



Net Express

入門書

Micro Focus NetExpress™

# 入門書

Micro Focus®

第 3 版

1998 年 10 月

Copyright © 1999 Micro Focus Limited. All rights reserved.

本書ならびに使用されている固有の商標と商品名は国際法で保護されています。

Micro Focus は、このマニュアルの内容が公正かつ正確であるよう万全を期しておりますが、このマニュアルの内容は予告なしに随時変更されることがあります。

このマニュアルに述べられているソフトウェアはライセンスに基づいて提供され、その使用および複製は、ライセンス契約に基づいてのみ許可されます。特に、Micro Focus 社製品のいかなる用途への適合性も明示的に本契約から除外されており、Micro Focus はいかなる必然的損害に対しても一切責任を負いません。

Micro Focus®、Animator®、および Visual® は、Micro Focus Limited の登録商標です。Data Access Wizard™、Dialog System™、Form Designer™、Form Express™、Micro Focus COBOL™、NetExpress™、Object COBOL™、Solo™、UNIX Option™、および WebSync™ は、Micro Focus Limited の商標です。

Microsoft®、Windows®、Win32® および MS-DOS® は、Microsoft Corporation の登録商標です。AutoPlay™、Internet Explorer™、および Windows NT™ は、Microsoft Corporation の商標です。

# 目次

第1章 概要.....	1-1
第2章 補足的なソフトウェア .....	2-1
第3章 情報の検索.....	3-1
第4章 ここからチュートリアルをスタート.....	4-1
4.1 概要 .....	4-1
4.2 用語と表記規則.....	4-1
4.3 ヒント.....	4-2
4.4 チュートリアル マップ.....	4-3
第5章 NetExpress の使用方法.....	5-1
5.1 概要 .....	5-1
5.1.1 プロジェクトとプロジェクト フォルダ .....	5-1
5.1.2 デモンストレーション アプリケーション .....	5-2
5.2 具体的な手順.....	5-2
5.2.1 IDE の起動.....	5-2
5.2.2 プロジェクトの読み込み.....	5-4
5.2.3 プロジェクトのビルド.....	5-5
5.2.4 アプリケーションの実行.....	5-6
5.2.5 アプリケーションのデバッグ .....	5-8
5.2.6 IDE オプションの設定 .....	5-9
5.2.7 アニメートの終了.....	5-10
5.2.8 コンテキスト ヘルプの使用法 .....	5-10
5.3 次へ進む前に.....	5-10

第6章 データ ファイルの維持と作成.....	6-1
6.1 概要 .....	6-1
6.2 準備 .....	6-1
6.3 具体的な手順.....	6-1
6.3.1 ファイルの変換.....	6-2
6.3.2 書式設定されていないデータ ファイルの表示.....	6-3
6.3.3 レコード レイアウト ファイルの作成 .....	6-5
6.3.4 デフォルトのレコード レイアウトの作成 .....	6-6
6.3.5 条件付きレコード レイアウトの作成 .....	6-6
6.3.6 レコード レイアウト ファイルの保存 .....	6-7
6.3.7 書式設定されたデータ ファイルの表示 .....	6-7
6.3.8 ファイルの 16 進表示.....	6-8
6.3.9 データ ファイルの印刷.....	6-8
6.3.10 複数のデータ ファイルの編集.....	6-9
6.3.11 新しいデータ ファイルの作成.....	6-10
6.4 次へ進む前に.....	6-11
第7章 Web アプリケーションの概要.....	7-1
7.1 Solo .....	7-1
7.2 サーバー側プログラム.....	7-2
7.2.1 構造.....	7-2
7.2.2 対称と非対称.....	7-2
7.3 次へ進む前に.....	7-3
第8章 Web アプリケーションの作成.....	8-1
8.1 概要 .....	8-1
8.2 準備 .....	8-1

8.3 具体的な手順.....	8-2
8.3.1 プロジェクトとフォームの作成.....	8-2
8.3.2 この段階の IDE 画面.....	8-3
8.3.3 フォームへのコントロールの追加.....	8-5
8.3.4 サーバー側プログラムの作成.....	8-7
8.3.5 作成されたファイル.....	8-8
8.3.6 フォームのテスト.....	8-9
8.4 次へ進む前に.....	8-10
第9章 Web アプリケーションの完成と実行.....	9-1
9.1 概要.....	9-1
9.2 準備.....	9-1
9.3 具体的な手順.....	9-2
9.3.1 COBOL プログラムの編集.....	9-2
9.3.2 アプリケーションのビルド.....	9-3
9.3.3 アプリケーションの実行.....	9-3
9.3.4 アプリケーションのデバッグ.....	9-4
9.4 次へ進む前に.....	9-5
第10章 COBOL アプリケーションから Web アプリケーションへの変換.....	10-1
10.1 概要.....	10-1
10.2 準備.....	10-2
10.3 具体的な手順.....	10-2
10.3.1 プロジェクト、フォーム、およびプログラムの作成.....	10-2
10.3.2 フィールド プロパティの追加.....	10-3
10.3.3 生成されたファイル.....	10-5
10.3.4 アプリケーションのビルド.....	10-6

10.3.5 アプリケーションの実行.....	10-6
10.4 次へ進む前に.....	10-7
第11章 Web データベース アプリケーションの作成 .....	11-1
11.1 概要 .....	11-1
11.1.1 用語.....	11-2
11.2 準備 .....	11-2
11.3 具体的な手順.....	11-2
11.3.1 プロジェクト、フォームおよびプログラムの作成.....	11-2
11.3.2 作成されたファイル.....	11-5
11.3.3 アプリケーションのビルド .....	11-6
11.3.4 アプリケーションの実行.....	11-6
11.3.5 データベースのナビゲート .....	11-6
11.3.6 データベースの更新.....	11-8
11.3.7 データベースのフィルタ処理 .....	11-8
11.4 次へ進む前に.....	11-10
第12章 Windows GUI アプリケーションの作成 .....	12-1
12.1 概要 .....	12-1
12.1.1 スクリーンセットの作成.....	12-1
12.2 準備 .....	12-2
12.3 具体的な手順.....	12-2
12.3.1 プロジェクトとスクリーンセットの作成 .....	12-2
12.3.2 この段階の画面表示.....	12-3
12.3.3 スクリーンセット用データの定義 .....	12-5
12.3.4 スクリーンセットへのコントロールの追加 .....	12-7
12.3.5 スクリーンセットへのダイアログの追加 .....	12-10

12.3.5.1 グローバル ダイアログ .....	12-10
12.3.5.2 ローカル ダイアログ .....	12-11
12.3.6 作成されたファイル .....	12-12
12.3.7 アプリケーションのビルド .....	12-13
12.3.8 スクリーンセットのテスト .....	12-13
12.4 次へ進む前に .....	12-15
第13章 Windows GUI アプリケーションの完成と実行 .....	13-1
13.1 概要 .....	13-1
13.1.1 関連付けられたプログラムの構造 .....	13-1
13.2 準備 .....	13-1
13.3 具体的な手順 .....	13-2
13.3.1 COBOL プログラムの編集 .....	13-2
13.3.2 アプリケーションのビルド .....	13-3
13.3.3 アプリケーションの実行 .....	13-3
13.4 次へ進む前に .....	13-3
第14章 UNIX でのアプリケーションの実装 .....	14-1
14.1 概要 .....	14-1
14.2 準備 .....	14-1
14.3 具体的な手順 .....	14-2
14.3.1 プロジェクトの作成 .....	14-2
14.3.2 アプリケーションのビルド .....	14-3
14.3.3 UNIX 端末としての PC の使用 .....	14-3
14.3.4 パブリッシュ オプションの設定 .....	14-4
14.3.5 アプリケーションのパブリッシュ .....	14-7
14.3.6 作成されたファイル .....	14-8

14.3.7 アプリケーションの実行.....	14-8
14.4 次へ進む前に.....	14-8
第15章 実行時ライセンス.....	15-1
15.1 概要.....	15-1
15.2 実行時ライセンスが必要な場合.....	15-2
15.3 Application Server ライセンスまたは Run Time System ライセンスが必要な場合.....	15-2
15.3.1 NetExpress Run Time System のコンポーネント.....	15-3
15.3.2 NetExpress Application Server と Web Application Server のコンポーネント.....	15-4
15.4 アプリケーションの実装、パッケージ化、出荷に関する注意.....	15-4
15.5 追加製品の実行時ライセンス.....	15-5
15.5.1 ODBC ドライバ.....	15-5
15.5.2 ProtoView データ テーブル グリッド コントロール.....	15-5
15.5.3 その他のサード パーティのコンポーネント.....	15-5
付録A: Windows のヒント.....	A-1
A.1 ウィザードの標準ボタン.....	A-1
A.2 フォルダの変更.....	A-1
A.3 フォルダの操作.....	A-2
A.4 ツリー ビュー.....	A-3
A.5 ファイル拡張子の表示または非表示.....	A-3
A.6 コンテナの列見出し.....	A-3
A.7 リスト項目の選択.....	A-4
A.8 ドロップダウン リスト.....	A-5
A.9 ポップアップ メニュー.....	A-5
A.10 切り取りと貼り付け.....	A-6
A.11 ドラッグ & ドロップ.....	A-6



A.12 ウィンドウのサイズ変更とドッキング .....	A-7
付録B: その他の機能 .....	B-1
B.1 機能説明 .....	B-1
B.1.1 機能リスト .....	B-1
B.1.1.1 Btrieve インターフェイス .....	B-1
B.1.1.2 呼び出し可能ファイル ハンドラ .....	B-1
B.1.1.3 呼び出し可能なリビルド .....	B-1
B.1.1.4 呼び出し可能な整列 .....	B-2
B.1.1.5 Cblink .....	B-2
B.1.1.6 クライアント サーバー結合 .....	B-2
B.1.1.7 COBOL システム ライブラリ ルーチン .....	B-2
B.1.1.8 コマンド プロンプト .....	B-2
B.1.1.9 Common Gateway Interface (CGI) サポート .....	B-2
B.1.1.10 埋め込み HTML .....	B-3
B.1.1.11 Fileshare .....	B-3
B.1.1.12 統合プリプロセッサ サポート .....	B-3
B.1.1.13 Microsoft Transaction Server サポート .....	B-3
B.1.1.14 マルチスレッド .....	B-3
B.1.1.15 各国語サポート .....	B-3
B.1.1.16 オブジェクト指向 COBOL 構文 .....	B-4
B.1.1.17 オブジェクト リクエスト ブローカ (ORB) .....	B-4
B.1.1.18 ODBC ドライバ .....	B-4
B.1.1.19 OLE オートメーション サポート .....	B-5
B.1.1.20 OpenESQL .....	B-5
B.1.1.21 OpenESQL アシスタント .....	B-5

B.1.1.22 PVCS .....	B-5
B.1.1.23 リビルド .....	B-5
B.1.1.24 整列ユーティリティ .....	B-6
B.1.1.25 ソース コード管理システム サポート.....	B-6
B.1.1.26 WebSync リンク .....	B-6
B.1.1.27 Win32 固有の API プログラミング .....	B-6
B.2 移行のための機能 .....	B-6
B.2.1 機能リスト .....	B-6
B.2.1.1 COBSQL .....	B-6
B.2.1.2 ダイアログ エディタ .....	B-7
B.2.1.3 オンライン ヘルプ システム.....	B-7
B.2.1.4 パネル .....	B-7
B.2.1.5 画面とキーボードのハンドラ (文字モード).....	B-7
B.2.1.6 画面とキーボードの構成ツール.....	B-8
B.2.1.7 スクリーン節 .....	B-8
B.2.1.8 ウィンドウ操作サポート .....	B-8
付録C: Web アプリケーション .....	C-1
付録D: Web サーバーの構成.....	D-1

# 第1章 概要

NetExpress をお買い上げいただき、ありがとうございます。NetExpress は、最も効率的な分散アプリケーション開発ツールです。また、NetExpress は、インターネット アプリケーションのための完全な統合開発環境です。Web フォームを作成し、それらを処理する完全な作業プログラムを生成することができます。その場合、ビジネス ロジックをプログラムに追加するだけで済みます。また、既存のアプリケーションから完全な Web アプリケーションを生成することもできます。

NetExpress には、Web アプリケーションの作成、開発、ビルドおよびテストを行うツールと機能がすべて揃っています。また、個人用 Web サーバーも組み込まれています。

NetExpress は、Windows ユーザー インターフェイスを持つ分散アプリケーションの開発にも使用できます。

また、NetExpress では、Microsoft 社の COM/DCOM や ActiveX、OMG の CORBA ベースの技術、Sun Microsystem 社の Java Bean のようなコンポーネント ベースの新技术を使用して、企業用コンポーネントをビルドすることができます。

このマニュアルでは、通常、Windows 95、Windows 98、および Windows NT 4.0 を省略して "Windows" と呼んでいます。

この『入門書』以外のマニュアルは、すべてオンライン マニュアルです。オンライン マニュアルは、Windows の [スタート] メニューか、NetExpress の [ヘルプ] メニューから参照できます。この『入門書』は、オンライン マニュアルと書籍の両方で参照できます。NetExpress のインストールに必要な説明は、インストール ユーティリティ Setup にすべて表示されています。

まず、NetExpress の主要部分についての概説から始めます。これらの内容を習得すると、NetExpress をすぐに使い始めることができます。「*NetExpress の使用方法*」の章から先は、すべて、次のような主要部分についてのチュートリアルです。

- 開発環境

コードの編集、コンパイルおよびデバッグが可能なプルダウン メニューを持つメイン ウィンドウです。このウィンドウを経由して、NetExpress のほとんどのツールにアクセスします。これを、統合開発環境 (IDE) と呼びます。

- プロジェクト

アプリケーションのすべてのファイルの詳細、コンパイル方法、リンク方法を説明するファイルです。プロジェクトは、簡単に作成でき、マウスを 1 回クリックするだけでアプリケーション全体をコンパイルしたり、リンクさせたりすることができます。ただし、どんなに簡単なアプリケーションでも、それぞれにプロジェクトを作成する必要があります。

- データ ツール

アプリケーションで使用するデータ ファイルの変換、参照、編集および作成を可能にするツールのセットです。データ ファイルを検証してアプリケーションによる更新内容を確認することができます。また、アプリケーションのテスト データを提供するファイルの作成や編集もできます。ファイルは、COBOL 形式で、レコードとフィールドの両方のレベルで表示できます。

- HTML ページ ウィザード

Web アプリケーションのユーザー インターフェイスを作成するためのツールです。Web フォームを指定すると、必要なファイルが生成されます。

- インターネット アプリケーション ウィザード

Web アプリケーションの作成に使用するツールです。HTML ページの作成と HTML ページを処理する COBOL コードの作成は、事実上すべてインターネット アプリケーション ウィザードが行います。アプリケーションは、次の 3 つの方法で作成できます。

- HTML ページ ウィザードで作成した HTML フォームから作成する方法
- 既存の COBOL プログラムの連絡節から作成する方法
- 既存のデータベースの構造体から作成する方法 (この方法は、ODBC ドライバを使用するデータベースに適用できます)

- Form Designer

Web を基本とするアプリケーションのフォーム設計に使用する WYSIWYG フォーム エディタです。

- Solo

ユーザーのコンピュータで実行する Web サーバー ソフトウェアです。Web アプリケーションのテストに使用します。

- Windows GUI アプリケーション ウィザード

Windows ベースのアプリケーションを作成するためのツールです。このツールを使用してスクリーンセットを作成します。このスクリーンセットには、ウィンドウとダイアログ ボックスの詳細が保存されます。また、このツールを使用して、ウィンドウやダイアログ ボックスを処理する COBOL プログラムも生成します。

- Dialog System

Windows ベースのアプリケーションで使用するウィンドウとダイアログ ボックスを設計するためのツールです。また、イベントを処理するプロシージャを書き込むこともできます。

- PVCS

一般的なソースコード管理システム (SCCS) です。このシステムは、中央のサーバーに保存されたソースコードに対する複数ユーザーからのアクセスを制御するものです。ファイル変更ごとのバージョンを再作成できるように、変更は差分形式で保存されます。NetExpress には、一般に使用される PVCS の機能のほとんどが含まれています。

- UNIX オプション

UNIX サーバーで NetExpress アプリケーションを実装するためのツールです。また、NetExpress を使用して完全な UNIX アプリケーションを開発するためのツールもあります。

## 第2章 補足的なソフトウェア

NetExpress のライセンスにより、NetExpress CD-ROM や付属 CD-ROM で提供される Micro Focus 以外のソフトウェアを使用することができます。

- Microsoft インターネット エクスプローラ

Microsoft インターネット エクスプローラは、一般的な Web ブラウザです。Form Designer では、フォームを設計するときにこのブラウザの機能を使用します。また、NetExpress のマニュアルを参照したり、Web アプリケーションをテストする場合にも Web ブラウザが必要です。そのため、Microsoft インターネット エクスプローラが必要になります。まだこの Web ブラウザがインストールされていない場合、または Form Designer に必要なバージョンがインストールされていない場合は、NetExpress のセットアップ時にインストールを勧めるメッセージが表示されます。

- Win32 ソフトウェア開発キット

Win32 ソフトウェア開発キット (SDK) は、Windows アプリケーションを作成する場合に便利なツールとシステム ライブラリのセットです。NetExpress CD-ROM に付属する Win32 SDK CD-ROM のルート ディレクトリにある setup.exe を使用してインストールします。

- Adobe Acrobat Reader

Adobe Acrobat Reader を使用すると、Adobe Acrobat がサポートする形式のオンライン マニュアルを表示することができます。NetExpress のオンライン マニュアルは、Web ページ形式と Adobe Acrobat 形式で作成されています。詳細は、「情報の検索」の章を参照してください。

- Server Control Program (SCP)

Server Control Program (SCP) は、UNIX で使用するサーバー ソフトウェアです。NetExpress の UNIX オプションを使用する場合は、これを UNIX マシンにインストールする必要があります。詳細は、オンライン マニュアル『UNIX オプション ユーザー ガイド』を参照してください。

- Samba

Samba を使用すると、PC から UNIX サーバーのドライブを参照できます。これは、UNIX マシンにインストールしてください。詳細は、オンライン マニュアル『UNIX オプション ユーザー ガイド』を参照してください。

## 第3章 情報の検索

この『入門書』以外のマニュアルは、すべてオンラインだけで提供されます。オンライン マニュアルは、Windows の [スタート] メニュー、または、NetExpress の [ヘルプ] メニューから参照することができます。この『入門書』は、オンライン マニュアルと書籍の両方で提供されます。


多くのマニュアルは、Win95 ヘルプの形式で作成されています。それ以外は、Web ページ形式のオンライン マニュアルです。これらのオンライン マニュアルは、Windows の [スタート] メニュー、または、NetExpress の [ヘルプ] メニューから参照します。ヘルプで本型アイコンをクリックすると、そのマニュアルが Web ブラウザに表示されます。

情報を検索する場合、次の順序でさまざまなマニュアルを参照することをお勧めします。

1. ヒント情報があるか確認する。

フィールドやコントロールに関する情報が必要な場合は、それらのフィールドやコントロールにマウス ポインタを合わせたまましばらく待ちます。すると、フィールドやコントロールの横や NetExpress ウィンドウの最下部に簡単な用途説明が表示されます。

2. コンテキスト ヘルプを使用する。

メニュー項目やコントロールの情報が必要な場合は、ツールバーで [コンテキスト ヘルプ] ボタン  をクリックしてから、メニュー項目やコントロールをクリックします。

3. ヘルプのキーワードを使用して検索する。

NetExpress の実行中は、[ヘルプ] メニューで [ヘルプ トピック] をクリックします。NetExpress を実行していない場合は、[スタート] [プログラム] [Micro Focus NetExpress] [NetExpress ヘルプ] の順にクリックします。開いたウィンドウで、[キーワード] タブをクリックし、検索する語句を入力します。カーソルは、検索条件に一致するエントリがあると自動的にそこへ移動します。また、検索する情報が見つかるまでキーワードをスクロールさせて参照することもできます。

[スタート] [プログラム] [Micro Focus NetExpress] [NetExpress オンライン マニュアル] の順にクリックします。総合索引 (オンライン マニュアル) を選択します。見たい本を選択して開き、[キーワード] タブをクリックして検索する語句を入力します。

4. まだ、必要な情報が見つからない場合は、その他の語句を使用して手順 3 から 手順 5 を繰り返します。

その他の情報は、次のとおりです。

- オンライン ヘルプは、クイック ヘルプとして使用します。特定の題材について深く知るには、まず該当するオンライン マニュアルを参照してください。オンライン マニュアルで情報が検索できない場合は、

ヘルプを参照してください。すべてのオンライン マニュアルへは、ヘルプの目次からアクセスすることができます。オンライン マニュアル、NetExpress ヘルプ、Readme ヘルプ、Dialog System のような個々のオプションのヘルプには、個別の索引があります。オンライン マニュアルの索引は、すべてのオンライン マニュアルを対象としているので総合索引と呼ばれます。オンライン マニュアルには、個別の索引もあります。

- オンライン マニュアルは、Web ブラウザから印刷することができます。Web ページ 1 ページ分が 1 つの章になっているので、ブラウザの印刷機能で 1 ページを印刷すると、1 章全体が印刷されることとなります。目次ではなく、テキストを確実に印刷するには、テキストが表示されているフレームをクリックしてから印刷してください。
- NetExpress の CD-ROM には、Adobe Acrobat Reader と各マニュアルの Adobe Acrobat ファイルが収録されています。これらを使用すると、マニュアル 1 冊分を 1 回の操作で印刷することができます。使用許諾契約上、個人使用を前提とする場合は各マニュアルにつき 1 冊のコピーを正当に印刷することができます。
- オンライン マニュアルのリンクは、他の章やマニュアル、オンライン用語集などにリンクします。前のページに戻るには、Web ブラウザの [戻る] ボタンを使用します。正しい目次に戻るために [戻る] ボタンを 2 回クリックする必要がある場合があります。
- この章を読んだ後で、特に 2 つのオンライン マニュアル『ソリューション ガイド』と『移行ガイド』を参照すると便利です。『ソリューション ガイド』では、頻繁に必要とされるアプリケーションの開発について解説されています。『移行ガイド』では、他の COBOL システムを NetExpress に移行する方法が説明されています。
- 項目に対応した機能を持つポップアップ メニューを表示するには、項目を右クリックします。この機能には、ヒント情報も含まれています。
- IDE の「テキスト」ウィンドウで COBOL 予約語に対する構文を表示するには、この予約語にカーソルを合わせ、[F1] キーを押します。または、この予約語を右クリックし、ポップアップ メニューで [ヘルプ] を選択します。
- NetExpress のインストールや実行に問題が発生した場合は、Readme ヘルプで、その問題がすでに把握されている問題であるかを確認します。Windowsで、[スタート] [プログラム] [Micro Focus NetExpress] [Readme] の順にクリックします。また、NetExpress ヘルプの目次からも Readme ヘルプにアクセスすることができます。Readme ヘルプの情報は、NetExpress ヘルプの索引ではなく、Readme ヘルプの索引に一覧表示されています。
- NetExpress に関するサポート、更新、情報には、WebSync を使用してアクセスします。WebSync は NetExpress の機能で、これを使用すると Micro Focus の Web にアクセスできます。WebSync を起動するには、[ヘルプ] メニューの [WebSync] をクリックします。
- Micro Focus 社の連絡先の所在地と電話番号については、Readme ヘルプを参照してください。



## 第4章 ここからチュートリアルをスタート

NetExpress のチュートリアル『入門書』へようこそ。

### 4.1 概要

ヘルプの「ここからスタート」という画面からこのマニュアルにアクセスしている場合は、前の章をスキップしているはずですが、この章は、第 1 部の最終章です。この章からは、チュートリアルで構成される他のパートに直接進むことができます。NetExpress の主要分野ごとにまとめられています。各分野について、いくつか章があり、各章がチュートリアルになっています。

(第 1 部の第 1 章から第 3 章までは、読んでおく必要があります。これらの章には、NetExpress の概要と、マニュアルで情報を検索するときの実用的なアドバイスが多数掲載されています。)

これらのチュートリアルは、COBOL と使用するオペレーティング システム (Windows 95、Windows 98、または Windows NT 4.0) に精通しており、NetExpress や他の Micro Focus ソフトウェアに関する知識を必要としている読者を対象としています。

必ず、次の項目を読んでから開始するようにしてください。

### 4.2 用語と表記規則

Windows	このマニュアルでは、通常、Windows 95、Windows 98、Windows NT 4.0 を総称して "Windows" と呼びます。
バージョン番号	例では、メニュー項目の本文のバージョン番号は省略されます。
マウス ボタン	メインのマウス ボタンは左のボタンです。マウスのボタンを逆に設定している場合は、マウス ボタンに関する説明で「左」と「右」をすべて逆に読み替えてください。
クリック	「クリック」とは、メイン (左側) のマウス ボタンを 1 回クリックすることをいいます。ダブルクリックまたは右クリックする必要がある場合は、そのように指示されます。
大文字小文字の区別	これらのチュートリアルでは、通常、表示されている大文字と小文字の組み合わせについて厳密に考える必要はありません。Windows では大文字と小文字を区別しません。大文字小文字の区別が重要になる場合はほとんどありませんが、その場合は、説明が添付されています。
ディレクトリとフォルダ	Windows では、ディレクトリとフォルダという用語は、同じように使用できます。

	ディレクトリを入力する場合、サブディレクトリを使用できます。同様に、フォルダを開くと、他のフォルダを格納できます。NetExpress のマニュアルでは、両方の用語を使用します。このマニュアルでは、ほとんどの場合、「フォルダ」という用語を使用します。
ドライブ	NetExpress でフォルダ名を書き込む場合、ドライブ名として <i>d:</i> を使用します。このドライブ名は、NetExpress がインストールされているドライブの名前に置換してください。

### 4.3 ヒント

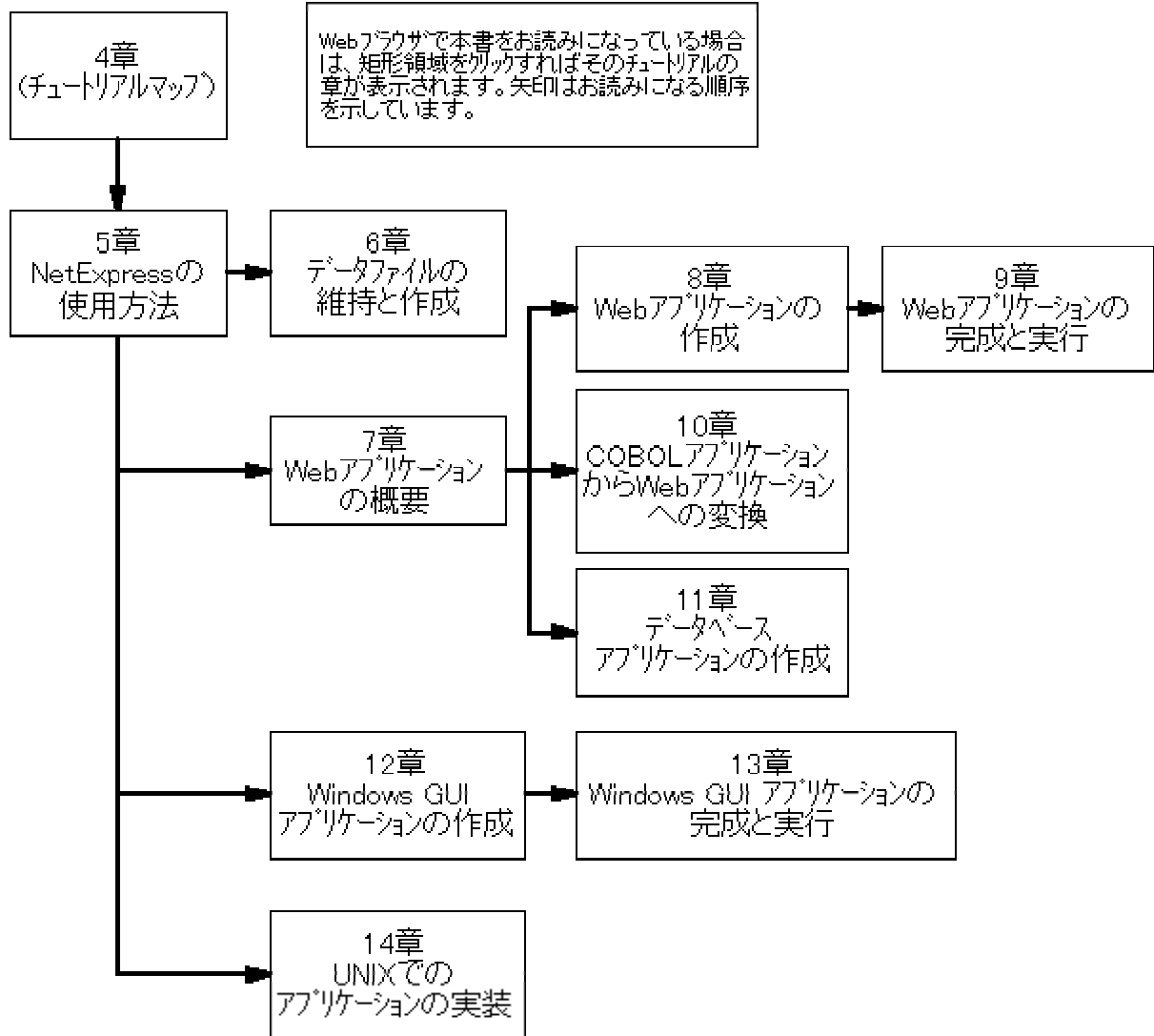
このマニュアルの使用方法	デモ操作中には、ウィンドウを数個開くことが多いため、画面が見にくくなる場合があります。このマニュアルを Web ブラウザでオンライン表示している場合は、他のウィンドウが上に重なることがよくあります。そのため、印刷されたマニュアルを使用することもできます。ただし、オンラインバージョンには、多くのリンクがあり、マニュアル中で高速に移動できます。ブラウザで各章を印刷し、ウィンドウを縮小または最小化して開いておき、リンクを使用する必要がある場合に拡大するのが最善の方法です。
リンク後に元のページへ戻る方法	別の章へのリンクをクリックした後で元に戻る場合は、ブラウザの [戻る] ボタンを 2 回クリックします。この操作により、本文と目次の両方で元のページに戻ることができます。
クロスリファレンス	チュートリアルでは、すぐに参照する必要があるクロスリファレンスにはリンクが作成されています。後で参照する箇所を示すクロスリファレンスには、リンクが使用されていません。
Windows の技法	頻繁に使用する Windows の技法については、付録「Windows のヒント」で簡単に説明しています。
プロジェクトの再読み込み	連続する章では、同じプロジェクトを使用する場合があります。各章の間で NetExpress を終了する場合は、次のようにプロジェクトの再読み込みを行うと便利です。  NetExpress が起動すると前回 NetExpress のメイン ウィンドウを閉じたときに開いていたプロジェクトがロードされるオプションがあります。このオプションがオンになっていることを確認するには、[オプション] メニューの [カスタマイズ] をクリックします。次に、[ワークスペース] タブをクリックして「起動時にファイルと前回のプロジェクトを再読み込みする」にチェック マークが付いていることを確認

	します。[OK] をクリックしてダイアログボックスを閉じます。  他には、[開く] または [最近のプロジェクト] を使用して既存のプロジェクトを開く方法があります。これらは、ともに [ファイル] メニューにあります。
--	---

## 4.4 チュートリアル マップ

各デモ操作は、1 つの章になっています。このマニュアルのオンライン バージョンでは、下のボックスをクリックして別の章に進みます。1 番目のデモ操作を最初に実行することが重要です。それ以降のデモ操作の相互関係は、次のツリー図のとおりです。2 つのチュートリアルが矢印で結合されている場合は、1 番目のデモ操作を実行してから、2 番目のデモ操作に進みます。

『入門書』のこれらのチュートリアルは、最も一般的な作業を通して、各分野について短時間で紹介するように作成されています。これらの分野すべてに関する詳細は、「情報の検索」の章で説明するとおり、ヘルプとオンライン マニュアルを参照してください。すべてのヘルプとオンライン マニュアルには、Windows の [スタート] メニューからアクセスできます。



# 第5章 NetExpress の使用方法

統合開発環境 (IDE) では、すべてのアプリケーションのコンパイル、編集およびデバッグを行います。

---

これは、『入門書』で説明するチュートリアル最初のデモ操作です。他のデモ操作を行う前にこのデモ操作を終了する必要があります。まず、「ここからチュートリアルをスタート」の章を読み終えてから、このデモ操作を行ってください。

---

## 5.1 概要

IDE は、COBOL アプリケーションの編集、コンパイルおよびデバッグ (NetExpress では「アニメート」と呼びます) に必要なすべてのツールを統合したものです。IDE には、ツールと COBOL 言語に関する膨大なオンライン マニュアルも含まれています。

IDE を最も簡単に学習する方法は、簡単な作業をいくつか実際に行うことです。このデモ操作では、プロジェクトの読み込み、コンパイル、実行、およびアニメート (デバッグ) について作業します。つまり、ソース コードを最後までステップ実行していきます。

### 5.1.1 プロジェクトとプロジェクト フォルダ

プロジェクトは、アプリケーションに含まれるすべてのファイルの詳細情報とそのコンパイル方法を説明するファイルです。プロジェクトは、簡単に作成できます。プロジェクトを使用すると、簡単かつ高速にコンパイルすることができます。どんなに簡単なアプリケーションでも、アプリケーションごとにプロジェクトを作成する必要があります。ディスクでは、プロジェクトは名前の末尾に付く拡張子 `.app` で識別されます。ただし、NetExpress の多くの機能と同様、プロジェクトは、すべて IDE を使用して作成し、維持するため、プロジェクトを直接参照する必要はありません。

アプリケーションのプロジェクトを保持するフォルダは、プロジェクト フォルダと呼ばれます。プロジェクトにはファイルに対するポインタが記述されているので、アプリケーションの他のファイルを保存する場所は指定されていません。通常は、すべてプロジェクト フォルダに保存すると便利です。

NetExpress をインストールする場合、セットアップ作業により NetExpress が格納されているシステム フォルダ内にフォルダが作成されます。このフォルダのデフォルト名は、`d:\NetExpress\Base\Workarea` です。この作業領域用フォルダにすべてのプロジェクト フォルダを格納することをお勧めします。

## 5.1.2 デモンストレーション アプリケーション

NetExpress には、多くのデモンストレーション アプリケーションが含まれています。この中には、この『入門書』のチュートリアルで使用されるものもあれば、オンライン ヘルプからアクセスする高度なチュートリアルで使用されるものもあります。これらのアプリケーションは、すべて `d:\NetExpress\Base\Demo` 内のフォルダに格納されています。

インストール時に、フォルダ NetExpress に別の名前を指定している場合は、デモ操作では NetExpress の代わりにその名前を使用してください。

チュートリアルの途中まで進み、最初からやり直すときは、NetExpress ヘルプの完全なチュートリアル リストにあるデモの [初期化] ボタンをクリックすると、デモ ファイルを初期状態に戻すことができます。この場合、NetExpress をインストールした CD-ROM またはネットワークへの接続が必要です。このリストを参照するには、[スタート] [プログラム] [Micro Focus NetExpress] [NetExpress ヘルプ] の順にクリックします。次に、ヘルプの [目次] で [ここからスタート] [チュートリアル] をダブルクリックし、必要な項目を選択します。

この初期化によって、デモで作成したファイルやフォルダが削除されることはありませんが、供給されたファイルの元のバージョンが再インストールされます。

このデモ操作で使用するデモ アプリケーションは、「Locking」です。このアプリケーションにはプロジェクト ファイルが含まれているので、このデモ操作では、プロジェクトを読み込んでアプリケーションを実行します。


## 5.2 具体的な手順

この項では、次の作業を行います。

- IDE の起動
- プロジェクトの読み込み
- プロジェクトのビルド
- COBOL コードの実行
- COBOL コードのアニメート
- IDE オプションの設定
- アニメートの終了

### 5.2.1 IDE の起動

インストールの終わりで [実行] ボタンをクリックして IDE を起動した場合は、インストール ユーティリティのメイン画面が表示されたままになります。この画面には、[インストール]、[インストール上の注意]、および [終了] の

各ボタンが付いています。ただし、この画面は IDE の後ろに隠れていることがあります。この画面は、 ボタンをクリックすると、いつでも閉じることができます。この画面が表示されていても、このデモ操作に影響はありません。

上記の場合以外は、次のように IDE を起動します (これは、今後 IDE を起動するときに必ず使用する方法です)。

1. Windows の [スタート] メニューで [プログラム] をクリックし、[Micro Focus NetExpress]、[NetExpress] の順に選択します。

「Micro Focus 保護システム」というタイトルのダイアログ ボックスが表示され、ライセンスの有効期限が数日後に切れることを警告することがあります。このデモ操作では、この警告は無視します。ただし、後で NetExpress を再起動し、この画面の [ヘルプ] ボタンをクリックして、完全なライセンスの取得方法について詳細を確認する必要があります。

2. ライセンスの警告が表示されたら、[OK] をクリックして無視します。

最初に IDE を読み込むと、「IDE」ウィンドウとともに「Welcome」画面が表示されます。この画面には、毎回この画面を表示するかどうかを設定できるチェックボックスがあります。この設定によって、ヘルプ ファイルを見るか、そのまま NetExpress に進むかを選択できます。

3. 「Welcome」画面で [続ける] をクリックします。


「Welcome」画面が閉じます。これで IDE が画面に表示されます。IDE は、 5-1 のような画面です。



図 5-1 統合開発環境 (IDE)

大きなペインには、「プロジェクト」ウィンドウや「編集」ウィンドウなどの各種ウィンドウが開きます。その下のペインは「出力」ウィンドウで、ここに IDE やコンパイラからのメッセージが表示されます。このペインにはタブがいくつかありますが、最も使用頻度が高いのは、[ビルド] タブです。[ビルド] タブが強調表示されている（白くなっている）ときは、このウィンドウにビルドの進捗状況を示すメッセージが表示されています。

IDE の端や角をドラッグすると、IDE のサイズや形を変更できます。ペインの中には、切り離して IDE 内の別の場所や、画面で IDE とは別の場所に配置できるものがあります。この操作をドッキングまたはアンドッキングといいます。この操作を行う場合は、付録 Windows のヒント」を参照してください。

このマニュアルの手順を実行する場合、以前に同じコンピュータで本製品が使用されている場合は、標準の位置からペインが移動されている可能性があることに注意してください。


## 5.2.2 プロジェクトの読み込み

プロジェクト「Locking」を読み込むには、次の手順にしたがいます。

1. 「ファイル」メニューの [開く] をクリックします。



これによって、「開く」ダイアログ ボックスが開きます。インストール後、初めて NetExpress を起動する場合は、`d:\NetExpress\Base\Demo` に格納されたファイルやフォルダが表示されます。表示されない場合は、このディレクトリに変更します。

- サブディレクトリ Locking に移動し、プロジェクト locking.app を選択してから [開く] をクリックします。(通常、プロジェクト ファイルには、拡張子 .app とアイコン が表示されます。Windows の設定により、拡張子が表示されていない場合は、アイコンでファイルを識別できます。)

IDE によって「プロジェクト」ウィンドウが開き、プロジェクトのファイルが表示されます。左側のペインは、ファイルの従属関係を示すツリー ビューです。右側のペインは、ファイルのリストです。ウィンドウの境界はドラッグできます。たとえば、ファイルをより詳細に参照するために、右側のペインを拡大することができます。


- 右側のペインを右クリックして、ポップアップ メニューを表示します。このメニューの「ソース ファイルだけを表示」がチェックされていないことを確認してください。チェックされていたら、それをクリックして、このペインに、コンパイル可能なファイルだけでなく、すべてのファイルが表示されるようにします。

.cbl ファイルは COBOL ソース ファイルです。

- .cbl ファイルをダブルクリックします。

この操作によって、「テキスト」ウィンドウが開き、ソース コードが表示されます。

- [ファイル] メニューの [閉じる] をクリックします。

これによって、「プロジェクト」ウィンドウが開いたまま、「テキスト」ウィンドウが閉じます。または、右上の [閉じる] アイコン をクリックして、ウィンドウを閉じることもできます。

.int ファイルは、中間コード形式 (Micro Focus の実行可能ファイル形式) の実行可能ファイルです。

IDE では、業界標準の .exe ファイルおよび .dll ファイルを作成することもできます。ただし、.int には、リンクが不要であるという利点があります。このため、新規アプリケーションをデバッグするときに、個別のプログラムの再コンパイルが高速化されます。


プロジェクトに表示するファイルはまだ必要ありません。プロジェクトは、ビルドを指示する Makefile やバッチ ファイルに似ています。まだプロジェクトをビルドしていないので、.int ファイルはありません。

### 5.2.3 プロジェクトのビルド

プロジェクトのビルドは、ファイルを実行可能ファイル形式にコンパイルすることを意味します。作成されたファイルの形式は、ビルド タイプによって、また、プロジェクトの種類に応じて選択されたオプションによって決まります。各プロジェクトには、標準的ビルド タイプであるデバッグやリリースを適用できます。また、独自のビルド タ


イプも作成できます。

プロジェクトをビルドするには、次の手順にしたがいます。

1. [プロジェクト] メニューの [リビルド] をクリックするか、ツールバーの  をクリックします。

この操作によって、前回プロジェクトがビルドされた後に変更されたファイルがすべてリビルドされます。これまでプロジェクトをビルドしていなかったため、すべてのファイルがリビルドされます。(同じメニューの [すべてをリビルド] 機能を使用すると、必要性に関係なく、プロジェクトに含まれるすべてのファイルがリビルドされます。)

この 1 つの機能ですべてのビルドを実行できます。各ソース ファイルに対して正しいコンパイラまたはトランスレータが自動的に呼び出されます。NetExpress では、「コンパイル」という用語は、通常、すべてのコンパイル、翻訳、変換またはプリプロセッシングに対して使用されます。また、「ソース ファイル」という用語は、そのようなコンパイルに入力するすべてのファイルに対して使用されます。

ツールバーの  ボタンに注目してください。[プロジェクト] メニューの [リビルド] にも、同じマークが表示されます。メニューで頻繁に使用される機能の多くには、ツールバーに同じボタンが表示されるようになっています。ボタンを使用するとより速く作業できます。デモ操作では、メニューを使用することもあれば、ツールバーを使用することもあります。マウス ポインタを少しの間ツールバーのボタンに置いたままにすると、ボタンの簡単な説明が表示されます。このような説明は、ツール情報といえます。

「出力」ウィンドウには、ビルドの進捗状況に関するメッセージが表示されます。「リビルド完了」というメッセージが表示されると、ビルドは終了しています。


## 5.2.4 アプリケーションの実行

実行とデバッグには [アニメート] メニューを使用します。デバッグしないでアプリケーションを実行するには、次のようにします。

1. [アニメート] メニューの [実行] をクリックします。

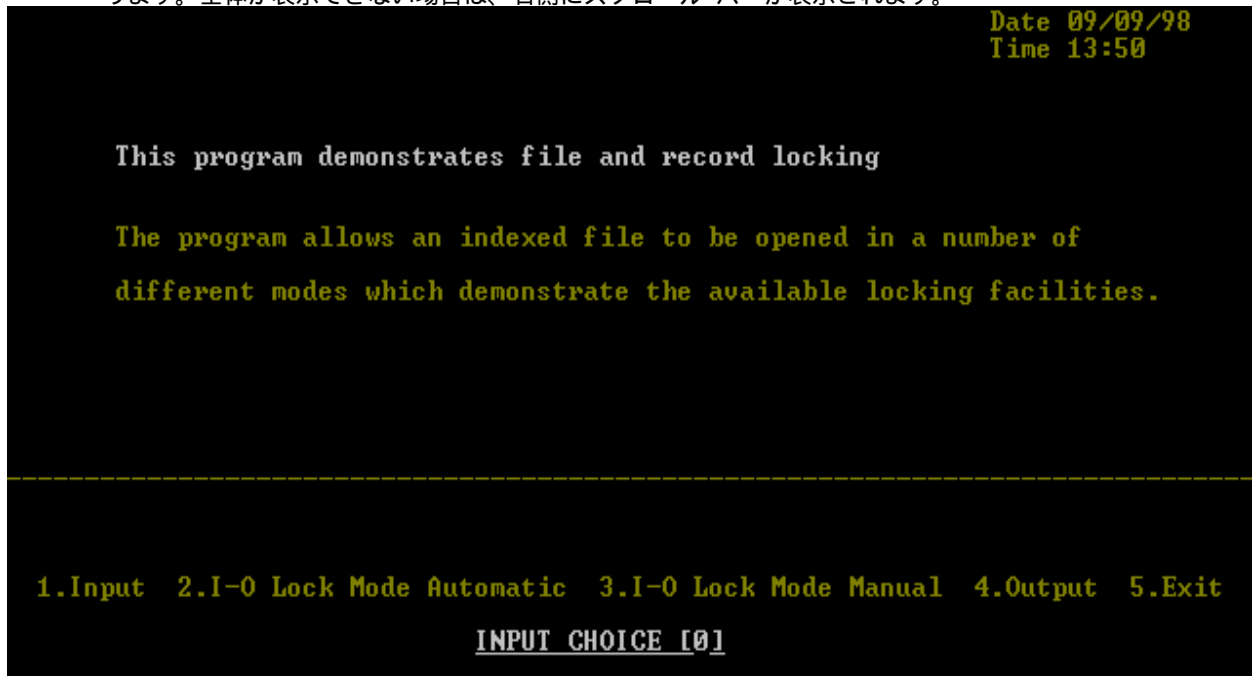
「アニメーションの起動」ダイアログ ボックスが表示されます。このダイアログ ボックスを使用して、実行開始位置を指定できます。

このダイアログ ボックスは、実行とデバッグの両方で表示されます。IDE は、実行とデバッグの基礎に同じ機構を使用します。IDE 内でアプリケーションを実行する場合、実際には、デバッグ機能の切り替えを解除して、デバッグ エンジンを使用します。そのため、実行は、[アニメート] メニューを経由して制御され、デバッグと同様に、通常、実行についても進捗状況が表示されます。

ツールバーの  ボタンは、[デバッグ] メニューの [実行] と同じです。

2. [OK] をクリックして、Locking プログラムの開始を確認します。

これは、文字列ベースのアプリケーションなので、新しいウィンドウが IDE の下に表示されます。このウィンドウは、文字列を表示するアプリケーションに対して必ず表示されます。これは、「アプリケーション出力」ウィンドウ、または、文字列ウィンドウといいます。このウィンドウは、図 5-2 に表示されます。全体を表示できるように、IDE 内の境界線をドラッグしてこのウィンドウを拡大する必要がある場合があります。全体が表示できない場合は、右側にスクロールバーが表示されます。



```
Date 09/09/98
Time 13:50

This program demonstrates file and record locking

The program allows an indexed file to be opened in a number of
different modes which demonstrate the available locking facilities.

-----

1.Input  2.I-O Lock Mode Automatic  3.I-O Lock Mode Manual  4.Output  5.Exit
      INPUT CHOICE [0]
```

図 5-2 「アプリケーション出力」ウィンドウ

この手順では、アプリケーションのビルド方法と実行方法を説明するだけなので、アプリケーションの詳細な使用方法については触れません。

3. 「アプリケーション出力」ウィンドウをクリックして選択し、「5」と入力してから、[Enter] キーを押します。

「5」は、このアプリケーションを終了するコードです。「アプリケーション出力」ウィンドウの出力内容はそのまま残るので、最後に表示されたアプリケーションを参照できます。

[アニメート] メニューをクリックすると、アプリケーションが終了したかどうかを確認できます。アプリケーションの実行中に灰色表示される [アニメート開始] および [実行] が灰色表示されていない場合は、アプリケーションが終了しています。画面で別の場所をクリックしてメニューを再び閉じます。

(この Locking アプリケーションは、オンライン ヘルプからアクセスする高度なチュートリアルでも使用されます。そのチュートリアルでは、別のファイル ロック方法について説明しています。)

4. 「アプリケーション出力」ウィンドウの任意の場所を右クリックし、ポップアップ メニューの [非表示] を選択します。

この操作により、「アプリケーション出力」ウィンドウが閉じます。

## 5.2.5 アプリケーションのデバッグ


IDE では、わかりやすいグラフィックを使用してコードの実行を追跡することができます。また、IDE では、さまざまなデバッグ機能 (アニメート) を使用できます。これは、アプリケーションが期待どおりに機能しない場合、または、不慣れたアプリケーションに精通する必要がある場合に便利です。この項では、これらのデバッグ機能をいくつか理解するために、アプリケーションをアニメートします。

1. [アニメート] メニューの [アニメート開始] をクリックします。

「アニメーションの起動」ダイアログ ボックスが表示されます。

2. [OK] をクリックして、Locking プログラムのデバッグを開始することを確認します。

IDE によって「テキスト」ウィンドウが開き、locking.cbl のソース コードが表示されます。最初の文が強調表示された場合、実行準備が整ったことを示します。

3. [アニメート] メニューの [ステップ実行] をクリックするか、ツールバーの  をクリックします。

[ステップ実行] によって、次の文が実行されます。この文により、テキスト コンソールに空白の領域が表示され、このアプリケーション用の「アプリケーション出力」ウィンドウが開きます。「アプリケーション出力」ウィンドウ用を表示するためにウィンドウを移動することもできます。また、ソース コードが表示されるように、IDE の境界をドラッグする必要がある場合もあります。

4. 次の文もステップ実行します (DISPLAY LOCKING01-00)。

5. [アニメート] メニューの [指定範囲を実行] をクリックします。

[指定範囲を実行] によって、PERFORM 文または CALL 文に関するすべてのコードを 1 回の手順で実行することができます。

6. EVALUATE CHOICE 文 (段落 RE-ENTER-CHOICE の 2 番目の文) を右クリックし、ポップアップ メニューの [カーソルまで実行] を選択します。(「テキスト」ウィンドウを拡大して、より多くのソースを表示する必要がある場合があります。)


[カーソル位置まで実行] を選択すると、ブレークポイントを設定しなくても、特定の文を高速に実行できます。この段階では、まだ EVALUATE 文の実行段階には達していません。実行したコードの一部が ACCEPT 文なので、アプリケーションはその位置で一時停止して入力を待っています。

7. 「アプリケーション出力」ウィンドウをクリックして選択し、「5」と入力してから、[Enter] キーを押します。

EVALUATE CHOICE 文が強調表示され、実行準備が整ったことを示します。

8. データ項目 CHOICE をダブルクリックして、その値を確認します。

この操作で「確認リスト」が開き、値が 5 であることがわかります。このダイアログ ボックスでは、データ項目に対する個別のモニタ ウィンドウの作成、値の変更、または「ウォッチ リスト」(画面の下部に開くウィンドウ) への追加ができます。

9. 「確認リスト」の右上にある  ボタンをクリックします。

## 5.2.6 IDE オプションの設定

IDE の動作を設定する方法について、いくつか例をあげます。

1. [オプション] メニューの [アニメート] をクリックします。
2. ダイアログ ボックスで、「データ確認用ツール情報を表示」チェックボックスにチェックマークが付いていることを確認し (付いていない場合はクリックします)、[OK] をクリックします。
3. マウス ポインタをクリックしないでデータ項目 CHOICE の上に置いたままにして、しばらく待ちます。

このオプション設定により、データ項目の上にマウス ポインタを置くと、項目の値が表示されます。この方法は、「確認リスト」を使用するより簡単です。ただし、このようにデータ項目にマウス ポインタを置くたびに表示されるのがわずらわしい場合もあります。このオプションを設定したままにするかどうかは、判断してください。

4. マウス ポインタをソース コードの行の上に置き、ボタンを押したまま、行に沿って少しドラッグします。

すると、その行全体が選択されます。この方法は、編集でよく使用します。ただし、必ずしも行全体を選択する必要がないこともあります。


5. 行の選択を解除するには、ソース コードのどこかをクリックします。
6. [オプション] メニューの [編集] をクリックします。
7. ダイアログ ボックスで、[ブロック/クリップボード] タブをクリックしてから、「プリフィクス領域以外でマウスをドラッグして列範囲を選択」オプション ボタンが選択されていることを確認し (選択されていない場合は、クリックします)、[OK] をクリックします。
8. マウス ポインタをソースの任意の行に置き、マウス ボタンを押したまま、行に沿って少しドラッグします。

行でマウスをドラッグした部分だけが選択されます。この場合も、このオプションを設定するかどうかは判断してください。

設定できるオプションは他にも多数あります。これらのオプションについては、IDE の使用方法をよく学




んでから実行することをお勧めします。

## 5.2.7 アニメートの終了


1.  をクリックして、アプリケーションの残りを実行します。  
アプリケーションの末尾にある STOP RUN 文で実行が停止します。
2. 「Stop Run 文に達しました」というメッセージに対して [OK] をクリックし、[アニメート] メニューの [アニメート停止] を選択します。


## 5.2.8 コンテキスト ヘルプの使用方法


NetExpress の使用中にヘルプを参照する方法について説明します。

1. ツールバーの  をクリックしてから、「プロジェクト」ウィンドウの任意の場所をクリックします。  
  
ポップアップ ウィンドウが表示され、「プロジェクト」ウィンドウの目的を説明します。通常、 または  という [コンテキスト ヘルプ] ボタンが表示される場合は、このような方法で使用すると、簡単に状況に応じたヘルプを参照できます。多くの画面には、メイン ヘルプに進める [ヘルプ] メニューや [ヘルプ] ボタンも付いています。
2. ポップアップを非表示にするには、任意のキーを押すか、任意の場所でマウスをクリックします。

## 5.3 次へ進む前に

プロジェクトをクリックして選択し、[ファイル] メニューの [閉じる] をクリックするか、 ボタンをクリックして、プロジェクトを閉じます。「プロジェクト」ウィンドウを閉じると、それに従属するすべてのウィンドウが閉じます。この手順では、ソースを表示している「テキスト」ウィンドウが閉じます。

「アプリケーション出力」ウィンドウと「出力」ウィンドウのような標準ウィンドウは、[ファイル] メニューの [閉じる] で閉じることはできません。また、これらのウィンドウには、 ボタンもありません (アンドッキングしている場合はこの限りではありません)。かわりに、「アプリケーション出力」ウィンドウを右クリックして、ポップアップ メニューの [非表示] をクリックします。または、[表示] メニューの [ドッキングできるウィンドウ] をクリックし、ダイアログ ボックスの「アプリケーション出力」をクリックしてチェックマークを削除します。[閉じる] をクリックしてダイアログ ボックスを閉じます。これらのウィンドウは、閉じないで非表示にしておくと、再びウィンドウを開いたときに、前の表示内容をそのまま表示することができます。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままでかまいません。それ以外の場合は、[ファイル] メニューの [終了] をクリックするか、IDE の  ボタンをクリックします。

「ここからチュートリアルをスタート」の章の チュートリアル マップに戻り、目的に沿って次のデモ操作を選択してください。

# 第6章 データ ファイルの維持と作成

NetExpress のデータ ツールを使用して、アプリケーションで使用するデータ ファイルの変換、参照、編集、作成を行います。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」の作業を終了してから、この章に進んでください。

---

## 6.1 概要

データ ツールを使用すると、データ ファイルを確認し、アプリケーションによってそれらのファイルがどのように更新されたかがわかります。また、ファイルを作成、編集して、アプリケーションのテスト データを作成することもできます。ファイルは、COBOL 形式で、レコード ビューとフィールド ビューの両方で表示できます。形式と文字セットの変換ができます。

このデモ操作で使用されるデモ アプリケーションは、データ ファイルとそれを維持する COBOL ソース プログラムで構成されています。データ ファイルは可変長順ファイルで、レコードは従業員、管理者および役員の 3 種類の社員に関する詳細情報です。このデモ操作では、このファイルを可変長索引順ファイルに変換します。また、データ ファイル エディタを使用して、書式設定する方法や書式設定しない方法などさまざまな方法で表示します。

## 6.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。「プロジェクト」ウィンドウや他のウィンドウが開いている場合は、閉じます。

プロジェクトを作成する手間を省くため、NetExpress には、1 つのプロジェクトが用意されています。[ファイル] メニューの [開く] をクリックして、`d:\NetExpress\%Base%\Demo\%Dtoldemo%\dtoldemo.app` を開きます。「プロジェクト」ウィンドウに `.cbl` ファイルと `.dat` ファイルが表示されます。

## 6.3 具体的な手順

この手順では、次の操作を行います。

- ファイルの変換
- 書式設定されていないデータ ファイルの表示
- レコード レイアウト ファイルの作成

- デフォルトのレコード レイアウトの作成
- 条件付きレコード レイアウトの作成
- レコード レイアウト ファイルの保存
- 書式設定されたデータ ファイルの表示
- ファイルの 16 進表示
- データ ファイルの印刷
- 複数のデータ ファイルの編集
- 新しいデータ ファイルの作成

### 6.3.1 ファイルの変換

用意された順ファイルを索引順ファイルとしてコピーするには、次の手順にしたがいます。

1. [ツール] メニューの [データ ツール] をクリックし、[変換] を選択します。

入力ファイルの詳細を書き込むには、[参照] ボタンをクリックし、[開く] ダイアログ ボックスでファイル `datavseq.dat` を選択します (ダイアログ ボックスは、右のフォルダ `d:\¥NetExpress¥Base¥Demo¥Dtoldemo` で開く必要があります)。入力ファイルと出力ファイルの詳細は、`datavseq.dat` のファイル ヘッダーから読み込まれた情報に初期化されます。

2. ダイアログ ボックスの下部に、次の出力ファイル情報を入力します (残りのフィールドは変更しません)。

ファイル名	<code>d:\¥NetExpress¥Base¥Demo¥Dtoldemo¥staff.dat</code>
形式	Micro Focus
編成	索引順

3. [キーの定義] をクリックしてから、[キーの挿入] をクリックします。

キーとして使用するフィールドはレコードの始めにあり、長さは 7 バイトです。

4. 「キーのオフセット」を 0 に、「Key Lengthキーの長さ」を 7 にそれぞれ設定し、[OK] をクリックします。
5. [変換] をクリックします。



6. データ ファイルの変換が終了したことを示すメッセージが表示されたら、[OK] をクリックします。次に、変換結果を報告するメッセージが表示されたら、[OK] をクリックします。

「データ ファイルの変換」ダイアログ ボックスが再表示され、さらにファイル変換を行うことができます。

7. [キャンセル] をクリックします。
8. 「プロジェクト」ウィンドウを参照します。NetExpress によって、プロジェクトに新しいファイル staff.dat が自動的に追加されています。

### 6.3.2 書式設定されていないデータ ファイルの表示

作成した上記のファイルを表示するには、次の手順にしたがいます。

1. 「プロジェクト」ウィンドウから、ファイル staff.dat をダブルクリックします。

索引ファイルと相対ファイルの編集内容がすぐに適用されることを示すメッセージ ボックスが表示されます。

2. 次回からこのメッセージが再表示されないようにするには、チェックボックスをクリックして、「次回からこのメッセージを表示しない。」にチェック マークを付けます。
3. [OK] をクリックします。

「データ ファイル エディタ」ウィンドウが表示され、図 6-1 のように、staff.dat の内容が表示されます。

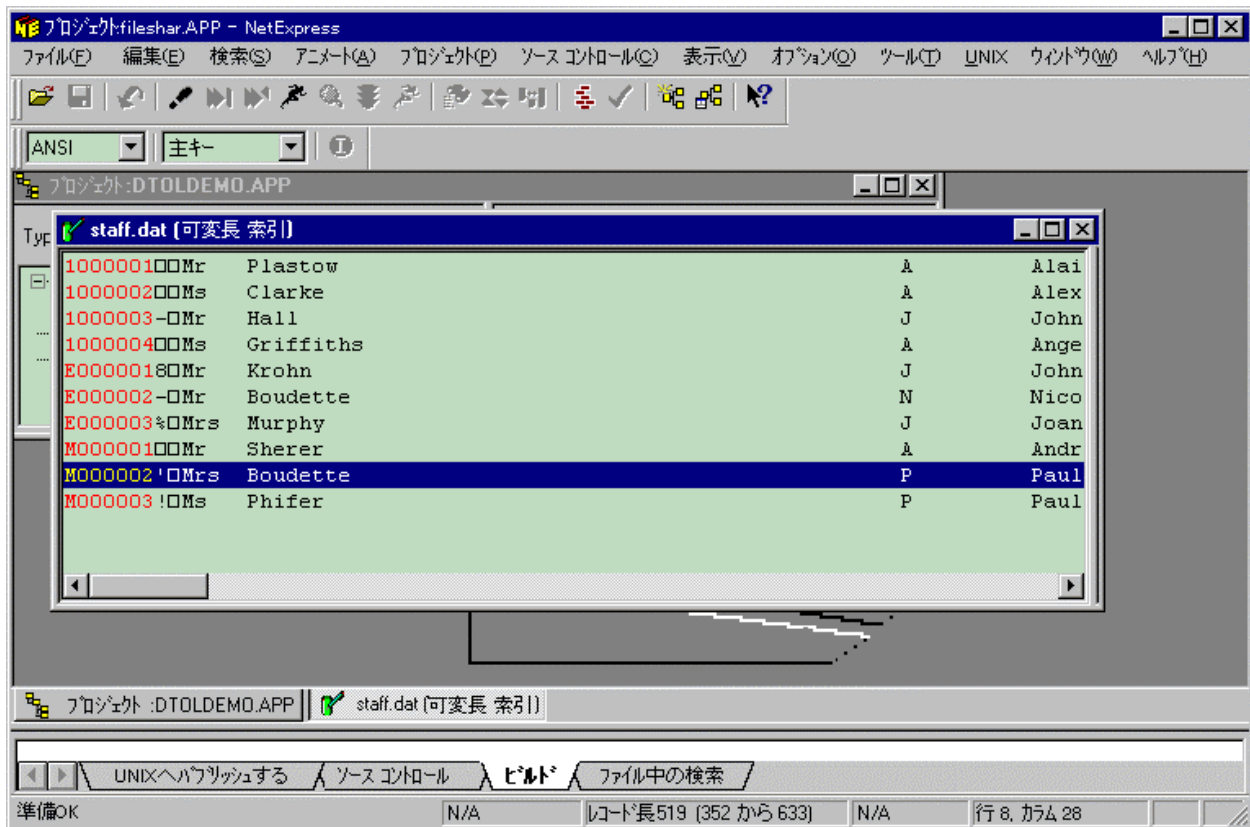


図 6-1 「データ ファイル エディタ」ウィンドウが表示された IDE

各レコードは 1 行で表示されます。データ ファイル エディタは、レコードのフィールド長を認識しないため、各行は単なる連続したテキスト行になります。レコードは書式設定されていません。COMP 項目のような文字列以外のフィールドでは、通常、標準以外の文字で表示されるか、印刷可能な文字に対応していない場合は、□として表示されます。

既存のデータに上書きで入力してファイルを編集することができます。使用可能な編集機能は多数あります。たとえば、[検索] メニューの「データ ツール」オプションには、次の 3 つのデータ ファイル検索方法があります。

- データ ファイルの検索と置換
- 索引キーの検索
- レコードへの移動

IDE の下のステータス行には、ファイルの詳細情報に応じて次のようなさまざまな統計が表示されます。

- 相対レコード番号 (相対ファイルのみ)
- 選択されたレコードの長さ。可変長の場合、後に (最小長、最大長) が続きます。
- レコード数 (順ファイルのみ)

- カーソル位置 - とともに 0 で始まる行番号と列位置

「データ ファイル エディタ」ウィンドウでは、右側のペインを表示したときに、「このウィンドウを使用するには、先にレコード レイアウト ファイルを読み込む必要があります」というメッセージが表示されることがあります。この用途については、後の項で説明します。

ファイルのプロパティを完全に表示するには、次の手順にしたがいます。

1. 「データ ファイル エディタ」ウィンドウを右クリックします。
2. ポップアップ メニューの「ファイル情報」をクリックします。  
  
「ファイル情報」 ウィンドウには、ファイルの形式や編成などの情報が表示されます。
3. ファイルの詳細を表示するには、[キー] および [一般] タブをクリックします。
4. 終了したら、[キャンセル] をクリックして「ファイル情報」ウィンドウを閉じます。
5. 「データ ファイル エディタ」ウィンドウは、このチュートリアルの後半で使用するので開いたままにします。

### 6.3.3 レコード レイアウト ファイルの作成

データ ファイル エディタにより、フィールドのレイアウトが認識される場合は、書式設定されたファイル内容が表示されます。このファイルを使用する COBOL プログラムのデータ部から、レコード レイアウト ファイルを作成します。必要な情報を作成するために、プログラムをコンパイルしておく必要があります。

ここで使用するデータ ファイル staff.dat には、従業員、管理者、役員の 3 種類のレコードがあります。データ ファイル エディタにレコードの種類ごとのレイアウトと種類の識別方法を認識させる必要があります。1 種類をデフォルト レイアウトとして定義し、残りを条件付きレイアウトとして定義します。

レコード レイアウト ファイルは、次のように作成します。

1. 「プロジェクト」ウィンドウを選択します。
2. 「プロジェクト」ウィンドウの左側のペインの dfdstaff.cbl を右クリックしてから、ポップアップ メニューの [コンパイル] をクリックします。
3. 再び dfdstaff.cbl を右クリックしてから、ポップアップ メニューの [レコード レイアウトの生成] をクリックします。

「レコード レイアウト エディタ」ウィンドウが開きます。このウィンドウには、dfdstaff.cbl に記述されたデータ部のツリー ビューが表示されます。

### 6.3.4 デフォルトのレコード レイアウトの作成

デフォルトのレコード レイアウトは、次のように作成します。

1. "+" をクリックして、FD MF-FILE エントリを展開します。
2. 01 EMPLOYEE-REC を右クリックしてから、ポップアップ メニューの [レイアウトの新規作成] をクリックします。

すでにデフォルトのレイアウトを定義しているため、[省略時のレイアウト] は、自動的に設定されているはずですが。

3. 未設定の場合は、[省略時のレイアウト] をクリックして、EMPLOYEE-REC をデフォルトのレコード レイアウトに設定します。
4. [次へ] をクリックしてから、[終了] をクリックします。

EMPLOYEE-REC-DEFAULT フォルダが右側のペインに表示されます。

### 6.3.5 条件付きレコード レイアウトの作成

条件付きレコード レイアウトは、次のように作成します。

1. 01 MANAGER-REC を右クリックしてから、ポップアップ メニューの [レイアウトの新規作成] をクリックします。

すでにデフォルトのレイアウトを定義しているため、[条件付きレイアウト] は自動的に設定されます。

2. [次へ] をクリックしてから、[完了] をクリックし、MANAGER-REC を条件付きレイアウトとして追加します。

MANAGER-REC のフォルダが右側のペインに表示されます。

レコードの種類を識別するフィールドと条件を指定する必要があります。

3. 右側のペインで、"+" をクリックして MANAGER-REC フォルダを展開します。続いて、"+" をクリックして 01 MANAGER-REC エントリを、"+" をクリックして 02 MN-CODE エントリを、それぞれ展開します。
4. 右側のペインで 03 MN-POSITION を右クリックしてから、ポップアップ メニューの [属性] をクリックします。

「フィールドの属性」ダイアログ ボックスが表示されます。

5. プルダウン リストから選択して「条件」を「IS = TO」に設定します。

6. 左側の「条件」フィールドに、M(大文字)と入力します。

M は、管理者のレコードを示します。点線の下 の 2 つの文字は 16 進表現に相当し、垂直に整列されます。ANSI の M は 16 進 で 4D なので、この場合は "4" と "D" になります。

7. [OK] をクリックします。

右側のペインの 03 MN-POSITION の横に、小さく赤で IF と表示されます。

8. 前述の説明で MANAGER を EXECUTIVE に、MN-POSITION を EX-POSITION に、さらに M を E に読み替えてこの手順を繰り返し、役員レコードの条件付きレコード レイアウトを作成します。

### 6.3.6 レコード レイアウト ファイルの保存

レコード レイアウト ファイルは、次のように保存します。

1. [ファイル] メニューの [名前を付けて保存] をクリックします。

「名前を付けて保存」ダイアログ ボックスが表示されます。デフォルトのフォルダは、`d:\¥NetExpress¥Base¥Demo¥Dtoldemo` です。

デフォルトのファイル名は、`dfdstaff.str` で、これは COBOL プログラム名に拡張子 `.str` が付いたものです。

データ ファイル エディタは、データ ファイルを読み込むと、同じフォルダの中にデータ ファイルと同じ名前でも拡張子 `.str` を持つレコード レイアウト ファイルがないかを調べます。拡張子以外の名前が COBOL プログラムと同じデータ ファイルを持つ簡単なアプリケーションでは、拡張子 `.str` を使用したこのデフォルトのプログラム名を使用するのが適切です。

2. データ ファイル名が `staff.dat` であるので、`staff.str` という名前のレコード レイアウト ファイルが必要になります。(データ ファイル コンバータを使用して `datavseq.dat` を `staff.dat` に変換したことを思い出してください。) そのため、レコード レイアウト ファイルの「ファイル名」フィールドを `staff.str` に変更します。その後 [保存] をクリックします。
3. [ファイル] メニューの [閉じる] をクリックして「レコード レイアウト エディタ」ウィンドウを閉じます。

### 6.3.7 書式設定されたデータ ファイルの表示

レコード レイアウト ファイルは、次のように使用します。

1. 前に開いたままにしておいた「データ ファイル エディタ」ウィンドウをクリックして、前面に表示させます。
2. [ファイル] メニューの [レコード レイアウトのロード] をクリックします。
3. `staff.str` を選択して、[開く] をクリックします。

新しく右側にペインが表示されます。左側のペインには、以前と同様に複数のレコードが行ごとに書式設定されていない状態で表示されます。

右側のペインには、データ ファイル エディタによりレコード レイアウト ファイルの書式情報が適用された状態でデータが表示されます。このペインには、書式設定された単一のレコードが表示されます。レコード レイアウト名は、ペインの最上部に表示されます。右上には、2 つのナビゲーション コントロールがあります。小さい上向きの矢印で前のレコードに、小さい下向きの矢印で次のレコードに移動することができます。

[値] 列をクリックして既存のデータに上書き入力すると、レコードを編集できます。前述の書式設定されていないビューでの編集機能は、書式設定されたビューでも使用できます。

4. 左側のペインをクリックして、カーソルを書式設定されていないビューのレコードに移動します。

ファイル内でカーソルを移動させたときに、カーソル位置のレコードと一致する書式設定されたビューがどのように変化するか確認します。

### 6.3.8 ファイルの 16 進表示

データ ファイル エディタを使用し、データを 16 進で表示して編集する方法は、次のとおりです。

1. 「データ ファイル エディタ」ウィンドウ (またはペイン) を右クリックして、ポップアップ メニューの [16 進表示] をクリックします。

ウィンドウの下部に新しいペインが表示されます。左下のペインに、左上のペインで選択したレコードが表示されます。その下の行に 16 進表現が表示されます。基本的なフィールド項目については、右上のペインで選択したフィールドが右下のペインに表示されます。その下に 16 進表現が表示されます。基本的なフィールド項目以外については、右側のペインに「Group item」などのテキスト説明が表示されます。

16 進数の組み合わせは、垂直に整列されます。たとえば、ANSI データを編集する場合、M を 16 進表示すると、上の行に "4"、下の行に "D" が表示されます。16 進の行のどちらかを表示できない場合は、下端を下にドラッグしてウィンドウを拡大してください。

2. 「データ ファイル エディタ」ウィンドウを右クリックして、ポップアップ メニューの [16 進表示] をクリックします。

このメニュー項目に付けられたチェックマークが削除され、16 進のペインが表示されなくなります。

### 6.3.9 データ ファイルの印刷

印刷機能を使用してデータ ファイル エディタのデータ ファイルを印刷したり、印刷プレビュー機能を使用して印刷ページを画面に表示することができます。表示方法としては、書式設定されたビューか、書式設定されていないビューを選択できます。現在のレコード、すべてのレコード、または部分的レコードの印刷を選択できます。

書式設定されていないデータ ファイルの印刷をプレビューする方法は、次のとおりです。

1. 「データ ファイル エディタ」ウィンドウの左側のペインをクリックしてから、[ファイル] メニューの [印刷設定] をクリックします。
2. [用紙の向き] を「横置き」に設定し、[OK] をクリックします。
3. [ファイル] メニューの [印刷プレビュー] を選択します。
4. 「ヘッダ (テキスト)」の横のチェックボックスをチェックして、見出しテキストとして「データ ファイル STAFF.DAT の印刷」と入力します。
5. 次の項目にチェックマークを付けます。
  - ページ番号
  - 定規
  - レコード番号
  - 16 進の値

続いて、次の項目を選択します。

- すべてのレコード
  - 全レコード
6. [プレビュー] をクリックします。

「プレビュー」ウィンドウが表示されます。ツールバーのボタンを使用して拡大、縮小、ページ間の移動、印刷を行うことができます。

7. 「プレビュー」ウィンドウのボタンバーで [閉じる] をクリックします。


### 6.3.10 複数のデータ ファイルの編集

データ ファイル エディタでは、NetExpress で使用する多くのエディタと同様に、さまざまな方法で複数のウィンドウを IDE に配置し、複数のファイルを一度に編集することができます。


ウィンドウを重ねて 2 つのデータ ファイルを編集する方法は、次のとおりです。

1. 「プロジェクト」ウィンドウで、ファイル `datavseq.dat` をダブルクリックします。

`datavseq.dat` の新しい「データ ファイル エディタ」ウィンドウが開きます。このファイルに関連付けられているレコード レイアウトがないので、書式設定されていないペインだけが表示されます。

2.  ボタンをクリックして、「プロジェクト」ウィンドウを最小化します。
3. [ウィンドウ] メニューで [横に並べて表示] を選択します。

これで、datavseq.dat と staff.dat の両方の「データ ファイル エディタ」ウィンドウを参照できます。

4. NetExpress ウィンドウの下部にある datavseq.dat と staff.dat のタグを交互にクリックして、ウィンドウを切り替えます。
5. 両方の「データ ファイル エディタ」ウィンドウを閉じます。「プロジェクト」ウィンドウを通常の大きさに戻します ( ボタンをクリックします)。

### 6.3.11 新しいデータ ファイルの作成

固定長順ファイルは、次のように作成します。

1. [ファイル] メニューの [新規作成] を選択し、[データ ファイル] を選択して、[OK] をクリックします。

「ファイルの作成」ダイアログ ボックスが表示されます。

2. 「ファイル名」フィールドに `d:\NetExpress\Base\Demo\Doldemo\newfile.dat` を指定します。次に、「最大の長さ」フィールドを 20 に設定し、「文字集合」フィールドを ANSI に設定します。他のフィールドではデフォルト値を使用して、[作成] をクリックします。

固定長順ファイルにはファイル ヘッダーがないのでプロファイル ファイルに入力したファイル ヘッダーの詳細情報を保存するように NetExpress によりメッセージが表示されます。プロファイルを作成すると、次にデータ ファイルを開くときに、ファイル ヘッダーの詳細情報を入力しないで済みます。

3. [はい] をクリックします。

NetExpress によって、データ ファイルと同じディレクトリに作成されるプロファイル ファイルにプロファイルが保存されます。プロファイル ファイルの名前は、データ ファイルと同じファイル名に拡張子 `.pro` が付いたものです。そのため、この新しいデータ項目のプロファイル ファイル名は、`newfile.pro` になります。

「データ ファイル エディタ」ウィンドウが表示されます。このウィンドウには、「ファイルが空です。」と表示されます。

このファイルにレコードの追加と削除を行います。

4. 「データ ファイル エディタ」ウィンドウを右クリックし、ポップアップ メニューの [後にレコードを挿入する] を選択します。

カーソルは、ファイルの開始位置に配置されます。



5. abc と入力します。
6. 「データ ファイル エディタ」ウィンドウを右クリックし、ポップアップ メニューの [レコードの繰り返し] を選択します。

前述のレコードのコピーがファイルに追加されます。

7. 「データ ファイル エディタ」ウィンドウを右クリックし、ポップアップ メニューの [レコードの削除] を選択します。次に、「削除の警告」メッセージ ボックスの [はい] をクリックします。

新規レコードがファイルから削除されて、カーソルが前のレコードに移動します。

8. [ファイル] メニューの [上書き保存] をクリックします。
9. 「データ ファイル エディタ」ウィンドウを閉じます。

## 6.4 次へ進む前に

プロジェクトを閉じます。次のセッションに進む前に、作業を中断する場合は、NetExpress を閉じてかまいません。

「ここからチュートリアルをスタート」の章の チュートリアル マップに戻り、目的に沿って次のデモ操作を選択します。

# 第7章 Web アプリケーションの概要

この章では、Web アプリケーションの作成に関するチュートリアルに共通する基本的な概念について説明します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了してから、この章を読んでください。

---

必ず World Wide Web の基本を理解してから、先へ進んでください。World Wide Web については、付録「Web アプリケーション」で説明しています。付録には、この章と Web に関する一連のデモ操作で使用される多くの用語の説明が記載されています。

NetExpress では、次の 3 つの方法で Web アプリケーションを作成できます。

- 最初から作成する方法 - データを受け付け、処理し、結果を表示するサーバー側プログラムと、ユーザーのブラウザに表示する HTML フォームを作成する必要があります。
- 既存の COBOL プログラムから作成する方法 - データを受け付け、既存のプログラムに渡して処理し、結果を表示するサーバー側プログラムと、ユーザーのブラウザに表示する HTML フォームを作成する必要があります。
- 既存のデータベースから作成する方法 - データを受け付け、データベースの情報を検索し、結果を表示するサーバー側プログラムと、ユーザーのブラウザに表示する HTML フォームを作成する必要があります。

3 つの方法すべてにインターネット アプリケーション ウィザードを使用します。最初の方法では、まず HTML ページ ウィザードを使用して独自のフォームを設計します。フォーム、プログラム、またはデータベースなどの既存の項目を設定すると、残りのアプリケーションが自動作成されるという意味では、上記の 3 種類の方法におけるインターネット アプリケーション ウィザードの使用方法は同じです。

この方法で生成されたフォームとフォーム処理コードは、NetExpress に組み込まれた標準テンプレートを基礎としています。条件に合わせて独自にカスタマイズする場合は、後でフォームとフォームを処理するコードを編集することができます。

このパートのデモ操作では、インターネット アプリケーション ウィザードを使用して上記の 3 種類の方法を紹介します。これらのデモ操作でアプリケーションをテストする場合は、どの Web ブラウザでも使用できます。

## 7.1 Solo

アプリケーションをテストする場合、どの Web サーバー ソフトウェアでも使用できます。これらのデモ操作では、NetExpress に同梱されている Web サーバー ソフトウェアである Solo を使用します。Solo は、NetExpress を使用

して開発されたインターネット アプリケーションをテストするために特別に設計されたものです。プロダクション Web サーバー用には設計されていません。Solo は、構成も設定もする必要がありません。(Solo の自動構成方法を確認するには、付録「Web サーバーの構成」を参照してください。)

## 7.2 サーバー側プログラム

NetExpress を使用して作成されたサーバー側プログラムでは、Micro Focus の 2 つの COBOL 拡張機能である拡張 ACCEPT 文と EXEC HTML 文 (埋め込み HTML) によってフォームの入出力が行われます。埋め込み HTML により、文の中に書き込まれた HTML が出力されます。フォームには、[送信] ボタンが含まれている必要があります。エンドユーザーがこのボタンをクリックすると、Web サーバーにあるプログラムが実行されます。ACCEPT 文により、フォームからのデータ入力が可能になります。

Micro Focus COBOL には拡張 DISPLAY 文もありますが、EXEC HTML 文の方がより実効性があります。

### 7.2.1 構造

サーバー側プログラムは、通常、次の要素を中心に構成されています。

1. 入力フォームのデータを入力するための ACCEPT 文
2. 入力フォームに関連付けられたデータ項目を作業領域に転記するためのコード
3. 挿入されたビジネス ロジック
4. 作業領域の結果をフォームに関連付けられたデータ項目に転記するためのコード
5. 出力フォームを作成するための EXEC HTML 文

ただし、次のことに注意してください。

- メイン ループはありません。エンドユーザーが入力フォームを送信するたびに、プログラムは再実行されます。
- ACCEPT 文によりフォームが読み込まれるわけではありません。また、フォームの読み込みを待つこともありません。エンドユーザーが [送信] ボタンをクリックすると、Web ブラウザによって記入済みフォームがプログラムに送信されます。そのときに ACCEPT 文がある場合は、データを受け付けますが、それ以外は何の操作も行いません。
- EXEC HTML 文は、フォームを作成するだけで出力しません。プログラムが終了すると、フォームは Web サーバーによってブラウザに送信されます。

### 7.2.2 対称と非対称

アプリケーションでは、入力と出力の両方に同じフォームを使用することも、また、別のフォームを使用すること

も可能です。

1 つのフォームだけを使用するアプリケーションを対称アプリケーションといいます。このアプリケーションは、通常、実行可能ファイルを実行中のエンドユーザーにより起動されます。実行可能ファイルにリンクする Web ページを作成する場合、エンドユーザーがこのリンクをクリックすると、プログラムがいったん実行され、ブラウザにフォームが表示されます。エンドユーザーがフォームに入力して [送信] ボタンをクリックすると、プログラムが再起動してデータが処理され、結果が表示されます。再び、フォームが準備され、エンドユーザーが使用できるようになります。

入力フォームと出力フォームが異なるアプリケーションを非対称アプリケーションといいます。このアプリケーションは、通常、入力フォームを読み込むエンドユーザーにより起動されます。このフォームにリンクする Web ページを作成する場合、エンドユーザーがリンクをクリックすると、フォームが読み込まれます。エンドユーザーがフォームに記入して [送信] ボタンをクリックすると、プログラムが起動してデータが処理され、結果とともに出力フォームが表示されます。次のクエリーを入力するには、入力フォームを再度読み込みます。(入力フォームに含まれるすべての入力フィールドが出力フォームに含まれている場合、入力フォームを再読み込みせずに、このフォームを使用して入力することができます。プログラムは、ブラウザからデータを受信しているだけなので、2 つのフォームの違いは認識されません。)

複数の入力フォームを使用するアプリケーションでは、フォームごとに個別のサーバー側プログラムを作成します。

## 7.3 次へ進む前に

「ここからチュートリアルをスタート」の章のチュートリアル マップに 戻り、目的に従って次のデモ操作を選択します。

# 第8章 Web アプリケーションの作成

この章と次の章では、Web アプリケーションを作成します。

この章では、HTML ページ ウィザードと Form Designer を使用して、ユーザー インターフェイスを作成します。その後、インターネット アプリケーション ウィザードを使用してデータを処理するためのプログラムを作成します。IDE、Web ブラウザ、NetExpress に組み込まれている Web サーバー ソフトウェアである Solo を使用して、アプリケーションを実行し、インターフェイスを表示します。

次の章「Web アプリケーションの完成と実行」では、プログラムにビジネス ロジックを追加し、完成したアプリケーションを実行します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了してください。その後で、「Web アプリケーションの概要」の章を読み、この章へ進んでください。

---

## 8.1 概要

HTML ページ ウィザードを使用して、Web ブラウザで表示するためのフォームを作成します。フォームは、アプリケーションのユーザー インターフェイスになります。このウィザードから Form Designer を起動し、レイアウトの設計や編集を行います。

フォームを処理するサーバー側 COBOL プログラムを生成するには、インターネット アプリケーション ウィザードを使用します。

HTML フォームでは、コントロールの順序を指定できますが、通常、正確なレイアウトはフォームを表示するブラウザで決定されます。ただし、Form Designer には、レイアウトを詳細に設定する機能があります。

サーバー側プログラムは、Web アプリケーションの中核です。このプログラムは、入力フォームのデータを受け付け、1 つ以上の出力フォームにデータを出力します。1 つのフォームを入力と出力に使用するのは、特殊な場合です。この章では、このようなアプリケーションを作成します。

## 8.2 準備

NetExpress が閉じている場合は、再起動します。「プロジェクト」ウィンドウや他のウィンドウが開いている場合は、閉じます。

## 8.3 具体的な手順

この手順では、次の操作を行います。

1. プロジェクトとフォームの作成
2. フォームへのコントロールの追加
3. サーバー側プログラムの作成
4. フォームのテスト

### 8.3.1 プロジェクトとフォームの作成

まず、「*NetExpress の使用方法*」の章の説明にしたがい、プロジェクトを作成します。ただし、直接フォームを作成する場合は、途中で NetExpress によりプロジェクトを作成する機会があります。フォームは、次のように作成します。

1. [ファイル] メニューの [新規作成] をクリックし、「新規作成」ダイアログ ボックスで「HTML ページ」を選択して、[OK] をクリックします。
2. プロジェクトを作成するかどうかを質問するメッセージが表示されたら、[OK] をクリックします。
3. 「HTMLプロジェクト」が選択されていることを確認してください (デフォルトでは、HTMLプロジェクトが選択されています)。プロジェクト名として Goform を、プロジェクトを格納するフォルダとして `d:\NetExpress\Base\Demo\Goform` を入力し、[作成] をクリックします。
4. このディレクトリを作成するかどうかをたずねるメッセージに対して、[はい] をクリックします。(この手順を以前に実行している場合は、かわりに既存のプロジェクトに上書きするかどうかをたずねられます。その場合、[はい] をクリックします。)

このプロジェクトの「プロジェクト」ウィンドウが表示されます。また、HTML ページ ウィザードの最初のページも表示されます。

5. 「位置フォーム.htm」(拡張子 .htm は設定によって表示されません。)をクリックし、[次へ] を選択します。

通常、HTML フォームを設計するときに指定できるのはフォームに配置するオブジェクトの順序だけです。正確な位置は、フォームを表示するブラウザによって決定されます。ただし、NetExpress では、フォームを部分的に位置フォームにすることができます。位置フォームのオブジェクトは、決められた位置に正確に配置されます。ここでは、[位置フォーム.htm] をクリックして、フォームに、その大部分を占める位置フォームを 1 つ作成するように選択します。

Web 用語では、ユーザーが入力するフィールドを持つ Web ページをフォームと呼ぶことを覚えておいてください。このフォームとフォーム内の位置フォームを混同しないでください。

6. デフォルトの HTML ファイル名に mypage を上書きします。[クロスプラットフォーム(テーブル形式)]を選択したまま、[次へ] をクリックします。

これによって、HTML フォームの名前が mypage.htm に設定されます。標準の HTML ファイルの拡張子は、.htm です。

このクロスプラットフォーム を指定すると、NetExpress の位置フォームを使用してオブジェクトを確実に目的の場所に配置することができます。クロスプラットフォームでは、このために HTML テーブルを使用します。ダイナミック HTML では、スタイル属性を使用します。HTML テーブルはより多くのブラウザでサポートされています。ダイナミック HTML を使用すると、より正確に位置を指定できますが、これをサポートしているのはインターネット エクスプローラ 4.0 だけです。エンドユーザーがフォームの表示に使用するブラウザはさまざまであることに注意してください。

7. mypage.htm または mypage.mff がすでに存在していることを示すメッセージ ボックスが表示されたら、[はい] をクリックして上書きします。

この手順を以前に実行している場合は、これらのメッセージが表示されることがあります。それ以外は、直接最後のダイアログ ボックスが表示されます。

最後のダイアログ ボックスは、選択した項目をまとめたものです。

8. [完了] をクリックします。

HTML ウィザードにより、フォームが生成され、作業が終了します。生成されたファイルの名前がプロジェクトに追加され、プロジェクトの従属関係が確認されます。

続いて、IDE によって、「Form design」ウィンドウと、関連する 3 つのウィンドウが開かれます。「Form design」ウィンドウは、背景が点線のグリッドになっているので判別できます。タブが 3 つある追加のツールバーが表示されます。これはオブジェクト ツールバーといい、フォームに追加するオブジェクトが含まれています。追加のメニュー [ページ] と [整列] がメニュー バーに表示されます。

### 8.3.2 この段階の IDE 画面

この段階では、IDE は図 8-1 のように表示されます。

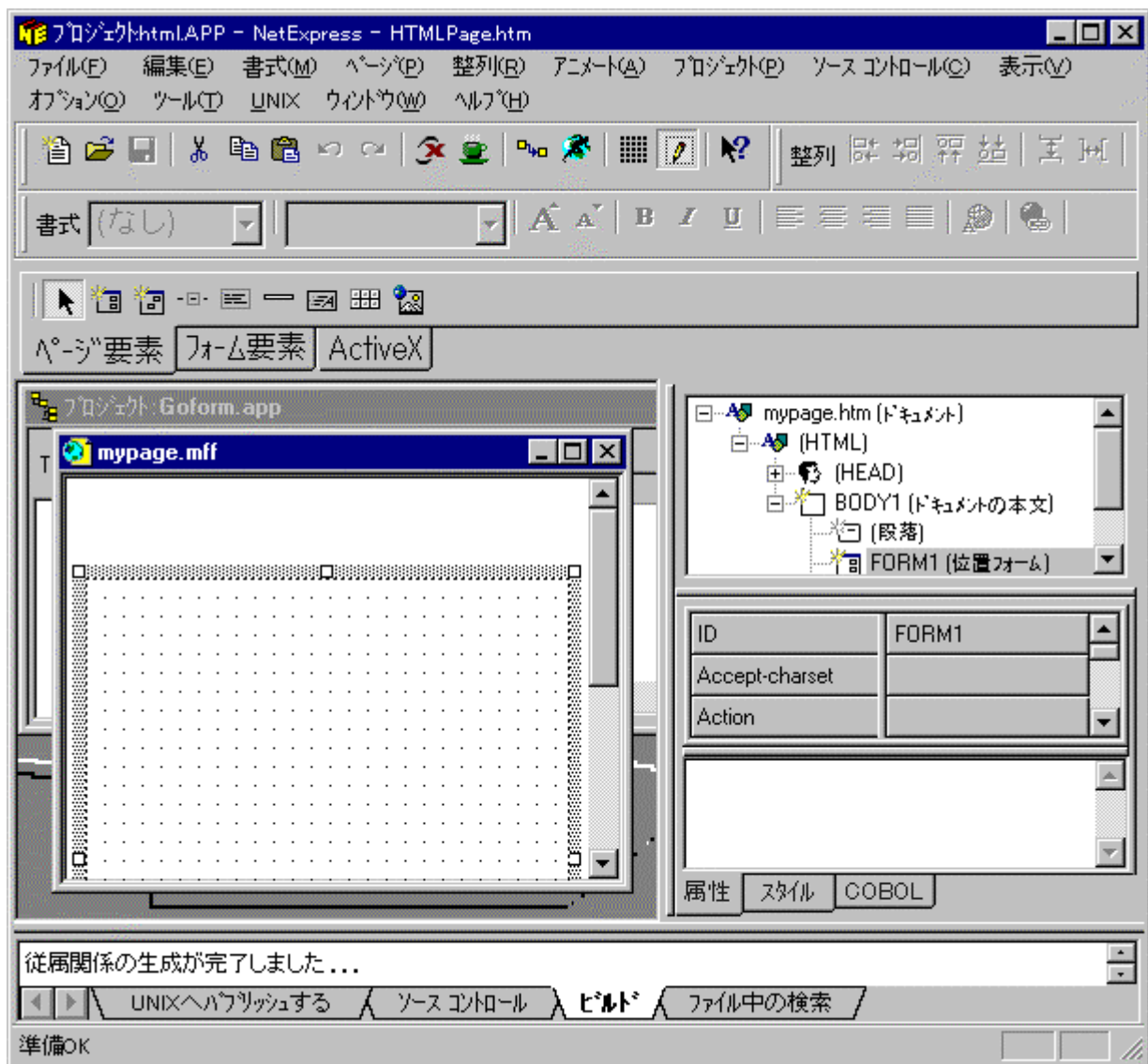


図 8-1 Form Designer のウィンドウを表示する IDE

次のような新しいウィンドウが 4 つ表示されます。

- 「Form design」ウィンドウ

背景が点線のグリッドになっているウィンドウで、フォームを設計するために使用します。作成したフォーム mypage.htm を読み込んで、フォームの設計準備をします。大部分の領域が強調表示された四角形で覆われていることに注意してください。これが位置フォームです。

- 「コントロール ツリー」ウィンドウ

右上のウィンドウには、フォームのコントロールがすべて表示されます。コントロールを右クリックしてコンテキストメニューから [ID設定] を選択すると、コントロール名を変更できます。



- 「プロパティ リスト」ウィンドウ

「コントロール ツリー」ウィンドウの下ウィンドウには、現在「Form design」ウィンドウで選択されているオブジェクトのプロパティが表示されます。プロパティ名が左下に、対応する値が右下に、表示されます。値フィールドをクリックすると、プロパティを編集できます。

- 「ヘルプ」ウィンドウ

右下ウィンドウには、「プロパティ リスト」ウィンドウのヘルプが表示されます。「プロパティ リスト」ウィンドウのフィールドをクリックすると、そのフィールドのヘルプが「ヘルプ」ウィンドウに表示されます。

プロジェクトの変更を確認するために、「Form design」ウィンドウをドラッグして脇に移動させることができます。また、ウィンドウを IDE 内で移動させたり、サイズを変更して、すべてのウィンドウが表示されるようにすることもできます。

### 8.3.3 フォームへのコントロールの追加

フォームは、次のように設計します。

1. オブジェクト ツールバーの [フォーム要素] タブをクリックして、他のタブより前に表示します。
2. 位置フォーム、つまり、「Form design」ウィンドウの点線の四角形をクリックして、作成するオブジェクトがこの中に配置されるように確実にこのフォームを選択します。
3. オブジェクト ツールバーの [テキスト入力] ボタン **abi** を選択し、「Form design」ウィンドウをクリックします。(マウス ポインタをしばらくツールバー ボタンに置いたままにすると、ツール情報が表示され、ボタン名がわかります。) フィールドに表示された初期状態のテキスト「テキスト入力」を削除します。

これで、クリックした位置にエントリ フィールドが作成されます。エントリ フィールドとは、エンドユーザーがデータを入力したり表示できるフィールドのことです。フィールドの配置後も、マウス ポインタをその境界に合わせてマウス ボタンを押したままにすると、ドラッグすることができます。ここでは、このフィールドをウィンドウの右側に配置します。

「プロパティ リスト」ウィンドウの「Name」プロパティに注意してください(「Name」プロパティを表示するには、リストをクリックして、矢印キーでリストを上下に移動する必要がある場合があります)。この場合、「Name」プロパティの値は、input1 です。これは、後で作成する COBOL プログラムでこのフィールドに対するデータ名として使用されます。

4. 再度、点線の四角形をクリックして、選択します。この後に作成するオブジェクトが中に配置されます。
5. 同様に、[テキスト] ボタン **Aa** を選択し、エントリ フィールドの左をクリックして、エントリ フィールドのラベルを作成します。ラベルは、エントリ フィールドに重ならないようにしてください。重なった場

合は、フィールドの境界をドラッグして移動するか、サイズ変更してください。

6. ラベル フィールドで、デフォルトのキャプション「テキスト新規」に上書きで「Enter Name」と入力します（改行してしまうので、[Enter] キーを押さないでください）。
7. マウス ポインタをキャプションの上でクリックし、その左側をエントリ フィールドの右下にドラッグして、キャプションとエントリ フィールドの両方を選択します。まず、「Form design」ウィンドウを拡大して、両方を表示できるようにする必要があります。


四角形がしばらく表示された後、両方のフィールドが強調表示されます。

8. [整列] メニューの [配置] を選択し、[上端] をクリックします。

これで 2 つのフィールドが整列します。

9. 再び点線の四角形を選択してから、前述のとおり 2 番目のエントリ フィールドとラベルを追加し、ラベルのキャプションを「Greeting」に設定します。

2 番目のエントリ フィールドの「プロパティ リスト」では「Name」プロパティが input2 になっていることに注意してください。

10. 必要に応じて、2 つのキャプションと、2 つのエントリ フィールドをそれぞれ整列させることができます。どちらの場合も、2 つのフィールドを選択します（両方のフィールドがこの四角形に含まれていることを確認してください。それ以外の場合、フィールドは選択されません）。[整列]メニューの [配置] をクリックしてから、[左端] をクリックします。
11. 再び点線の四角形を選択してから、ツールバーの [実行ボタン]  をクリックします。続いて、フォームをクリックして、ウィンドウの下部にボタンを配置します。

この段階では、フォームは、図 8-2 のように表示されます。

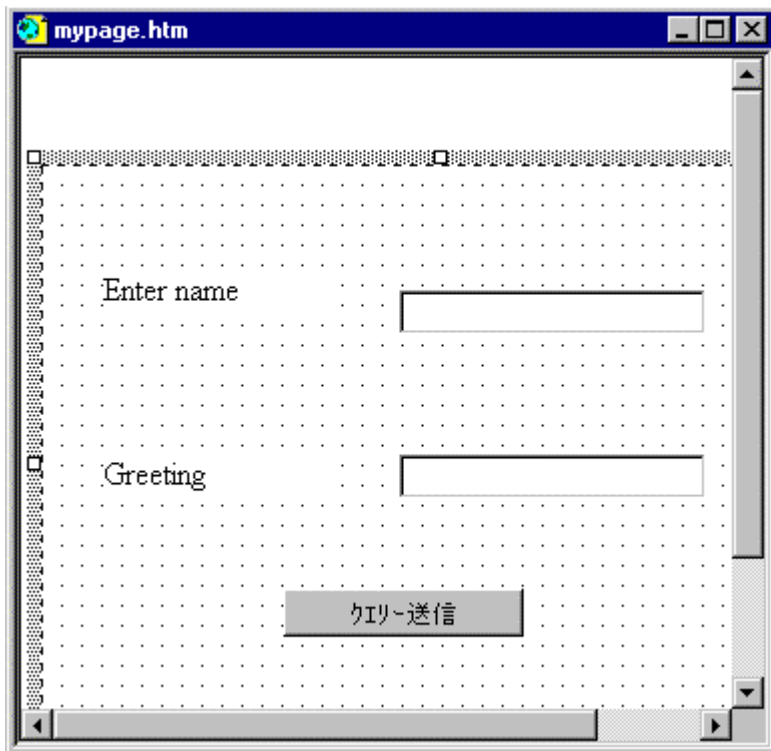


図 8-2 「Form design」ウィンドウで作成したフォーム

12. [ファイル] メニューの [上書き保存] をクリックして、フォームを保存します。
13. 「Form design」ウィンドウの  ボタンをクリックします。

「Form design」ウィンドウとそれに関連付けられたウィンドウをすべて閉じます。厳密にはここでウィンドウを閉じる必要はありません。ウィンドウを開いたままにして、次のデモ操作に進むこともできますが、閉じた方が画面が煩雑にならずに済みます。

### 8.3.4 サーバー側プログラムの作成

インターネット アプリケーション ウィザードを使用して、サーバー側プログラムを作成します。

1. [ファイル] メニューの [新規作成] をクリックしてから、「新規作成」ダイアログ ボックスで「インターネット アプリケーション」を選択し、[OK] をクリックします。

インターネット アプリケーション ウィザードの最初のページが表示されます。このページでは、「Web アプリケーションの概要」の章で説明する 3 種類のアプリケーション作成方法から使用する方法を 1 つ選択します。

2. 「NetExpress が作成した HTML からサーバープログラムを生成」を選択します。

ダイアログ ボックスの下部では、「HTMLフォーム」が自動的に選択されます。

3. [次へ] をクリックします。

表示されたページでは、サーバー側プログラムに名前を付け、使用する入力フォームと 1 つ以上の出力フォームを指定することができます。

4. 「ファイル名」フィールドに myprog.cbl と入力します。

この名前がサーバー側プログラムのファイル名になります。

5. 「入力ファイル/フォーム」と「出力フォーム」の両方のフィールドで、mypage が選択されていること（強調表示されていること）を確認してください。

プロジェクト フォルダが以前に使用されていない場合、存在するフォームは mypage だけです。「入力ファイル/フォーム」フィールドは、自動的に mypage に設定されます。

サーバー側プログラムには 1 つの入力フォームと、1 つ以上の出力フォームがあります。このように入力と出力に 1 つのフォームを使用するプログラムを対称サーバー側プログラムといいます。

6. [次へ] をクリックします。

これによって、選択した項目をまとめたダイアログ ボックスが表示されます。

7. [完了] をクリックします。

インターネット アプリケーション ウィザードは、プログラムを生成して終了します。生成されたファイルの名前がプロジェクトに追加され、プロジェクトの従属関係が確認されます。その後、インターネット アプリケーション ウィザードが閉じます。

### 8.3.5 作成されたファイル

これでユーザー インターフェイスが作成されました。ウィザードと Form Designer によって次のファイルが作成され、プロジェクトに追加されています（すべてのファイルを表示するには、「+」記号をクリックして「プロジェクト」ウィンドウのツリー ビューを展開する必要がある場合があります）。

- mypage.htm - HTML フォームです。
- myprog.cbl - サーバー側プログラムです。

次のファイルは、右側のペインだけに表示されます。

- mypage.mff - サーバー側プログラムの生成に必要な追加情報が含まれています。

また、プロジェクトをビルドするときに作成される次のオブジェクト ファイルもプロジェクトに追加されています。

- myprog.obj - コンパイルされたサーバー側プログラムです。

- myprog.exe - 実行可能ファイルです。

コピーファイルもいくつか作成されます (拡張子は .cp\* です)。これらのファイルの用途は、.cbl ファイルのコメントで説明されています。

上記の説明では、ファイル名を小文字で表記していますが、プロジェクトでは大文字で表記されるものもあります。ファイル名では、大文字と小文字を区別しません。

フォームを更新する必要がある場合は、「プロジェクト」ウィンドウの mypage.htm をダブルクリックするだけで、Form Designer が開きます。更新したフォームを保存するときに、コピーファイルが再生成されるので、コピーファイルを直接編集しないでください。ただし、ウィザードで生成された .cbl プログラムは編集できます。

### 8.3.6 フォームのテスト

このマニュアルを Web ブラウザで参照している場合は、ブラウザを新しく開き (ブラウザで [ファイル] メニューの [新規作成] を選択します)、この項の作業中にもマニュアルを表示できるようにします。

Web ブラウザでフォームの表示形式を確認するには、次のようにします。

1. 「プロジェクト」ウィンドウの mypage.htm をダブルクリックします。


これで、フォームが開いて再編集できます。ここでは、新たに変更はしません。ここで、編集用のフォームを開いた理由は、メニュー バーに [ページ] メニューと [整列] メニューを再表示するためです。

2. [ページ] メニューの [プレビュー] をクリックします。

この操作によって、Web ブラウザが実行されていない場合は、Web ブラウザが起動され、フォームが読み込まれます。また、Soloが実行されていない場合は、Soloも起動されます。

この操作については、次のデモ操作「Web アプリケーションの完成と実行」でより詳しく説明します。ここでは、ブラウザでフォームの表示形式を確認するだけにします。

2 つのエントリ フィールドには ":f-Input1" と ":f-Input2" が表示されます。これらは、サーバー側プログラムがフォームを送信したときに表示されるデータのプレースホルダなので、無視してかまいません。

3. 「Form design」ウィンドウの  ボタンをクリックします。

必要な場合、ビジネス ロジックを追加しなくても、サーバー側プログラムが実行されていることを確認できます。

4. ツールバーの  をクリックします。

NetExpress により .exe ファイルがビルドされます。

5. NetExpress の [アニメート] メニューの [実行] をクリックします。

6. 「アニメーションの起動」ダイアログ ボックスの [OK] をクリックします。

ここでサーバー側プログラムが起動し、ブラウザにフォームが表示されます。プレースホルダがすでに表示されていないことに注意してください。IDE は自動的に最小化されます。


7. 最初のフィールドに名前を入力し、[クエリ送信] をクリックしてください。

プログラムが起動し、フォームが再表示されますが、まだビジネス ロジックを追加していないので、2 番目のフィールドには何も表示されません。これについては、次のデモ操作「*Web アプリケーションの完成と実行*」で実行します。

8. 最小化された IDE を元に戻し、NetExpress の [アニメート]メニューの [アニメート停止] をクリックします。

## 8.4 次へ進む前に

フォームを表示するために Web ブラウザを別に開いている場合は、閉じます。

次のデモ操作「*Web アプリケーションの完成と実行*」で、このアプリケーションを完成させます。このまま次のデモ操作に進む場合は、プロジェクトを開いたままにしておいてかまいません。いったん中止する場合は、プロジェクトを閉じ、次のデモ操作を開始するときに再度開くことができます。または、NetExpress 全体を閉じてかまいません。その場合、プロジェクトは閉じても閉じなくてもかまいません。また、Windows タスクバー トレイの Solo アイコン  を右クリックし、ポップアップ メニューの [終了] を選択して、Solo を閉じることもできます。

## 第9章 Web アプリケーションの完成と実行

前の章では、HTML ウィザードと Form Designer を使用してフォームを設計し、インターネット アプリケーション ウィザードを使用して COBOL プログラムを生成しました。この章では、生成された COBOL プログラムを編集し、ビジネス ロジックを追加します。その後、Web ブラウザと、NetExpress に組み込まれた Web サーバー ソフトウェア Solo を使用して、完成したプログラムを実行し、デバッグします。Web ブラウザと Solo は、どちらも 1 台のマシンで実行できます。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了します。次に、「Web アプリケーションの概要」の章を読んでください。また、前のデモ操作「Web アプリケーションの完成」を終了してから、この章へ進んでください。

---

### 9.1 概要

インターネット アプリケーション ウィザードにより、フォームの入出力用コードを記述した COBOL ソース プログラムが .cbl ファイルに生成されます。この .cbl ファイルを編集し、フォームのデータを処理して出力データを作成するビジネス ロジックを追加します。また、CALL 文を追加し他のプログラムを呼び出すだけでもかまいません。この場合、フォームを処理するこのプログラムとビジネス ロジックを完全に別ファイルに分離できます。

### 9.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。プロジェクトが閉じている場合は、「ここからチュートリアルをスタート」の章の説明にしたがい、開きます。[開く] を使用する場合は、ディレクトリ `d:\NetExpress\Base\Demo\Goform` のプロジェクト Goform を開きます。デフォルトでは「開く」ダイアログ ボックスの「ファイルの種類」フィールドに .app、.cbl、および .cpy が設定されているので、ファイル Goform が表示されます。

コンピュータで Solo 以外の Web サーバー ソフトウェアを実行している場合は、それを閉じます。閉じない場合、Solo を実行するときに問題が生じる可能性があります。

先に Web ブラウザが開いていても問題ありません。

このマニュアルを Web ブラウザで参照している場合は、ブラウザを新しく開き (ブラウザで [ファイル] メニューの [新規作成] を選択します)、この章の作業中にもマニュアルを表示できるようにします。

## 9.3 具体的な手順

この手順では、次の操作を行います。

1. COBOL プログラムの編集
2. アプリケーションのビルド
3. Solo の起動
4. アプリケーションの実行
5. アプリケーションのデバッグ

### 9.3.1 COBOL プログラムの編集

COBOL プログラムにビジネス ロジックを追加するには、テキスト エディタを使用します。ここでは、IDE を使用します。

1. 「プロジェクト」ウィンドウで `myprog.cbl` をダブルクリックします。

プログラムのソースを表示する「テキスト」ウィンドウが開きます。ソースを明確に表示するためにウィンドウのサイズを変更する必要がある場合があります。[表示] メニューの「コピーファイルをすべて非表示」を確認し、項目の横にあるボタンのようなアイコンが押された状態になっていることを確認します（このような状態でない場合はクリックします）。

手続き部に目を通し、手続きについてのコメントを読みます。

2. [表示] メニューの [コピーファイルをすべて非表示] をクリックします。

この操作によって、[コピーファイルをすべて非表示] が解除され、ファイルが表示されます。コピーファイルは、インターネット アプリケーション ウィザードにより、`.cbl` ファイルとともに生成されます。

([コピーファイルをすべて非表示] を解除したときに影響を受けるのは、前に [ファイル] メニューの [コピーファイル] を使用して展開された COPY 文、またはプログラムが以前コンパイルされたときにソースに記述されていた COPY 文だけです。前章の「フォームのテスト」の項でプロジェクトがビルドされたときにプログラムがコンパイルされているので、ファイルはすべて展開されます。)

展開されたコピーファイルを参照します。

Convert-Input Section は、Input-Conversion Section を実行します。その結果、フォームに関連付けられたデータ項目のデータがプログラムの作業領域に転記されます。同様に、Mypage-Cvt Section により、フォームに関連付けられたデータ項目にデータが転記されます。ソースのコメントは、各コピーファイルの内容を説明するものです。



このプログラムでは、入力と出力に同じフォームを使用することに注意してください。2つのエン트리 フィールドの名前は、F-INPUT1 と F-INPUT2 です。入力の場合、2つのエン트리 フィールドに入力されたデータは、INPUT1 と INPUT2 に転記されます。一方、出力の場合は、INPUT1 と INPUT2 の内容が2つのエン트리 フィールドに転記されることがわかります。

3. Process-Business-Logic Section では、カーソルを EXIT 文の上の空白行に移動させ、[Enter] キーを押して空白の行を作成します。カーソルを 12 列目の EXIT の "e" 上に移動し、次のコードを挿入します。

```
string "Hello " Input1 delimited by size into Input2
```

実際のアプリケーションでは、通常、ここで、フォームのデータを処理して表示データの計算や検索を行うサブプログラムを呼び出します。

4. 「テキスト」ウィンドウを閉じます。ファイルを保存するかどうかをたずねるメッセージに対して [はい] をクリックします。

### 9.3.2 アプリケーションのビルド

アプリケーションは、次のようにビルドします。

1. [プロジェクト] メニューの [すべてリビルド] をクリックします。

NetExpress は、プログラムの保存とコンパイルを行い、実行可能ファイルをビルドします。「出力」ウィンドウに "リビルド完了" というメッセージが表示されるまで待ってから、先に進みます。

### 9.3.3 アプリケーションの実行

アプリケーションの実行方法は、次のとおりです。

1. NetExpress の [アニメート] メニューで [実行] をクリックします。

このアプリケーションでは、フォームを1つしか使用しないため、「アニメーションの起動」ダイアログボックスには、実行の開始場所として実行可能ファイルが表示されます。

2. 「アニメーションの起動」ダイアログボックスの [OK] をクリックします。

この操作により Solo と Web ブラウザが起動し、エンドユーザーが実行可能ファイルへのリンクをクリックしたかのようにプログラムが実行されます。プログラムによりフォームが表示されます。IDE は自動的に最小化されます。

Solo の実行に問題が生じた場合は、最小化された IDE を元のサイズに戻し、NetExpress の [ヘルプ] メニューで [ヘルプ トピック] をクリックします。ヘルプの [キーワード] タブで「Solo トラブルシューティング」を参照します。

3. Web ブラウザで、「Enter Name」フィールドをクリックして名前を入力し、[クエリ送信] をクリックしま

す。

プログラムが実行されます。この操作により、「Greeting」フィールドの "Hello " の後に名前が配置され、Web ブラウザにフォームが表示されます。

4. 「Enter Name」フィールドをクリックし、自分の名前を別の名前に置換して [クエリ送信] をクリックします。

プログラムが再実行され、"Hello " に続いて新しい名前が表示されます。

5. 最小化された IDE を元のサイズに戻し、NetExpress の [アニメート] メニューで [アニメート停止] をクリックします。

### 9.3.4 アプリケーションのデバッグ

アプリケーションは、次のようにデバッグします。

1. NetExpress の [アニメート] メニューで [アニメート開始] をクリックします。
2. 「アニメーションの起動」ダイアログ ボックスの [OK] をクリックします。

Web ブラウザが起動し、IDE に myprog.cbl のソースが表示され、デバッグの準備が整います。通常、プログラムの PERFORM 文やサブプログラム呼び出しに関係する部分だけを高速で実行してフォームを表示し、その後で実際にデバッグを開始します。


3. NetExpress の [アニメート] メニューで [実行] をクリックします。

IDE が最小化され、アニメートせずにプログラムが実行されます。その結果、Web ブラウザにフォームが表示されます。フォームは初期状態になっています (前項の出力データが残ったままのフォームは、置換されます)。

[実行] は、IDE でアプリケーションを実行するための機能です。この機能は、デバッグ中にアニメートしないでブレークポイントまで実行する場合にも使用されます。サーバー側プログラムをデバッグするとき、ブレークポイントとして最初にプログラムへの再入を処理します。

4. Web ブラウザで、「Enter Name」フィールドをクリックし、自分の名前を入力して [クエリ送信] をクリックします。

最小化された IDE を元のサイズに戻し、再び myprog.cbl のソースをデバッグする準備が整います。

5. ツールバーの  を使用して、STOP RUN 文を実行する直前まで myprog.cbl をステップ実行します。

プログラムはそれほど長くないので、通常はステップ実行が終了するまで 1 分かかりません。埋め込み HTML (EXEC HTML 文) は数秒間で実行されます。


6. STOP RUN 文をステップ実行します。

IDE が最小化され、完成された Web フォームが表示されます。「Greeting」フィールドは更新されています。

7. 最小化された IDE を元のサイズに戻し、NetExpress の [アニメート] メニューで [アニメート停止] をクリックします。

## 9.4 次へ進む前に

プロジェクトを閉じます。

タスクバー トレイの Solo アイコン  を右クリックし、ポップアップ メニューで [終了] を選択して Solo を閉じます。

フォームを表示するために別の Web ブラウザを開いている場合は、それを閉じます。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままにしておいてかまいません。

「ここからチュートリアルをスタート」の章の「チュートリアル マップ」に戻り、目的に沿って次のデモ操作を選択します。

# 第10章 COBOL アプリケーションから Web アプリケーションへの変換

この章では、インターネット アプリケーション ウィザードを使用して、既存の COBOL アプリケーションを Web アプリケーションに変換します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了します。次に、「Web アプリケーションの概要」の章を読んでから、この章に進んでください。

---

## 10.1 概要

インターネット アプリケーション ウィザードを使用して、既存の COBOL プログラムの連絡節から、フォームとフォームを処理する COBOL プログラムを生成します。フォーム処理プログラムと元のプログラムは、ともにサーバー側プログラムです。

既存の COBOL プログラムとは、レガシー プログラム、または、この新規アプリケーションを作成する第一段階としてすでに作成したプログラムを指します。

既存のプログラムは、サブプログラムであることが必要です。このサブプログラムでは、ユーザー インターフェイス用のデータは連絡節で定義されます。通常、サブプログラムはファイル処理プログラムまたはデータ処理プログラムで、既存のアプリケーションの画面処理はこのプログラムを呼び出す別のプログラムによって実行されます。このプログラムは、サブプログラムにユーザー クエリーを渡し、ユーザーに応答を返します。新しい Web ベースのアプリケーションでは、生成されたフォームとプログラムが元の画面処理プログラムの役割を果たします。

インターネット アプリケーション ウィザードは、Windows GUI アプリケーションから Web アプリケーションへの変換には適していません。この作業については、オンライン マニュアル『移行ガイド』の「はじめに」の章を参照してください。

この章では、まず、既存の COBOL サブプログラムである `acctopen.cbl` を使用します。このプログラムは、COBOL 索引ファイルに保存されているアカウント データにアクセスします。変換されたアプリケーションを実行する場合、エンドユーザーでは 1 つのフィールドを持つ HTML フォームが表示されます。エンドユーザーは、アカウント番号を入力して [クエリーの送信] をクリックします。その結果、アカウントとそのアカウントに関する最新の注文情報を処理する営業担当者名を示すフォームが返されます。アプリケーションは、`d:\NetExpress\Base\Demo\Formx` のデータ ファイル `acctasc.vsm` と `acctasc.idx` で上記の情報を検索します。

## 10.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。「プロジェクト」ウィンドウや他のウィンドウが開いている場合は、閉じます。

最初に Web ブラウザや Solo が実行されていても問題はありません。

このマニュアルを Web ブラウザで参照している場合は、ブラウザを新しく開き (ブラウザで [ファイル] メニューの [新規作成] を選択します)、この章の作業中にもマニュアルを表示できるようにします。

## 10.3 具体的な手順

この手順では、次の操作を行います。

1. プロジェクト、フォーム、およびプログラムの作成
2. フィールド プロパティの追加
3. アプリケーションのビルド
4. アプリケーションの実行

### 10.3.1 プロジェクト、フォーム、およびプログラムの作成

まず、「*NetExpress の使用方法*」の章の説明にしたがい、まずプロジェクトを作成します。ただし、直接アプリケーションを作成する場合は、途中で NetExpress によりプロジェクトを作成する機会があります。アプリケーションは、次のように作成します。

1. [ファイル] メニューの [新規作成] をクリックしてから、「新規作成」ダイアログ ボックスの「インターネット アプリケーション」を選択し、[OK] をクリックします。

インターネット アプリケーション ウィザードの最初のページが表示されます。このページでは、「*Web アプリケーションの概要*」の章で説明する 3 種類のアプリケーション作成方法から使用する方法を 1 つ選択します。

2. 「アプリケーション全体 (HTML クライアントとサーバー プログラム)」を選択します。

ダイアログ ボックスの下部では、デフォルトの「COBOL ソース ファイル」が選択された状態になっています。

3. [次へ] をクリックします。
4. プロジェクトを作成するかどうかをたずねるメッセージに対して、[はい] をクリックします。
5. 「空プロジェクト」が選択されていることを確認してください (デフォルトでは、空プロジェクトが選択さ

れています)。プロジェクト名に Account を、プロジェクトを格納するフォルダに `d:\NetExpress\Base\Demo\Formx` を入力し、[作成] をクリックします。

このフォルダは、NetExpress のインストール時に作成されたものです。また、このチュートリアルで使用する COBOL プログラム `acctopen.cbl` は、このフォルダにインストールされます。このデモ操作を以前に実行している場合は、既存のプロジェクトに上書きするかどうかをたずねるメッセージが表示されるので、[はい] をクリックします。

ここで表示されるページでは、フォームとフォーム処理プログラムを生成する既存のプログラムを指定できます。

6. [参照] をクリックしてから `acctopen.cbl` をダブルクリックして、「ソース プログラム」フィールドを設定します。続いて [次へ] をクリックします。
7. 「個別入出力フォーム」を選択して、デフォルト設定の「クロスプラットフォーム(テーブル形式)」をそのまま使用し、[次へ] をクリックします。

このクロスプラットフォームを選択すると、NetExpress の位置フォームを使用してオブジェクトを確実に目的の場所に配置することができます。クロスプラットフォームでは、このために HTML テーブルを使用します。ダイナミック HTML では、スタイル属性を使用します。HTML テーブルはより多くのブラウザでサポートされています。ダイナミック HTML を使用すると、より正確に位置を指定できますが、これをサポートしているのは インターネット エクスプローラ 4.0 だけです。エンドユーザーがフォームの表示に使用するブラウザはさまざまであることに注意してください。

ここで表示されるページには、作成されるファイルが示されます。2 つは データを入力し結果を表示する HTML フォームです。もう 1 つは、フォームを処理し、`Acctopen` を呼び出す COBOL プログラムです。

8. [次へ] をクリックします。

### 10.3.2 フィールド プロパティの追加

ここで表示されるページには、元のプログラムのデータ項目が示され、右側の生成されるフォーム上には対応するフィールドが示されます。最初に表示されるのは、最高位の項目だけです。

1. まず、`LS-OLD-MF-REC` の横にある "+" をクリックし、次に、`OMF-ORDER` の横にある "+" をクリックします。

図 10-1 に示されるとおり、ツリー ビューが展開され `ls-old-mf-rec` に属するすべての項目が表示されます。



図 10-1 データ項目と対応するフィールド

左側のペインには、acctopen.cbl の連絡節で宣言されたデータ項目が表示されます。右側のペインには、インターネット アプリケーション ウィザードが生成するフォームに含まれることになる対応するフィールドが表示されます。「ラベル名」列には、フィールドごとに表示されるラベルが列挙されます。ラベル名は、宣言されたデータ名に設定されています。これらのラベルをよりわかりやすいラベル名に変更し、その他のプロパティを設定します。

- 右側のペインで、「ラベル名」フィールドの OMF-ACCTNO をクリックし、かわりに Account Number と入力します。[Enter] キーを押します。

設定するコントロールの種類は、デフォルトの Edit なので、[コントロールの種類] に属するエントリを変更する必要はありません。Edit は、このデータ項目がフォームのテキスト入力フィールドに表示されることを示します。

設定する I/O はデフォルトの Both なので、「I/O」に属するエントリを変更する必要はありません。この設定は、このデータ項目が入力を受け取り、アプリケーションのフォームにデータを送信することを示します。このアプリケーションは、個別の入力フォームと出力フォームを持つ非対称アプリケーションであるため、対応するフィールドは入力フォームと出力フォームの両方に表示されます。

- 右側のペインで、「ラベル名」フィールドの OMF-SALESREP をクリックし、かわりに Representative と

入力します。[Enter] キーを押します。

設定するコントロールの種類は、デフォルトの Edit なので、「コントロールの種類」に属するエントリを変更する必要はありません。

4. OMF-SALESREP (この段階では Representative に変更されています) の「I/O」フィールドをクリックして、ドロップダウン リストから Out を選択します。

この設定では、このデータ項目がデータをアプリケーションのフォームに送信します。このアプリケーションは、非対称アプリケーションであるため、対応するフィールドは出力フォームだけに表示されます。

5. 左側のペインの OMF-ORDER を右クリックし、ポップアップメニューで Out を選択します。

この方法でグループ フィールドのプロパティを設定すると、従属する項目が影響を受けます。右側のペインの「I/O」列が変更されて、これらの項目のプロパティが表示されます。

6. [次へ] をクリックします。

ここで表示されるページは、選択した項目をまとめたものです。

7. [完了] をクリックします。

インターネット アプリケーション ウィザードはフォームとプログラムを生成して、終了します。生成されたファイルの名前がプロジェクトに追加され、プロジェクトの従属関係が確認されます。その後で、インターネット アプリケーション ウィザードが閉じます。

### 10.3.3 生成されたファイル

これでユーザー インターフェイスが作成されました。インターネット アプリケーション ウィザードによって、次のファイルが自動生成され、プロジェクトに追加されています (すべてのファイルを表示するには、"+" 記号をクリックして「プロジェクト」ビューのツリー ビューを展開する必要がある場合があります)。

- acctopen\_input.htm - 入力フォームです。
- acctopen\_output.htm - 出力フォームです。
- acctopen\_server.cbl - コンパイルされ、acctopen.cbl にリンクされるとサーバー側プログラムになります。

次のファイルは、右側のペインにだけ表示されます。

- acctopen\_input.mff - サーバー側プログラムの生成に必要な追加情報が含まれます。
- acctopen\_output.mff - サーバー側プログラムの生成に必要な追加情報が含まれます。

元のプログラム acctopen.cbl もプロジェクトに追加されます。



また、プロジェクトをビルドするときに作成される次のオブジェクト ファイルもプロジェクトに追加されています。

- acctopen\_server.obj - コンパイルされたフォーム処理プログラムです。
- acctopen\_server.exe - サーバー側プログラムの実行可能ファイルです。
- acctopen.obj - コンパイルされた元のプログラムです。

コピーファイルもいくつか作成されます (拡張子は .cp\* です)。その用途は、.cbl ファイルのコメントに記述されています。

ファイル名は前述のように小文字で表記されていますが、プロジェクト内では大文字で表記されるものもあります。ファイル名には大文字と小文字の区別はありません。

フォームを更新する場合は、「プロジェクト」ウィンドウで acctopen\_input.htm または acctopen\_output.htm をダブルクリックするだけで、Form Designer が開きます。更新したフォームを保存するときに、コピーファイルが再生成されるので、これらのコピーファイルを直接編集しないでください。ただし、ウィザードで生成された .cbl プログラムは編集できます。

### 10.3.4 アプリケーションのビルド

アプリケーションは、次のようにビルドします。

1. [プロジェクト] メニューの [すべてをリビルド] をクリックします。

NetExpress により、プログラムがコンパイルされ、実行可能ファイルがビルドされます。「出力」ウィンドウに "リビルド完了" というメッセージ が表示されるまで待機してから、先へ進みます。

### 10.3.5 アプリケーションの実行

アプリケーションは、次のように実行します。

1. NetExpress の [アニメート] メニューで [実行] をクリックします。

このアプリケーションでは個別の入力フォームと出力フォームが使用されるため、「アニメーションの起動」ダイアログ ボックスに、実行の開始場所として入力フォームが表示されます。

2. 「アニメーションの起動」ダイアログ ボックスの [OK] をクリックします。

この操作により、Solo と Web ブラウザが起動し、エンドユーザーがフォームへのリンクをクリックしたかのように入力フォーム acctopen\_input.htm がブラウザに読み込まれます。

Solo の実行に問題が生じた場合は、(最小化されている場合は、IDE を元のサイズに戻し、) NetExpress の [ヘルプ] メニューで [ヘルプ トピック] をクリックします。ヘルプの [キーワード] タブで「Solo トラブルシューティング」を参照します。

3. Web ブラウザで「Account Number」フィールドをクリックし、「A1」と入力してから [フォームの送信] をクリックします (このアプリケーションでは、大文字と小文字を区別するので、「a1」ではなく「A1」と入力してください)。このとき、IDE は自動的に最小化されます。

プログラムが実行されます。その結果、アカウント "A1" の営業担当者名と、そのアカウントに関する最新の注文情報が出力フォームに記載されます。このフォームは、Web ブラウザに表示されます。


4. 「Account Number」フィールドに "A1" から "A9" までの他の番号を入力してみてください。

ファイルには、A1 から A9 までのアカウント レコードがあります。既存のプログラム acctopen.cbl には、B1 などの他の番号を入力すると、「Representative」フィールドに "Not Found" と表示されるコードが記述されています。

5. 最小化された IDE を元のサイズに戻し、NetExpress の [アニメート] メニューで [アニメート停止] をクリックします。

## 10.4 次へ進む前に

プロジェクトを閉じます。

タスクバー トレイの Solo アイコン  を右クリックし、ポップアップ メニューの [終了] を選択して Solo を閉じます。

フォームを表示するために別の Web ブラウザを開いている場合は、それを閉じます。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままにしておいてかまいません。

「ここからチュートリアルをスタート」の章の「チュートリアル マップ」に戻り、目的に沿って次のデモ操作を選択します。

# 第11章 Web データベース アプリケーションの作成

インターネット アプリケーション ウィザードを使用して、既存のデータベースに対する Web インターフェイスを作成します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了します。次に、「Web アプリケーションの概要」の章を読んでから、この章へ進んでください。

---

## 11.1 概要

インターネット アプリケーション ウィザードを使用して、フォームとサーバー側プログラムを作成し、Web を経由して既存のデータベースにアクセスできるようにします。

データベースは、Open Database Connectivity (ODBC) をサポートするデータベース システムで作成されている必要があります。このようなシステムには、Oracle、Sybase、Informix、DB2、Microsoft SQL (「シークエル」と発音します) Server、Microsoft Access などの最も一般的なデータベースがあります。

インターネット アプリケーション ウィザードを使用すると、ユーザーがフォーム設計やプログラミングを行う必要はありません。ウィザードによってデータベース構造体が参照され、フォームとサーバー側プログラムが自動的に作成されます。ユーザーは、可能な操作に関するオプションを設定するだけです。

生成されたアプリケーションは、データベースへのクエリーが可能です。また、選択しているオプションによっては、データベースの更新もできます。元のデータベースのツールを使用して、データベースのクエリーと更新を行うこともできます。

NetExpress には、卸売り業者の顧客情報に関する Microsoft Access データベースをはじめとして、いくつかのデータベース例が含まれています。この卸売り業者の顧客情報に関するデータベースの顧客とは、小売りを指します。この章では、Web アプリケーションを作成してこのデータベースのクエリーと更新を行います。

このデータベースには、データソース名 (DSN) NetExpress Sample2 を使用してアクセスします。この名前は、Microsoft Access データベースの例 `d:\NetExpress\Base\Demo\SMPLDATA\access\sample.mdb` を表します。(ODBC と、DSN の割り当て方法の詳細については、ODBC Administrator のヘルプを参照してください。このヘルプには、Windows のデスクトップにあるコントロール パネルからアクセスできます。)

### 11.1.1 用語

この章では、データベースについて説明する場合にデータベース用語に従い、*行*と*列*という用語を使用します。Web ブラウザでフォームにデータが表示された場合は、対応する用語*レコード*と*フィールド*を使用します。

## 11.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。「プロジェクト」ウィンドウや「テキスト」ウィンドウが開いている場合は閉じます。

最初に Web ブラウザや Solo が実行されていても問題ありません。

このマニュアルを Web ブラウザで参照している場合は、ブラウザを新しく開き (ブラウザの [ファイル] メニューで [新規作成] を選択します)、この章の作業中にもマニュアルを表示できるようにします。

## 11.3 具体的な手順

この手順では、次の操作を行います。

1. プロジェクト、フォーム、およびプログラムの作成
2. アプリケーションのビルド
3. アプリケーションの実行
4. データベースのナビゲート
5. データベースの更新
6. データベースのフィルタ処理

### 11.3.1 プロジェクト、フォームおよびプログラムの作成

まず、「*NetExpress の使用方法*」の章の説明にしたがい、プロジェクトを作成します。ただし、直接アプリケーションを作成する場合は、途中で NetExpress によりプロジェクトを作成する機会があります。アプリケーションは、次のように作成します。

1. [ファイル] メニューの [新規作成] をクリックしてから、「新規作成」ダイアログ ボックスの「インターネット アプリケーション」を選択し、[OK] をクリックします。

インターネット アプリケーション ウィザードの最初のページが表示されます。このページでは、「*Web アプリケーションの概要*」の章で説明する 3 種類のアプリケーション作成方法から使用する方法を 1 つ選択します。

2. 「アプリケーション全体 (HTML クライアントとサーバー プログラム)」を選択し、ダイアログ ボックスの下部で「SQL データベース」を選択します。
3. [次へ] をクリックします。
4. プロジェクトを作成するかどうかをたずねるメッセージに対して、[はい] をクリックします。
5. 「空プロジェクト」が選択されていることを確認してください (デフォルトでは、空プロジェクトが選択されています)。プロジェクト名に Daw、プロジェクトを格納するフォルダ名に d:\NetExpress\Base\Demo\Daw と入力し、[作成] をクリックします。
6. このディレクトリを作成するかどうかをたずねるメッセージに対して、[はい] をクリックします (このデモ操作を以前に実行している場合は、かわりに既存のプロジェクトに上書きするかどうかをたずねるメッセージが表示されるので、[はい] をクリックします)。
7. 生成されるすべてのファイルの基本名として Retail を、すべてのフォームに表示されるタイトルに Retail Outlets を入力し、[次へ] をクリックします。

ここで表示されるページには、ユーザーのコンピュータに保存されたデータ ソースが一覧表示されます。右側には各種のタブがあります。必ず [クエリー] タブが最前列にくるようにしてください (デフォルトでは、[クエリー] タブが最前列にくるようにになっています)。

8. NetExpress Sample2 をダブルクリックします。

リストが展開され、このデータベースのテーブルが表示されます。

9. Customer をダブルクリックします。

リストが展開され、「Customer」という名のテーブルが表示されます。

SQL (「シーケル」と発音する) の SELECT 文が生成され、右側に表示されます。SQL は、データベース クエリー言語です。NetExpress には、OpenESQL という機能も含まれています。OpenESQL は、コンパイル プリプロセッサで、SQL を COBOL ソースに埋め込むことができます。インターネット アプリケーション ウィザードを使用すると SQL を作成できるので、ウィザードを使用するがぎり SQL の知識は必要ありません。

この段階では、文が未完成です。文は次の手順で完成し、インターネット アプリケーション ウィザードによって生成される COBOL ソース プログラムに埋め込まれる SQL 文の基礎として使用されます。

10. ツリー ビューの Customer を右クリックし、ポップアップ メニューの [すべてのカラムを選択] をクリックします。

すべての列名にチェックマークが表示され、「クエリー」フィールドの文が完成します。この文によって、「Customer」テーブルの列がすべて表示されます。

このクエリーは、COBOL プログラムに埋め込むために作成していますが、この段階で、次のように対話式に実行してみます。

11. [クエリーの実行] (「インターネット アプリケーション ウィザード」ダイアログ ボックスの左上付近) をクリックします。

[実行結果] タブが最前列に表示され、「Customer」テーブルの行と列がすべて表示されます。

ソース プログラムに埋め込むために作成中の SQL 文の実行内容は上記の操作内容と同じですが、実行時に、表示する行 (情報を表示する顧客) を選択する条件を指定できます。これらの文により、テーブルのデータがプログラムの作業場所節に読み込まれます。読み込まれたデータは、他の COBOL 文によって Web フォームに転記されます。

このダイアログ ボックスは、NetExpress の別の機能である OpenESQL アシスタントでも使用されます。OpenESQL アシスタントは、対話式に実行するための SQL クエリーや独自に作成した COBOL ソース プログラムに埋め込むための SQL クエリーを作成する場合に使用します。

ここで作成したクエリーより、さらに複雑なクエリーを作成する方法を学習するには、このマニュアルを読んだ後でオンライン マニュアル『データベース アクセス』の「OpenESQL アシスタント」のチュートリアルを読む必要があります。

12. [次へ] をクリックします。

ここで表示されるページでは、アプリケーションの Web フォームを指定します。

クロスプラットフォームを選択すると、NetExpress の位置フォームを使用してオブジェクトを確実に目的の場所に配置することができます。クロスプラットフォームでは、このために HTML テーブルを使用します。ダイナミック HTML では、スタイル属性を使用します。HTML テーブルはより多くのブラウザでサポートされています。ダイナミック HTML を使用すると、より正確に位置を指定できますが、これをサポートしているのは インターネット エクスプローラ 4.0 だけです。エンドユーザーがフォームの表示に使用するブラウザはさまざまであることに注意してください。この場合、この[クロスプラットフォーム(テーブル形式)]をデフォルトとして使用します。

標準フォームは、2 種類あります。単一レコード ビュー (単に「レコード ビュー」ともいう) フォームでは、テーブル 1 行分のデータが表示されます。この例では、1 件の顧客に関する詳細情報が表示されます。一方、テーブル ビュー フォームでは、複数行のデータが表形式で表示されます。ここでは、デフォルトのとおり、両方のフォームで表示するように設定します。

13. [次へ] をクリックします。

ここで表示されるページでは、エンドユーザーが行う操作をデータベースのクエリーだけに限定するか、または、更新も可能にするかを指定します。この手順では、デフォルトのとおり、ユーザーがデータベースを更新できるように設定します。

14. [次へ] をクリックします。

ここで表示されるページは、選択した項目をまとめたものです。

15. [完了] をクリックします。

インターネット アプリケーション ウィザードは、フォームとプログラムを生成して、終了します。生成されたファイルの名前がプロジェクトに追加され、プロジェクトの従属関係が確認されます。その後でインターネット アプリケーション ウィザードが閉じます。

### 11.3.2 作成されたファイル

これでユーザー インターフェイスが作成されました。インターネット アプリケーション ウィザードにより、次のファイルが自動作成され、プロジェクトに追加されています (すべてのファイルを表示するには、"+" 記号をクリックしてツリービューを展開する必要がある場合があります)。

- retailform.htm - HTML のレコード ビュー フォームです。
- retailform.cbl - このフォームのサーバー側プログラムです。
- この他に、テーブル ビュー フォームについても同じようなファイル セットがあります。これらのファイル名には、すべて基本名 `retaillist` が使用されています。

次のファイルは、右側のペインだけに表示されます。

- retailform.mff - サーバー側プログラムの生成に必要な追加情報が含まれています。
- retaillist.mff - サーバー側プログラムの生成に必要な追加情報が含まれています。

プロジェクトをビルドするときには作成される次のオブジェクト ファイルもプロジェクトに追加されています。

- retailform.obj - コンパイルされたサーバー側プログラムです。
- retailform.exe - 実行可能ファイルです。
- この他に、テーブル ビュー フォームについても同じようなファイル セットがあります。これらのファイル名には、すべて基本名 `retaillist` が使用されています。

また、コピーファイルもいくつか作成されます (拡張子は `.cp*` です)。その用途は、`.cbl` ファイルのコメントに記述されています。

上記のファイル名は小文字で表記されていますが、プロジェクトでは大文字で表記されるものもあります。ファイル名には大文字と小文字の区別はありません。

フォームを更新する場合は、「プロジェクト」ウィンドウで `retailform.htm` または `retaillist.htm` をダブルクリックするだけで、Form Designer が開きます。更新したフォームを保存するときに、コピーファイルが再生成されるので、

コピーファイルを直接編集しないでください。ただし、ウィザードで生成された .cbl プログラムは編集できます。

### 11.3.3 アプリケーションのビルド

アプリケーションのビルド方法は、次のとおりです。

1. [プロジェクト] メニューの [すべてをリビルド] をクリックします。

NetExpress は、プログラムを保存、コンパイルして、実行可能ファイルをビルドします。「出力」ウィンドウに "リビルド完了" というメッセージが表示されるまで待機してから、先へ進みます。

### 11.3.4 アプリケーションの実行

アプリケーションの実行方法は、次のとおりです。

1. NetExpress の [アニメート] メニューで [実行] をクリックします。
2. 「アニメーションの起動」ダイアログ ボックスで [OK] をクリックします。

この操作により Solo と Web ブラウザが起動し、プログラム Retailform が実行されます。プログラムによりフォーム (単一レコード ビュー フォーム) retailform.htm が表示されます。IDE は自動的に最小化されます。

まず、フォームには、テーブルの 1 行目にある CustID = ALWAO を持つ顧客の詳細情報が表示されます。

Web ブラウザのフォント サイズによっては、レコード ビュー フォームに表示される 5 文字の顧客 ID を確認できないことがあります。たとえば、ALWAO は ALWA、MERRG、または MERR と表示されることがあります。「CustID」フィールドをクリックして左右の矢印キーを使用し、フィールドを横にスクロールすると、完全な顧客 ID を参照できます。

3. フォーム右上の [テーブル ビュー] をクリックします。

テーブル ビュー フォーム retailist.htm が、ブラウザに読み込まれます。このフォームには、テーブルにある最初の 10 件の顧客 ALWAO から CONSH ままで表示されます。

4. フォーム右上の [レコード ビュー] をクリックします。

単一レコード ビュー フォーム retailform.htm が再度読み込まれます。

### 11.3.5 データベースのナビゲート

両方のフォームには、テーブル内を前後に移動する標準コントロールがあります。レコード ビュー フォーム retailform.htm には、テーブルの更新とクエリーを行う機能もあります。機能によっては、フォームを作成するときに選択できるものもあります。この手順では、これらをすべて選択するためにすべてデフォルト値を使用します。



この項では、これら標準コントロールをいくつか使用します。単一レコード ビュー フォームが表示されていることを確認してください。デフォルトでは、「ソート順」フィールドは「CustID」に設定されています。この設定では、テーブルの行が CustID 順に表示されます。

表示されたフォームを正しく参照するために、Web ブラウザを全画面表示する必要がある場合があります。

1. フォーム上部の [ > ] をクリックします。

フォームには、テーブルの次の行にある CustID = ANDRC を持つ顧客が表示されます。

2. [テーブル ビュー] をクリックします。

テーブル ビュー フォームが読み込まれ、顧客 ANDRC を含めて上から 10 件の顧客が表示されます。

3. [ >> ] をクリックします。

フォームには、テーブルの末尾から 10 件、つまり TOPOF から WITAE までの顧客が表示されます。

4. [ > ] をクリックします。

フォームは再表示されますが、変更されていません。テーブル ビューでは、テーブルの末尾まで進むとそれ以上先には移動できなくなります。

5. [ << ] をクリックします。

フォームには、テーブルの先頭、つまり上から 10 件の顧客 ALWAO から CONSH までが表示されます。

6. [ > ] をクリックします。

フォームには、次の 10 件の顧客 EASTC から HANOP までが表示されます。

7. [レコード ビュー] をクリックします。

レコード ビューが読み込まれ、顧客 EASTC が表示されます。

8. [ >> ] をクリックします。

フォームには、テーブル末尾の顧客 WITAE が表示されます。

9. [ > ] をクリックします。

フォームには、テーブルの先頭の顧客 ALWAO が表示されます。レコード ビューでは、テーブルの末尾より先に進むとすると、テーブルの先頭に戻ります。

10. 「CustID」フィールドで、"BE" と入力して現在入力されている ALWAO と置換し、[クエリー] をクリックします。

フォームには、CustID をアルファベット順に整列したときに、「CustID」フィールドに入力した文字以降に最初に登場する CustID を持つ顧客が表示されます。この場合、顧客 BERGS が表示されます。

11. [テーブル ビュー] をクリックします。

テーブル ビューが読み込まれ、BERGS で始まる顧客が 10 件分表示されます。

12. 顧客 EASTC の最初の列にある [EASTC] をクリックします。

顧客 EASTC の単一レコード フォームが表示されます。

### 11.3.6 データベースの更新

このデモ操作では、アプリケーションの作成時に、エンドユーザーによるデータベース更新を可能にするデフォルト設定を使用しました。この手順では、単一レコード ビューを使用して、レコードの挿入、削除、または更新を行います。

1. 「Address」フィールドで、"35 King George" の後に "Street" を追加し、フォームの下部にある [レコードの更新] をクリックします。

レコードが表示されます。

2. [テーブル ビュー] をクリックします。

テーブル ビューが再び読み込まれ、データベースが変更を反映して更新されたことを確認できます。

### 11.3.7 データベースのフィルタ処理

レコード ビュー フォームの最上部にあるフィールドを使用すると、表示しないレコードをフィルタ処理したり、フィールドをレコード整列条件として指定したりできます。

カナダの顧客だけを表示するには、次のように都市順に表示します。

1. [レコード ビュー] をクリックします。

レコード ビュー フォームが表示されます。[画面をクリア] をクリックすると、フィルタを設定する前に顧客データ フィールドを削除できます。ただし、表示されるのは指定したフィールドだけなので、この処理は不要です。

2. ドロップダウン リストを使用して、「ソート順」フィールドを「City」に、「フィルタタイプ」フィールドを「=」（デフォルト）に、「フィルタ」フィールドを「Country」に設定します。

3. 「Country」フィールドに Canada と入力して現在の入力内容と置換し、[<<] をクリックします。

フォームには、都市順に整列したテーブルの先頭にあるカナダの顧客が表示されます。この例では、East

Vancouver の MERRG が表示されます。

「フィルタ」は自動的に「(existing)」に変更されています。これは、次のいくつかの手順で（このフィールドを変更するまで）選択済みの行セットを確実に使用するためです。これらの手順ではテーブル内の位置だけを変更し、新しいフィルタを設定しません。

4. 「City」フィールドに T と入力して現在の入力内容と置換し、[クエリー] をクリックします。

フォームには、カナダの顧客のうち、都市名が都市順で "T" 以降にある最初の顧客が表示されます。この例では、Tsawassen の BOTTM が表示されます。

5. [テーブル ビュー] をクリックします。

テーブル ビュー フォームが読み込まれます。都市順で Tsawassen 以降のカナダの顧客が表示されます。この例では、3 件の顧客が表示されます。

ここで新しいフィルタを設定します。フィルタを設定するには、「フィルタ」フィールドを「(existing)」から指定するフィールド名に変更します。

6. [レコード ビュー] をクリックします。

7. 「フィルタ」フィールドを Country に変更し、「Country」フィールドに UK と入力して [<<] をクリックします。

フォームには、イギリスの顧客のうち、都市順で最初にある顧客 Hedge End の ISLAT が表示されます。

8. [テーブル ビュー] をクリックします。

テーブル ビュー フォームが読み込まれます。フォームには、イギリスの顧客のうち、都市順で上から 10 件分顧客が表示されます。

ここでテーブル全体を元に戻し、再び顧客 ID 順に整列させます。

9. [レコード ビュー] をクリックします。

10. 「ソート順」フィールドを CustID に、「フィルタ」フィールドを (none) にそれぞれ変更し、 [<<] をクリックします。

フォームが再表示され、顧客 ID 順でテーブルの先頭にある顧客が表示されます。

11. [テーブル ビュー] をクリックします。


テーブル ビュー フォームが読み込まれ、上から 10 件の顧客が表示されます。この段階では、テーブル全体が顧客 ID 順で再表示されています。

12. 最小化された IDE を元のサイズに戻し、NetExpress の [アニメート] メニューで [アニメート停止] をク

リックします。

## 11.4 次へ進む前に

プロジェクトを閉じます。

タスクバー トレイの Solo アイコン  を右クリックし、ポップアップ メニューで [終了] を選択して Solo を閉じます。

フォームを表示するために別の Web ブラウザを開いている場合は、それを閉じます。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままにしておいてかまいません。

「ここからチュートリアルをスタート」の章の チュートリアル マップに戻り、目的に沿って次のデモ操作を選択します。

# 第12章 Windows GUI アプリケーションの作成

この章と次の章では、Windows のグラフィカル アプリケーションを作成します。

この章では、Windows GUI アプリケーション ウィザードと Dialog System を使用して、ユーザー インターフェイスとデータを処理するプログラムを作成します。また、IDE を使用してアプリケーションを実行し、インターフェイスの表示状態を確認します。

次の章「Windows GUI アプリケーションの完成と実行」では、プログラムにビジネス ロジックを追加し、完成したアプリケーションを実行します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了してから、この章へ進んでください。

---

## 12.1 概要

Windows GUI アプリケーション ウィザードを使用して、Windows システムで表示するウィンドウとダイアログ ボックスを作成します。これらは、アプリケーションのユーザー インターフェイスになります。このウィザードから、Dialog System に移動し、レイアウトの設計や編集を行います。設計するウィンドウとダイアログ ボックスのセットを、スクリーンセットと呼びます。Dialog System により、作成したウィンドウとダイアログ ボックスを定義するスクリーンセット ファイルが生成されます。

また、Dialog System を使用して、スクリーンセットを処理する COBOL プログラムを生成することもできます。これは、関連付けられたプログラムといます。

### 12.1.1 スクリーンセットの作成

スクリーンセットを作成するには、次の 4 つの手順にしたがいます。

1. ウィンドウとダイアログ ボックスを定義します。また、ステータス バーやツールバーのようなウィンドウとダイアログ ボックスのプロパティを定義します。
2. ユーザー インターフェイスと COBOL プログラムの間で受け渡されるデータを格納するデータ項目を定義します。
3. ウィンドウとダイアログ ボックスに表示されるコントロール (エントリ フィールドやプッシュボタンなど) を定義します。

4. プッシュボタンのクリックなど、ユーザーがインターフェイスで何かアクションを起こした場合に発生するイベントを定義するためのダイアログを作成します。

手順 4 に示されるダイアログは、Dialog System 独自のスクリプト言語で作成します。

通常、Dialog System に関するマニュアルでは、コントロールのことをグラフィカル オブジェクト、または単にオブジェクトと呼びます。

## 12.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。「プロジェクト」ウィンドウや他のウィンドウが開いている場合は、閉じます。

## 12.3 具体的な手順

この手順では、次の操作を行います。

1. プロジェクトとスクリーンセットの作成
2. スクリーンセットのデータ項目の定義
3. スクリーンセットへのコントロールの追加
4. スクリーンセットへのダイアログの追加
5. アプリケーションのビルド
6. スクリーンセットのテスト

### 12.3.1 プロジェクトとスクリーンセットの作成

まず、「*NetExpress の使用方法*」の章の説明にしたがい、プロジェクトを作成します。ただし、直接スクリーンセットを作成する場合は、途中で NetExpress によりプロジェクトを作成する機会があります。スクリーンセットは、次のように作成します。

1. [ファイル] メニューの [新規作成] をクリックしてから、「新規作成」ダイアログ ボックスの「Dialog System スクリーンセット」を選択し、[OK] をクリックします。
2. プロジェクトを作成するかどうかをたずねるメッセージに対して、[はい] をクリックします。
3. 「Windows GUI プロジェクト」を選択します。プロジェクト名として Welcome を、プロジェクトを格納するフォルダとして `d:\NetExpress\Base\Demo\Welcome` をそれぞれ入力してから、[作成] をクリックします。
4. 「Window GUIアプリケーションの新規作成ウィザード—スクリーンセット名」ウィンドウが表示されます。

(このデモ操作を以前に実行している場合は、既存のプロジェクトに上書きするかどうかをたずねるメッセージが表示されるので、[はい] をクリックします。)

5. デフォルトのスクリーンセット名 NewSet.gs に Welcome.gs を上書きし (パス名は変更しません)、[次へ] をクリックします。(このデモ操作を以前に実行している場合は、既存の Welcome.gs を上書きするかどうかをたずねるメッセージが表示されます。[はい] をクリックします。)
6. フレームベースのMDI アプリケーションの作成については、[はい] をクリックし、MDIの子ウィンドウの個数として 2 を入力します。[次へ] をクリックします。

Single Document Interface (SDI) は、1 つのウィンドウだけで構成されます。MDI は、1 つのウィンドウが親になり、多くのウィンドウやダイアログ ボックスで構成されます。

7. 「ステータス バー」と「主ウィンドウのツールバー」を選択し、[次へ] をクリックします。

この設定により、メイン ウィンドウにステータス バーとツールバー、さらにメニュー バーが作成されます。ツールバーとメニュー バーは一緒に作成されます。

8. [次へ] を 2 回クリックします。

スキップした 2 つのダイアログ ボックスでは、Dialog System の拡張機能を使用したり、実行時の動作を変更したりできます。ここでは、デフォルト設定を使用します。

9. 「COBOL スケルトン プログラムを生成する」がチェックされていることと、生成する COBOL プログラムの名前として Welcome.cbl が指定されていることを確認して、[次へ] をクリックします。

次に表示されるダイアログ ボックスは、選択した項目をまとめたものです。

10. [完了] をクリックします。

Windows GUI アプリケーション ウィザードはスクリーンセットとプログラムを生成して、終了します。生成されたファイルの名前がプロジェクトに追加され、プロジェクトで従属関係が確認されます。

次に、Dialog System が読み込まれます。

### 12.3.2 この段階の画面表示

この段階では、画面は図 12-1 のように表示されます。3 つの各ウィンドウは、必ずしもこの位置に表示されるとは限りません。ウィンドウを画面で移動させ、すべてのウィンドウを見やすく表示することもできます。

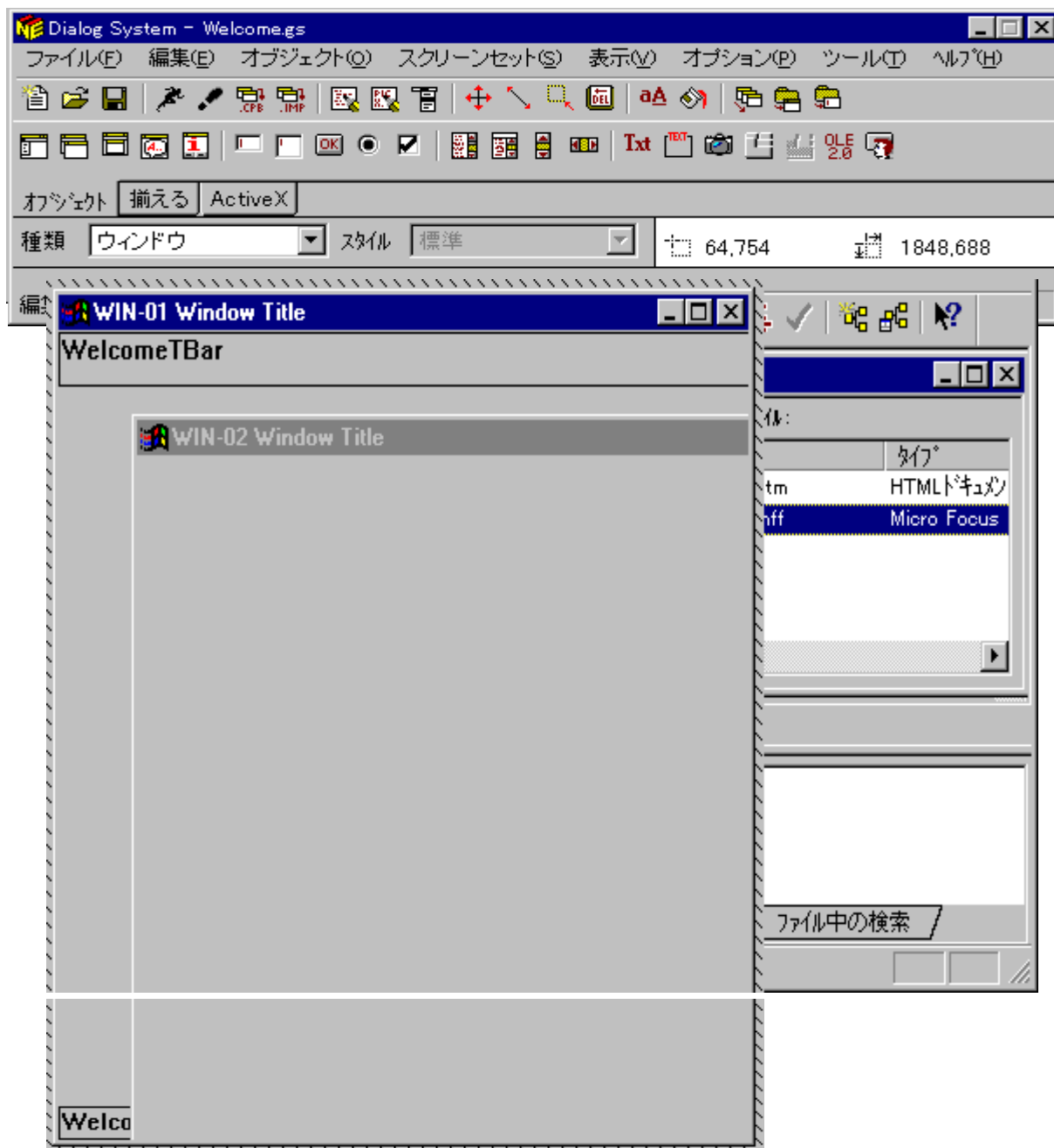


図 12-1 Dialog System が読み込まれた画面

画面には、次の 2 種類の新しいウィンドウが表示されます。

- WIN-01

「WIN-01 ウィンドウ タイトル」というタイトルのウィンドウは、スクリーンセットを設計するウィンドウです。このウィンドウ「WIN-01」はスクリーンセットのメイン ウィンドウです。スクリーンセットを MDI として指定したので、子ウィンドウの「WIN-02」は、ウィンドウ「WIN-01」の内部に表示されます。



- Dialog System のウィンドウ

このウィンドウは、メニュー バー、ツールバーおよびステータス バーで構成されます。上のウィンドウのタイトルは "Dialog System - Welcome.gs" で、下のウィンドウのタイトルは "編集 WIN-01" です。最初は、画面で「WIN-01」と重なっている場合があります。

プロジェクトの変更内容を確認するために、「プロジェクト」ウィンドウを前景に移動することができます（ウィンドウをクリックします）。その後で、Dialog System と WIN-01 をクリックして、前景に移動させます。

### 12.3.3 スクリーンセット用データの定義

エン트리 フィールドのようにデータを格納する画面上のオブジェクトは、データ項目に関連付ける必要があります。データは、これらのデータ項目と画面上のオブジェクトとの間で受け渡されます。

このチュートリアルでは、まずデータ項目を定義し、後の項でオブジェクトを定義します。必ずしもこのような方法を使用する必要はありません。後で示すように、データがない状態でもスクリーンセットの表示状態と動作をテストすることができます。ただし、一般的にまずデータを定義する方が便利です。データが定義されていると、エンドユーザーとの間で受け渡しする必要がある情報を明確に判断できます。データ定義を変更せずに、スクリーンセットの表示状態を変更することも可能です。

定義するデータ項目は、データ ブロックと呼ばれます。生成されたプログラムにより Dialog System のランタイム サポート モジュール DSGRUN が呼び出されると、データ ブロックがパラメータとして渡されます。データは、COBOL プログラムとスクリーンセットの間でこのように受け渡しされます。

データ項目を定義するには、「Dialog System」ウィンドウを使用します。ここでは、「WIN-01」は使用しません。

Dialog System には、データ フィールド定義の入力モードが 2 種類あります。プロンプト モードでは、表示される選択肢リストとダイアログ ボックスを使用してデータ フィールドを定義します。このモードは、Dialog System で使用するデータ宣言の構文に慣れていない場合に便利です。非プロンプト モード（編集モードとも呼ばれます）では、宣言を直接入力します。この手順では、非プロンプト モードを使用します。

1. Dialog System の [スクリーンセット] メニューで [データ ブロック] をクリックします。

「Dialog System:データの定義」ウィンドウが表示されます。このような特殊な編集ウィンドウは、Dialog System で異なる種類の情報を入力する場合に常に使用されます。これには固有のメニューがあります。次の手順では、このメニューを使用します。

データ宣言がすでに存在しているのがわかります。これらのデータ宣言は、スクリーンセットの作成時に自動的に生成されたものです。これらデータ宣言は、MDI、ツールバー、メニュー バー、およびステータス バーの維持に使用されますが、無視してかまいません。

2. 「Dialog System:データ定義」ウィンドウの [オプション] メニューで「プロンプト モード」にチェックマークが付いていないことを確認します（付いている場合は、クリックします）。

- 「Dialog System:データ定義」ウィンドウをクリックして選択し、既存の項目の終わりまで表示して、[Enter] キーを押します。
- I-NAME X 12.0 と入力してから [Enter] キーを押します。

行は、自動的に書式設定されます。小文字で入力した場合でも、自動的に大文字に変換されます。ここま  
 でで、最初のデータ項目 I-NAME が定義されています。X は、英数字を表し、12.0 はデータの長さが 12 文  
 字で小数点以下を空白文字にしないことを表します。フィールドが英数字であっても、".0" は必要です。  
 ただし、0 を入力しなくても自動的に挿入されるので、省略してもかまいません。

これらのデータ項目は生成された COBOL プログラムからも、Dialog System からもアクセスできるため、  
 COBOL 予約語は使用しないでください (使用しても警告は表示されません)。データ項目名を NAME で  
 はなく I-NAME としたのは、このためです。

- GREETING X 22.0 と入力し、[Enter] キーを押します。

これは、2 番目のデータ項目 GREETING になります。データは 22 文字の英数字になります。

この段階では、「Dialog System:データ定義」ウィンドウは、図 12-2 のように表示されます。



図 12-2 データ宣言が入力された「Dialog System:データ定義」ウィンドウ

- 「データ定義」ウィンドウの [編集] メニューで [データの定義の終了] をクリックします。

データ定義は、画面に表示されませんが、メモリに読み込まれた状態になっています。後で [上書き保存] を  
 実行してスクリーンセットを保存するまで、ディスクには保存されません。

以前、Dialog System によりスクリーンセットと COBOL プログラムが生成されたときに、データ ブロッ  
 クの定義を含む COBOL コピーファイル (.cpb) が生成されています。このコピーファイルは、生成された  
 プログラムの作業場所節で使用されます。作業場所節には、データが実際に格納されるます。データ プロ

ックを変更しているため、このコピーファイルを再作成する必要があります。

- Dialog System の [ファイル] メニューで [生成] をクリックし、[データブロックCopyファイル] を選択して、デフォルトの Welcome.cpb をコピーファイル名として使用します。次に [保存] をクリックします。[はい] をクリックして既存のコピーファイルに上書きすることを確認します。

データ ブロックを変更する場合は、必ずこのコピーファイルを再作成してください。

次の項に進む前に、これまでの作業データを損失した場合に備え、スクリーンセットを保存します。

- Dialog System の [ファイル] メニューで [上書き保存] をクリックします。

### 12.3.4 スクリーンセットへのコントロールの追加

次に、ユーザー インターフェイスの表示内容を指定します。ステータス バー、ツールバーおよびメニュー バーは、スクリーンセットを作成したときにすでに追加してありますが、ここで、テキスト フィールドとエントリ フィールドを追加する必要があります。Dialog System では、このようなコントロールを、通常「オブジェクト」と呼びます。

画面に現在表示されているウィンドウ「WIN-01」は、スクリーンセットの親、またはメイン ウィンドウです。最上部の四角形の枠には Welcome TBar と表示され、メニューバーとツールバーが配置されることを示しています。また、最下部の四角形の枠には Welcome SBar と表示され、ステータスバーが配置されることを示しています。

MDI では、親ウィンドウは子ウィンドウを表示するフレームにすぎません。そのため、親ウィンドウにオブジェクトを配置することはできません。エンドユーザーが実際に使用するのは子ウィンドウです。ウィンドウの作成方法を説明するために 2 つの子ウィンドウを持つ MDI を作成しましたが、使用するのはこのうちの 1 つだけです。


[オブジェクト] メニューまたはオブジェクト ツールバーを使用してオブジェクトを追加します。オブジェクト ツールバーは、マウス ユーザーにとっては高速で簡単です。一方、メニューは、キーボード ユーザーによって使用されます。これらの実行内容はまったく同じです。この章ではオブジェクト ツールバーを使用します。

Dialog System では、初期状態のウィンドウ サイズはすべて同じになります。まず、ウィンドウのサイズを変更して、「WIN-01」内で「WIN-02」全体が表示されるようにします。

- 「WIN-02」ウィンドウの右下端が表示されるように、画面で「WIN-01」ウィンドウを上に移動させ、右下端をドラッグして拡大します。次に、「WIN-02」ウィンドウの右下端をドラッグして「WIN-02」を縮小し、Welcome TBar と Welcome SBar が表示された 2 つの四角形の間に収まるようにします。「WIN-01」ウィンドウを元のサイズに戻して、これら 2 つの四角形を囲むようにします。
- 「WIN-02」ウィンドウ内をクリックして、確実に選択し、Dialog System の [編集] メニューで [属性] をクリックします (その他の方法としては、「WIN-02」ウィンドウを右クリックしてポップアップ メニューの [属性] を選択するか、単に「WIN-02」ウィンドウをダブルクリックします)。

「WIN-02」ウィンドウの属性を編集するための、「ウィンドウの属性」ダイアログ ボックスが表示されま  
す。

3. デフォルトのタイトル WIN-02 ウィンドウ タイトルに Welcome を上書きします。それ以外は、デフォ  
ルト設定を使用し、[OK] をクリックします。

4. オブジェクト ツールバーの [入力フィールド] ボタン  をクリックし、「WIN-02」ウィンドウをクリッ  
クします。

これで、クリックした位置に入力フィールドが作成されます。入力フィールドとは、エンドユーザーがデ  
ータの入力や表示を行うフィールドのことです。フィールドを配置すると、マウスでドラッグして移動さ  
せることができます。フィールドをウィンドウの右側に移します。

5. 入力フィールドをダブルクリックします。

入力フィールドのプロパティを定義するための「入力フィールドの属性」ダイアログ ボックスが表示され  
ます。

6. デフォルト名の EF1 に I-NAME-DISP を上書きします。

入力フィールド名は、I-NAME-DISP になります。


7. [マスタ フィールド] をクリックします。

「マスタ フィールド」ダイアログ ボックスが表示されます。このダイアログ ボックスには、デー  
タ ブロックのフィールドが一覧表示され、適切なデータ フィールドを入力フィールドにリンクできます。

8. 「マスタ フィールド」ダイアログ ボックスで、フィールド「I-NAME」を表示し、これをクリックします。  
[OK] をクリックします。

「マスタ フィールド」フィールドに I-NAME が表示され、「PICTURE文字列」フィールドが自動的に X(12)  
に設定されます。これは、入力フィールド「I-NAME-DISP」が 12 文字の英数字で、データ項目 I-NAME に  
格納されたデータを表示することを意味します。

9. 残りの属性については、デフォルト設定を使用し、[OK] をクリックします。


10. オブジェクト ツールバーの [テキスト] ボタン  をクリックし、入力フィールドの左側をクリックして  
入力フィールドのラベルを作成します。

11. ラベルをダブルクリックして、「テキストの属性」ダイアログ ボックスを表示します。

12. デフォルトのテキスト「テキスト」に「Enter Name」を上書き入力してから、[OK] をクリックします。

13. 同様に、2 つ目の入力フィールドとラベルを追加します。フィールド名に "GREETING-DISP" を使用し、

そのマスタ フィールドに GREETING を選択します。ラベルを「Greeting」に設定します。

- オブジェクト ツールバーの [OK] ボタン  をクリックし、「WIN-02」ウィンドウをクリックしてボタンを配置します。ボタンは、ウィンドウの下部に置きます。
- ボタンをダブルクリックします。

ボタンの属性を定義するための、「プッシュボタンの属性」ダイアログ ボックスが表示されます。

- デフォルト名 PB1 に START-BUTTON を上書き入力し、デフォルトのテキスト ~Pushbutton に Start を上書き入力します。残りのプロパティについては、デフォルト設定を使用し、[OK] をクリックします。

この段階で、スクリーンセットは、図 12-3 のように表示されます。

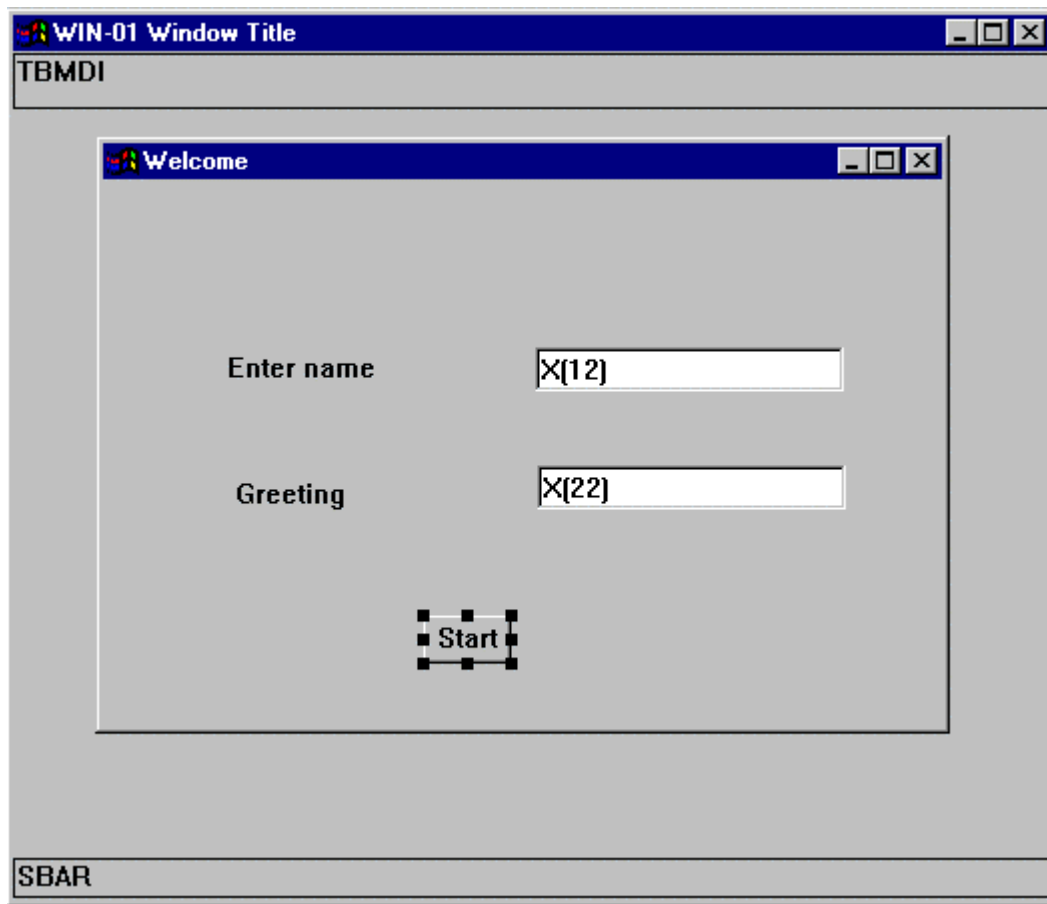


図 12-3 入力フィールド、テキスト フィールドおよびプッシュボタンを持つスクリーンセット

次の項に進む前に、これまでの作業データを損失した場合に備え、スクリーンセットを保存します。

- Dialog System の [ファイル] メニューで [上書き保存] をクリックします。

### 12.3.5 スクリーンセットへのダイアログの追加

ここまでで、ウィンドウの表示内容と、ウィンドウのフィールドから受け取るデータ（または、フィールドで表示するデータ）を格納するデータ項目を定義しました。ただし、ユーザーがウィンドウを使用しても（たとえば、[Start] ボタンをクリックしても）、何も起こりません。通常ダイアログと呼ばれるダイアログ スクリプトを作成し、ユーザー アクションの効果を定義します。

キーを押したり、マウス ボタンをクリックするなどのユーザー アクションによって、イベントが生成されます。各イベントにตอบสนองして実行されるプロシージャを作成します。

ダイアログは、コントロールやウィンドウなどの特定のオブジェクトに結合することも、全体的に適用することもできます。このような適用範囲を、ダイアログの **範囲** といいます。オブジェクトに結合されたダイアログは、ローカルといいます。オブジェクトでイベントが発生した場合、Dialog System は、まずそのオブジェクトに結合されたダイアログを探します。このようなダイアログがない場合、Dialog System は、オブジェクトが属するウィンドウに結合されたダイアログを探します。また、このようなダイアログが検出されない場合、Dialog System は、グローバルなダイアログを検索します。

ユーザーが [Start] ボタンをクリックしたときに、DSGRUN が呼び出しプログラムに戻るようにダイアログを作成する必要があります。ユーザーが入力したデータをまずデータ ブロックのフィールドに転記することが必要ですが、以前にマスタ フィールドの関連付けを定義しているため、この処理は自動的に行われます。そのため、この処理に関するダイアログを作成する必要はありません。

Dialog System では、データの定義とともに、プロンプト モードまたは非プロンプト モードの選択が可能です。ここでは、非プロンプト モードを使用します。

#### 12.3.5.1 グローバル ダイアログ

この例では、グローバル ダイアログを作成する必要はありませんが、自動生成されたダイアログを参照すると役立つ場合があります。

1. Dialog System の [スクリーンセット] メニューで [グローバル ダイアログ] をクリックします。

「ダイアログの定義：グローバル」ウィンドウが表示されます。すでに存在しているダイアログを参照することができます。これらのダイアログは、ウィザードで選択された実行時のデフォルト動作を実装するために、スクリーンセットの作成時に自動生成されたものです。

最初のプロシージャは、イベント ESC と 2 つの関数 MOVE と RETC で構成されます。このプロシージャでは、イベント ESC が発生すると、DEGRUN がデータ項目 EXIT-FLAG に 1 を転記して呼び出し側プログラムに戻ります。イベント ESC は、ユーザーが [Esc] キーを押すと生成されます。

このダイアログはグローバルなので、[Esc] キーを押すたびにこのプロシージャが実行されます。これは、オブジェクト、またはオブジェクトがあるウィンドウに ESC 用の固有のダイアログがない限り、現在フォ

ーカスを持っているオブジェクトとは関係なく適用されます。

2 番目のプロシージャでは、ユーザーがウィンドウを閉じるときに生成されるイベント CLOSED-WINDOW と同じ処理を行うように指定します。

このダイアログの残りの部分で、MDI、メニュー バー、ツールバー、およびステータス バーを維持します。このダイアログは無視してかまいません。

Dialog System のプロシージャを、ダイアログ テーブルと呼びます。ダイアログ テーブルは、イベントやプロシージャ名とこれに続く関数で構成されます。これらの関数は、イベントが発生したり、このプロシージャ名が別のプロシージャから明示的に起動された場合に実行されます。Dialog System のイベントや関数の完全なリストは、ヘルプに記載されています。

プロシージャが表示される順序は重要ではありません。COBOL の場合とは異なり、制御に失敗して、あるプロシージャの終わりから次のプロシージャが実行されることはありません。DSGRUN によりプロシージャの終わりで停止し、次のイベントが検索されます (ただし、最後の関数が、呼び出しプログラムに戻る RETC である場合は除きます)。

2. 「ダイアログの定義」ウィンドウの [編集] メニューで [ダイアログの定義の終了] をクリックします。

### 12.3.5.2 ローカル ダイアログ

ダイアログを作成し [Start] ボタンを機能させます。ダイアログを適用するオブジェクトはこれ 1 つだけなので、このダイアログをこのオブジェクトに結合されたローカル ダイアログにします。

1. 「Welcome」ウィンドウで [Start] ボタンを右クリックしてから、ポップアップ メニューの [ダイアログ] をクリックします。

「ダイアログの定義」ウィンドウが表示されます。この種のオブジェクトでよくダイアログが作成されるイベントの名前が挿入され、Dialog System でダイアログが作成されています。この場合、エンドユーザーがこのボタンをクリックすると生成される BUTTON-SELECTED を挿入する必要があります。

強調表示は、すでにイベントの下の方に配置され、すぐに入力できるようになっています。

2. 「\* [Start] ボタンで呼び出しプログラムに戻ります。」 と入力してから、[Enter] キーを押します。

行のアスタリスク以降の部分は、すべてコメントとして処理されます。

3. RETC と入力して [Enter] キーを押します。

行には、自動的に書式が設定されます。小文字で入力した場合でも、自動的に大文字に変換されます。Dialog System では、RETC は関数名として認識され、インデントが挿入されます。

4. REFRESH-OBJECT GREETING-DISP と入力して [Enter] キーを押します。

ダイアログは、次のように作成されているはずですが。

```
BUTTON-SELECTED
```

\* [Start] ボタンで呼び出しプログラムに戻ります。

```
RETC
```

```
REFRESH-OBJECT GREETING-DISP
```

RETC によって、DSGRUN が COBOL プログラムに返されます。DSGRUN が次に呼び出されたとき、DSGRUN は停止したところから実行されます。多くの場合、RETC は、プロシージャの最後の関数として配置されます。そのため、DSGRUN が再入したときにプロシージャが終了している場合、DSGRUN は次のイベントまで待機します。ただし、この例では、DSGRUN が再入したときに、何らかの処理を実行させます。たとえば、COBOL プログラムがマスタ フィールド GREETING に格納する更新データを使用して、GREETING-DISP を更新します。この処理は、REFRESH-OBJECT 関数により実行されます。

関数の構文は、Dialog System のヘルプ トピック「ダイアログ文：関数」で説明されています。

5. 「ダイアログの定義」ウィンドウの [編集] メニューで [ダイアログの定義の終了] をクリックします。
6. Dialog System の [ファイル] メニューで [上書き保存] をクリックします。

### 12.3.6 作成されたファイル

これでユーザー インターフェイスが作成されました。ウィザードと Dialog System により、次のファイルが自動作成され、プロジェクトに追加されています (すべてのファイルを表示するには、"+"記号をクリックし、「プロジェクト」ウィンドウのツリー ビューを展開して表示する必要がある場合があります)。

- welcome.gs - スクリーンセットです。
- welcome.cbl - 関連付けられたプログラムです。
- welcome.cpb - COBOL のデータ ブロックを定義するコピーファイルです。
- welcomeTBar.cbl - ツールバーとメニューを処理するプログラムです。
- welcomeSBar.cbl - ステータス バーを処理するプログラムです。

また、プロジェクトをビルドするときに作成される次のオブジェクト ファイルもプロジェクトに追加されています。

- welcome.int - コンパイル済みの関連付けられたプログラムです。
- welcomeTBar.int - ツールバーやメニューを処理するコンパイル済みのプログラムです。
- welcomeSBar.int - ステータス バーを処理するコンパイル済みのプログラムです。

コピーファイルもいくつか作成されます (拡張子は .cp\* です)。



上記のファイル名は小文字で表記されていますが、プロジェクトでは大文字で表記されるものもあります。ファイル名には大文字と小文字の区別はありません。

スクリーンセットを更新する必要がある場合は、「プロジェクト」ウィンドウの Welcome.gs をダブルクリックするだけで、Dialog System が開きます。更新されたスクリーンセットを保存するときに、コピーファイルが再生成されるので、コピーファイルを直接編集しないでください。ただし、ウィザードで生成された .cbl プログラムは編集できます。

### 12.3.7 アプリケーションのビルド

アプリケーションは、次のようにビルドします。

1. [プロジェクト] メニューの [すべてをリビルド] をクリックします。

NetExpress によってプログラムがコンパイルされ、実行可能ファイルがビルドされます。「出力」ウィンドウに "リビルド完了" というメッセージが表示されるまで待機してから、先に進みます。

### 12.3.8 スクリーンセットのテスト

スクリーンセットは、関連付けられたプログラムがなくても Dialog System で実行できます。そのため、スクリーンセットの動作の評価やテストを行い、必要な場合は、スクリーンセットの定義に戻って変更することができます。これは、インターフェイスのプロトタイプを作成する場合に役立ちます。

ツールバー、メニュー、ステータス バーなどのプログラムはテスト中に実行されるので、ビルドしてからテストを実行する必要があります。

1. Dialog System の [ファイル] メニューで [実行] をクリックします。

Dialog System ウィンドウと「WIN-01」ウィンドウは画面に表示されなくなりますが、「WIN-01」ウィンドウは実行時に表示されます。このウィンドウには、ウィザードで設定したツールバー、メニュー バーおよびステータス バーがあることに注意してください。「WIN-02」ウィンドウはその中に表示されています。

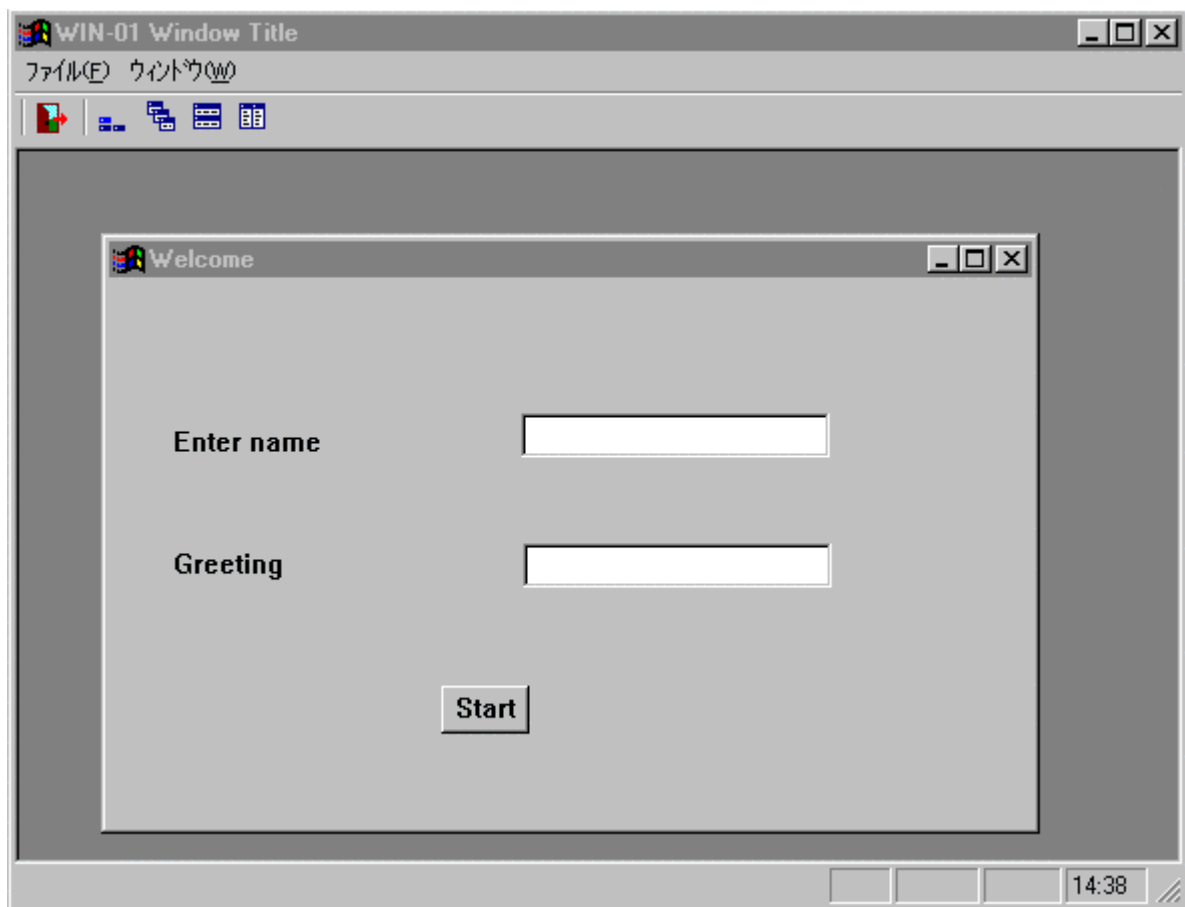


図 12-4 実行関数によって表示されるスクリーンセット

ウィザードで作成したツールバーとメニューバーは、生成されたコントロールプログラムによって、NetExpress のクラス ライブラリを使用して表示されます。詳細情報およびウィザードが作成したツールバーとメニューバーの修正方法については、『*Dialog System ユーザーガイド*』を参照してください。ウィンドウのメニューを独自に作成する場合は、メニュー作成に使用する Dialog System 固有の統合システムに関する追加オプションがあります。詳細は、『*Dialog System ユーザーガイド*』の「ウィンドウとオブジェクト」の章を参照してください。

画面の右下に [デバッグ] ボタンと [中止] ボタンを持つ小さなダイアログボックスが表示されます。Dialog System のこのスクリーンセット テスト機能を使用する場合、スクリーンセットはスクリーンセットアニメータによって制御されます。スクリーンセットアニメータは、ダイアログのデバッガで、IDE の COBOL デバッグ機能の概念と似ています。ダイアログボックスを使用すると、次の操作が可能です。

- デバッグ - シングル ステップ実行などのスクリーンセットアニメータの機能を使用できるようにダイアログを表示します。
- 中止 - スクリーンセットのテストを停止し、Dialog System に戻ります。

2. 定義した [Start] ボタンをクリックします。

図 12-5 のようにダイアログ ボックスが表示されます。

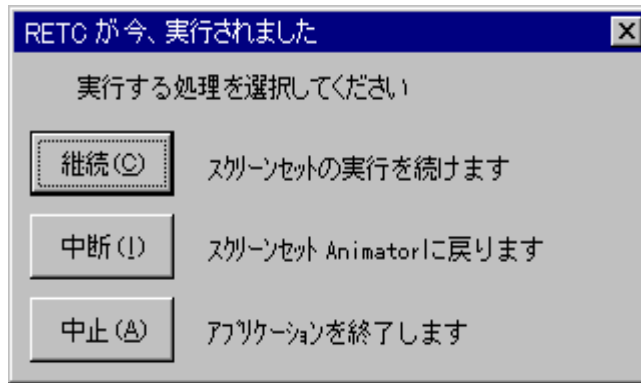


図 12-5 RETC ダイアログ ボックス

前述のとおり、[Start] をクリックすると、このボタン用に作成したダイアログが実行されます。このダイアログには、RETC コマンドが含まれていることも覚えておいてください。RETC が実行されるたびにスクリーンセット アニメータがこのダイアログ ボックスに変わるので、次に行う操作を指定する必要があります。

- 継続 - アニメーションなしでスクリーンセットのテストを続けます。
- 中断 - シングル ステップ実行などのスクリーンセット アニメータの機能を使用できるようにダイアログを表示します。
- 中止 - スクリーンセットのテストを停止し、Dialog System に戻ります。

アプリケーションでスクリーンセットを実行しているときにダイアログをデバッグする場合、スクリーンセット アニメータを有効化する DSGRUN のパラメータを使用します。スクリーンセット アニメータについては、Dialog System のヘルプで説明されています。

3. [中止] をクリックします。
4. Dialog System の [ファイル] メニューで [終了] をクリックします。

## 12.4 次へ進む前に

次の章「Windows GUI アプリケーションの完成と実行」では、このアプリケーションを完全に機能させます。このまま次のデモ操作を続ける場合は、プロジェクトを開いたままにしておきます。中止する場合は、プロジェクトを閉じ、次のデモ操作の最初に再起動することができます。また、プロジェクトを閉じて、閉じなくても、NetExpress 全体を終了することができます。

# 第13章 Windows GUI アプリケーションの完成と実行

前の章では、Windows GUI アプリケーション ウィザードとダイアログ システムを使用してスクリーンセットを設計し、COBOL プログラムを作成しました。この章では、作成した COBOL プログラムを編集し、ビジネス ロジックを追加します。その後で、IDE を使用して完成したアプリケーションを実行します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了します。次に、前の章「Windows GUI アプリケーションの作成」の作業を終了してから、この章に進んでください。

---

## 13.1 概要

この章では、Windows GUI アプリケーション ウィザードを使用して、.cbl ファイルにスクリーンセットの入出力用コードを含む COBOL ソース プログラムを生成します。また、この .cbl ファイルを編集し、スクリーンセットからのデータの処理と出力データの作成を行うビジネス ロジックを追加します。CALL 文を追加して他のプログラムを呼び出すこともできます。この方法では、スクリーンセット処理プログラムとビジネス ロジックを完全に分離できます。

### 13.1.1 関連付けられたプログラムの構造

関連付けられたプログラムには、通常、次の処理を実行するメイン ループがあります。

1. DSGRUN の呼び出し
2. DSGRUN から返された状態コードのテスト
3. 挿入したビジネス ロジックの実行
4. 出力を表示し、さらに入力を受け付けるための DSGRUN の呼び出しの反復

Windows GUI アプリケーション ウィザードにより生成されたプログラムは、このような構造になっています。他の例については、後でディレクトリ `d:\NetExpress\DialogSystem\Demo\Customer` の顧客に関するデモを参照してください。

## 13.2 準備

NetExpress が閉じている場合は、前述のとおり再起動します。プロジェクトが閉じている場合は、「ここからチュ

ートリアルをスタート」の章の説明にしたがって開きます。[開く] を使用する場合、ディレクトリ `d:\¥NetExpress¥Base¥Demo¥Welcome` のプロジェクト Welcome を開きます。デフォルトでは、「開く」ダイアログ ボックスの「ファイルの種類」フィールドに `.app`、`.cbl`、および `.cpy` が設定されているので、ファイル Welcome が表示されます。

## 13.3 具体的な手順

この手順では、次の操作を行います。

1. COBOL プログラムの編集
2. アプリケーションのビルド
3. アプリケーションの実行

### 13.3.1 COBOL プログラムの編集

COBOL プログラムにビジネス ロジックを追加するには、テキスト エディタを使用します。この章では、IDE を使用します。

1. 「プロジェクト」ウィンドウの `Welcome.cbl` をダブルクリックします。

「テキスト」ウィンドウが開き、プログラムのソースが表示されます。ソースを見やすく表示するためにウィンドウのサイズを変更する必要があることがあります。[表示] メニューの [コピーファイルをすべて非表示] の横にあるボタンに似たアイコンが押されたように表示されていることを確認します (そうでない場合はクリックします)。

プログラムを一通り調べ、実行内容を確認します。

2. 作業場所節の行 `COPY "Welcome.CPB"` にカーソルを置き、[ファイル] メニューの [コピーファイル] をクリックします。[表示] をクリックします。

ファイル `welcome.cpb` はデータ ブロックです。データ項目 `I-NAME` と `GREETING` はファイルの末尾にあります。

もう一方のコピーファイル `ds-cntrl.mf` は、制御ブロックといえます。制御ブロックは NetExpress で作成され、ここにすべてのダイアログ システム アプリケーションに必要な標準定義が記述されています。

3. Program-Body Section で、下側のピリオドと上側の文 `PERFORM Call-Dialog-System` の間に次の行を挿入します。

```
string "Hello " I-Name delimited by size into Greeting
```

実際のアプリケーションでは、通常、ここでサブプログラムを呼び出し、スクリーンセットのデータを処理してから、表示するデータを計算したり検索したりします。DSGRUN から返された値によって決められ

たアクションを実行する EVALUATE 文を使用することもできます。これには、通常、分岐 WHEN EXIT-FLAG-TRUE CONTINUE が記述されています。

### 13.3.2 アプリケーションのビルド

アプリケーションは、次のようにビルドします。

1. [プロジェクト] メニューの [すべてをリビルド] をクリックします。

NetExpress は、プログラムの保存とコンパイルを行い、実行可能ファイルをリビルドします。「出力」ウィンドウに "リビルド完了" というメッセージが表示されるまで待機してから、先へ進みます。

### 13.3.3 アプリケーションの実行

アプリケーションは、次のように実行します。

1. NetExpress の [アニメート] メニューで [実行] をクリックします。

「アニメーションの起動」ダイアログ ボックスが表示されます。

2. 「アニメーションの起動」ダイアログ ボックスの [OK] をクリックします。

プログラムが実行され、スクリーンセットが表示されます。

3. 「Enter Name」フィールドをクリックし、自分の名前を入力して [Start] をクリックします。I-NAME-DISP を 12 文字に定義したため、文字数は 12 文字に制限されています。

このプログラムにより、「Greeting」フィールドの "Hello" の後に名前が配置され、スクリーンセットが再表示されます。

4. 「Enter Name」フィールドをクリックして名前を削除してから、別の名前を入力して [Start] をクリックします。

プログラムは再度メイン ループを回り、"Hello" に続いて新しい名前が表示されます。

5. [Esc] キーを押してウィンドウを閉じ、プログラムを終了します。

## 13.4 次へ進む前に

プロジェクトを閉じます。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままにしておいてかまいません。

「ここからチュートリアルをスタート」の章の チュートリアル マップに戻り、目的に沿って次のデモ操作を選択します。

# 第14章 UNIX でのアプリケーションの実装

UNIX オプションを使用して、NetExpress で開発したアプリケーションを UNIX システムに実装します。

---

まず、「ここからチュートリアルをスタート」の章を読み、最初のデモ操作「NetExpress の使用方法」を終了してから、この章へ進んでください。

---

## 14.1 概要

UNIX オプションを使用して、NetExpress で作成したアプリケーションを、Micro Focus COBOL for UNIX V3.1 以降がインストールされた UNIX に実装します。

この章では、既存の COBOL アプリケーションで作業を開始します。UNIX オプションを使用して UNIX システムにアプリケーションを実装する場合、アプリケーションを UNIX でリビルドし、実行します。このアプリケーション例は、1 つのレコードを出力するファイルを作成するだけの 1 つの COBOL プログラムで構成されます。

Web アプリケーションを UNIX で実装する方法を参照するには、このマニュアルを終了した後で、オンライン マニュアル『インターネット アプリケーション』を参照してください。

Windows GUI アプリケーション (Dialog System を使用して作成された Windows GUI アプリケーションを含みます) は、UNIX では実装できません。ただし、UNIX オプションには、文字によるユーザー インターフェイスを作成するための文字バージョンの Dialog System があります。これについては、『Dialog System 文字 モード ユーザー ガイド』で説明しています。

## 14.2 準備

サーバー コントロール プログラム (SCP) を UNIX システムにインストールしてから、このデモ操作を実行してください。このプログラムは、UNIX オプションの パブリッシュ 機能を実行するために必要です。SCP は、NetExpress CD に収録されています。インストール説明については、『UNIX オプション ユーザー ガイド』の付録「SCP と Samba のインストール」を参照してください。

UNIX システムに接続する必要があります。UNIX システムにログインして、.rhosts ファイルを構成し、パスワードを必要としない透過ログインを有効化します。この方法の詳細については、UNIX のマニュアルを参照してください。

NetExpress が閉じている場合は、前述のとおり再起動します。「プロジェクト」ウィンドウや「テキスト」ウィンドウが開いている場合は閉じます。

## 14.3 具体的な手順

この手順では、次の操作を行います。

1. プロジェクトの作成
2. アプリケーションのビルドと移植性のチェック
3. UNIX 端末としての PC の使用 (オプション)
4. パブリッシュ オプションの設定
5. UNIX システムでのアプリケーションのパブリッシュ
6. アプリケーションの実行

### 14.3.1 プロジェクトの作成

アプリケーションのプロジェクトは、次のように作成します。

1. [ファイル] メニューの [新規作成] をクリックしてから、「新規作成」ダイアログ ボックスで「プロジェクト」を選択し、[OK] をクリックします。
2. 「プロジェクトの新規作成」ダイアログ ボックスの [既存アプリケーションから作成するプロジェクト] をクリックします。
3. プロジェクト名として UNIXO を入力します。また、プロジェクトを格納するフォルダとして `d:\NetExpress\Base\Demo\UNIXO` を入力し、[作成] をクリックします。
4. このチュートリアルを以前に実行している場合は、既存のプロジェクトに上書きするかどうかをたずねるメッセージが表示されます。[はい] をクリックします。
5. [ファイルの追加] をクリックします。
6. UNIXO.CBL を選択し、[追加] をクリックします。
7. [次へ] をクリックします。
8. 次のダイアログ ボックスで [次へ] をクリックします。
9. [完了] をクリックしてプロジェクトを作成します。

プロジェクトの「プロジェクト」ウィンドウが表示され、プロジェクトの従属関係が確認されます。このプロジェクトには、現在アプリケーションのビルドに使用する COBOL プログラムだけが格納されています。



## 14.3.2 アプリケーションのビルド

このアプリケーションをビルドする場合、UNIX に移植できない機能が含まれている場合に警告するオプションを設定する必要があります。

1. [プロジェクト] メニューの [プロパティ] をクリックします。
2. 「プロジェクト指令」フィールドで、末尾のセミコロンの前に WARNINGS(2); と入力します(ENSUITE(3) の後は空白文字を残します)。[OK] をクリックします。
3. [プロジェクト] メニューの [リビルド] をクリックして、プログラムをコンパイルします。

プロジェクトの指令の変更で、全プロジェクトの変更を促すことを警告するメッセージが「出力」ウィンドウに表示されます。場合によっては、メッセージを見やすくするために「出力」ウィンドウを拡大してください。ファイルを編集して、移植できない構文を変更する必要があります。

4. 「出力」ウィンドウのエラー メッセージをダブルクリックします。

「編集」ウィンドウが開き、問題のある行の余白に赤い×印が表示されます。

5. 行を編集して、OUT FILE を OUTFILE に置換します。

ファイル名に空白文字が含まれているために、コンパイラによってこの行にフラグが立てられています。空白文字を含むファイル名は Windows では使用できますが、UNIX では使用できません。

6. [プロジェクト] メニューの [リビルド] をクリックします。

これで、警告が表示されずにプログラムをコンパイルできるはずです。アプリケーションを UNIX システムで実装することができます。

7. UNIXO.CBL を表示する「テキスト」ウィンドウを閉じます。

## 14.3.3 UNIX 端末としての PC の使用

NetExpress には、端末エミュレーション ソフトウェアである Telnet が組み込まれているので、PC を UNIX 端末として使用できます。このチュートリアルの実行中に Telnet を使用して、UNIX ホスト システムの端末として PC に別のウィンドウを開くと便利です。他の端末エミュレーション ソフトウェアがあるため、または、別の UNIX 端末を使用しているために、この方法を使用しない場合は、この項をスキップしてかまいません。

1. [UNIX] メニューの [ターミナル] をクリックします。
2. Telnet の [接続] メニューで [リモートシステム] をクリックします。
3. 「接続」ダイアログ ボックスで、「ホスト名」フィールドに UNIX ホスト マシンの名前を入力します。「ポート」と「ターミナルの種類」フィールドに適切な値が設定されているかどうかを確認します。次に[接続]

をクリックします。

4. 通常の方法で UNIX システムにログインします。

#### 14.3.4 パブリッシュ オプションの設定

実装とは、UNIX オプションのパブリッシュのことで、パブリッシュは、UNIX システムの正しいディレクトリにファイルをコピーし、そこでアプリケーションをビルドすることを指します。このプロセスを処理するツール、パブリシャは、NetExpress のメニューからアクセスできます。

まず、パブリシャに、パブリッシュ操作に関する必要な詳細情報を次のように設定します。

1. [Unix] メニューの [セットアップ] をクリックします。
2. 「UNIXオプションへようこそ」画面で [OK] をクリックします。前述の「準備」の項を実行するときに、この画面に表示される準備を完了しておいてください。

その結果、図 14-1 のようなフォームが表示されます。

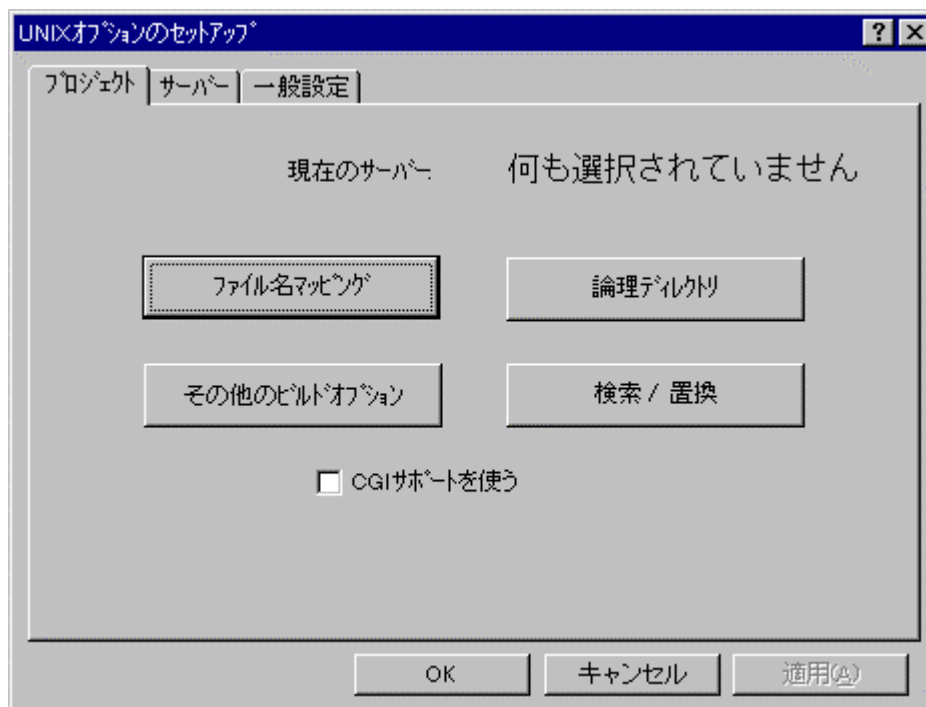


図 14-1 「セットアップ」ダイアログ ボックス

3. [サーバー] タブをクリックします。これによって、図 14-2 のようなフォームが表示されます。



図 14-2 「サーバー」ダイアログ ボックス

リストボックスには、現在定義されているサーバーが表示されます。これらの 1 つを選択し、このタブのボタンを使用して変更を加えると、その変更内容が選択したサーバーに適用されます。エントリ「新しいサーバー」では、新しいサーバーを定義できます。現在構成されているサーバーがない場合は、デフォルトで「新しいサーバー」が選択され、「設定」を除くダイアログ ボックスのボタンはすべて非アクティブになります（上記の図を参照）。

4. [設定] をクリックします。次のようなダイアログが表示され、新しいサーバー名を指定できます。



図 14-3 サーバーの指定

5. 有効なサーバー名を入力して [OK] をクリックします。次のようなダイアログボックスが表示されます。

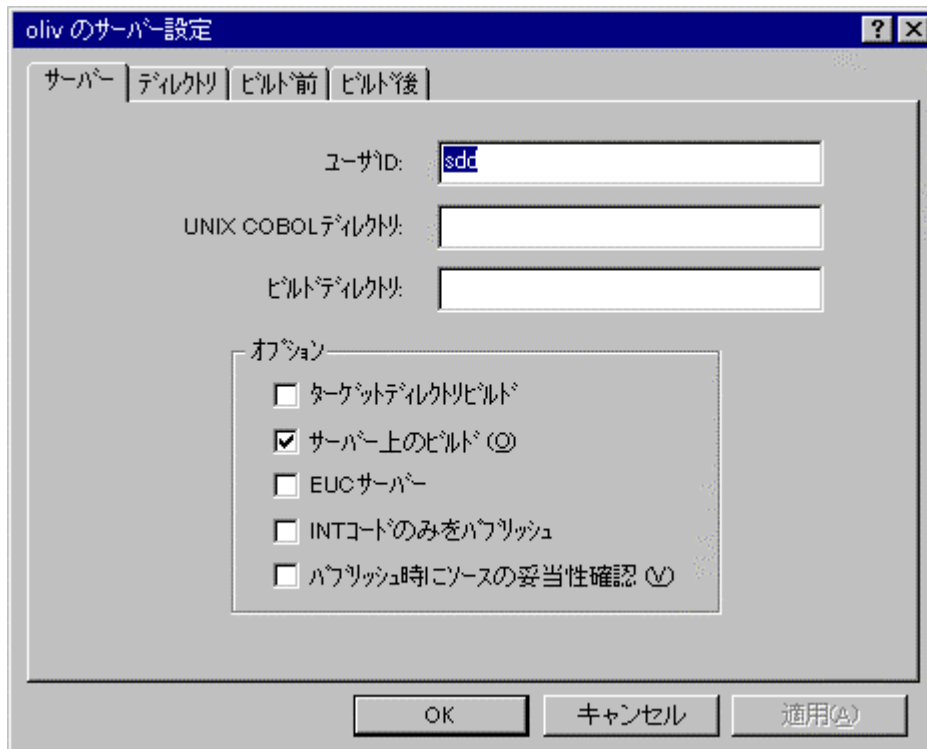


図 14-4 サーバーの設定

6. 「ユーザー ID」フィールドに、UNIX システムにログオンするためのユーザー ID を入力します。
7. 「UNIX COBOL ディレクトリ」フィールドに、UNIX システムの COBOL システム ディレクトリの名前を入力します。UNIX システムで `echo $COBDIR` と入力すると、このディレクトリを検索できます。このフィールドを空白のままにすると、デフォルトにより `/usr/lib/cobol` が設定されます。
8. アプリケーションをパブリッシュする UNIX システムのディレクトリを決定し、「ビルド ディレクトリ」フィールドに名前を入力します。
9. [OK] をクリックします。
10. 再び [OK] をクリックします。

### 14.3.5 アプリケーションのパブリッシュ

アプリケーションは、次のようにパブリッシュします。

1. 「UNIXオプションのセットアップ」ダイアログ ボックスの「ビルド ディレクトリ」フィールドに名前を入力したディレクトリが UNIX システムにあることを確認します。
2. NetExpress の [UNIX] メニューで [パブリッシュ] をクリックします。

その結果、PC でプロジェクトがビルドされ、UNIX で同じ処理を実行するための Makefile というファイルが作成されます。また、UNIX システムにプロジェクト ファイルがコピーされ、UNIX システムで Makefile が実行されます。

問題が生じた場合は、再度 [UNIX] メニューの [セットアップ] を選択して設定を確認します。サーバー名、UNIX ログイン名、および、ディレクトリ名が正しいことを確認してから、再試行します。

これで問題が解決しない場合は、『UNIX オプション ユーザー ガイド』の「ヒントとトラブルシューティング」の章を参照してください。

3. UNIX マシンでリスト ユーティリティを使用してログ ファイル `Make.log` を参照します。

ログ ファイルにエラーが表示されている場合、または次の項「作成されたファイル」に一覧表示されているファイルのどれかが UNIX のビルド ディレクトリにない場合は設定を確認して再試行します。

[パブリッシュ] によってパブリッシュされるのは、前回プロジェクトをパブリッシュした後で変更されたファイルだけです。[すべてをパブリッシュ] では、すべてのファイルが無条件でパブリッシュされます。このチュートリアルを実行するのが 2 回目である場合は、先に UNIX のビルド ディレクトリからファイルを削除しないでください。[パブリッシュ] では、ファイルが削除されたことが認識されないため、置換されません。UNIX ディレクトリからファイルを削除する場合は、[すべてをパブリッシュ] を使用します。

### 14.3.6 作成されたファイル

UNIX オプションにより、PC のプロジェクト ディレクトリに次のファイルが作成され、UNIX のビルド ディレクトリにコピーされます (このコピーは、PC で使用されないため、プロジェクトに追加されていません)。

- Makefile - UNIX でビルドする場合にガイドするスクリプト ファイルです。

次のファイルも PC のプロジェクト ディレクトリから UNIX のビルド ディレクトリにコピーされます。

- UNIXO.CBL - COBOL プログラムです。

ビルド ディレクトリには、UNIX のビルド処理で作成された次のファイルも格納されています。

- Make.log - UNIX でのビルド結果を示すログです。

ビルド プロセスで使用するために作成されたいくつかのファイルも、UNIX のビルド ディレクトリに格納されています。

UNIX では、ファイル名の大文字と小文字を区別します。

### 14.3.7 アプリケーションの実行

アプリケーションは、次のように実行します。

1. UNIX システムで、通常 UNIX COBOL アプリケーションを実行するのと同じ方法を使用してアプリケーションを実行します。たとえば、次のように実行します。

```
cobrun UNIXO.int
```

2. ls コマンドなどを使用して、ビルド ディレクトリを一覧表示し、ファイル OUTFILE があるかどうかを確認します。リスト コーティリティを使用して、"Example output" を含むレコードが 1 つだけであることを確認します。

UNIXO.CBL の実行内容は、このファイルの作成だけです。そのため、このファイルがある場合は、アプリケーションが正しくビルドされ、実行されていることを示します。

## 14.4 次へ進む前に

プロジェクトを閉じます。

Telnet を使用している場合は、通常の方法で UNIX からログオフし、Telnet の [接続] メニューで [Telnetの終了] をクリックします。

そのまま別のデモ操作に進む場合は、NetExpress を開いたままにしておくことができます。

「ここからチュートリアルをスタート」の章の チュートリアル マップに戻り、目的に沿って次のデモ操作を選択します。

# 第15章 実行時ライセンス

## 15.1 概要

ご購入頂いた Micro Focus NetExpress のライセンスでは、ご購入者の個人使用を目的としたアプリケーションの開発と実行が可能です。NetExpress を使用して開発および変更されたアプリケーションを第三者による使用を目的として配布する場合には、Micro Focus 社の実行時ライセンスを、同時に使用するユーザー毎にご購入頂く必要があります。

Micro Focus NetExpress には、開発システムのコンポーネントとして追加製品が含まれています。これらの追加コンポーネントをアプリケーションの一部として配布する場合は、「追加製品の実行時ライセンス」の項で説明されており、追加ライセンスが必要になります。

UNIX にアプリケーションを実装し、Operating System Extension (OSX) または Application Server for UNIX のライセンスが必要になった場合は、Micro Focus COBOL for UNIX の使用許諾契約を参照してください。

NetExpress を Microsoft Windows Terminal Server あるいはそれに類似するターミナルソフトウェアの上で使用することはサポートされていません。もしユーザがそのような環境で製品を使用する場合には、たとえソフトウェア製品がひとつのマシンの上でしか稼動していなくても、NetExpress にアクセスする個々のユーザの数分のライセンスを購入する必要があります。同様のことが、本章の後半で述べる実行時ライセンスに対しても成り立ちます。

NetExpress の実行時ライセンスを取得すると、NetExpress の機能を Windows や Web プラットフォームに実装できます。NetExpress に対する Micro Focus 社の実行時ライセンスには、次の 3 種類があります。

### 1. Micro Focus NetExpress Run Time System

NetExpress Run Time System には、NetExpress を使用して作成されたクライアント アプリケーションの実装に必要なすべての NetExpress 開発システム ファイルが含まれます (Dialog System も含まれます)。

### 2. Micro Focus NetExpress Application Server

NetExpress Application Server は、NetExpress Run Time System の上位セットです。NetExpress Application Server には、NetExpress を使用して作成されたクライアント サーバー アプリケーションやイントラネット アプリケーションの実装をサポートする追加コンポーネントが含まれています。

### 3. Micro Focus NetExpress Web Application Server

NetExpress Web Application Server には、NetExpress Application Server と同じ機能が含まれますが、これは、アプリケーションを World Wide Web (インターネット) に実装する場合に必要なになります。

次に、各ライセンスが必要になる場合について説明します。

## 15.2 実行時ライセンスが必要な場合

NetExpress によって作成または変更されたアプリケーションを実行するには、アプリケーションの使用目的（社内用であるか、または、市販用であるか）に関係なくアプリケーションの実行時ライセンスを購入する必要があります。

マルチユーザー アプリケーションを実行する場合、同時にアプリケーションを使用するユーザーの最大数（同時にアプリケーションにアクセスする最大ユーザー数）を予測し、その人数分のライセンスを購入する必要があります。このライセンスは、Windows システムやイントラネット システムで実装する場合にも適用されます。

WWW（インターネット）に実装するアプリケーションがある場合、NetExpress Web Application Server ライセンスが必要になります。NetExpress Web Application Server ライセンスは、Web サーバーごとに 1 つ必要です。このライセンスを取得すると、無制限のユーザーをもつサーバーからインターネットに実装することができます。

アプリケーションを単独でインターネットに実装する場合、必要なライセンスは、NetExpress Web Application Server のライセンスだけです。一方、アプリケーションをインターネットに実装し、さらにイントラネットや Windows システムにも実装した場合、NetExpress Web Application Server のライセンスと、同時に使用するユーザー数分の NetExpress Run Time System ライセンスまたは Application Server ライセンスの両方が必要になります。

実行時ライセンスが必要ないのは、次の場合だけです。

1. アプリケーションのユーザーがすでに NetExpress Development ライセンスを取得している場合、追加の実行時ライセンスは必要ありません。
2. ユーザーがすでに別のアプリケーションについて NetExpress の実行時ライセンスを取得している場合、同じライセンスを複数のアプリケーションに使用することができます。

ライセンス条件の詳細については、「NetExpress 使用許諾契約」を参照してください。

## 15.3 Application Server ライセンスまたは Run Time System ライセンスが必要な場合

アプリケーションが NetExpress のクライアント サーバー機能またはイントラネット機能を使用している場合、アプリケーションを実行するには、Application Server ライセンスが必要になります。それ以外の場合には、アプリケーションの実行時に NetExpress Run Time System ライセンスが必要になります。インターネットで実装するアプリケーションでは、常に NetExpress Web Application Server ライセンスが必要になります。

---

ライセンス契約 : NetExpress Application Server 製品と NetExpress Run Time System 製品を構成するコンポーネントは、該当するライセンスを購入したカスタマに出荷することができます。これらの製品の上記以外のアイテムやコ



ンポーネントを配布および実装することはできません。

---

NetExpress Application Server と NetExpress Run Time System のパッケージに含まれるファイルの正確な詳細情報については、「NetExpress 実装リスト」を参照してください。各ライセンスのコンポーネントの概要については、次に説明します。

### 15.3.1 NetExpress Run Time System のコンポーネント

Net Express Run Time System には、Net Express アプリケーションの実行中に必要なすべてのコンポーネントや、静的にリンクされたアプリケーションにリンクするすべてのコンポーネントが含まれています。ただし、クライアント サーバーとイントラネットをサポートするコンポーネントは含まれません。Net Express Run Time System には、主に次のコンポーネントが含まれています。

- Micro Focus 共有ランタイム システムおよびランタイム環境
- Micro Focus 静的ランタイム システム
- COBOL システム ライブラリ ルーチン
- 呼び出し可能ファイル ハンドラ (MFFH)
- Btrieve インターフェイス
- 索引ファイル保守ツール (Rebuild)
- 画面とキーボードのハンドラ (ADIS)
- 画面属性構成ファイル (ADISCF)
- パネル バージョン 2
- クラス ライブラリ
- MFSORT (呼び出し可能モジュール)
- COBSQL サポート
- OpenESQL サポート
- Dialog System ランタイム

## 15.3.2 NetExpress Application Server と Web Application Server のコンポーネント

NetExpress Application Server と NetExpress Web Application Server には、次のような NetExpress Run Time System のすべてのコンポーネントと補足的なクライアント サーバー コンポーネントが含まれます。

- クライアント サーバー結合 (CSBIND)
- 共通通信インターフェイス(CCI)
- Fileshare バージョン 2
- CGI サポートのための拡張 ACCEPT/DISPLAY文
- 埋め込み HTML サポート

および

- NetExpress Run Time System のすべてのコンポーネント

## 15.4 アプリケーションの実装、パッケージ化、出荷に関する注意

ライセンスの購入とは別に、アプリケーションの実装を計画する際に考慮すべき点が多数あります。

1. アプリケーションを個人的に開発してデバッグする場合と 実装する場合は、アプリケーションの最適なビルド方法が異なります。アプリケーションのビルド方法の相違点については、「アプリケーションのビルド」の章を参照して下さい。
2. NetExpress Application Server ライセンスまたは NetExpress Run Time System ライセンスをご購入頂くと、通常は、ソフトウェア パッケージを入手されることとなります。このパッケージのファイルは、ライセンスに基づき、アプリケーションとともに配布することができます。また、このパッケージには、アプリケーションに必要なすべての NetExpress システム ファイルが含まれています。現在 NetExpress Application Server または NetExpress Run Time System のパッケージを入手していない場合、「NetExpress 実装リスト」で全ファイル リストを参照してください。
3. NetExpress Application Server ライセンスを購入すると、通常 NetExpress Application Server ソフトウェア パッケージを入手します。このパッケージに含まれるファイルは、インターネット アプリケーションを実行しているマシンにコピーし、そのアプリケーションだけで使用することができます。これらのファイルは、インターネットからダウンロードすることはできません。ウェブでの実装については、オンライン マニュアル『インターネット アプリケーション』の「アプリケーションの実装」の章を参照してください。
4. NetExpress Application Server パッケージまたは NetExpress Run Time System パッケージの全体をアプリケーションに含める必要がある場合とない場合があります。アプリケーションが実装のために静的にリンクされている場合、別に NetExpress のシステム ファイルを配布する必要はありません (該当する実行時ライセンスは必要です)。アプ

リケーションのビルドと配布の詳細については、ヘルプの「出荷する実行時ファイルの選択」を参照してください。配布するファイルを選択する場合は、NetExpress のパッケージからではなく、NetExpress Application Server パッケージまたは NetExpress Run Time System パッケージからこれらのファイルを選択することをお勧めします。これは、NetExpress パッケージでは、一部のファイルが、インストールした開発システムでしか実行できないように保護されていることがあるためです。

## 15.5 追加製品の実行時ライセンス

Micro Focus NetExpress には、開発システムのコンポーネントとして追加製品が含まれています。追加製品の中には Micro Focus 社に所有権のないコンポーネントもあります。これらの追加コンポーネントをアプリケーションの一部として配布する場合、製品の使用許諾契約に基づいて別のライセンスが必要になる場合があります。

### 15.5.1 ODBC ドライバ

Micro Focus NetExpress 開発システムには、Intersolv 社の ODBC ドライバも含まれています。Intersolv 社の ODBC ドライバをアプリケーションとともに配布するには、別のライセンスが必要です。日本で Intersolv 社の ODBC ドライバのライセンスを取得する場合、詳細については、Intersolv KK で確認してください。

他の ODBC ドライバを実装している場合は、使用許諾契約の内容について、該当するベンダにお問い合わせください。

### 15.5.2 ProtoView データ テーブル グリッド コントロール

Micro Focus NetExpress には、ProtoView Inc. が開発したデータ テーブル ActiveX グリッド コントロールが含まれています。このコントロールをアプリケーションとともに配布する場合は、別のライセンスが必要です。日本におけるデータ テーブルのライセンスについては、次の場所にお問い合わせください。

Protoview UK Ltd  
または  
文化オリエント株式会社

### 15.5.3 その他のサード パーティのコンポーネント






Micro Focus NetExpress には、他に Microsoft SDK のようなサード パーティのソフトウェアが含まれています。これらの詳細については、NetExpress 使用許諾契約のサード パーティのソフトウェアに関する契約条件の項を参照し、該当するベンダにお問い合わせください。

# 付録A: Windows のヒント

このマニュアルは、Windows オペレーティング システムの使用方法に関する基礎知識を持つ読者を対象としています。この付録では、オペレーティング システムのインターフェイスとアプリケーションで一般的に使用される技法について簡単に説明します。

## A.1 ウィザードの標準ボタン

多くの場合、ウィンドウの右上端にあるタイトルバーには、ボタンが 3 つまで配置されます。ここに表示されるボタンは、次のとおりです。

	閉じる	ウィンドウとウィンドウで実行されている内容をすべて閉じます。
	最小化	ウィンドウを画面の最下部にあるタスクバーのエントリとして縮小します。これによって、画面スペースが節約されます。ウィンドウとウィンドウで実行されている内容はそのまま存在します。
	最大化	ウィンドウを拡大して全画面表示します。画面の他の内容は起動されたままですが、このウィンドウの後にあるため表示されません。
	元のサイズに戻す	ウィンドウを全画面表示から部分画面表示に縮小します。
	コンテキスト ヘルプ	マウス ポインタが疑問符に変わります。このポインタで項目をクリックすると、その項目の通常の機能が実行されずに、クリックした項目に関する簡単なヘルプが表示されます。

## A.2 フォルダの変更

ファイルを選択する場合に、特定のフォルダに格納されたファイルとフォルダがすべて表示されるダイアログ ボックスが開きます。この状態を、ダイアログ ボックスがこのフォルダで開いているといいます。一般に使用される図 A-1 の「開く」ダイアログ ボックスや [プロジェクトへのファイルの追加] を選択すると開く「ファイルの追加」ダイアログ ボックスなどが例にあげられます。

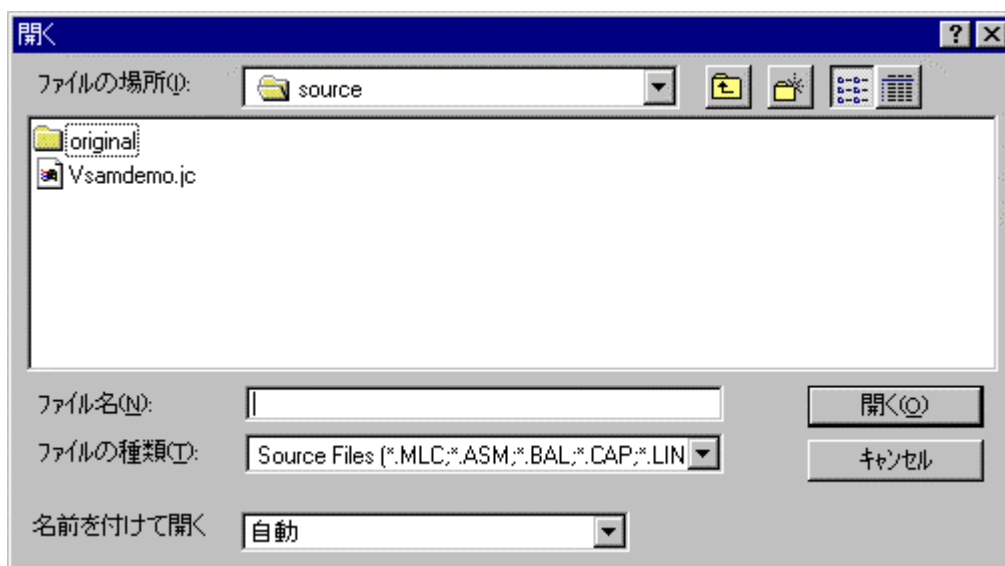




図 A-1 「開く」ダイアログ ボックス


フォルダの階層を移動するために次の方法（一部またはすべて）を使用できます。

- 開いているフォルダ内にあるフォルダを開くには、そのフォルダをダブルクリックします。
- 開いているフォルダを格納しているドライブまたはフォルダに移動するには、 をクリックします。
- ドライブかフォルダに移動するには、「ファイル名」フィールドにそのパスを入力して、該当するボタン（たとえば、[開く] または [追加]）をクリックします。完全パスを「ファイル名」フィールドに入力すると、ファイル自体が開くか、追加されます。
- ドライブとフォルダの階層をすべて一覧表示するには、「ファイルの場所」フィールドの右側にある下向きの矢印  をクリックします。その結果、開いているフォルダの名前が表示されます。目的のフォルダをクリックしてフォルダを変更することもできます。マウス ポインタを移動するだけで、選択項目が変更されます。そのため、フォルダを変更しないでこのリストを閉じるには、強調表示が移動していないことを確認するか、リスト以外をクリックします。

### A.3 フォルダの操作

「フォルダの変更」の項で説明するダイアログ ボックスでは、通常、フォルダの作成、フォルダとファイルの名前変更と削除などの機能を実行できます。

これらの操作は、たまたま使用しているソフトウェアからも実行できるので、Windows の別のセッションを行うより便利です。たとえば、IDE を使用していて、フォルダを作成する必要がある場合は、IDE の [ファイル] メニューで [開く] をクリックして、「開く」ダイアログ ボックスで操作します。その後は、[キャンセル] をクリックするだけで、このダイアログ ボックスが閉じます。

- 開いたフォルダ内に新しいフォルダを作成するには、 をクリックします。
- 他の機能を使用する場合は、開いたフォルダ内のフォルダ名を右クリックしてポップアップ メニューを使用します。

## A.4 ツリー ビュー

ディレクトリとそのサブディレクトリを一覧表示する場合のように、階層データは主にツリー ビューで表示されます。初期状態のツリー ビューは、最上位のリストで、各項目の左側に "+" が付いています。"+" をクリックしてエントリを展開し、すぐ下に従属する項目を表示します。このとき、"+" は "-" に変わります。 "-" をクリックすると、展開されたツリーは再び圧縮されます (閉じます)。または、項目をクリックして、数字キーで "\*" (アスタリスク) を入力し、その項目に従属する項目をすべて表示します。

Windows エクスプローラでは、左側のペインがコンピュータのドライブとフォルダを示すツリー ビューになっており、この良い例です (エクスプローラを起動するには、デスクトップの [マイ コンピュータ] をクリックし、[エクスプローラ] をクリックします)。

## A.5 ファイル拡張子の表示または非表示

(たとえば)「開く」ダイアログ ボックスまたは Windows エクスプローラ で表示されるファイルのリストでは、Windows が認識するファイル拡張子が表示されないことがあります。その場合は、ファイルの種類がアイコンや説明として表示されます。拡張子を表示するように Windows の設定を変更する場合は、Windows エクスプローラで次のように操作します。

1. デスクトップの [マイ コンピュータ] をクリックしてから、[エクスプローラ] をクリックします。
2. [表示] メニューの [オプション] をクリックします。
3. [表示] タブをクリックします。
4. 「登録されているファイルの拡張子は表示しない」がチェックされていないことを確認してください。表示されているチェックを解除するには、チェックボックスをクリックします。
5. [OK] をクリックします。

Windows エクスプローラ を閉じます。

この操作中にダイアログ ボックスが開いている場合は閉じ、再び開いてから変更を確認します。



## A.6 コンテナの列見出し

リストによっては、見出しの背景が灰色で表示された列を持つものがあります。Windows では、このようなリストは コンテナ といいます。図 A-2 は、このようなリストの一部です。

DS名	DS編成	PC名	レコード形式	論理レコード長	ブロックサイズ	ボリューム	Cat	ジョブ
MFMVS.SYSLOG	PS	E:MFUSER#PRO...	LSEQ	132	0		はい	A...
SORTWRK	PS	E:MFUSER#PRO...	FB	37	0		はい	A...
SYS1998.S0305.S105907.J01001...	PS	E:MFUSER#PRO...	LSEQ	132	0		はい	A...
SYS1998.S0305.S111254.J01002...	PS	E:MFUSER#PRO...	LSEQ	132	0		はい	A...
SYS1998.S0305.S111836.J01003...	PS	E:MFUSER#PRO...	LSEQ	132	0		はい	A...
SYS1998.S0305.S112248.J01004...	PS	E:MFUSER#PRO...	LSEQ	132	0		はい	A...
VSAMDEMO.BADDATA	PS	E:MFUSER#PRO...	V	536	0		はい	A...
VSAMDEMO.BADDATA.DATA	PS	E:MFUSER#PRO...	V	536	540	MVS...	はい	VS.
VSAMDEMO.EASTVSAM	VSAM	E:MFUSER#PRO...	KS	535	0		はい	VS.
VSAMDEMO.EASTVSAM.AX1	VSAM	E:MFUSER#PRO...	KS	4086	0		はい	VS.
VSAMDEMO.EASTVSAM.AX2	VSAM	E:MFUSER#PRO...	KS	4086	0		はい	VS.
VSAMDEMO.FRGVNSAM	VSAM	E:MFUSER#PRO...	KS	535	0		はい	VS.

図 A-2 コンテナの一部

レイアウトを再配置するには、次のどれかの方法を使用します（これらの方法は、使用できない場合もあります）。

- 列幅を拡大または縮小するには、列見出しで右の列との境界線をクリックして、左右にドラッグします。列は、縮小して非表示にできることに注意してください。目的の列が表示されていない場合は、表示されている列をドラッグすると、非表示になっていた列を見つけられることがあります。
- 特定のエントリの列幅を最大列幅のエントリに自動的に合わせるには、列見出しで右の列との境界線をダブルクリックします。
- テーブルの列リストを参照するには、列見出しを右クリックします。このリストのエントリをクリックし、表示されているテーブルに対して列を挿入または削除します。
- 列の順序を再配置するには、列見出しをクリックして他の列の列見出しにドラッグします。最初の列は、ドラッグした先の列のすぐ右に移動します。
- 特定の列を整列させるには、その列見出しをクリックします。
- エントリの順序を逆にするには、リストの整列条件であるフィールドの列見出しをクリックします。列見出しの横にある小さい矢印は、リストが昇順 、または降順  に整列されていることを示します。

## A.7 リスト項目の選択

ボタンをクリックして、選択した項目に対してアクションを実行する前に、表示されたリストから項目を選択する作業が必要な場合があります。たとえば、リストボックス、上図 A-1 のような一般的に使用される「開く」ダイアログ ボックス、「ファイルの追加」ダイアログ ボックスなどでは、このような操作が必要です。

単一の項目を選択するには、その項目を左クリックします。また、多くの場合、複数の項目を選択することもできます。複数の項目を選択する方法は、次のとおりです（これらの方法は、使用できない場合もあります）。

- [Ctrl] キーを押しながら複数の個別項目をクリックします。[Ctrl] キーを押さずにクリックすると、それまでに選択した項目がすべて選択解除されます。

- 1 つの項目をクリックし、[Shift] キーを押しながら 2 番目の項目をクリックします。この場合、選択した 2 つの項目とその間にある項目がすべて選択されます。
- [Shift] キーを押しながらスクロール バーの下向きの矢印をクリックします。これによって、エントリが連続して選択されます。
- [Ctrl] を押しながら A と入力します。これによって、リスト全体が選択されます。

## A.8 ドロップダウン リスト


右側に下向きの矢印  の付いたフィールドが表示されることがあります。この矢印は、フィールドに入力できる値を一覧表示するドロップダウン リストを示します。図 A-3 はドロップダウン リストを持つフィールドを示します。図 A-4 は同じフィールドでドロップダウン リストが開いている状態を示します。



図 A-3 ドロップダウン リストを持つフィールド

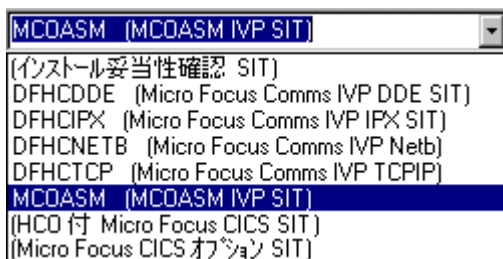



図 A-4 ドロップダウン リストが開いている同じフィールド

フィールドに値を入力する方法は、次のとおりです(これらの方法は、使用できない場合もあります)。

- 必要な値を入力します。
-  をクリックしてから、ドロップダウン リストで必要な値をクリックします。
- フィールドに、必要な値の最初の文字を入力すると、その文字で始まる値のうちリストの最初にある値がフィールドに挿入されます。実装内容によっては、さらに文字を入力して、必要な値をより詳細に指定することもできます。また、最初に入力した文字がその後の文字に置換され、後の文字が必要な値の最初の文字になることもあります。

## A.9 ポップアップ メニュー

画面のウィンドウまたはオブジェクトを右クリックすると、そのウィンドウやオブジェクトに対応した機能を示すメニューが表示されることがあります。これを、ポップアップ メニューまたはコンテキスト メニューといいます。



## A.10 切り取りと貼り付け

多くのアプリケーションでは、次のように項目を移動できます。

1. マウスで項目を強調表示します。ファイル名の場合、通常 1 回のクリックで強調表示されます。テキストの場合、テキストに沿ってマウス ボタンをドラッグして、必要な部分を強調表示します。
2. テキストをコピーする場合は、[Ctrl+C] キーを押します。テキストを移動する場合は、[Ctrl+X] キーを押して現在の位置から削除します。
3. 項目を配置する場所を強調表示します。項目を挿入する場合は、その場所をクリックします。既存のテキストを置換する場合は、置換先のテキストに沿ってマウス ポインタをドラッグして、強調表示します。
4. [Ctrl+V] キーをクリックします。

[Ctrl+C] キー、[Ctrl+X] キー、および [Ctrl+V] キーは、それぞれ通常メニューで使用できる [コピー] 機能、[切り取り] 機能、および [貼り付け] 機能に対するショートカット キーです。ショートカット キーを使用するとより速く操作できます。アプリケーションによっては、これらの機能に対して別のショートカット キーを使用することがあります。

このオンライン バージョンの『入門書』からテキストを切り取って IDE に貼り付けると、フィールドにパスを入力しないで済みます。

## A.11 ドラッグ & ドロップ

多くのアプリケーションでは、項目、リストやツリー ビューのアイコンなどを「ドラッグ & ドロップ」して、移動したりコピーしたりできます。マウス ポインタを項目やアイコンに合わせ、左マウス ボタンを押したまま、目的の位置にポインタを移動させます。これを「ドラッグする」といいます。目的の位置でボタンをはなすと項目が配置され、これを「ドロップする」といいます。

ドラッグ & ドロップで項目をコピーすると、新しい位置にコピーが配置されますが、コピー元の項目は元の場所に残っています。また、ドラッグ & ドロップで項目を移動させると、元の位置にあった項目は削除されます。この方法は、アプリケーションによって異なります。

この方法で、2 つのリスト間またはツリー ビュー間で項目を転送できます。Windows エクスプローラがよい例で、Windows エクスプローラを数回開くと、同じツリー ビューのコピーを複数表示することができます。ツリー ビュー全体が大きすぎて、画面の 1 つのツリー ビューでソースとターゲットを一度に表示できない場合は、コピーされたツリー ビューの新しい位置にファイルをドラッグできます。

項目グループもドラッグ & ドロップできます。前述の「リスト項目の選択」の項で説明するとおりに項目を選択し、そのうちの 1 項目で左マウス ボタンを押してドラッグします。マウス ポインタが目的の位置にくるまで、ボタンをはなさないように注意してください。途中でボタンをはなすと、すべて選択解除されてしまいます。

## A.12 ウィンドウのサイズ変更とドッキング

ウィンドウの形やサイズの変更方法や複合ウィンドウ内のペインの再配置方法は、IDE を使用した短いチュートリアルで詳細に説明されています。

1. マウス ポインタを「出力」ウィンドウの灰色の境界に置きます。

マウス ポインタの形は、置かれた位置によって変わります。Windows の設定によって使用される形が異なるので、図示することはできません。ただし、この境界では、標準形 (通常は矢印) になります。たとえば、ポインタが「出力」ウィンドウ内にある場合と、「出力」ウィンドウとその上のメイン ペインが接する位置にある場合では、ポインタの形は異なります。

2. 左マウス ボタンを押したまま上にドラッグし、さらに右にドラッグします。

「出力」ウィンドウを表すアウトラインがポインタとともに移動します。ポインタが IDE の右の境界と交差すると、このアウトラインの形が変わります。

「ドッキングさせない場合は、[Ctrl] キーを押したままにしてください。」というヒントが IDE のステータス バーに表示されます。

3. マウス ボタンをはなします。

「出力」ウィンドウは、右側の境界に結合されます。これをドッキングといいます。

4. 「出力」ウィンドウの左側にあるメイン ペインとの境界線にマウス ポインタを置き、使いやすい幅になるまで左にドラッグします。

このように、通常の方法でペインを広げることができます。

5. 「出力」ウィンドウの左側にある灰色の境界にマウス ポインタを置き、IDE の外側に出るまで右にドラッグします。

この操作により、「出力」ウィンドウが IDE の外側に分離します。この状態を「アンドッキング」またはフローティングといいます。

6. 「出力」ウィンドウのタイトル バーにマウス ポインタを置き、元の位置にドラッグして、ドッキングさせます。

# 付録B: その他の機能

この付録では、「概要」の章で説明されている主な機能に加え、注目すべき機能を紹介します。この付録は、NetExpress のすべての機能について説明するものではありません。

主に他の COBOL システムからの移行に利用する機能については、この付録の「B.2 移行のための機能」項で説明します。

NetExpress の補完用に同梱されている追加ソフトウェアについては、「補足的なソフトウェア」の章で説明します。

## B.1 機能説明

NetExpress の機能には、アプリケーションで使用する機能と開発ツールがあります。

### B.1.1 機能リスト

機能については、アルファベット順に説明します。

---

#### B.1.1.1 Btrieve インターフェイス

Novell 社の Btrieve ファイル処理システムへのインターフェイスで、COBOL のファイル入出力構文を使用して Btrieve ファイルへアクセスすることができます。このインターフェイスにより、Btrieve ファイルと NetExpress 固有の形式のファイルを同じプログラムで使用できます。

---

#### B.1.1.2 呼び出し可能ファイル ハンドラ

呼び出し可能ファイル ハンドラは NetExpress のファイル ハンドラへのインターフェイスで、これにより、CALL 文を使用してプログラムからファイル ハンドラを呼び出せます。COBOL で使用するすべての編成のファイルに対して低レベルの制御が可能になり、高度なファイル処理プログラムや高度なデータベース処理プログラムを作成できるようになります。また、このファイル ハンドラを使用して C 言語やアセンブラ言語などの他の言語から COBOL 形式のファイルにアクセスすることもできます。

---

#### B.1.1.3 呼び出し可能なリビルド

呼び出し可能なリビルドは、リビルド ユーティリティへのインターフェイスで、これにより、プログラムで CALL 文を使用してリビルド ユーティリティを呼び出すことができます (リビルドの説明を参照してください)。

---

#### B.1.1.4 呼び出し可能な整列

呼び出し可能整列モジュールは、データ ファイルを整列させ、再配列するためのスタンド アロンの整列ルーチンです。デフォルトの COBOL 整列機構よりも高速に処理されます。呼び出しインターフェイスを使用すると、データをより柔軟に整列させることができます。また、別の整列モジュールとの置換も可能になります。

---

#### B.1.1.5 Cblink

Cblink は、システム リンカへの高度なコマンド行インターフェイスです。通常、アプリケーションのプロジェクトにより自動的に呼び出されます。

---

#### B.1.1.6 クライアント サーバー結合

クライアント サーバー結合により、さまざまなプロトコルを使用したネットワーク間で通信するための標準機構が提供されます。クライアント サーバー結合を使用するように作成されたプログラムは、モジュールを交換するだけで、この機構でサポートされるどのプロトコルでも使用できます。

---

#### B.1.1.7 COBOL システム ライブラリ ルーチン

COBOL システム ライブラリ ルーチンは、プログラムで呼び出すことのできるルーチン セットで、COBOL 言語にはない多くの操作を使用できるようになります。

---

#### B.1.1.8 コマンド プロンプト

NetExpress のコマンド プロンプトは、NetExpress の環境設定で実行するコマンド行セッションです。コマンド プロンプトから NetExpress の機能を実行できます。Windows の [スタート] メニューから [プログラム] をクリックします。続いて [Micro Focus NetExpress]、[NetExpress コマンド プロンプト] の順にクリックします。

---

#### B.1.1.9 Common Gateway Interface (CGI) サポート

CGI サポートにより、CGI、ISAPI、NSAPI などの規格に準拠したプログラムを実行できるようになります。その結果、これらのプログラムをリモートの Web ブラウザから起動して Web サーバー ソフトウェアの制御下で実行することができます。

---

#### B.1.1.10 埋め込み HTML

埋め込み HTML は、プログラムの手続き部に HTML を作成できるようにする COBOL の拡張機能で、これにより Web ページの作成と表示が可能になります。

---

#### B.1.1.11 Fileshare

Fileshare を使用し、入出力要求をバケットに圧縮してネットワーク回線に送信すると、ネットワーク入出力が高速化されます。そのため、ファイルを格納するサーバーでファイルを処理することができます。また、いくつかのファイルを論理データベースにリンクさせることもできます。このデータベースにアクセスするときに回復ログを設定すると、データの完全性を高レベルで実現できます。また、エンドユーザーがファイルの変更情報をすべて入力するまでファイルへの書き込みを保留できるトランザクション ログ機能があります。これらの変更は、COMMIT 文により実行し、ROLLBACK により取り消すことができます。

---

#### B.1.1.12 統合プリプロセッサ サポート

統合プリプロセッサ サポートは、コンパイラの拡張機能です。これにより、コンパイラがユーザー定義の言語プリプロセッサを起動して、COBOL 構文以外を COBOL 構文に変換できるようになります。デバッグする場合、プリプロセッサにより文が変更される前の元のソースが表示されます。

---

#### B.1.1.13 Microsoft Transaction Server サポート

Microsoft Transaction Server は、インターネットとイントラネットのアプリケーションを開発、実装、管理するための、コンポーネントに基づくトランザクション処理システムです。NetExpress を使用して COBOL によるトランザクション サーバー コンポーネントを作成できます。

---

#### B.1.1.14 マルチスレッド

マルチスレッド環境では、同じアプリケーションに対して複数の実行用並列パスを設定できます。

---

#### B.1.1.15 各国語サポート

各国語サポート (NLS) を採用すると、プログラムは、実行時にユーザーの居住国に合った文字セット、通貨記号、

および編集記号を自動的に使用します。その国の文字（たとえば、アクセントの付いた文字など）が確実に正しく照合され、メッセージ ファイルから適切な言語でメッセージを取り込むためのライブラリ ルーチンを使用できます。

---

#### B.1.1.16 オブジェクト指向 COBOL 構文

オブジェクト指向構文は、オブジェクト指向 (OO) プログラミングを可能にする COBOL の拡張機能セットです。これは、OO プログラムで使用する定義済みの関数セット、OO クラス ライブラリによりサポートされます。

---

#### B.1.1.17 オブジェクト リクエスト ブローカ (ORB)

オブジェクト リクエスト ブローカ (ORB) は、分散コンピューティング環境でコンポーネント間のクライアント サーバー関係を確立させるミドルウェアです。クライアントが分散コンポーネントのサービスを呼び出すと、ORB がその呼び出しを受信して、処理します。クライアントは、コンポーネントの場所、作成に使用されたプログラム言語、実行に使用されているオペレーティング システム、その他コンポーネントのインターフェイス以外の面を認識する必要がありません。

NetExpress では、OMG の CORBA 規格に準拠した IONA 社の製品 Orbix がサポートされています。COBOL アプリケーションを Orbix のクライアントやサーバーとして機能させるために、Orbix 用の CORBA ウィザードが含まれています。新しい NetExpress Orbix プロジェクトを作成したり、既存の NetExpress プロジェクトを NetExpress Orbix プロジェクトに変換することができます。

---

#### B.1.1.18 ODBC ドライバ

ODBC ドライバを使用すると、アプリケーションは Open Database Connectivity (DOBC) 規格に準拠するデータベースにインターフェイスできるようになります。NetExpress のデモンストレーション プログラムの中には、ODBC ドライバを使用しているものがあります。データ アクセス ウィザードを使用して作成されたアプリケーションでも ODBC ドライバが使用されています。

Microsoft Access データベース用のドライバは、Microsoft 社から Micro Focus 社に提供されています。その他のドライバは、Intersolv 社から Micro Focus 社に提供されています。

注記: サードパーティに対して NetExpress に含まれている ODBC ドライバを配布すること、または、そのライセンスを提供することはできません。ライセンス供与または配布を希望する場合は、まず Microsoft Corporation または Intersolv 社からライセンスを取得する必要があります。別の方法としては、顧客にこのようなドライバのライセンスを直接取得するように要請することもできます。

---

#### B.1.1.19 OLE オートメーション サポート

Object Linking and Embedding (OLE) は Microsoft Windows の機能で、これにより、アプリケーションは、メッセージを読み込んで、オペレーティング システムに OLE オブジェクトとして登録された他のアプリケーションに送信することができます。OLE オートメーション サポートは、NetExpress の機能で、これにより、NetExpress のプログラムやクラスから OLE オブジェクトにメッセージを送信することができます。

クラス ウィザードを使用して、スケルトン オブジェクト COBOL クラスを生成することができます。このクラスには、アプリケーションが OLE オートメーションまたは Microsoft Transaction Server (MTS) をサポートできるようにするクラスも含まれます。また、他に、起動ファイル、レジストリ ファイル、タイプ ライブラリ ファイルなどの OLE に必要なファイルも生成されます。

メソッド ウィザードを使用すると、オブジェクト COBOL クラスにスケルトン メソッドを追加できます。このウィザードでは、メソッドが MTS をサポートしているかどうかについてパラメータと戻り値の種類を入力する機会があります。また、OLE クラスに必要なタイプ ライブラリ情報が追加されます。

---

#### B.1.1.20 OpenESQL

OpenESQL は、NetExpress で提供されるプリプロセッサです。ソース コードに SQL 文を埋め込むことにより、COBOL プログラムから ODBC データ ソースへアクセスできるようになります。

---

#### B.1.1.21 OpenESQL アシスタント

OpenESQL アシスタントは、COBOL プログラムに組み込む SQL 文を生成するウィザードです。

---

#### B.1.1.22 PVCS

PVCS は広く使用されるソース コード管理システム (SCCS) です。中央のサーバーに保存されたソース コードに対する複数ユーザーからのアクセスが管理されます。変更ごとのファイル バージョンを再作成できるように、変更は差分形式で蓄積されます。

---

#### B.1.1.23 リビルド

リビルド ユーティリティは、索引ファイルを維持するためのツールです。このツールを使用すると、キーとデータを再編成し、パフォーマンスを改善できます。既存のファイルに新しいキー構造体を設定したり、破損した索引を再作成したり、順ファイルと相対ファイルを索引編成に変換することができます。このユーティリティでは、大きなファイルを処理できるように 64 ビットのアドレスが使用されています。

---

#### B.1.1.24 整列ユーティリティ

整列ユーティリティにより、オペレーティング システムのプロンプトから呼び出せるファイル整列機能を使用できるようになります。この機能では、呼び出し可能整列モジュールが使用されます。

---

#### B.1.1.25 ソース コード管理システム サポート

ソース コード管理システム (SCCS) サポートにより、NetExpress でプロジェクトが開いている間は、IDE のメニューを使用して SCCS にアクセスすることができます。SCCS サポートで使用できる SCCS は、PVCS と MS Visual SourceSafe です。PVCS は、NetExpress に含まれています。

---

#### B.1.1.26 WebSync リンク

WebSync は、Micro Focus 社の Web サーバーにアクセスするために NetExpress ヘルプに設けられたリンクです。このリンクを使用して、NetExpress に関するサポート、更新、および情報を取得することができます。

---

#### B.1.1.27 Win32 固有の API プログラミング

きわめて低レベルの制御を必要とする場合に、Windows オペレーティング システムのルーチンを呼び出すことができます。

---

## B.2 移行のための機能

NetExpress で提供される次の補足機能は、主に NetExpress への移行に使用されます。オンライン マニュアル『移行ガイド』では、新規アプリケーションについては別の機能を使用するように説明されています。

### B.2.1 機能リスト

機能については、アルファベット順に説明します。

---

#### B.2.1.1 COBSQL

COBSQL は、リレーショナル データベース ベンダから提供される COBOL プリコンパイラと併用するように設計



された統合プリプロセッサです。Oracle Pro\*COBOL Version 1.8 または Sybase Open Client Embedded SQL/COBOL Version 11.1 で使用します。Micro Focus COBOL 製品の旧バージョンでこれらのプリコンパイラのどちらかを使用してアプリケーションを NetExpress に移行する場合、または、UNIX プラットフォームで実装するアプリケーションを作成して Oracle または Sybase のリレーショナル データベースにアクセスする必要がある場合は、COBSQL を使用します。

上記以外のタイプの埋込み SQL アプリケーションを開発する場合は、OpenESQL を使用してください。

---

### B.2.1.2 ダイアログ エディタ

ダイアログ エディタは、ウィンドウとダイアログ ボックスを作成するためのスクリーン ペインタです。Micro Focus の Visual Object COBOL V1.0 に同じツールが含まれています。このシステムを使用して作成されたアプリケーションは、NetExpress のダイアログ エディタを使用して実行したり、さらに開発したりすることができます。

---

### B.2.1.3 オンライン ヘルプ システム

オンライン ヘルプ システム (Hyhelp と Ohbld) は、文字モードのオンライン マニュアルを作成し、画面に表示するためのシステムです。このシステムには、情報の拡張ナビゲーション機能があります。最小限のプログラミングで、プログラムがこのシステムを呼び出してヘルプを表示するように指定できます。文字とグラフィックを使用したオンライン マニュアルや Windows 固有のマニュアルも作成することができます。

---

### B.2.1.4 パネル

パネルは、文字モード画面でウィンドウを操作するためにアプリケーションが呼び出すアプリケーション プログラミング インターフェイス (API) です。ウィンドウの重複、テキストと属性の個別更新または同期更新、ウィンドウでのテキストのスクロール、ポップアップ メニューやプルダウン メニューの使用などが可能になります。

---

### B.2.1.5 画面とキーボードのハンドラ (文字モード)

Adis として知られる画面とキーボードのハンドラは、拡張 Accept 機能または拡張 Display 機能を実行時にサポートするモジュールです。データ部のスクリーン節を使用して全画面文字モードに ACCEPT 文と DISPLAY 文を適用します (これを Web ページを処理するための拡張 ACCEPT 文や拡張 DISPLAY 文と混同しないでください)。

---

#### B.2.1.6 画面とキーボードの構成ツール

画面とキーボードの構成ツール (Adiscf および Keybcf) は、他の COBOL システムとの互換性など、独自の環境や必要性に合わせて文字モード画面とキーボードの処理を構成する 2 つのユーティリティです。

---

#### B.2.1.7 スクリーン節

スクリーン節は、データ部の節で、この節を使用して文字モード画面で表示するフォームを定義します。

---

#### B.2.1.8 ウィンドウ操作サポート

ウィンドウ操作サポートは、画面に文字モードの線や枠を描き、物理画面上に仮想ウィンドウを作成する COBOL 構文で構成されます。この構文を使用すると、これらの仮想ウィンドウをバックグラウンドで表示したり、元に戻したりできます。ACU COBOL と互換性があります。

## 付録C: Web アプリケーション

NetExpress は、従来型から最先端型まで、多様な COBOL アプリケーションの開発に使用できますが、特にクライアント サーバー アプリケーションに適しています。クライアント サーバー アプリケーションでは、データの多くは中央のコンピュータ (サーバー) に格納され、その他多くのコンピュータ (クライアント) からアクセスされます。このようなアプリケーションでは、クエリーは、クライアントで入力され、ローカルで実行されているソフトウェアによってサーバーに転送されます。サーバーでは、ソフトウェアによってデータの問い合わせや更新が行われ、応答が返されます。

今日、世界のコンピュータ ネットワークの多くは、相互接続されてインターネットを形成しています。インターネットは、主に *World Wide Web* として使用されます。これは通常、単に *Web* といいいます。Web サーバーというコンピュータは、Web ページというテキストやグラフィックを含むファイルを格納し、ソフトウェアを実行してこれらのファイルにインターネット経由でアクセスできるようにします。Web ブラウザという適切なソフトウェアを実行しているコンピュータであれば、すべてそれらのファイルにアクセスすることができます。ユーザーが入力できるフィールドを持つ Web ページは、Web フォームといいいます。Web サーバー ソフトウェアは、プログラム (これをサーバー側プログラムといいいます。または、最も一般的な規格である Common Gateway Interface に関連して *CGI プログラム* ともいわれます) を実行してデータを処理できます。また、Web ブラウザは Web サーバーから送信されたフォームを表示できます。

これは、クライアント サーバー アプリケーションの一例ですが、NetExpress は、Web のビジネス フォームとサーバー側プログラムの設計にきわめて適しています。

Web ページには、*HTML* (HyperText MarkupLanguage) という言語の書式設定コマンドが記述されています。手動で Web ページを作成する場合は、これらのコマンドをユーザーが記述しますが、NetExpress を使用すると、コマンドが自動作成されます。

一般的に普及している Web ブラウザには、いくつかの種類があるので、Web ページの表示状態も多少異なることがあります。

多くの企業や組織には、組織内のコンピュータからだけアクセスできる Web サーバーを備えた独自のイントラネットがあります。イントラネット向けのアプリケーションは、World Wide Web 向けのアプリケーションとまったく同じように機能するので、一般的には、どちらも *Web アプリケーション* と呼ぶ傾向にあります。

また、従来型のネットワークを持つ組織も多数あります。ネットワークサーバーにあるファイルは、サーバーのディスクがクライアント マシンにあるかのように、ネットワーク クライアントからアクセスできます。NetExpress を使用して従来型のクライアント サーバー アプリケーションを作成することもできます。この場合、インターネットは関係ないため、Web ページではなく、ウィンドウやダイアログ ボックスのような Windows 型のグラフィック オブジェクトを使用します。このようなアプリケーションを Web アプリケーションと区別して *Windows GUI アプリケーション* と呼びます (必要に応じて、文字モードのテキスト入出力も使用できます)。

Web フォーム、ウィンドウ、およびダイアログ ボックスには、プッシュボタン、エントリ フィールド、チェック ボックスなどのオブジェクトを配置することができます。これらはコントロールといいます。

World Wide Web に関する詳細な説明については、オンライン マニュアル『インターネット アプリケーション』で「*World-Wide Web の概要*」の章を参照してください。

## 付録D: Web サーバーの構成

Web サーバー ソフトウェアを使用して、Web アプリケーションをテストする必要があります。NetExpress で提供される Solo を使用することをお勧めします。後述のとおり、Solo は、自動的に構成されるため、手動で構成する必要はありません。Solo を使用する場合は、この付録をバックグラウンド情報として利用してください。他の Web サーバー ソフトウェアを使用する場合は、後述のとおり、手動で構成する必要があります。

アプリケーションが格納されているプロジェクト ディレクトリを指す Web 共有リソースを Web サーバー上で 2 つ設定する必要があります。1 つはプロジェクト ディレクトリ自体を指し、もう 1 つはプロジェクト ビルド ディレクトリというサブディレクトリを指すように設定します。Solo が起動されると、これら 2 つの Web 共有リソースが、NetExpress に読み込まれた最新のプロジェクトのディレクトリを指すように自動設定されます。他の Web サーバーを使用する場合は、これらを手動で設定する必要があります。

プロジェクトのビルド ディレクトリは、プロジェクトの作成時に自動生成されます。前述のとおり、Form Designer のようなツールで作成されたファイルは、適切なディレクトリに自動的に格納されます。

2 つの Web 共有リソースは、次のとおりです。

- /COBOL/

プロジェクトのソース ディレクトリ。これには、読み取りアクセス権が必要です。アプリケーションの HTML ページを保存する場所です。たとえば、「Web アプリケーションの作成」の章で作成された Goform プロジェクトの場合は、¥NetExpress¥Base¥Demo¥Goform です。

- /cgi-bin/

プロジェクトのビルド ディレクトリ。このディレクトリにアクセスするには、読み取りアクセス権と実行アクセス権が必要です。このディレクトリに、アプリケーションのサーバー側実行可能ファイルを保存します。Goform プロジェクトの場合は、c:¥NetExpress¥Base¥Demo¥Goform¥DEBUG です。

# 索引

[Ctrl+C].....	A-6	Cblink.....	B-2
[Ctrl+V].....	A-6	CGI サポート.....	B-2
[Ctrl+X].....	A-6	CGI プログラム.....	C-1
[コンテキスト ヘルプ] ボタン.....	A-1	COBOL 構文	
[最小化] ボタン.....	A-1	ヘルプの表示.....	3-2
[最大化] ボタン.....	A-1	COBOL システム ライブラリ ルーチン.....	B-2
[送信] ボタン.....	7-2	COBSQL.....	B-6
[閉じる] ボタン.....	A-1	Common Gateway Interface.....	B-2
[元のサイズに戻す] ボタン.....	A-1	CORBA.....	B-4
「データ型」列.....	10-3	CORBA ウィザード.....	B-4
「開く」		Dialog System	
ダイアログ ボックス.....	A-1	概説.....	1-2
『入門書』		基本チュートリアル.....	12-1
使用方法.....	4-2	生成ファイル.....	12-12
『入門書』のチュートリアル.....	4-1	Dialog System のダイアログ	
64 ビットのファイル アドレス.....	B-5	基本チュートリアル.....	12-10
ACCEPT/DISPLAY		DSN.....	11-1
拡張.....	7-2	Dtoldemo.....	6-1
Acrobat.....	3-2	EXEC HTML.....	7-2
Adis.....	B-7	Execution Environment (EE).....	15-1, 15-2, 15-3
Adobe Acrobat.....	3-2	Fileshare.....	B-3
app ファイル.....	5-1	Form Designer	
Brieve インターフェイス.....	B-1	概説.....	1-2

基本チュートリアル.....	7-1	起動.....	5-2
生成ファイル.....	8-8	基本チュートリアル.....	5-1
Form Express		IDE での実行.....	5-6
基本チュートリアル.....	10-1	IDE の起動.....	5-2
生成ファイル.....	10-5	int ファイル	
GUI		プロジェクト.....	5-5
使用するもの.....	1-2	IONA.....	B-4
GUI アプリケーション		Locking プログラム.....	5-7
基本チュートリアル.....	12-1	mfuser.....	5-1
GUI アプリケーションの作成		Micro Focus	
基本チュートリアル.....	12-1	Web による.....	3-2
GUI 用インターフェイス		所在地と電話番号.....	3-2
基本チュートリアル.....	12-1	Micro Focus への連絡	
GUI 用ユーザー インターフェイス		所在地と電話番号.....	3-2
基本チュートリアル.....	12-1	Microsoft Transaction Server サポート.....	B-3
HTML.....	B-3, C-1	NetExpress	
HTML フォーム.....	8-1	基本チュートリアル.....	5-1
HTML ページ ウィザード.....	8-2	NetExpress アプリケーション	
概説.....	1-2	説明.....	C-1
HTML 「ラベル」フィールド.....	10-4	NetExpress コマンド プロンプト.....	B-2
HyperText MarkupLanguage.....	C-1	ODBC.....	11-1
IDE		ODBC 使用許諾契約.....	B-4
Dialog System が読み込まれた.....	12-3	ODBC ドライバ.....	B-4
Form Designer を読み込んだ.....	8-3	OMG.....	B-4
概説.....	1-1	OO クラス ブラウザ.....	B-4

OO クラス ライブラリ .....	B-4
Open Database Connectivity .....	11-1, B-4
OpenESQL .....	B-5
OpenESQL アシスタント .....	B-5
Operating System Extension (OSX).....	15-1, 15-2, 15-4
ORB .....	B-4
Orbix .....	B-4
PICTURE文字列 .....	12-8
PVCS .....	1-3
Samba.....	2-1
Sample mdb データベース.....	11-1
SCCS.....	1-3
SCCS サポート .....	B-6
SCP .....	2-1
Server Control Program .....	2-1
Solo.....	7-1
概説 .....	1-2
基本チュートリアル.....	9-1
UNIX アプリケーション	
実行 .....	14-8
パブリッシュ.....	14-7
UNIX アプリケーションの実行.....	14-8
UNIX アプリケーションの展開	
使用するもの.....	1-3
UNIX アプリケーションのパブリッシュ .....	14-7

UNIX 移植性	
UNIX オプション .....	14-3
UNIX オプション	
概説 .....	1-3
基本チュートリアル .....	14-1
生成ファイル.....	14-8
UNIX サーバー	
使用するもの .....	2-1
UNIX での実装	
基本チュートリアル .....	14-1
UNIX 展開	
使用するもの .....	1-3
UNIX ネットワーク	
使用するもの .....	2-1
Web アプリケーション	
起動 .....	7-2
基本チュートリアル .....	7-1
作成 .....	8-1
実行 .....	9-3, 10-6
使用するもの .....	1-2
説明 .....	C-1
デバッグ .....	9-4
Web アプリケーションのアニメート .....	9-4
Web アプリケーションの起動 .....	7-2
Web アプリケーションの作成	



概要 .....	8-1
基本チュートリアル.....	7-1
Web アプリケーションの実行 .....	9-3, 10-6
Web アプリケーションのテスト	
使用するもの.....	1-2
Web アプリケーションのデバッグ.....	9-4
Web アプリケーションへのリンク.....	7-2
Web 共有リソース.....	D-1
Web サーバー .....	7-1, C-1
基本チュートリアル.....	9-1
構成 .....	D-1
Web サーバー Solo	
概説 .....	1-2
Web サーバーの構成 .....	D-1
Web データベース アプリケーション	
基本チュートリアル.....	11-1
実行 .....	11-6
Web データベース アプリケーションの作成.....	11-2
基本チュートリアル.....	11-1
Web データベース アプリケーションの実行.....	11-6
Web フォーム .....	C-1
Web ブラウザ .....	C-1
使用するもの.....	2-1
Web プログラム	
基本チュートリアル.....	9-1

Web ページ.....	C-1
戻る .....	4-2
Web 用インターフェイス	
概要.....	8-1
チュートリアル .....	7-1
Web 用ブラウザ	
使用するもの .....	2-1
Web 用ユーザー インターフェイス	
概要 .....	8-1
基本チュートリアル .....	7-1
Web を使用したデータベースのインターフェイス	
基本チュートリアル .....	11-1
Web を使用したデータベースのユーザー インターフェイス	
基本チュートリアル .....	11-1
WebSync.....	3-2, B-6
Win32 固有の GUI.....	B-6
Win32 ソフトウェア開発キット	
概要 .....	2-1
Windows.....	4-1
Windows 95.....	4-1
Windows 98.....	4-1
Windows GUI	
使用するもの .....	1-2
Windows GUI アプリケーション	

実行 .....	13-3
Windows GUI アプリケーションの実行.....	13-3
Windows GUI プログラム	
基本チュートリアル.....	13-1
Windows NT.....	4-1
Windows エクスプローラ.....	A-3
World Wide Web.....	C-1
アスタリスク	
ツリーの展開.....	6-6
アニメーション.....	5-6
データ確認.....	5-9
アニメーション コマンド	
カーソル位置まで実行.....	5-8
指定範囲を実行.....	5-8
ステップ実行.....	5-8
アニメート.....	5-1, 5-8
基本チュートリアル.....	5-1
使用するもの.....	1-1
「アプリケーション出力」ウィンドウ	
閉じないで非表示.....	5-10
アプリケーションの実装, パッケージ化と出荷15-4,	
15-5	
アプリケーションのビルド.....	9-3
移行ガイド	
概要.....	3-2

移植性	
UNIX オプション.....	14-3
移植性のチェック	
UNIX オプション.....	14-3
移動	
切り取りと貼り付けによる.....	A-6
ドラッグ & ドロップによる.....	A-6
リストの列.....	A-3
印刷	
データ ファイル.....	6-8
印刷物.....	3-2
インターネット.....	C-1
インターネット アプリケーション ウィザード11-1,	
11-2	
概説.....	1-2
基本チュートリアル.....	11-1
生成ファイル.....	11-5
インターネット エクスプローラ	
概要.....	2-1
イントラネット.....	C-1
ウィンドウ.....	8-1
境界.....	A-7
サイズ変更.....	A-7
ドッキング.....	A-7
フローティング.....	A-7

ウィンドウ ボタン.....	A-1
ウィンドウ操作サポート.....	B-8
埋め込み HTML.....	7-2, B-3
エクスペローラ	
インターネット.....	2-1
オートメーション サポート.....	B-5
大文字小文字の区別	
チュートリアル.....	4-1
オブジェクト	
定義.....	12-7
オブジェクト ツールバー.....	8-3, 12-7
オブジェクト リクエスト プロカー.....	B-4
オブジェクト指向 COBOL 構文.....	B-4
オブジェクトの定義.....	12-7
オンライン ヘルプ システム.....	B-7
オンライン マニュアル.....	3-1
カーソル位置まで実行.....	5-8
開発環境	
概説.....	1-1
起動.....	5-2
基本チュートリアル.....	5-1
開発環境の起動.....	5-2
概要	
インターネット アプリケーション ウィザード.....	8-1
拡張 ACCEPT/DISPLAY.....	7-2

拡張子	
表示と非表示.....	A-3
各国語サポート.....	B-3
画面	
Dialog System が読み込まれた.....	12-3
Form Designer を読み込んだ.....	8-3
画面とキーボードの構成ツール.....	B-8
画面とキーボードのハンドラ.....	B-7
関連付けられたプログラム	
基本チュートリアル.....	13-1
編集.....	13-2
関連付けられたプログラムの編集.....	13-2
関連付けられたマスタ フィールド.....	12-8
技術サポート	
Web による.....	3-2
既存の COBOL	
基本チュートリアル.....	10-1
使用するもの.....	1-2
既存の COBOL の変換	
基本チュートリアル.....	10-1
使用するもの.....	1-2
行.....	11-2
強調表示	
リスト項目.....	A-4
切り取りと貼り付け.....	A-6

クライアント.....	C-1	コンテキスト メニュー .....	A-5
クライアント サーバー.....	C-1	コンテナ	
クライアント サーバー 結合.....	B-2	再配置.....	A-3
クラス ウィザード.....	B-5	コントロール.....	C-2
クラス ブラウザ.....	B-4	コントロール, 追加.....	12-7
クラス ライブラリ.....	B-4	コントロールの追加.....	12-7
クリック		コンバータ	
チュートリアル.....	4-1	チュートリアル .....	6-1
グローバル ダイアログ.....	12-10	コンパイル	
クロスリファレンス.....	4-2	使用するもの.....	1-1
クロスリファレンスのクリック.....	4-2	サーバー.....	C-1
言語サポート.....	B-3	Web.....	C-1
構文		ネットワーク.....	C-1
ヘルプの表示.....	3-2	サーバー側プログラム	
国際化.....	B-3	基本チュートリアル.....	9-1
このマニュアルの使用方法.....	4-2	構造.....	7-2
コピー		編集.....	9-2
切り取りと貼り付けによる.....	A-6	サーバー側プログラムの編集.....	9-2
ドラッグ & ドロップによる.....	A-6	サイズ変更	
コピーファイル		ウィンドウ.....	A-7
表示.....	9-2	リストの列.....	A-3
コマンド プロンプト.....	B-2	再配置	
コメント		リストの列.....	A-3
Dialog System.....	12-11	索引	
コンテキスト ヘルプ.....	3-1	マニュアルとヘルプ.....	3-2

## 索引ファイル

維持 .....	B-5
作成	
条件付きレコード レイアウト .....	6-6
データ ファイル.....	6-10
デフォルトのレコード レイアウト .....	6-6
ファイルまたはフォルダ.....	A-2
レコード レイアウト ファイル.....	6-5
差分 .....	1-3
サポート	
Web による .....	3-2
実行時ライセンス.....	15-2
指定範囲を実行.....	5-8
使用許諾契約	
ODBC .....	B-4
条件付きレコード レイアウト .....	6-6
情報	
検索 .....	3-1
情報検索 .....	3-1
情報の検索.....	3-1
情報の検索方法.....	3-1
使用方法	
このマニュアル.....	4-2
書籍 .....	3-1
使用方法.....	3-1

## 新規データ ファイル

作成 .....	6-10
スクリーン節 .....	B-8
スクリーンセット	
作成 .....	12-2
スクリーンセットの作成 .....	12-2
スクリーンセットの処理	
基本チュートリアル .....	13-1
ステップ実行 .....	5-8
すべてをリビルド .....	5-6
生成されたファイル	
Dialog System によって.....	12-12
Form Designer によって .....	8-8
Form Express によって.....	10-5
UNIX オプションによって .....	14-8
インターネット アプリケーション ウィザード.....	11-5
整理.....	B-2, B-6
全画面表示 .....	A-1
選択	
リスト項目 .....	A-4
統合開発環境	
概説 .....	1-1
ソース コード管理	
使用するもの .....	1-3
ソース コード管理システム .....	1-3

ソース コード管理システム サポート .....	B-6	Windows GUI プログラム .....	13-1
ソリューション ガイド		インターネット アプリケーション ウィザード	11-1
概要 .....	3-2	既存の COBOL の変換 .....	10-1
ダイアログ エディタ .....	B-7	サーバー側プログラム .....	9-1
ダイアログ ボックス .....	8-1	統合開発環境 .....	5-1
「開く」 .....	A-1	編集 .....	5-1
ファイルの追加 .....	A-1	チュートリアル マップ .....	4-3
対称 .....	7-2	ツール情報 .....	5-6
タスクバー .....	A-1	ツールバー	
チュートリアル		IDE .....	5-6
『入門書』 .....	4-1	オブジェクト .....	8-3
Dialog System .....	12-1	ツリー ビュー .....	A-3
Dialog System のダイアログ .....	12-10	ディレクトリ	
Form Designer .....	7-1	チュートリアル .....	4-1
Form Express .....	10-1	低レベルのファイル処理 .....	B-1
GUI アプリケーション .....	12-1	データ	
Solo .....	9-1	デバッグ時に確認 .....	5-9
UNIX オプション .....	14-1	データ ツール	
UNIX での実装 .....	14-1	概説 .....	1-2
Web アプリケーション .....	7-1	概要 .....	6-1
Web アプリケーションの実行 .....	9-1	チュートリアル .....	6-1
Web アプリケーションのテスト .....	9-1	データ ファイル	
Web アプリケーションのデバッグ .....	9-1	16 進表示 .....	6-8
Web データベース アプリケーション .....	11-1	印刷 .....	6-8
Web への移行 .....	10-1	作成 .....	6-10

書式設定されたビュー .....	6-7	データベースの更新 .....	11-8
書式設定されていないビュー .....	6-3	データベースのナビゲート .....	11-6
複数データ ファイルの編集 .....	6-9	データベースのフィルタ処理 .....	11-8
変換 .....	6-2	データベースへのアクセス	
編集 .....	6-1	使用するもの .....	1-2
データ ファイル エディタ		テキスト	
チュートリアル .....	6-1	フォームへの追加 .....	8-5
データ ファイル コンバータ		テキストのプロパティ .....	12-8
チュートリアル .....	6-1	テキストボックス	
データ ブロック .....	12-5	定義 .....	8-5
データ確認 .....	5-9	フォームへの追加 .....	8-5
データソース .....	11-1	テスト	
データソース名 .....	11-1	基本チュートリアル .....	9-1
データベース		デバッグ .....	5-1, 5-8
クエリー .....	11-8	カーソル位置まで実行 .....	5-8
更新 .....	11-8	基本チュートリアル .....	5-1
使用するもの .....	1-2	指定範囲を実行 .....	5-8
ナビゲート .....	11-6	使用するもの .....	1-1
フィルタ処理 .....	11-8	ステップ実行 .....	5-8
データベース sample mdb .....	11-1	データ確認 .....	5-9
データベース アクセス用アプリケーション		デフォルトのレコード レイアウト .....	6-6
作成 .....	11-2	デモ アプリケーション	
データベース アプリケーション		Locking .....	5-7
基本チュートリアル .....	11-1	デモンストレーション アプリケーション .....	5-2
データベースのクエリー .....	11-8	統合開発環境の起動 .....	5-2

統合開発環境	リストの列 .....A-3
起動 .....5-2	パネル .....B-7
基本チュートリアル.....5-1	非対称 ..... 7-2
統合プリプロセッサ サポート ..... B-3	非表示
閉じる	「アプリケーション出力」ウィンドウ ..... 5-10
NetExpress .....5-10	ファイル拡張子 .....A-3
ウィンドウ.....5-10	表示
出力ウィンドウ.....5-10	書式設定されたデータ ファイル..... 6-7
閉じない「アプリケーション出力」ウィンドウ5-10	書式設定されていないデータ ファイル..... 6-3
プロジェクト.....5-10	データ ファイル プロパティ..... 6-5
ドッキング	データ ファイルの 16 進表示..... 6-8
ウィンドウ.....A-7	ファイル拡張子 .....A-3
ドライブ	フォルダ .....A-2
チュートリアル.....4-2	リストの列 .....A-3
ドラッグ & ドロップ ..... A-6	標準ボタン
ドロップダウン リスト.....A-5	ウィンドウの .....A-1
ネットワーク.....C-1	ファイル
ネットワークサーバー.....C-1	作成.....A-2
ネットワーク通信.....B-2	データ ファイルの変換 ..... 6-2
ネットワークでのファイル ロック ..... B-3	レコード レイアウト ..... 6-5
ネットワークでのファイル共有.....B-3	ファイル拡張子
ネットワークと UNIX.....2-1	表示と非表示 .....A-3
バージョン番号	ファイルの維持
チュートリアル.....4-1	使用するもの ..... 1-2
配置	ファイルの追加



ダイアログ ボックス.....	A-1	プラス記号 (+)	
フォーム		ツリー ビュー .....	A-3
Form Express で作成.....	10-2	プリプロセッサ	
コントロールの追加.....	8-5	統合された .....	B-3
作成 .....	8-2	フローティング	
テスト .....	8-7, 8-9	ウィンドウ .....	A-7
フォーム処理		プログラム例 .....	5-2
基本チュートリアル.....	9-1	Locking.....	5-7
フォームの作成.....	8-2	プロジェクト .....	5-1, B-5
Form Express .....	10-2	Dtoldemo .....	6-1
フォームの設計.....	8-5	概説 .....	1-1
フォームのテスト.....	8-7, 8-9	基本チュートリアル .....	5-1
フォームへのコントロールの追加 .....	8-5	作成 .....	8-2, 12-2, 14-2
フォルダ		ビルド .....	5-5
作成 .....	A-2	読み込み .....	5-4
チュートリアル.....	4-1	プロジェクト フォルダ .....	5-1
変更 .....	A-1	プロジェクト内に存在しないファイル.....	5-5
フォルダ オプション.....	A-3	プロジェクト内のファイル	
フォルダ階層.....	A-2	存在しないファイル .....	5-5
フォルダの移動.....	A-1	プロジェクトの再読み込み	
フォルダの階層.....	A-2	チュートリアル .....	4-2
複数列リスト		プロジェクトの作成 .....	8-2, 12-2, 14-2
再配置 .....	A-3	プロジェクトのビルド .....	5-5
プラス記号		基本チュートリアル .....	5-1
ツリー ビュー.....	A-3	プロジェクトの読み込み .....	5-4

チュートリアル.....	4-2	最小化.....	A-1
プロジェクトのリビルド		最大化.....	A-1
基本チュートリアル.....	5-1	閉じる.....	A-1
プロトタイプ作成.....	12-13	フォームへの追加.....	8-5
プロパティ		元のサイズに戻す.....	A-1
データ ファイル.....	6-5	ホットリンク.....	4-2
ページ		ポップアップ メニュー.....	3-2, A-5
戻る.....	4-2	マイナス記号	
ヘルプ.....	3-1	ツリー ビュー.....	A-3
COBOL 構文.....	3-2	マイナス記号 (-)	
画面の項目.....	3-2	ツリー ビュー.....	A-3
使用方法.....	3-1	マウス ポインタ	
変換		形.....	A-7
データ ファイル.....	6-2	マウス ボタン	
変更		チュートリアル.....	4-1
フォルダ.....	A-1	マスタ フィールド.....	12-8
編集		マニュアル	
基本チュートリアル.....	5-1	使用方法.....	3-1
使用するもの.....	1-1	マルチスレッド.....	B-3
複数のデータ ファイル.....	6-9	右クリック.....	A-5
保存		メイン ウィンドウ	
レコード レイアウト ファイル.....	6-7	Form Express.....	10-3
ボタン		メソッド ウィザード.....	B-5
ウィンドウの.....	A-1	メニュー	
コンテキスト ヘルプ.....	A-1	ポップアップとコンテキスト.....	A-5

目次	再配置.....A-3
戻る.....4-2	リビルド.....5-6, B-5
文字モード.....C-1	リリース ノート.....3-2
戻る	リンク.....4-2
Web ページへ.....4-2	使用するもの.....1-1
問題	マニュアル.....3-2
ヘルプの表示.....3-2	レガシー COBOL
ユーザー プロジェクトの場所 mfuser.....5-1	基本チュートリアル.....10-1
呼び出し可能な整列.....B-2	使用するもの.....1-2
呼び出し可能なリビルド.....B-1	レコード レイアウト ファイル
呼び出し可能ファイル ハンドラ.....B-1	作成.....6-5
ライブラリ ルーチン.....B-2	保存.....6-7
リスト	列.....11-2
項目の選択.....A-4	列見出し
ドロップダウン.....A-5	再配置.....A-3
複数列.....A-3	ローカル ダイアログ.....12-10
リストの列	