログラムでは、プログラムの終了に STOP RUN を使用することはできません。STOP RUN を使用している場合は、EXIT PROGRAM を使用するように修正します。

◆ MeFt/Web 運用時の追加エラーコード

MeFt/Web の運用時には、スタンドアロンでのエラーコードに加えて、次のエラーコードが通知されます。

エラーコードを判定して処理を分けているプログラムを MeFt/Web で運用する場合は、次のコードも考慮した判定を行うようにします。エラーコードの通知については、本テキストの「2.2 表示ファイルのプログラミング」の「エラー処理」に記載があります。

通知コード	FILE STATUS 句(4 桁)に通知される内容	エラー内容
N1	90N1	WWW サーバが正常に通信を行うことができなかったため、リモート実行処理を続行できなくなりました。または、クライアントマシンかサーバマシンでメモリ不足が発生しました。
N7	90N7	MeFt/Web サーバで通信監視時間のタイムアウトが発生しました。
N8	90N8	MeFt/Web コントロールの Quit メソッドが実行されました。

なお、本章で構築するアプリケーションでは、画面機能および帳票機能の各ファイル FILE STATUS 句に指定した 4 桁のデータ名の領域の値が”0000”であるかを判定して処理を分けています。そのため、MeFt/Web の運用時に追加されるエラーコードに対応するための修正は特に行いません。

3.3.2　アプリケーション資産の配置と環境設定

MeFt/Web でリモート実行するアプリケーションの資産を WWW サーバに配置し、資産に関する環境設定を行います。資産に関する環境設定としては、仮想ディレクトリの指定と画面帳票資産の格納先の設定があります。

◆　アプリケーション資産の配置

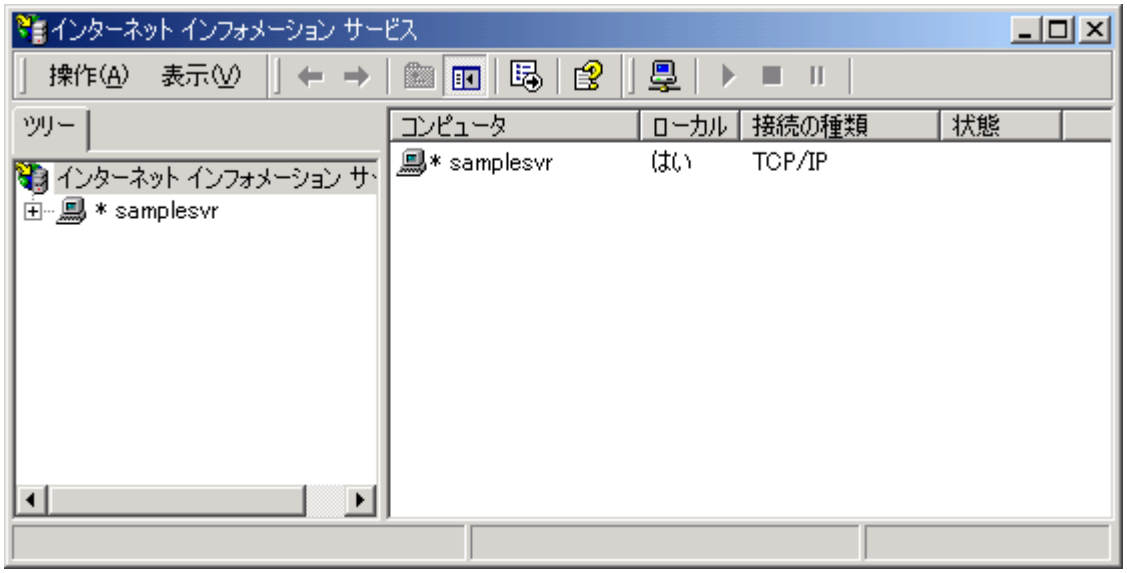
第 2 章で作成した画面帳票アプリケーションの資産を WWW サーバに配置します。本章で構築する MeFt/Web アプリケーションでは、次に示す資産を WWW サーバの「C:¥ADDRESS」に配置します。

資産の種類	ファイル名
実行可能ファイル	ADDR.EXE
画面帳票定義体	ADDRDSP.SMD
	ADDRPRT.SMD
オーバレイ定義体	ADDRPRT.OVD
実行用の初期化ファイル	COBOL85.CBR
ウィンドウ情報ファイル	DSP.ENV
プリンタ情報ファイル	PRT.ENV

◆　仮想ディレクトリの設定

配置されたアプリケーションの資産を WWW サーバから参照できるようにするため、アプリケーションの資産が格納されているディレクトリを仮想ディレクトリとして WWW サーバに設定します。
仮想ディレクトリを設定する方法を説明します。

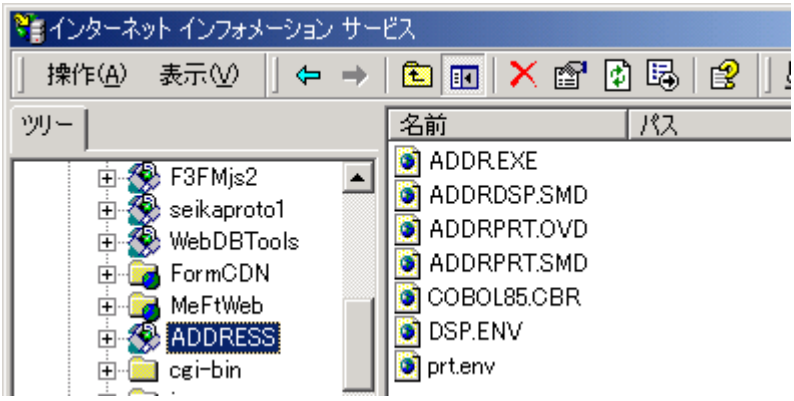
(1)[スタート]ボタン－[プログラム]－[管理ツール]から[インターネット サービス マネージャ]を選択して IIS の画面を表示します。



(2)ホスト名から「規定の Web サイト」を選択し、[操作]メニュー－[新規作成]から[仮想ディレクトリ]を選択します。

(3)「仮想ディレクトリの作成ウィザード」が表示されますので、次に示す内容を設定します。

設定箇所	設定内容
エイリアス	ADDRESS
ディレクトリへのパス	C:¥ADDRESS



参考

上記の説明は、IIS 5.0 での設定方法です。IIS 4.0 では、次の方法で設定します。

(1)[スタート]ボタン－[プログラム]－[Windows NT 4.0 Option Pack]－[Microsoft Internet Information Server]から[インターネット サービス マネージャ]を選択して IIS の画面を表示します。
(2)ホスト名から「規定の Web サイト」を選択し、[動作]メニュー－[新規作成]から[仮想ディレクトリ]を選択します。
(3)「新しい仮想ディレクトリのウィザード」が表示されますので、エイリアスとディレクトリへのパスを設定します。

参考

Windows Server 2003 の IIS6.0 を利用する場合には、以下の設定が必要です。
NetCOBOL の FAQ サイトのトラブルシューティング配下
http://software.fujitsu.com/jp/cobol/faq/trouble_sum.html
「異常」→「Windows/.NET」→「MeFt/Web」
→「Windows Server 2003 の IIS6.0 を利用してリモート実行できません。」をご参照ください。

◆ 画面帳票資産の格納先の設定

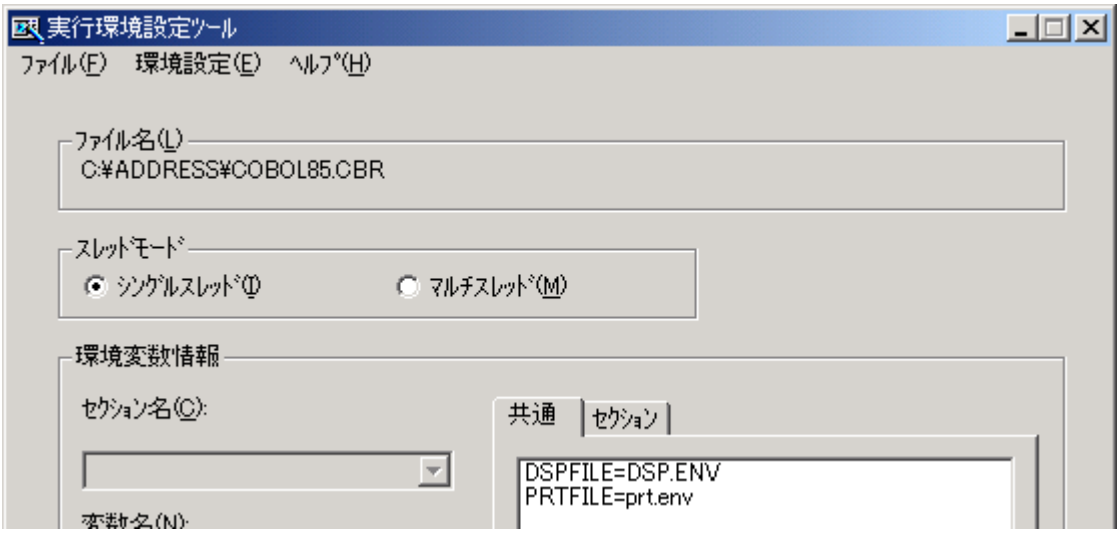
MeFt/Web アプリケーションが使用する、画面帳票定義体やウィンドウ情報ファイルといった MeFt 資産の格納先を指定します。MeFt/Web アプリケーションでは、資産の格納先の指定方法として、URL 指定とサーバのローカルパス指定がありますが、ここでは URL で指定します。

ウィンドウ情報ファイルおよびプリンタ情報ファイルの格納先の指定

第 2 章で作成したアプリケーションでは、COBOL 初期化ファイル（COBOL85.CBR）で、ファイル識別名に対応するウィンドウ情報ファイルおよびプリンタ情報ファイルをフルパスで指定しました。しかし、MeFt/Web 環境で運用する場合、ウィンドウ情報ファイルおよびプリンタ情報ファイルの指定は、ファイル名のみを推奨します。

ウィンドウ情報ファイルおよびプリンタ情報ファイルをファイル名のみの指定に変更する手順を説明します。

(1)COBOL85.CBR を開き、実行環境設定ツールのリストボックスに表示される実行環境情報を選択します。
(2)変数値のエディットボックスに表示されるウィンドウ情報ファイル名やプリンタ情報ファイル名からパス名を削除し、ファイル名のみの指定にします。



ウィンドウ情報ファイルの格納ディレクトリは、アプリケーション実行時の環境変数 MEFTWEBDIR に指定します。
一方、プリンタ情報ファイルについては、サーバに接続されているプリンタとクライアントに接続されているプリンタが異なるため、サーバ印刷用のプリンタ情報ファイルとクライアント印刷用のプリンタ情報ファイルを用意しますが、サーバ印刷用のプリンタ情報ファイルとクライアント印刷用のプリンタ情報ファイルとでは、格納ディレクトリの指定が異なります。
サーバ印刷用のプリンタ情報ファイルは、アプリケーション実行時の環境変数 MEFTDIR に指定します。それに対し、クライアント印刷用のプリンタ情報ファイルの格納ディレクトリは、アプリケーション実行時の環境変数 MEFTWEBDIR に指定します。
なお、サーバ印刷用のプリンタ情報ファイルのファイル名とクライアント印刷用のプリンタ情報ファイルのファイル名は同一である必要があります。

参考

アプリケーション実行時の環境変数を設定するには、以下のような方法があります。

- 実行用の初期化ファイルで設定する
- MeFt/Web をリモート実行する HTML で、MeFt/Web コントロールの environment プロパティで設定する
- システム環境変数で設定する
- SET コマンドで設定する

MeFt/Web コントロールの environment プロパティでの設定方法については、本テキストの「3.4 HTML の作成」および MeFt/Web 説明書「3.4.3 利用者プログラムの指定方法（pathname/argument/environment/funcname）」をご覧ください。
システム環境変数および SET コマンドで設定する方法については、NetCOBOL 使用手引書「5.3.2 実行環境情報の設定方法」をご覧ください。

画面帳票定義体の格納先の指定

第 2 章で作成したアプリケーションにおいては、画面帳票定義体の格納ディレクトリは、ウィンドウ情報ファイルおよびプリンタ情報ファイルの MEDDIR キーワードにてローカルパスで指定しましたが、ここでは次に示すように URL で指定します。

```
MEDDIR http://SampleSvr/ADDRESS
```

オーバレイ定義体の格納先の指定

第 2 章で作成したアプリケーションにおいては、オーバレイ定義体の格納ディレクトリは、プリンタ情報ファイルの OVLPDIR キーワードにてローカルパスで指定しましたが、ここでは次に示すように URL で指定します。

```
OVLPDIR http://SampleSvr/ADDRESS
```

画像ファイルの指定

画面帳票定義体には、画像ファイルを出力するための項目として「組み込みメディア項目」があり、プログラムの実行時にビットマップファイルなどを画面帳票定義体に出力することができます。組み込みメディア項目に使用するビットマップファイルの格納ディレクトリは、ウィンドウ情報ファイルおよびプリンタ情報ファイルの MEDIADIR キーワードで指定します。

また、画面定義体の背景にビットマップファイルを表示することもできます。画面定義体の背景に表示する画像ファイルを「背景メディア」といいます。背景メディアに使用するビットマップファイルの格納ディレクトリは、ウィンドウ情報ファイルの BACKMEDIA キーワードで指定します。

スタンドアロンで組み込みメディア項目および背景メディアを使用していたアプリケーションを MeFt/Web 環境へ移行する場合は、MEDIADIR キーワードおよび BACKMEDIA キーワードを使用して、ビットマップファイルの格納ディレクトリを URL で指定します。

第 2 章で作成したアプリケーションでは、組み込みメディア項目および背景メディアの定義を行わなかったため、ここでは、MEDIADIR キーワードおよび BACKMEDIA キーワードの指定は行いません。

参考

URL で指定する資産は、上記で説明したもののみです。初期化ファイルで指定するデータファイルなどは、通常通りローカルパスで指定します。

3.4 HTML の作成

WWW サーバ上のプログラムをリモート実行するには、HTML を作成する必要があります。
HTML では、次に示すような内容を記述します。

- MeFt/Web コントロールの指定
- アプリケーションの起動方法
- WWW サーバの指定
- リモート実行するアプリケーションの指定
- リモート実行時の動作に関する指定
- ページ移動時の動作に関する指定

ここでは、MeFt/Web のサンプルプログラムで提供されている HTML をコピーし、必要な部分を修正します。
サンプルプログラムの HTML は次に示す場所にある「denpyou1.htm」です。

```
C:¥Program Files¥NetCOBOL¥Samples¥MeFtWeb¥Sample.web¥
```

修正した HTML を以下に示します。下線付きの部分が修正した箇所です。
なお、この HTML に「address.htm」というファイル名をつけ、「C:¥ADDRESS」に格納することとします。

```
<HTML>
<HEAD>
<TITLE>住所録</TITLE>
</HEAD>
<BODY>
<INPUT TYPE=BUTTON VALUE="GO!" NAME="GO"><BR>
<OBJECT
ID="MeFtWeb1"
CLASSID="CLSID:61F12C43-5357-11D0-9EA0-00000E4A0F56"
WIDTH="670" HEIGHT="470"
CODEBASE="http://SampleSvr/meftweb/meftweb.cab#version=8,0,10,1">
</OBJECT>
<SCRIPT LANGUAGE="VBScript">
Sub GO_onClick()
MeFtWeb1.hostname = "SampleSvr"
MeFtWeb1.pathname = "C:¥ADDRESS¥ADDR.EXE"
MeFtWeb1.environment = "MEFTWEBDIR=http://SampleSvr/ADDRESS"
MeFtWeb1.message = TRUE
MeFtWeb1.usedcgi = FALSE
MeFtWeb1.ssl = FALSE
MeFtWeb1.displaywindow = 0
MeFtWeb1.hyperlink = 0
MeFtWeb1.printmode = 0
MeFtWeb1.previewwindow = 0
MeFtWeb1.previewdrawpos = 0
MeFtWeb1.previewdc = 0
MeFtWeb1.previewrate = 0
MeFtWeb1.submit()
end sub
Sub Window_onUnload()
MeFtWeb1.Quit()
end sub
</SCRIPT>
</BODY>
</HTML>
```

MeFt/Web コントロールの指定

MeFt/Web の画面を表示するため、MeFt/Web コントロールの指定を行います。

MeFt/Web コントロールの指定は、HTML の OBJECT タグに記述します。CODEBASE で指定する MeFt/Web コントロールの格納先には、WWW サーバのホスト名を記述します。

本章で作成する HTML では、以下の記述で MeFt/Web コントロールを指定しています。

```
<OBJECT
ID="MeFtWeb1"                                                  ←オブジェクト名
CLASSID="CLSID:61F12C43-5357-11D0-9EA0-00000E4A0F56"           ←クラス ID(固定)
WIDTH="670" HEIGHT="470"                                       ←HTML で表示する際のサイズ
CODEBASE="http://SampleSvr/meftweb/meftweb.cab#version=8,0,10,1">  ←格納先とバージョン
</OBJECT>
```

アプリケーションの起動方法

どのような操作によってアプリケーションを起動するのかを指定します。アプリケーションの起動には、MeFt/Web コントロールの submit メソッドを使用します。

本章で作成する HTML では、INPUT タグで指定されたボタンが押されたら起動するようにしています。

```
<INPUT TYPE=BUTTON VALUE="GO!" NAME="GO">
   :
<SCRIPT LANGUAGE="VBScript">
Sub GO_onClick()
   :
MeFtWeb1.submit()
end sub
   :
</SCRIPT>
```

参考

HTML の表示と同時にアプリケーションを起動するには、以下のように記述します。

```
<SCRIPT LANGUAGE="VBScript">
Sub Window_onload()
   :
MeFtWeb1.submit()
end sub
   :
</SCRIPT>
```

WWW サーバの指定

アプリケーションが格納されている WWW サーバのホスト名を指定します。

WWW サーバのホスト名は、MeFt/Web コントロールの hostname プロパティで指定します。WWW サーバ名を省略することはできません。なお、サーバとクライアントが異なるドメインに所属する場合は、hostname プロパティで指定するホスト名をフルドメイン形式で指定します。

本章で作成した HTML では、以下の記述で WWW サーバを指定しています。

```
MeFtWeb1.hostname = "SampleSvr"
```

リモート実行するアプリケーションの指定

どのアプリケーションをリモート実行するのかを指定します。

リモート実行するアプリケーション名は、MeFt/Web コントロールの pathname プロパティにサーバのローカルパスで指定します。アプリケーション名を省略することはできません。

また、アプリケーション実行時の環境変数は、MeFt/Web コントロールの environment プロパティで指定することができます。

本章で作成した HTML では、以下の記述でアプリケーション名と環境変数を指定しています。

```
MeFtWeb1.pathname = "C:¥ADDRESS¥ADDR.EXE"
MeFtWeb1.environment = "MEFTWEBDIR=http://SampleSvr/ADDRESS"
```

参考

ここでは、環境変数として MEFTWEBDIR を指定しています。環境変数 MEFTWEBDIR については、本テキストの「3.3.2 アプリケーション資産の配置と環境設定」をご覧ください。

また、サーバ印刷用のプリンタ情報ファイルの格納ディレクトリを指定する環境変数 MEFTDIR は指定していません。その場合、サーバ印刷時の出力プリンタは、MeFt/Web 動作環境の「サーバ印刷用の出力プリンタデバイス名」の設定に従います。

リモート実行時の動作に関する指定

画面の表示形式や印刷先の指定など、リモート実行時の動作に関する指定は、MeFt/Web コントロールの各プロパティで指定します。

本章で作成した HTML では、以下の記述でリモート実行時の動作に関する指定を行っています。MeFt/Web コントロールの各プロパティについては、MeFt/Web 説明書「3.4 プロパティ」をご覧ください。

```
MeFtWeb1.message = TRUE
MeFtWeb1.usedcgi = FALSE
MeFtWeb1.ssl = FALSE
MeFtWeb1.displaywindow = 0
MeFtWeb1.hyperlink = 0
MeFtWeb1.printmode = 0
MeFtWeb1.previewwindow = 0
MeFtWeb1.previewdrawpos = 0
MeFtWeb1.previewdc = 0
MeFtWeb1.previewrate = 0
```

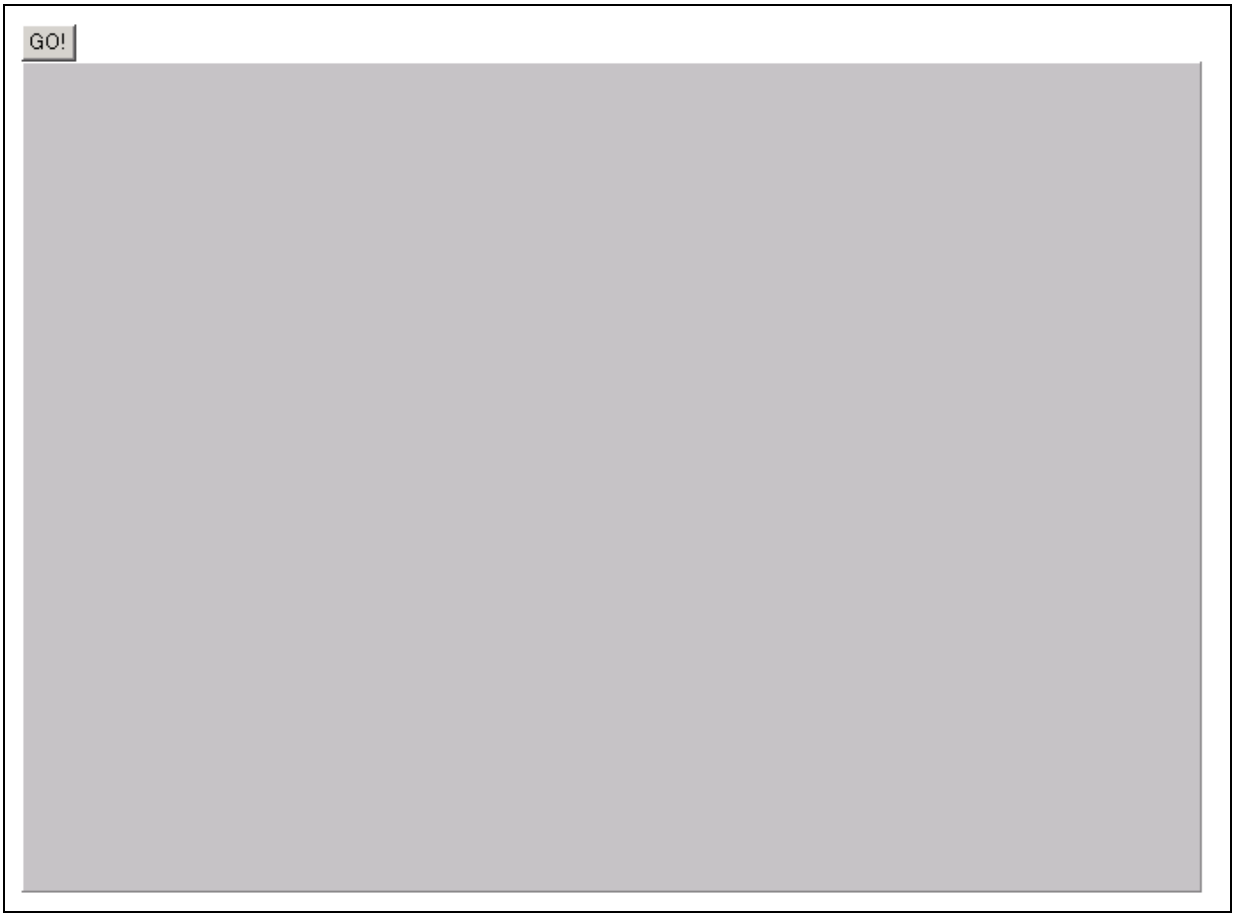
ページ移動時の動作に関する指定

リモート実行中にページが移動された場合の動作は、ページ移動時に発生する Window_onUnload イベントの処理として指定します。リモート実行中にページが移動された場合、そのことをアプリケーションに通知するため、MeFt/Web コントロールの Quit メソッドを記述します。

本章で作成した HTML では、以下の記述でページ移動時の動作に関する指定を行っています。詳細については、本テキストの「3.7 通信が切断されるパターンについて」をご覧ください。

```
<SCRIPT LANGUAGE="VBScript">
   :
Sub Window_onUnload()
MeFtWeb1.Quit()
end sub
   :
</SCRIPT>
```

作成した HTML を WWW ブラウザで表示すると、次のようになります。



3.5 リモート実行

WWW サーバマシン上に配置した COBOL アプリケーションを WWW ブラウザからリモート実行する方法を説明します。

3.5.1 HTML の表示

WWW ブラウザで、MeFt/Web をリモート実行する HTML を表示するために、次のような URL を指定します。

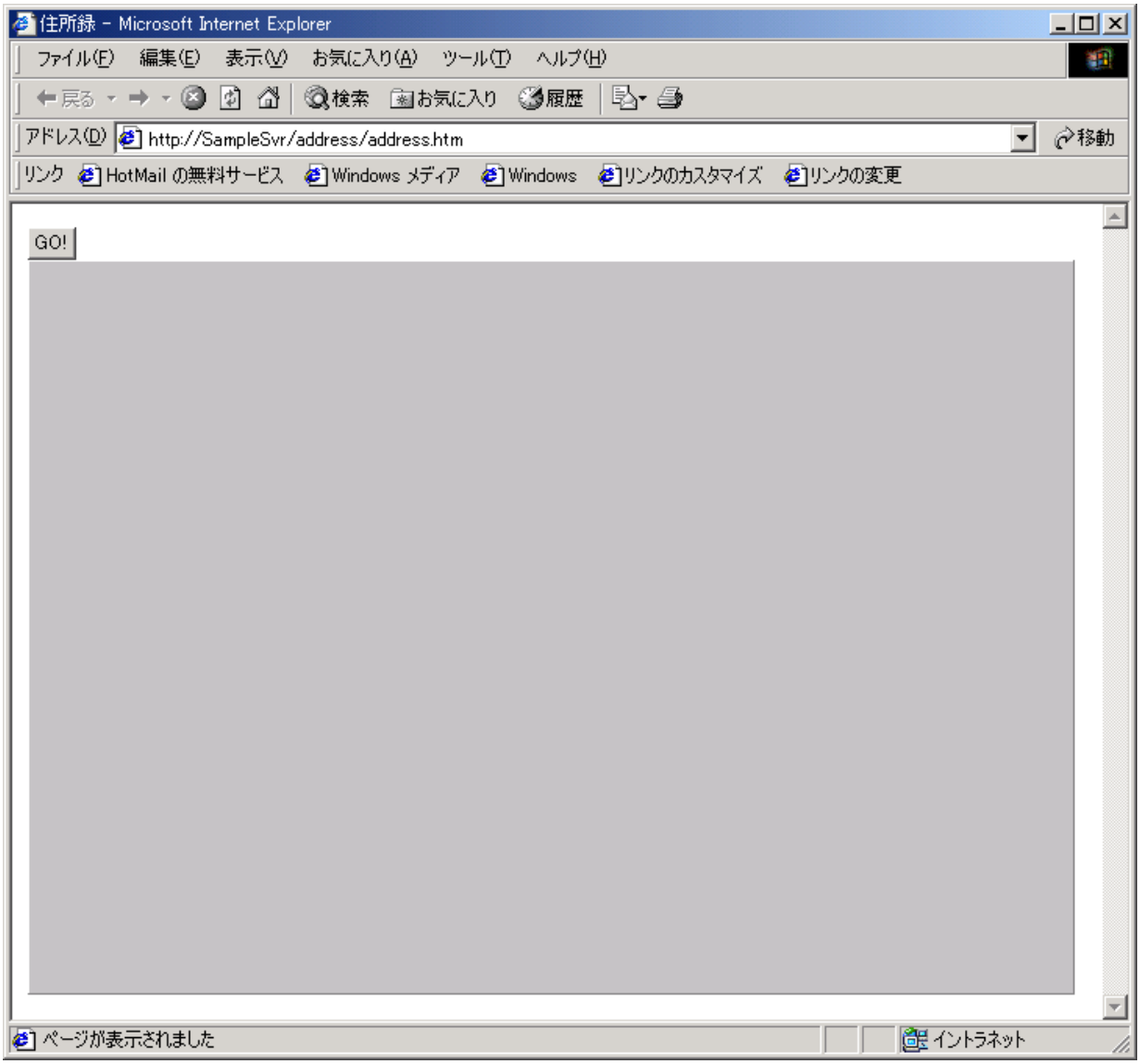
```
http://SampleSvr/ADDRESS/address.htm
```

リモート実行を行うには、クライアントに MeFt/Web コントロールをダウンロードする必要がありますが、WWW ブラウザでリモート実行するための HTML を表示すると、サーバマシンから MeFt/Web コントロールが自動的にダウンロードされます。

MeFt/Web コントロールのダウンロード時、ActiveX コントロールの認証を行うダイアログが表示されますので、「はい」を押してダウンロードします。

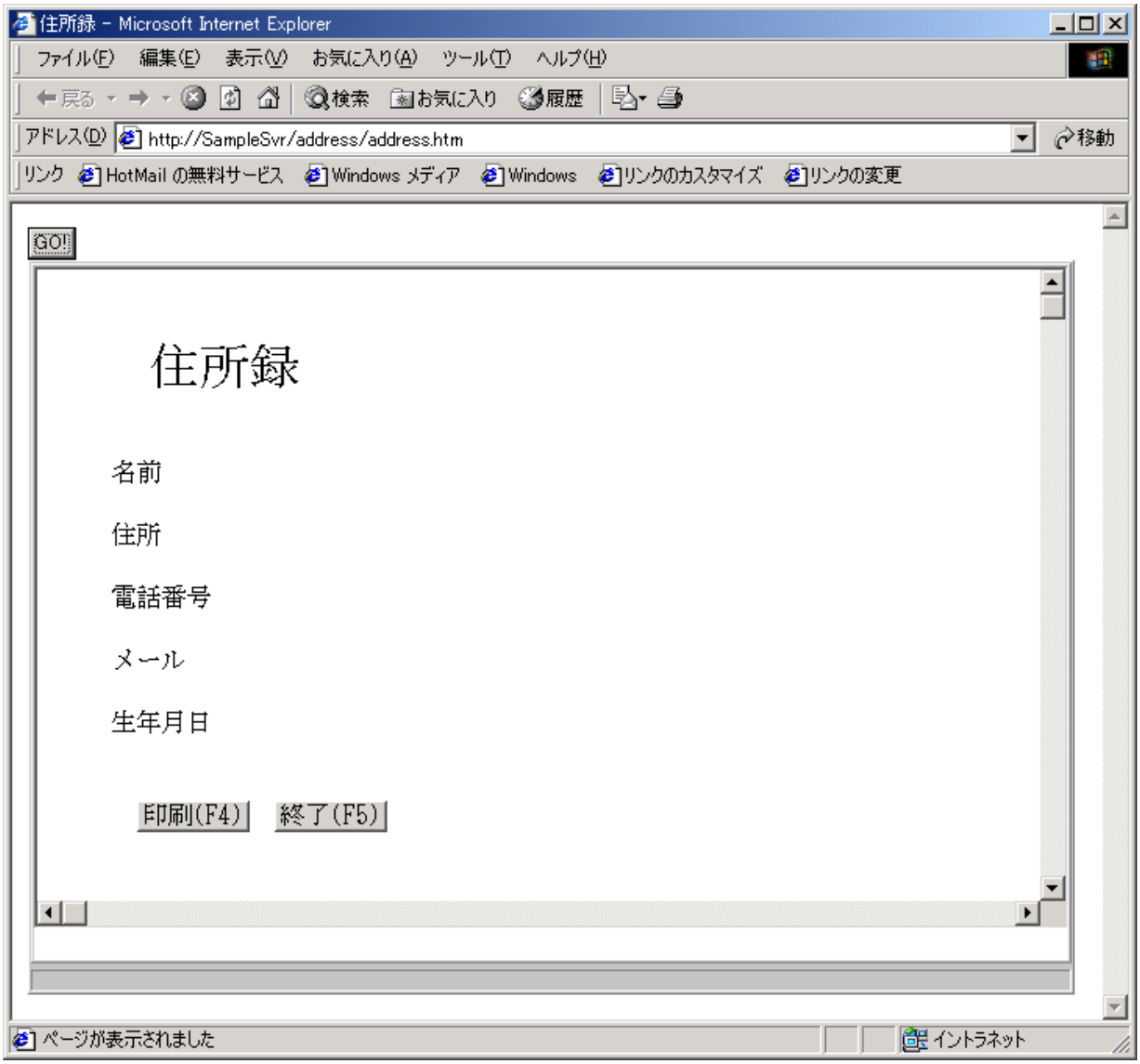
なお、MeFt/Web コントロールがダウンロードされるのは、MeFt/Web サーバへの初回の接続時のみです。2 回目以降は、ダウンロードされている MeFt/Web コントロールが使用されます。

MeFt/Web コントロールがダウンロードされると、次のような画面が表示されます。



3.5.2 プログラムの実行

HTML の「GO!」ボタンを押すと、プログラムが起動し、次のような画面が WWW ブラウザに表示されます。

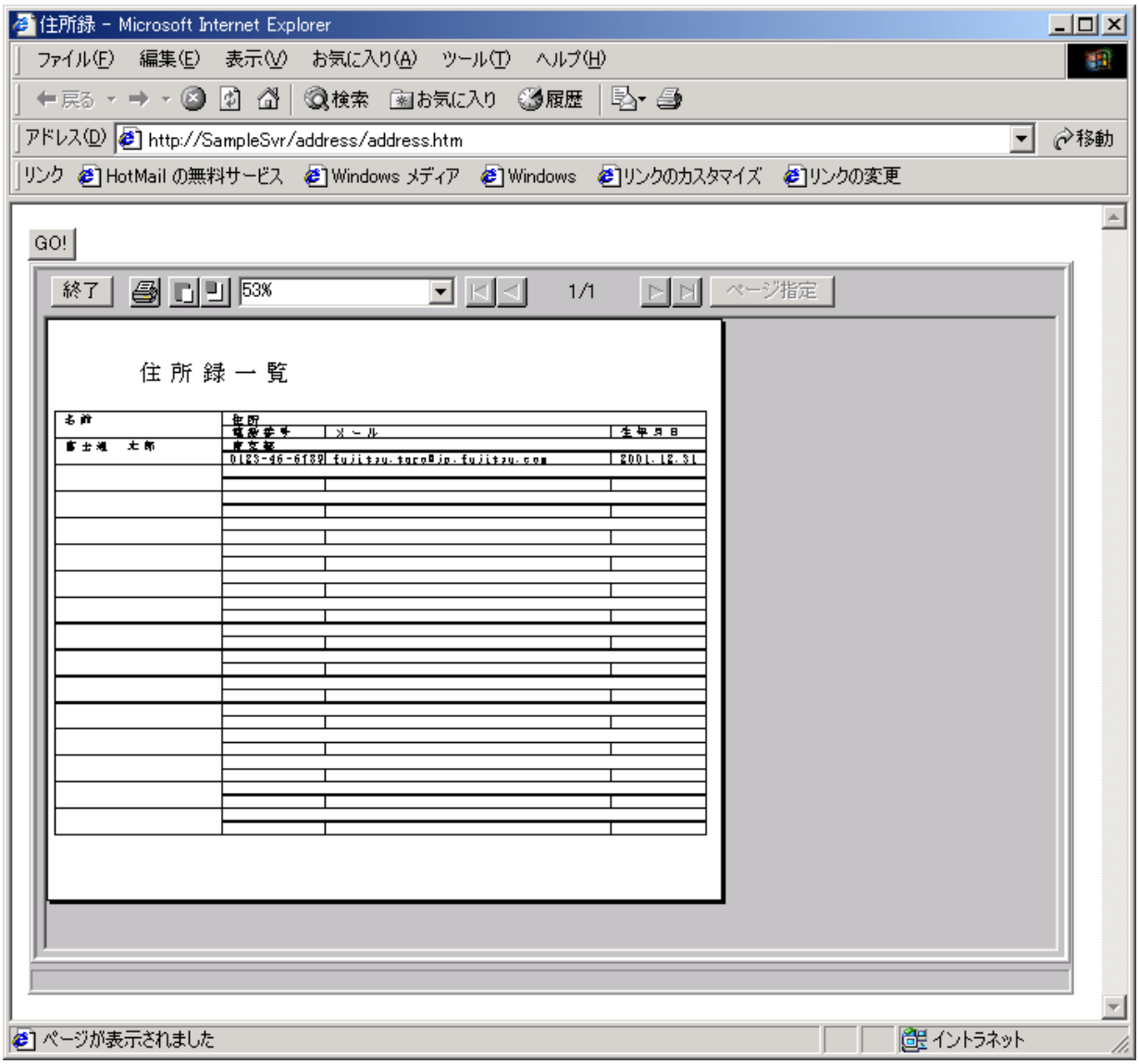


表示された画面に必要な情報を入力し、F4 キーまたは画面上の「印刷(F4)」というボタンを押すと、サーバのアプリケーションは印刷処理を行います。

MeFt/Web の印刷機能には次の 4 つがあります。

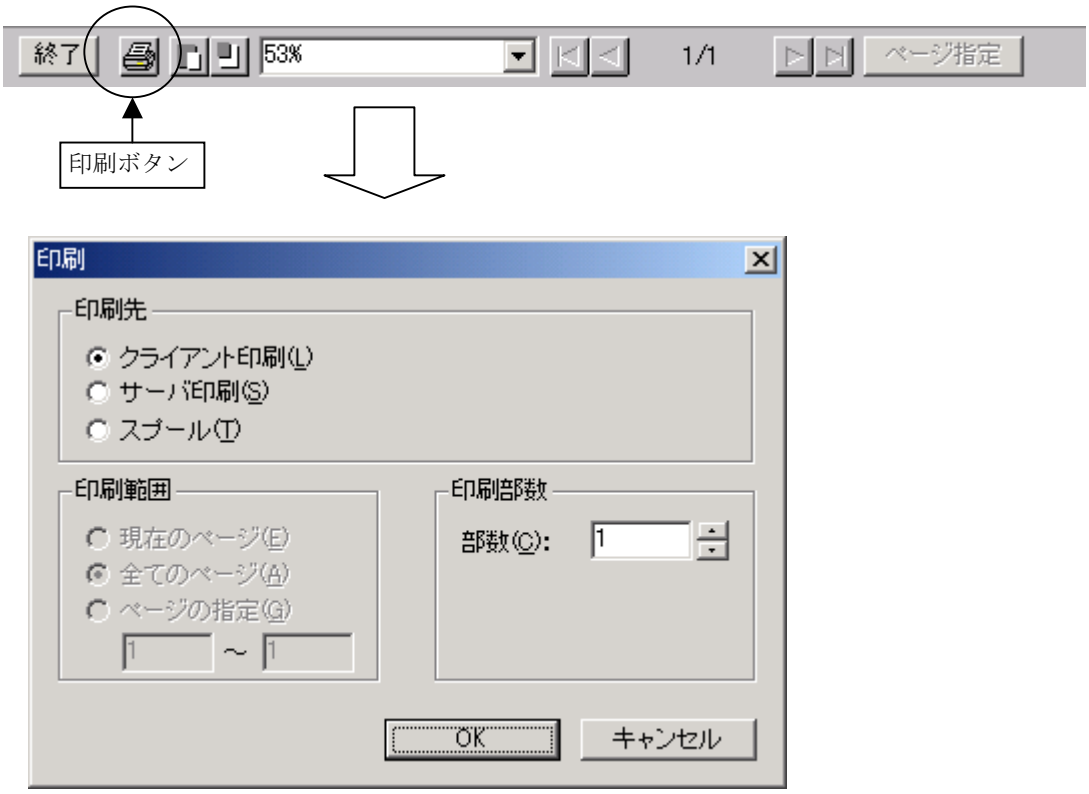
- プレビュー
- クライアント印刷
- サーバ印刷
- スプール

アプリケーションが印刷処理を行った場合、どのような印刷を行うかはリモート実行するための HTML で定義しますが、今回使用する HTML では、省略値であるプレビューが設定されています。
以下は、帳票印刷のプレビュー表示の例です。



プレビュー表示からの印刷

印刷プレビューの画面で印刷ボタンを押すと、印刷先などを設定する画面が表示されます。この画面で印刷先を設定すると、印刷プレビューの画面から指定した印刷先に印刷を行うことができます。



プレビューの終了

プレビュー表示を終了するには、プレビュー画面の「終了」ボタンを押します。

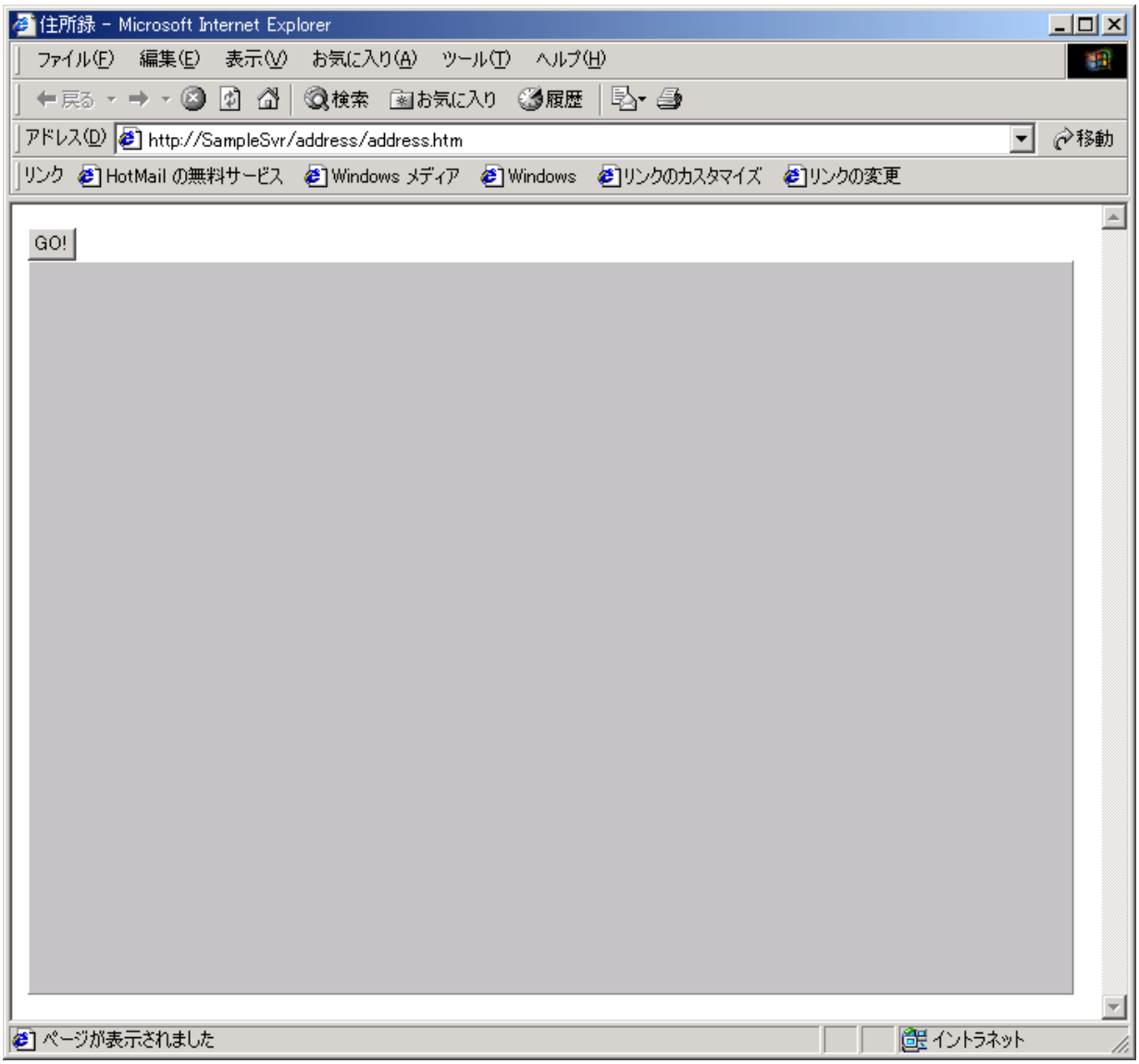
参考

帳票定義体を使用した印刷処理では、プレビューやクライアント印刷などの MeFt/Web の印刷機能が使用できます。

一方、帳票定義体を使用しない印刷処理では、MeFt/Web の印刷機能を使用しないサーバアプリケーションの印刷として処理されます。

プログラムの終了

ここでリモート実行されたアプリケーションでは、入力画面で F5 キーまたは画面上の「終了(F5)」というボタンを押すと、プログラムが終了します。プログラムが終了すると、WWW ブラウザは、リモート実行前と同じように表示されます。



3.6 デバッグ

MeFt/Web 環境のアプリケーションのデバッグ手順や方法について説明します。

3.6.1 MeFt/Web アプリケーションのデバッグについて

MeFt/Web 環境の COBOL 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。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では、プログラムを実行するとデバッガが自動的に起動し、デバッグ操作を行うことができます。

3.6.2 デバッグの準備

デバッグの準備として、次の 2 点の作業を行います。

- デバッグモジュールの作成とデバッグ資産の配置
- 実行環境情報の設定

◆ デバッグモジュールの作成とデバッグ資産の配置

デバッグモジュールを作成し、デバッグに必要となる資産を WWW サーバマシンに配置します。

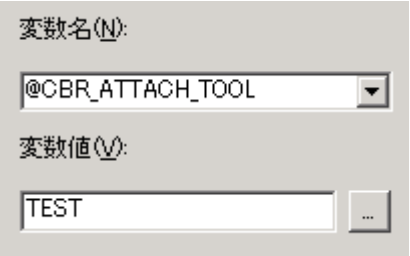
デバッグモジュールの作成には、MeFt/Web アプリケーション固有の部分はありません。第 1 章で作成したアプリケーションと同じ方法で作成します。

デバッグモジュールを作成するためのビルド、あるいはリビルドが終了したあと、次の資産を WWW サーバマシンの「C:¥ADDRESS」に配置します。

資産の種類	ファイル名
デバッグモジュール	ADDR.EXE
デバッグ情報ファイル	ADDR.SVD
COBOL ソースファイル	ADDR.COB

◆ 実行環境情報の設定

COBOL の実行環境情報で、デバッグしたいプログラムからデバッガを起動する設定を行います。
デバッグしたいプログラムからデバッガを起動する設定は、実行環境情報で実行環境変数「@CBR_ATTACH_TOOL」を使用します。「@CBR_ATTACH_TOOL」の変数値に「TEST」と指定することで、デバッグしたいプログラムからデバッガを起動できるようになります。「TEST」に続けてデバッガ起動時のパラメータ（デバッグ情報ファイルの格納フォルダなど)を指定できますが、本章で構築するアプリケーションでは必要ありません。

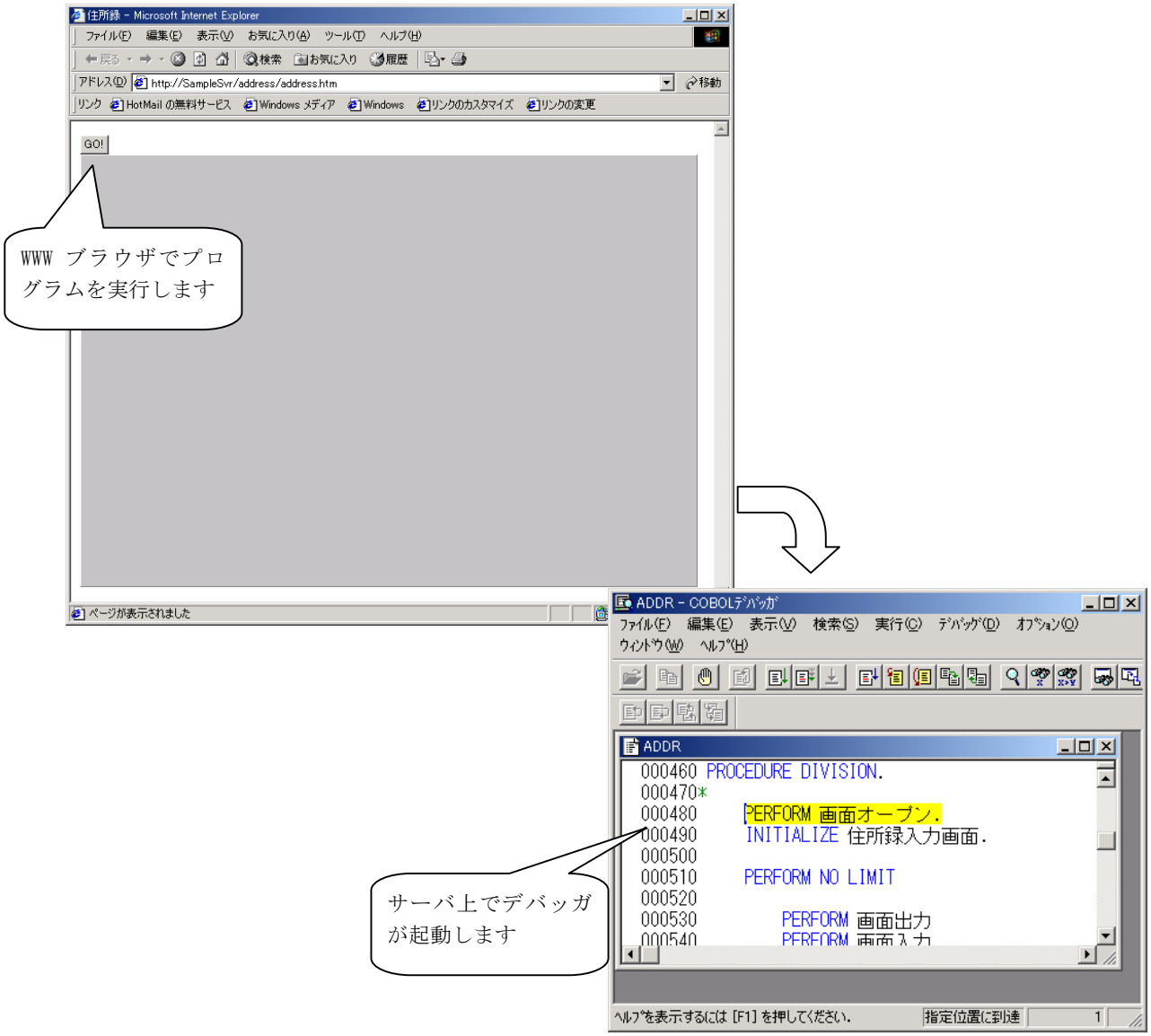


なお、本章で構築するアプリケーションでは、実行環境情報は実行時の初期化ファイルである C:¥ADDRESS¥COBOL85.CBR に設定します。

3.6.3 デバッガの起動

プログラムを通常に実行するときと同じように、WWW ブラウザから WWW サーバの COBOL プログラムをリモート実行します。このとき、WWW ブラウザはデバッグする COBOL プログラムが実行されるサーバマシンでなくてもかまいません。

COBOL プログラムが起動されたとき、サーバマシン上に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]ダイアログが表示されますので、「OK」ボタンを選択してデバッグを開始します。



3.6.4 デバッグ操作

デバッグ操作には、MeFt/Web アプリケーション固有の部分はありません。第 1 章で作成したアプリケーションと同じデバッグ操作が行えます。

3.6.5 デバッガの終了

デバッガの「ファイル」メニューから「デバッガの終了」を選択し、デバッガを終了します。

3.7 通信が切断されるパターンについて

MeFt/Web アプリケーションの運用中に、サーバとクライアントとの間の通信が切断されるパターンとして、次の 2 つがあります。

- WWW ブラウザで「戻る」ボタンが押されたり、WWW ブラウザのアドレスバーで別の URL が指定されたとき
- クライアントマシンの電源が切断されたり、ネットワークが不通になったとき

サーバとクライアントとの間の通信が切断された場合、サーバのアプリケーションはそのことを認識できず、クライアントからの入力を待ちつづける状態となります。しかし、通信切断によりクライアントから応答を返すことができないため、正常にプログラムを終了することができなくなります。プログラムを終了するには強制終了するしかありませんが、ファイルやデータベースが正常にクローズされないため、データに影響が出る可能性があります。

MeFt/Web アプリケーションを構築するにあたっては、このような通信切断時への対応を考慮する必要があります。以下に、とるべき対応について説明します。

◆ WWW ブラウザで「戻る」ボタンが押されたり、WWW ブラウザのアドレスバーで別の URL が指定されたとき

WWW ブラウザで「戻る」ボタンが押されたり、WWW ブラウザのアドレスバーで別の URL が指定されたりすると、他のページに移動します。その結果、それまで入出力を行っていたページとサーバとの通信が切断されてしまいます。この状況に対応するには、MeFt/Web コントロールの Quit メソッドを使用します。Quit メソッドを使用すると、WWW ブラウザでイベントが発生したことをアプリケーションに通知することができます。

リモート実行に使用する HTML で、WWW ブラウザのページが遷移したときに発生する Window_onUnload イベントの処理として MeFt/Web コントロールの Quit メソッドを実行するように記述すると、WWW ブラウザで「戻る」ボタンが押されるなどしてページが移動した場合、Quit メソッドが実行されてアプリケーションにコードが通知されます。

HTML での Quit メソッドの記述例を以下に示します。

```
<HTML>
  :
<OBJECT
ID="MeFtWeb1"
  :
</OBJECT>
<SCRIPT LANGUAGE="VBScript">
  :
Sub Window_onUnload()
MeFtWeb1.Quit()
end sub
  :
</SCRIPT>
</HTML>
```

COBOL アプリケーションには、表示ファイルの FILE STATUS 句に指定された 4 桁のデータ名の領域に"90N8"で通知されます。それを判定することによってページが移動されたことを知ることができるので、ファイルのクローズやデータベースの切断などの後処理を行うようにします。

MeFt/Web コントロールの Quit メソッドについては、本テキストの「3.4 HTML の作成」および、MeFt/Web 説明書「3.5.3 利用者プログラムの中断（Quit）」をご覧ください。

◆　クライアントマシンの電源が切断されたり、ネットワークが不通になったとき

クライアントマシンの電源が切断されたり、ネットワークが不通になったりして、サーバとクライアントとの間の通信が切断された状況に対応するには、MeFt/Web サーバの通信監視時間の機能を使用します。

MeFt/Web サーバに通信監視時間を設定すると、設定された通信時間を越えてクライアントからレスポンスがなかった場合、MeFt/Web サーバからアプリケーションにコードを通知します。COBOL アプリケーションには、表示ファイルの FILE STATUS 句に指定された 4 桁のデータ名の領域に"90N7"で通知されます。それを判定することによって通信監視時間を超えて応答がなかったことを知ることができるので、ファイルのクローズやデータベースの切断などの後処理を行うようにします。

MeFt/Web の通信監視時間の設定については、本テキストの「3.2.1 MeFt/Web 動作環境の設定」および、MeFt/Web 説明書「2.2 MeFt/Web の動作環境を設定する」をご覧ください。

<u>参考</u>

設定される通信監視時間が長すぎると、サーバとクライアントとの間の通信が切断されているにもかかわらずアプリケーションは起動し続けることになり、サーバの負荷が高まります。

一方、設定される通信監視時間が短すぎると、画面での作業に時間がかかった場合、突然アプリケーションが終了することになってしまいます。

通信監視時間は、業務の内容に応じて適切な値を設定してください。

MEMO

おわりに

今後、NetCOBOL を使用してプログラム開発を行うにあたり、参考となるマニュアルの箇所やサンプルプログラムについて紹介します。

マニュアル

- NetCOBOL 開発環境によるプログラムの作成について

NetCOBOL 使用手引書	第 2 章 プログラムを作成・編集する
	第 3 章 プログラムを翻訳する
	第 4 章 プログラムをリンクする
	第 5 章 プログラムを実行する
	第 6 章 プロジェクト管理機能

- 表示ファイルの画面機能について

NetCOBOL 使用手引書	第 9 章 画面を使った入出力 9.2 表示ファイル(画面入出力)の使い方
MeFt 説明書	第 3 章 開発と運用 3.1 画面機能
	第 4 章 画面機能

- 表示ファイルの印刷機能について

NetCOBOL 使用手引書	第 8 章 印刷処理 8.5 表示ファイル(帳票印刷)の使い方
MeFt 説明書	第 3 章 開発と運用 3.2 印刷機能
	第 5 章 印刷機能

- FORM による画面帳票定義体の作成について

FORM ヘルプ

- MeFt/Web の設定と運用について

MeFt/Web 説明書	第 2 章 セットアップ
	第 3 章 MeFt/Web コントロール
	第 4 章 リモート実行機能を利用する

- メッセージについて

NetCOBOL 使用手引書	付録 F メッセージ一覧
MeFt 説明書	第 8 章 リファレンス 8.4 通知コード
MeFt/Web 説明書	付録 - MeFt/Web コントロールのエラーメッセージ - MeFt/Web プラグインのエラーメッセージ - MeFt/Web サーバのイベントログ

サンプル

- 索引ファイルと連携した画面帳票プログラム

格納場所：NetCOBOL のインストールディレクトリ\SAMPLES\COBOL\SAMPLE03\

画面定義体に入力したデータおよび索引ファイルから読み込んだデータを画面に出力し、画面で入力されたデータを索引ファイルに出力したり、帳票定義体を使用して印刷するプログラムです。
このサンプルプログラムの説明については、サンプルプログラムが格納されているディレクトリにある「Sample3.txt」をご覧ください。

- ２つの画面定義体を使用するプログラム

格納場所：NetCOBOL のインストールディレクトリ\Samples\MeFtWeb\

２つの画面定義体を親画面、子画面として使用するプログラムです。親の画面定義体に入力するデータを子画面の画面定義体に表示されるデータから選択できます。また、入力されたデータは、帳票定義体を使用して印刷します。
このサンプルプログラムの説明については、MeFt/Web 説明書「2.6 サンプルプログラム」をご覧ください。

■ **NetCOBOL for Windows の体験版について**

NetCOBOL ホームページ

http://software.fujitsu.com/jp/cobol/

「体験版」より、お申込みください。

■ **本書に関するお問い合わせ先**

NetCOBOL ホームページ

http://software.fujitsu.com/jp/cobol/

「本製品に関するお問い合わせ」より、お願いします。

※　ご質問に関しては、記述箇所のページ、内容を明示してください。

※　ご質問の内容によっては、回答に数日またはそれ以上の期間を要する場合があります。

NetCOBOL

N e t C O B O L