



クイックスタート



Borland
C++Builder 6
for Windows

Borland International, Inc., 100 Borland Way
P.O. Box 660001, Scotts Valley, CA 95067-0001

C++Builder のライセンス既定および限定付き保証に従って配布が可能なファイルについては、C++Builder 製品のルートディレクトリにある DEPLOY ドキュメントファイルを参照してください。

Borland Software Corporation は、本書に記載されているアプリケーションに対する特許を取得または申請している場合があります。本書の提供は、これらの特許に関する権利を付与することを意味するものではありません。適用される特許については、製品 CD または [バージョン情報] ダイアログボックスを参照してください。

Borland のすべてのブランドおよび製品名は、米国 Borland Software Corporation の米国における商標または登録商標です。その他のブランドまたは製品名は、その著作権所有者の商標または登録商標です。

COPYRIGHT © 1983-2002 Borland Software Corporation. All rights reserved.

目次

第 1 章			
はじめに	1-1		
C++Builder の概要	1-1	SQL エクスプローラ (データベース エクスプローラ) 3-11	
C++Builder の登録	1-1	Database Desktop 3-12	
情報の検索	1-3	データディクショナリ 3-12	
オンラインヘルプ	1-3	カスタムコンポーネント 3-12	
[F1] ヘルプ	1-4	DLL 3-12	
開発者サポートサービスと Web サイト	1-6	COM と ActiveX 3-13	
このマニュアルで使われる書体と表記	1-6	タイプライブラリ 3-13	
第 2 章			
C++Builder の開発環境	2-1	第 4 章	
C++Builder の起動	2-1	チュートリアル: テキストエディタの 作成	4-1
IDE	2-1	新規アプリケーションの開始	4-1
メニューとツールバー	2-2	プロパティ値の設定	4-2
コンポーネントパレット, フォームデザイナー, オブジェクトインスペクタ	2-3	フォームへのコンポーネントの追加	4-3
オブジェクトツリー	2-4	メニューとツールバーのサポートを追加する	4-6
オブジェクトリポジトリ	2-5	アクションマネージャエディタと アクションリストエディタの違い	4-6
コードエディタ	2-6	メニューとツールバーのイメージの追加 (Enterprise/Professional)	4-7
支援機能	2-6	アクションマネージャにアクションを追加する (Enterprise/Professional)	4-8
コードのブラウズ	2-7	標準アクションの追加 (Enterprise/Professional)	4-11
[ダイアグラム] ページ	2-7	メニューの追加 (Enterprise/Professional)	4-12
フォームのコードの表示	2-8	ツールバーの追加 (Enterprise/Professional)	4-13
クラスエクスプローラ	2-9	イメージリストとイメージの追加 (Personal 版)	4-14
プロジェクトマネージャ	2-9	アクションリストにアクションを追加する (Personal 版)	4-15
To-Do リスト	2-10	アクションリストに標準アクションを追加する (Personal 版)	4-17
第 3 章		メニューの追加 (Personal 版)	4-19
C++Builder を使ったプログラミング	3-1	ツールバーの追加 (Personal 版)	4-21
プロジェクトの作成	3-1	テキスト領域のクリア (すべての版)	4-23
データモジュールを追加する	3-2	イベントハンドラの作成	4-23
ユーザーインターフェースの作成	3-2	新規作成コマンドのイベントハンドラの作成	4-24
フォームにコンポーネントを追加する	3-2	開くコマンドのイベントハンドラの作成	4-26
コンポーネントのプロパティを設定する	3-4	上書き保存コマンドのイベントハンドラの作成	4-27
コードの記述	3-5	名前を付けて保存コマンドの イベントハンドラの作成	4-28
イベントハンドラの作成	3-5	ヘルプファイルの作成	4-30
VCL/CLX ライブラリを使う	3-6	ヘルプ 目次コマンドの イベントハンドラの作成	4-30
プロジェクトのコンパイルとデバッグ	3-7	ヘルプ キーワードコマンドの イベントハンドラの作成	4-31
アプリケーションの配布	3-8	バージョン情報ダイアログボックスの作成	4-32
アプリケーションの国際化対応	3-8	アプリケーションを完成させる	4-34
プロジェクトの種類	3-9		
CLX アプリケーション	3-9		
Web サーバーアプリケーション	3-10		
データベースアプリケーション	3-10		
BDE Administrator	3-11		

第5章

チュートリアル：CLX データベース アプリケーションの作成	5-1
データベースアーキテクチャの概要	5-1
新しいCLX アプリケーションを作成する	5-2
データアクセスコンポーネントのセットアップ	5-3
データベース接続のセットアップ	5-3
データセットのセットアップ	5-4
プロバイダ、クライアントデータセット、 データソースのセットアップ	5-5
ユーザーインターフェースの設計	5-6
グリッドとナビゲーションバーの作成	5-6
メニューのサポートの追加	5-8
メニューの追加	5-10
ボタンを追加する	5-11
タイトルとイメージの表示	5-12
イベントハンドラの作成	5-13
[今すぐ更新] コマンドの イベントハンドラの作成	5-13
終了コマンドのイベントハンドラの作成	5-14
FormClose イベントハンドラの作成	5-15

第6章

デスクトップのカスタマイズ	6-1
作業領域の整理	6-1
メニューとツールバーの配置	6-1
ツールウィンドウのドッキング	6-2
デスクトップ配置の保存	6-5
コンポーネントパレットのカスタマイズ	6-5
コンポーネントパレットの再構成	6-5
コンポーネントテンプレートの作成	6-6
コンポーネントパッケージのインストール	6-7
フレームを使う	6-7
ActiveX コントロールの追加	6-8
プロジェクトオプションの設定	6-8
デフォルトプロジェクトオプションの設定	6-8
プロジェクトおよびフォームテンプレートを デフォルトに指定する	6-9
オブジェクトリポジトリへのテンプレートの 追加	6-10
ツールの環境設定	6-10
フォームデザイナのカスタマイズ	6-10
コードエディタのカスタマイズ	6-11

索引	I-1
----	-----

第1章

はじめに

この『クイックスタート』では、C++Builder をすぐに使い始めることができるように、その開発環境と機能の概要を説明します。また、C++Builder で使用できるツールと機能の詳細説明がどこに記載されているかも示しています。

第2章「C++Builder の開発環境」では、統合デスクトップ開発環境 (IDE) と呼ばれる C++Builder デスクトップの主要なツールについて説明しています。第3章「C++Builder を使ったプログラミング」では、そのツールを使ってアプリケーションを作成する方法を説明します。第4章「チュートリアル：テキストエディタの作成」では、演習としてテキストエディタプログラムの作成を順を追って説明しています。第5章「チュートリアル：CLX データベースアプリケーションの作成」では、データベースアプリケーションを作成する手順を説明します。第6章「デスクトップのカスタマイズ」では、開発のニーズに合わせて C++Builder IDE をカスタマイズする方法を説明しています。

C++Builder の概要

C++Builder は、高速アプリケーション開発 (RAD) 用のオブジェクト指向のビジュアルなプログラミング環境です。C++Builder を使うと、Microsoft Windows XP、Windows 2000、Windows 98 用の非常に効率的なアプリケーションを、最小限のコーディング作業で作成することができます。

C++Builder には、アプリケーションの開発、テスト、および配布のために必要な、再使用可能なコンポーネントのライブラリ、一連の設計ツール、アプリケーションとフォームのテンプレート、プログラミングウィザードなどがすべて含まれています。

C++Builder の登録

C++Builder を登録する方法はいくつかあります。C++Builder をインストール後に初めて起動すると、最初にインストール番号とインストールキーの入力を求められます。入力すると、登録のダイアログに3つの選択肢が示されます。

C++Builder の登録

- オンラインで取得する

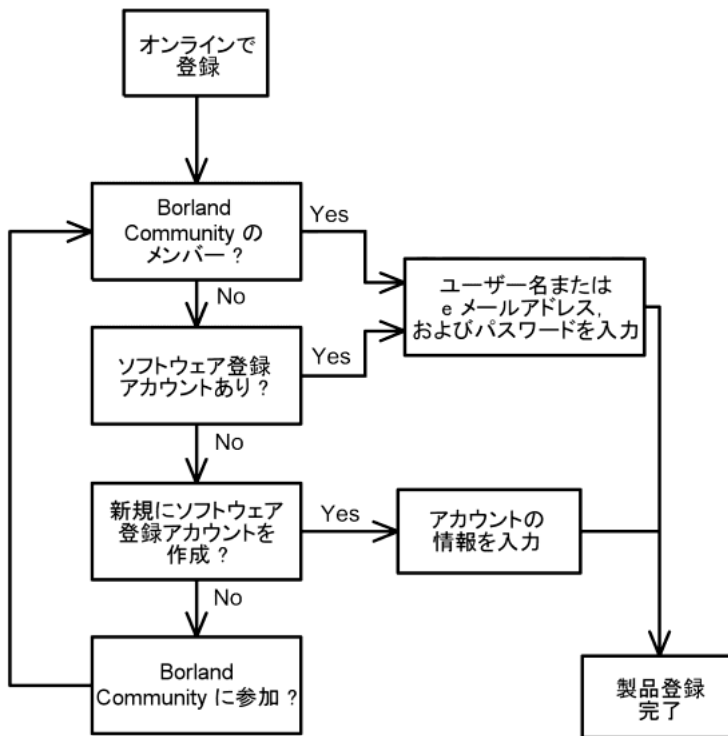
このオプションは、既存のインターネット接続を使ってオンラインで登録を行います。

- ボーランドへの問い合わせ

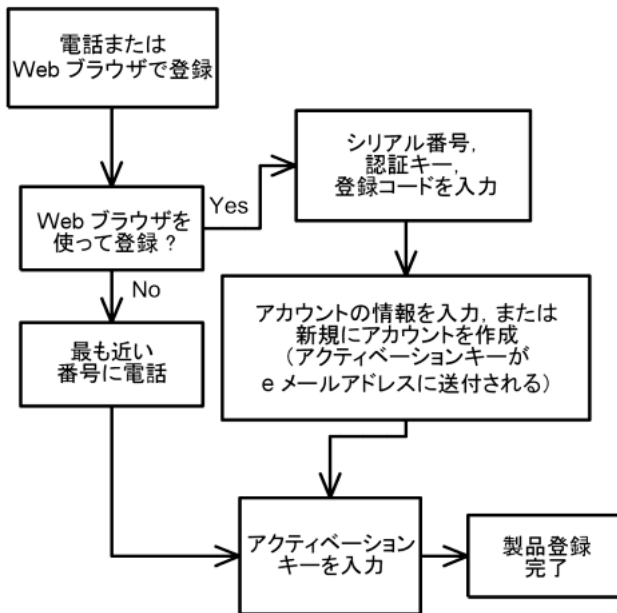
このオプションは、電話または Web ブラウザを使って登録を行います。Eメールでアクティベーションキーを取得した場合は、このオプションを使ってそのキーを入力します。

- 後で取得する

オンラインで C++Builder を登録するのが最も簡単な方法ですが、その時点でインターネットへの接続が有効になっている必要があります。すでに Borland Community のメンバーになっている場合、またはソフトウェア登録アカウントを持っている場合は、そのアカウント情報を入力するだけです。それによって C++Builder は自動的に登録されます。そうでない場合、登録のプロセスでこのアカウントを作成することになります。



2 番目のオプション（電話または Web ブラウザでの登録）は、インターネットに接続していないマシンにインストールした場合や、ファイアウォールによってオンライン登録がブロックされてしまう場合、またアクティベーションキーをすでに取得している場合に使用します。



メモ 特定の理由がない限り、オンライン登録を使うようにしてください。

情報の検索

C++Builder に関する情報は、次の方法で探すことができます。

- オンラインヘルプ
- 印刷マニュアル
- Borland 開発者サポートサービスおよび Web サイト

このバージョンの新しい機能については、オンラインヘルプの「新しい機能」を参照するか、Borland の Web サイトにアクセスしてください。

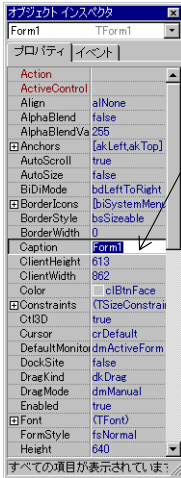
オンラインヘルプ

オンラインヘルプシステムでは、ユーザーインターフェースの機能、言語の実装、プログラミングのタスクに加えて、ビジュアルコンポーネントライブラリ (VCL) と Borland のクロスプラットフォーム用コンポーネントライブラリ (CLX) のコンポーネントに関する詳細な情報が提供されています。また、『開発者ガイド』の内容と、C++Builder に付属する他の機能のヘルプも含まれています。

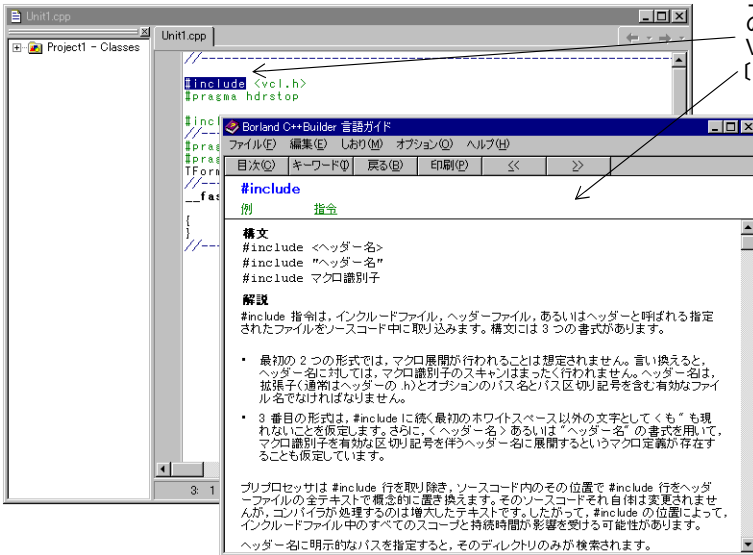
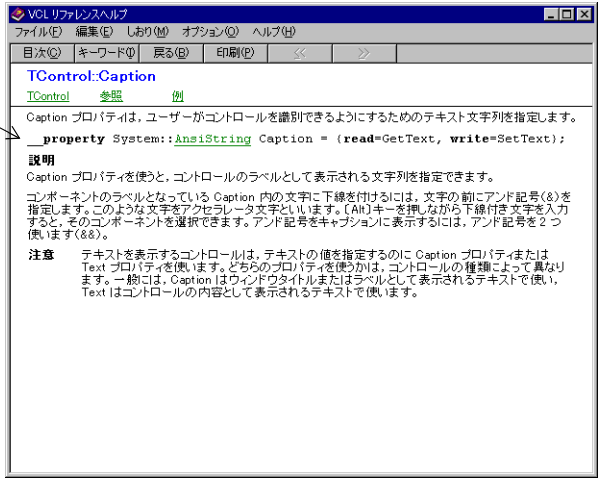
ヘルプの目次を表示するには、[ヘルプ | C++Builder ヘルプ] または [ヘルプ | C++Builder ツール] を選択して、[目次] タブをクリックします。VCL または CLX のオブジェクトや関連トピックを検索するには、[キーワード] または [テキスト検索] タブをクリックして、検索するテキストを入力します。

{F1} ヘルプ

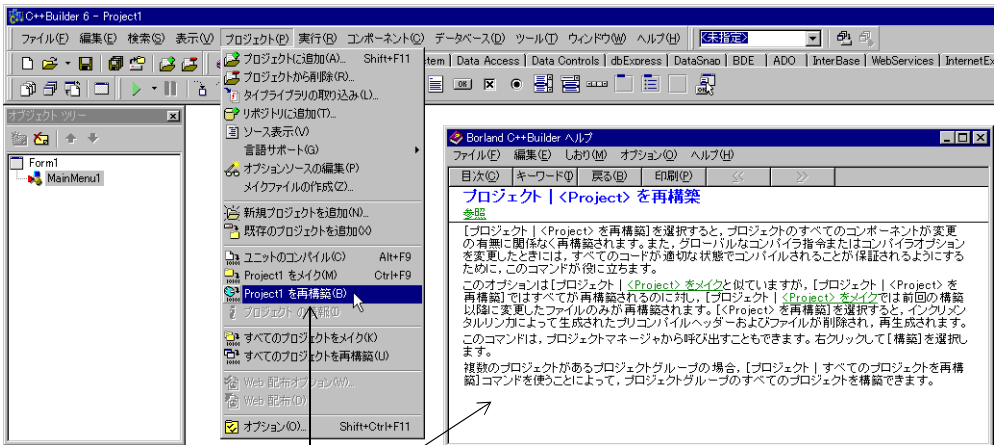
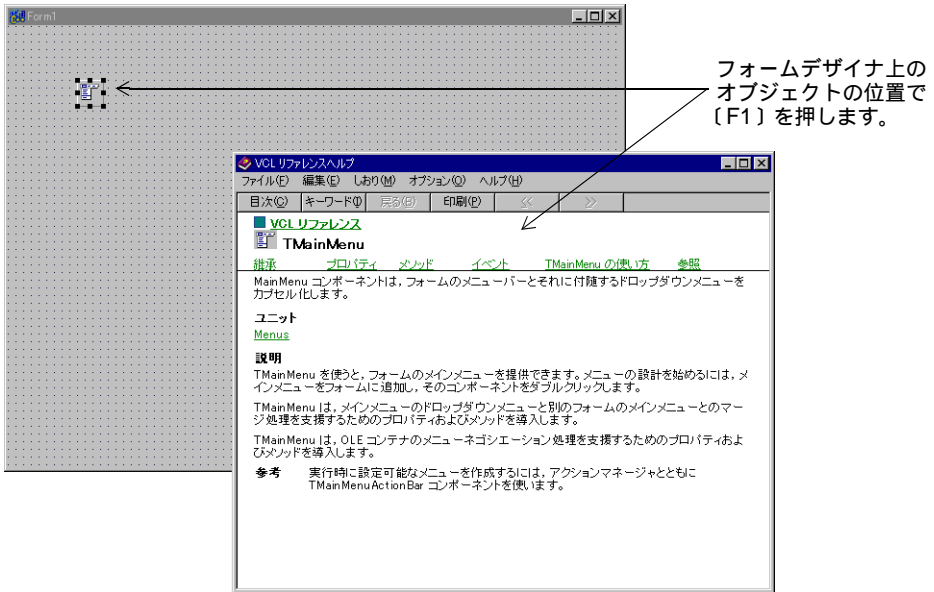
状況関知型ヘルプは、目的の項目を選択して {F1} を押すことにより、VCL と CLX、メニュー項目やダイアログボックス、ウィンドウ、ツールバー、コンポーネントなど、開発環境のどんな部分についても表示できます。



オブジェクトインスペクタ内のプロパティまたはイベント名を選択して {F1} を押すと、ヘルプが表示されます。



コードエディタで言語のキーワードまたは VCL の要素の位置で {F1} を押します。



メニューコマンド、ダイアログボックス、またはウィンドウを選択して〔F1〕を押すと、その項目に関するヘルプが表示されます。

どのダイアログボックスでも、[ヘルプ] ボタンを押せば状況に応じたオンラインヘルプが表示されます。

コンパイラとリンカのエラーメッセージは、コードエディタの下の特別なウィンドウに表示されます。コンパイラエラーに関するヘルプを見るには、リストからエラーメッセージを選択して〔F1〕を押します。

開発者サポートサービスと Web サイト

ポーランドでは、多様な開発者のニーズに合わせた各種のサポートも提供しています。開発者サポートの詳細については、次の URL を参照してください。 <http://www.inprise.co.jp/support/>

このサイトには、C++Builder に関する書籍の一覧や、C++Builder についての補足的な技術情報、FAQ（良くある質問と回答）などもあります。

このマニュアルで使われる書体と表記

このマニュアルでは、次の表に示した書体と記号を使って特殊なテキストを表します。

書体または記号	説明
Monospace type	等幅のこの書体は画面に表示される内容やコードの内容を表します。キーボードから入力する内容を示す場合もあります。
Boldface	本文またはコード内で太字で示された項目は、予約語またはコンパイラオプションです。
[メニュー名 コマンド名]	この表記はメニューバーのメニューとその中のコマンドを示します。 例：[ファイル 終了] を選択します。また、画面上の文字も [] で囲んで表記します。 例：[ファイルを開く] ダイアログボックスの [ファイル名] にファイル名を入力し、[OK] を選択します。
[Keycaps]	この表記はキーボードのキーを示します。 例：[Esc] を押してメニューを終了します。

第 2 章

C++Builder の開発環境

この章では、C++Builder の起動方法を説明し、統合開発環境（IDE）と呼ばれるデスクトップの主要な部分とツールの概要を説明します。

C++Builder の起動

C++Builder には次の起動方法があります。

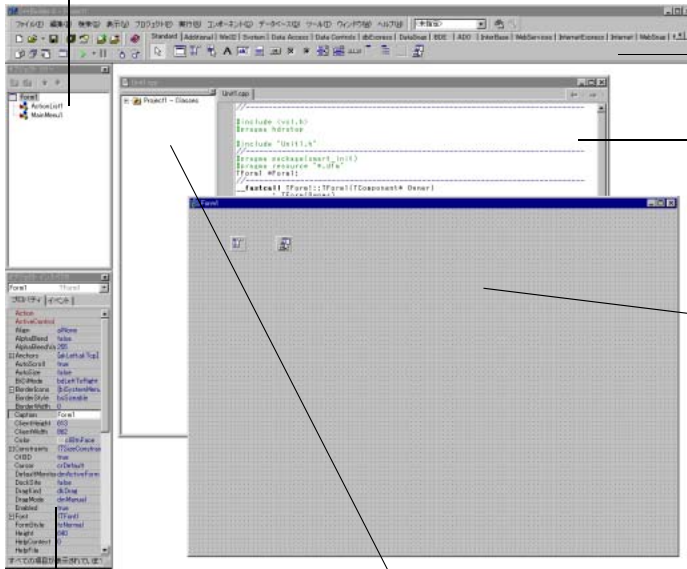
- C++Builder のアイコンをダブルクリックします（ショートカットを作成済みの場合）。
- Windows のスタートメニューから [プログラム | Borland C++Builder 6] を選択します。
- Windows のスタートメニューから [名前を指定して実行] を選択し、「Bcb」を入力します。
- CBuilder6¥Bin ディレクトリにある Bcb.exe をダブルクリックします。

IDE

C++Builder を起動すると、IDE の主要ツールのいくつかが表示されます。C++Builder の IDE には、ツールバー、メニュー、コンポーネントパレット、オブジェクトインスペクタ、オブジェクトツリー、コードエディタ、クラスエクスプローラ、プロジェクトマネージャなどのツールがあります。使用可能な機能とコンポーネントは、購入された C++Builder のバージョン（版）によって異なります。

メニューとツールバー

オブジェクトツリーには、コンポーネントの親子関係を示す階層構造が表示されます。



アプリケーションで使う既製のコンポーネントのパレット

コードを表示、編集するためのコードエディタ

フォームデザイナーには、アプリケーションのUI設計を開始するための空白のフォームがあります。アプリケーションには多くのフォームが含まれます。

オブジェクトインスペクタは、オブジェクトのプロパティの変更やイベントハンドラの選択に使用します。

クラスエクスプローラでは、ユニット内のクラス、変数、ルーチンが示され、各コードにすばやく移動できます。

C++Builderの開発モデルは、2-Way（双方向）ツールに基づいています。これによって、ビジュアルな設計ツールとテキストベースのコード編集ツールの間を行き来することができます。たとえば、フォームデザイナーのグラフィカルインターフェイスでボタンなどの要素を配置した後で、すぐにフォームの説明テキストの入っているフォームファイルを見ることができます。また、C++Builderが生成したコードをビジュアルな開発環境の中で手動で変更することもできます。

どのプログラミングツールも、IDEから簡単に呼び出すことができます。IDEの中にあるままで、グラフィカルインターフェイスの設計、クラスライブラリのブラウズ、コードの記述・コンパイル・テスト・デバッグ、およびプロジェクトの管理が行えます。

IDEの構成や設定の変更については、第6章「デスクトップのカスタマイズ」を参照してください。

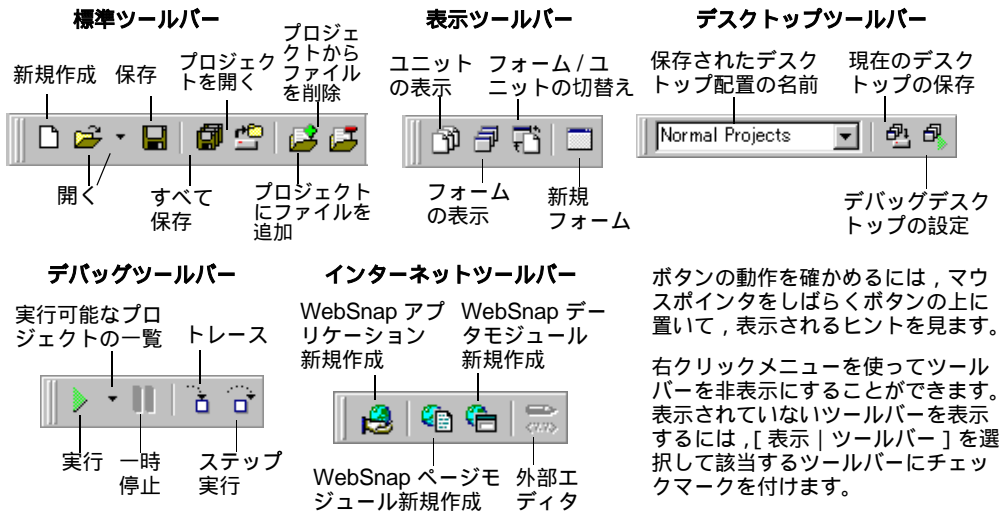
メニューとツールバー

画面上部を占めるメインウィンドウは、メインメニュー、ツールバー、およびコンポーネントパレットから構成されます。



デフォルト構成のメインウィンドウ

C++Builder のツールバーは、よく使用される操作やコマンドをすばやく実行できるように用意されています。ツールバーから呼び出せる操作は、すべてドロップダウンメニューからも実行できます。



多くの操作には、ツールバーボタンとともにキーボードショートカットが割り当てられています。キーボードショートカットが使用できる場合は、必ずドロップダウンメニューのコマンドの横に示されます。

たいていの場合、ツールやアイコンの上でマウスを右クリックすると、そのオブジェクトに応じたコマンドのメニューが表示されます。このようなメニューをコンテキストメニューと呼びます。

ツールバーもカスタマイズすることができます。ツールバーにコマンドを追加したり、別の位置に移動したりできます。詳細については、6-1 ページの「メニューとツールバーの配置」および 6-5 ページの「デスクトップ配置の保存」を参照してください。

詳細については、メニューオプションについてのヘルプが見たい場合は、マウスでポイントして [F1] キーを押します。

コンポーネントパレット, フォームデザイナー, オブジェクトインスペクタ

コンポーネントパレット, フォームデザイナー, オブジェクトインスペクタ, およびオブジェクトツリービューは、アプリケーションのユーザーインターフェースの構築を支援する機能を提供します。

コンポーネントパレットは、VCL と CLX のビジュアル/非ビジュアルコンポーネントを表すアイコンをグループ化したタブ付きページで構成されます。タブ付きページによって、コンポーネントはさまざまな機能別のグループに分類されています。たとえば、[Standard], [Additional], [Win32] の各ページには、編集ボックスやアップダウンボタンなどのウィンドウコントロールが入っており、

オブジェクトツリー

[Dialogs] ページには、開くファイルの選択やファイルの保存などの操作のためのコマンドダイアログボックスが入っています。



コンポーネントには、それぞれ固有の属性としてプロパティ、イベント、メソッドがあり、それらを使ってアプリケーションを制御することができます。

フォーム（フォームデザイナー）にコンポーネントを配置したら、コンポーネントのサイズや位置を自由に変更して、ユーザーインターフェースの外観を整えることができます。フォームに配置したコンポーネントに対して、オブジェクトインスペクタを使って、設計時プロパティの設定、イベントハンドラの作成、可視のプロパティとイベントの選別を行い、アプリケーションの外観とアプリケーションを動作させるコードとを連携させます。3-2 ページの「フォームにコンポーネントを追加する」を参照してください。

詳細については、ヘルプのキーワードで「コンポーネントパレット」を検索してください。

オブジェクトツリー

オブジェクトツリーは、コンポーネントの兄弟関係と親子関係を、階層構造のツリー図として図示します。ツリー図は、オブジェクトインスペクタおよびフォームデザイナーと同期しており、オブジェクトツリーでフォーカスを移動すると、それに合わせてオブジェクトインスペクタとフォームでのフォーカスも移動します。

オブジェクトツリーを使って、関連するコンポーネントの相互の関係を変更することができます。たとえば、パネルコンポーネントとチェックボックスコンポーネントをフォームに追加した場合、その2つのコンポーネントは兄弟になります。しかし、オブジェクトツリーでこのチェックボックスをパネルアイコンの上にドラッグすると、チェックボックスはパネルの子になります。

オブジェクトのプロパティの設定が完了していない場合、オブジェクトツリーには赤い疑問符が表示されます。ツリー図でオブジェクトをダブルクリックすると、コードエディタが開いて、イベントハンドラを記述できる場所に位置付けられます。

オブジェクトツリーが表示されていない場合は、[表示 | オブジェクトツリー] を選択します。

オブジェクトツリービューは、データベースオブジェクト間の関係の表示に特に便利です。

詳細については、ヘルプのキーワードで「オブジェクトツリー」を検索してください。

オブジェクトリポジトリ

オブジェクトリポジトリには、フォーム、データモジュール、ウィザード、DLL、サンプルアプリケーション、および開発を簡略化できる他のオブジェクトが含まれています。新しいプロジェクトを開始するときは、[ファイル | 新規作成 | その他] を選択して [新規作成] ダイアログボックスを表示します。[新規作成] ダイアログボックスは、オブジェクトリポジトリと同じダイアログボックスです。リポジトリを見て、必要なオブジェクトに似たオブジェクトがあるかどうかを確認めます。

リポジトリのタブ付きページには、フォーム、フレーム、ユニットなどのオブジェクトと、特定の目的の項目を作成するウィザードが入っています。

オブジェクトリポジトリに含まれている項目を基にして項目を作成する場合、項目をコピー、継承、または直接使用することができます。

[コピー] (デフォルト) を選択すると、項目のコピーがプロジェクト内に作成されます。[継承] を選択すると、プロジェクト内に作成された項目は、リポジトリ内の項目に対して加えられた変更を継承します。[直接使用] を選択すると、リポジトリ内のオブジェクトは、プロジェクト内に作成されたオブジェクトに対して加えられた変更を継承します。



オブジェクトリポジトリのオブジェクトを編集または削除するには、[ツール | リポジトリ] を選択するか、[新規作成] ダイアログボックスで右クリックして、[プロパティ] を選択します。

オブジェクトリポジトリのタブ付きページは、追加、削除、および名前の変更を行うことができます。

矢印をクリックすると、[新規作成] ダイアログボックスに表示されるタブ付きページの順序を変更することができます。

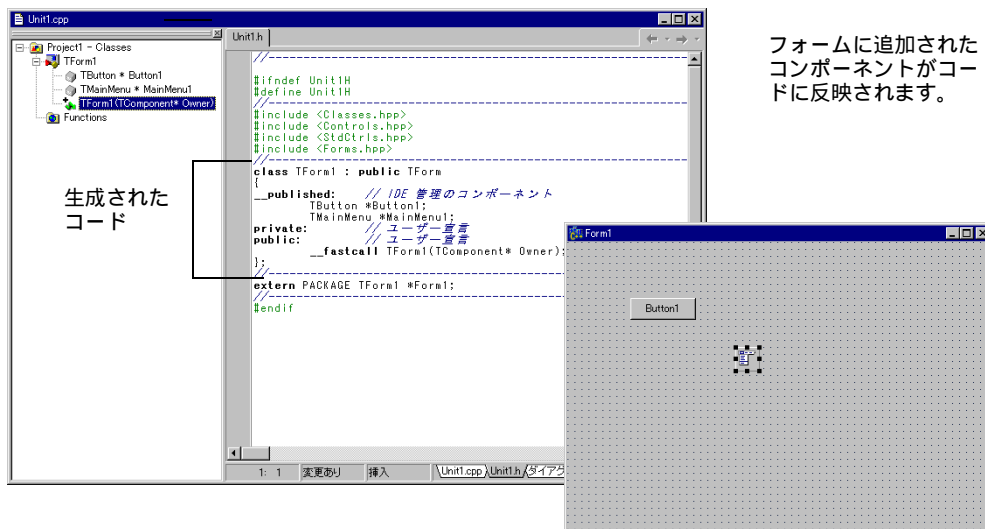


オブジェクトリポジトリにプロジェクトとフォームテンプレートを追加する方法については、6-10 ページの「オブジェクトリポジトリへのテンプレートの追加」を参照してください。

詳細については、ヘルプのキーワードで「オブジェクトリポジトリ」を検索してください。使用できるオブジェクトは、購入された C++Builder の版によって異なります。

コードエディタ

アプリケーションのユーザーインターフェースの設計に従って、C++Builderはその基礎となるオブジェクトコードを作成します。フォームとオブジェクトのプロパティを選択して変更すると、その変更は自動的にソースファイルに反映されます。機能の完備したテキストエディタである組み込みのコードエディタを使って、ソースファイルにコードを直接追加することができます。C++Builderは、コードを書くときの助けとなる支援ツール、クラス補完、コードブラウザなどのツールを提供しています。



支援機能

支援機能ツールは、状況に応じてポップアップウィンドウを表示します。

ツール	動作
コード補完	オブジェクトへのポインタを表す変数名に続けて矢印 (->) を入力するか、非 VCL オブジェクトを表す変数名に続けてピリオド (.) を入力すると、そのクラスまたは構造体に対応するプロパティ、メソッド、イベントの一覧が表示されます。代入文を代入演算子まで入力して [Ctrl] + [Space] を押すと、その変数に対して有効な値のリストが表示されます。関数またはメソッドの名前に続けて左カッコを入力すると、引数のリストが表示されます。
宣言表示	メソッド名に続いて左カッコを入力すると、そのメソッドの引数の構文が表示されます。
式評価	デバッグ中に一時停止しているときに、変数をポイントするとその現在の値が表示されます。
識別子定義の表示	コードの編集中に [Ctrl] を押しながら識別子をポイントすると、その宣言が表示されます。
コードテンプレート	[Ctrl] + [J] を押すと、コードに挿入可能な一般的なプログラミング文が表示されます。C++Builder で提供されるテンプレートに加えて、独自のテンプレートを作成することができます。

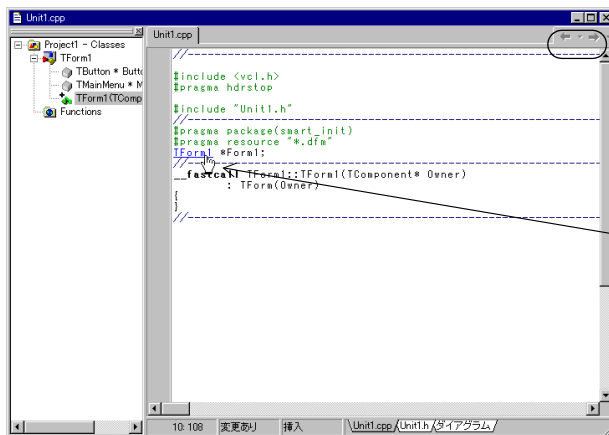
これらのツールのオン / オフを切り替えるには、[ツール | エディタオプション] を選択して [支援機能] タブをクリックします。[自動支援機能] セクションでツールをチェックするか、チェックを外します。

詳細については、ヘルプのキーワードで「支援機能」を検索してください。

コードのブラウズ

クラス、変数、プロパティ、メソッド、または他の識別子の名前にマウスを合わせると、識別子定義の表示のポップアップメニューが表示され、その識別子が宣言されている場所が示されます。[Ctrl] を押しとカーソルが手の形に変わり、識別子の色が青になって下線が付きます。ここでクリックすると、識別子を定義している場所へジャンプできます。

コードエディタには、Web ブラウザと同じような「戻る」ボタンと「進む」ボタンがあります。識別子を定義している場所などにジャンプしたとき、コードエディタはコード内での移動をトラッキングしています。[進む] ボタンと [戻る] ボタンの横にあるドロップダウンの矢印をクリックすると、この移動の履歴を使って前後に移動することができます。



エディタを Web ブラウザのように使うことができます。

[Ctrl] を押しながら識別子をポイントします。カーソルが手の形に変わり、識別子が青色に変わって下線が引かれます。

クリックすると識別子の定義が表示されます。

移動の後で、右上隅の左矢印をクリックすると、前の位置に戻ります。

コード編集環境のカスタマイズについては、6-11 ページの「コードエディタのカスタマイズ」を参照してください。

詳細については、ヘルプのキーワードで「コードエディタ」を検索してください。

[ダイアグラム] ページ

コードエディタの一番下に、使用している C++Builder の版によって、1 つまたは複数のタブがあります。デフォルトでは、コードの編集に使う [コード] ページが前面に表示されます。[ダイアグラム] ページには、フォームまたはデータモジュールに配置したコンポーネント間の関係が、アイコンと線によって表されます。表される関係としては、兄弟関係、親子関係、プロパティとコンポーネントの関係があります。

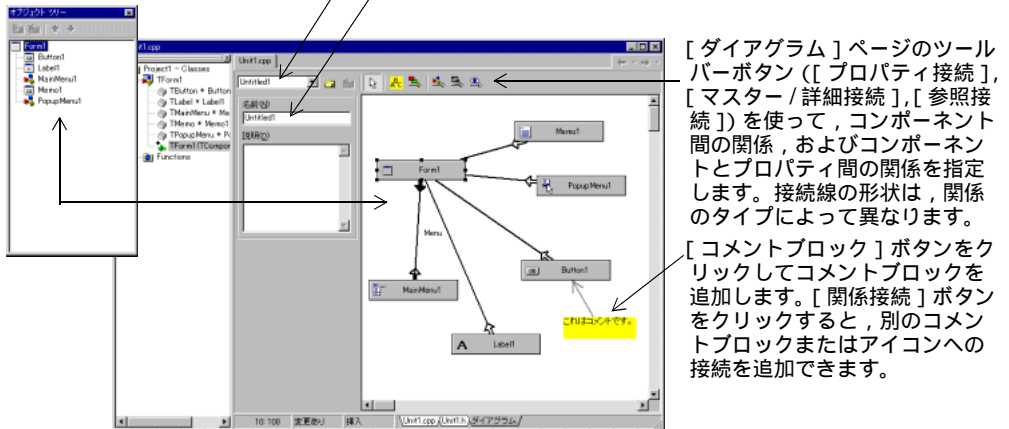
関係図を作成するには、[ダイアグラム] ページをクリックします。オブジェクトツリーからアイコンを [ダイアグラム] ページにドラッグすると、そのアイコンを縦に配置できます。アイコンを横に配置するには、[Shift] を押しながらドラッグします。親子関係またはコンポーネントとプロパティの依存関係にあるアイコンを一緒にドラッグすると、その依存関係を示す線（コネクタ）が自動的に追加されます。たとえば、データモジュールにデータセットコンポーネントを追加して、データセットアイコンとそのプロパティアイコンを [ダイアグラム] ページにドラッグすると、プロパティコネクタは自動的にデータセットアイコンに接続されます。

依存関係を持っていないコンポーネントの関係を表示したい場合は、[ダイアグラム] ページの一番上にあるツールバーボタンを使って、4種類のコネクタタイプ（関連、プロパティ、マスター / 詳細、参照）からコネクタを追加します。また、相互に接続するコメントブロック、または関連するアイコンに接続するコメントブロックを追加することもできます。

オブジェクトツリーから、コンポーネントのアイコンを [ダイアグラム] ページにドラッグします。

現在のプロジェクト内で名前を付けた他の関係図を表示するには、ドロップダウンリストボックスをクリックします。

関係図の名前と説明を入力します。



できあがった関係図は、名前と説明を入力し、保存し、印刷することができます。

詳細については、ヘルプのキーワードで「ダイアグラムページ」を検索してください。

フォームのコードの表示

フォームは、ほとんどの C++Builder プロジェクトで中心となるパーツであり、アプリケーションのユーザーインターフェースを設計する場所になります。通常、フォームは C++Builder のビジュアルツールを使って設計し、そのフォームはフォームファイルに格納されます。フォームファイル (.dfm, CLX アプリケーションの場合は .xfm) には、フォーム内の各コンポーネントと、すべての持続的プロパティの値が記述されます。フォームファイルをエディタで表示して編集するには、フォーム上で右クリックして [テキストとして表示] を選択します。フォームのグラフィック表示に戻るには、エディタ上で右クリックして [フォームとして表示] を選択します。

フォームファイルは、テキスト形式（デフォルト）またはバイナリ形式で保存できます。[ツール | 環境オプション] を選択して [デザイナ] ページをクリックし、新しく作成するフォームの保存形式を指定する [テキスト形式で作成] チェックボックスをチェックするかチェックを外します。

詳細については、ヘルプのキーワードで「フォームファイル」を検索してください。

クラスエクスプローラ

When you open C++Builder を起動すると、コードエディタウィンドウにクラスエクスプローラがドッキングした状態で表示されます（C++Builder の一部の版にはクラスエクスプローラは付属していません）。クラスエクスプローラは、コードエディタで開いているソースコードについて、作成したユニットで定義されているクラス、プロパティ、メソッド、グローバル変数、およびルーチンの一覧を、ツリー図の形式で「目次」のように表示します。

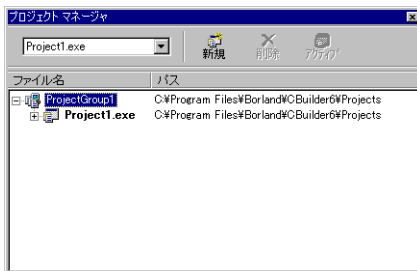
クラスエクスプローラを使うと、コードエディタの中を効率よく移動することができます。たとえば、クラスエクスプローラでメソッドをダブルクリックすると、コードエディタではカーソルが、ユニットのインターフェース部の該当するクラス宣言内のそのメソッドの定義にカーソルがジャンプします。

クラスエクスプローラの表示形式を設定するには、[ツール | 環境オプション] を選択して [エクスプローラ] タブをクリックします。

詳細については、ヘルプのキーワードで「クラスエクスプローラを参照してください。

プロジェクトマネージャ

初めて C++Builder を起動したときには、新しいプロジェクトが開きます。プロジェクトには、開発するアプリケーションまたは DLL を構成するいくつかのファイルが含まれます。これらは、フォーム、ユニット、リソース、オブジェクト、ライブラリなどのファイルで、プロジェクトマネージャというプロジェクト管理ツールによって表示したり構成を変更したりできます。プロジェクトマネージャを表示するには、[表示 | プロジェクトマネージャ] を選択します。



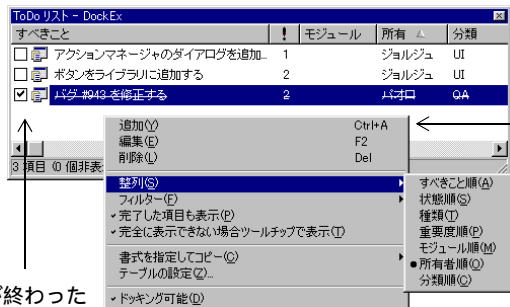
To-Do リスト

プロジェクトマネージャを使うと、関連する複数のプロジェクトの情報を1つのプロジェクトグループに結合して表示することができます。複数の実行形式ファイルなどの関連するプロジェクトをグループにまとめると、それらを1回の操作でコンパイルすることができます。プロジェクトのコンパイルなどのプロジェクトオプションの変更については、6-8 ページの「プロジェクトオプションの設定」を参照してください。

詳細については、ヘルプのキーワードで「プロジェクトマネージャ」を検索してください。

To-Do リスト

To-Do リストには、プロジェクトを完成させるために必要な項目を登録します。プロジェクト全体の項目をリストに直接追加することができ、ソースコード内の特定の項目を追加することもできます。プロジェクトに関連付けられる情報を追加または表示するには、[表示 | To-Do リスト] を選択します。



To-Do リスト上で右クリックすると、リストのソートやフィルタ処理を行うコマンドが表示されます。

項目の処理が終わったら、チェックボックスをクリックします。

詳細については、ヘルプのキーワードで「To-Do リスト」を検索してください。

第3章

C++Builder を使った プログラミング

以下の節では、C++Builder を使ったソフトウェア開発の概要として、プロジェクトの作成、フォームの操作、コードの記述、そしてアプリケーションのコンパイル、デバッグ、配布、および国際化対応について説明します。また、開発できるプロジェクトの種類についても説明します。

プロジェクトの作成

プロジェクトは、設計時に作成したファイルと、プロジェクトソースコードをコンパイルしたときに生成されたファイルのコレクションです。初めて C++Builder を起動したときには、新しいプロジェクトが開きます。自動的に、プロジェクトファイル (Project1.bpr)、ユニットファイル (Unit1.cpp および Unit1.h)、およびリソースファイル (Unit1.dfm、CLX アプリケーションの場合は Unit1.xfm) が生成されます。

プロジェクトがすでに開いているときに新しいプロジェクトを開く場合は、[ファイル | アプリケーションの新規作成] を選択するか、[ファイル | 新規作成 | アプリケーション] を選択するか、[ファイル | 新規作成 | その他] を選択して [アプリケーション] アイコンをダブルクリックします。[ファイル | 新規作成 | その他] を選択するとオブジェクトリポジトリが開きます。オブジェクトリポジトリには、プロジェクトに追加できるフォーム、モジュール、フレームとともに、ダイアログボックスなどの設計済みのテンプレートがあります。オブジェクトリポジトリの詳細については、2-5 ページの「オブジェクトリポジトリ」を参照してください。

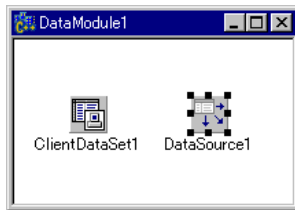
プロジェクトを開始するときには、開発する対象が何なのか、アプリケーションか DLL かなどが分かっている必要があります。C++Builder で開発できるプロジェクトの種類については、3-9 ページの「プロジェクトの種類」を参照してください。

詳細については、ヘルプのキーワードで「プロジェクト」を検索してください。

データモジュールを追加する

データモジュールは、非ビジュアルコンポーネントを含む特別なフォームの一種です。非ビジュアルコンポーネントは、通常のフォームにビジュアルコンポーネントとともに配置することができます。しかし、データベースとシステムオブジェクトのグループの再利用を考慮する場合や、アプリケーションのデータベース接続の部分とビジネスルールの部分を分離したい場合には、データモジュールはそのような構成を行うための便利なツールとなります。

データモジュールを作成するには、[ファイル | 新規作成 | データモジュール] を選択します。C++Builder は、空のデータモジュールを開き、コードエディタにそのモジュールのためのユニットファイルを表示し、現在のプロジェクトにそのモジュールを新規ユニットとして追加します。データモジュールに非ビジュアルコンポーネントを追加する場合も、フォームと同じ方法で行います。



コンポーネントパレットで非ビジュアルコンポーネントをダブルクリックして、データモジュールにそのコンポーネントを配置します。

既存のデータモジュールを開くと、C++Builder はそのコンポーネントを表示します。

詳細については、ヘルプのキーワードで「データモジュール」を検索してください。

ユーザーインターフェースの作成

C++Builder でユーザーインターフェース (UI) の作成を始めるには、コンポーネントパレットからコンポーネントを選択し、メインフォームに配置します。

フォームにコンポーネントを追加する

フォームにコンポーネントを配置するには、次のいずれかの方法を使います。

1. コンポーネントをダブルクリックする
2. コンポーネントを1回クリックしてから、フォーム上でコンポーネントを配置したい場所をクリックする

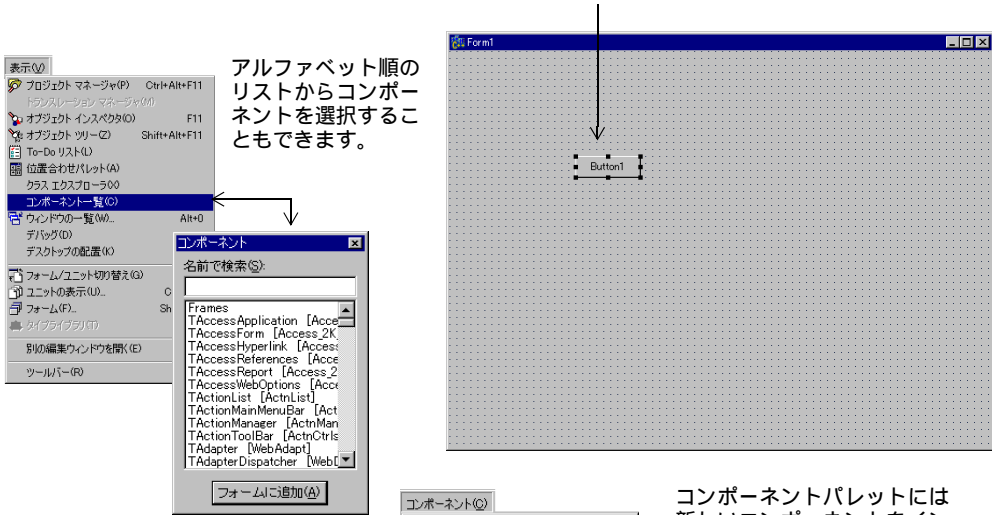
コンポーネントを選択して、フォーム上の好きな場所にドラッグします。

コンポーネントパレットには、多数のコンポーネントが機能別にグループ化されています。

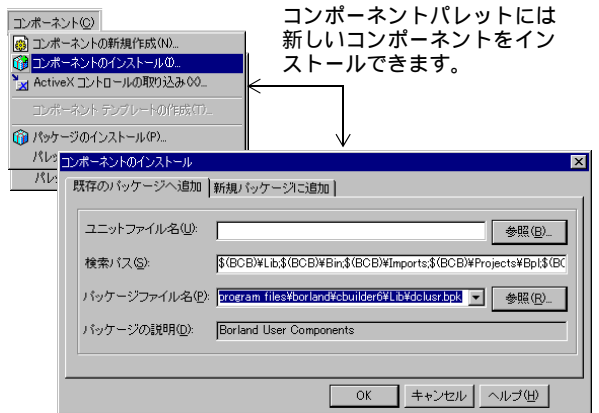


コンポーネントパレットのコンポーネントをクリックします。

次に、コンポーネントを配置したい場所をクリックします。



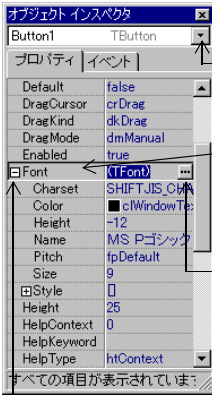
パレットの再構成や新しいページの追加もできます。[ツール|環境オプション]を選択し、[[パレット]タブを選択します。



詳細については、ヘルプのキーワードで「コンポーネントパレット」を検索してください。

コンポーネントのプロパティを設定する

フォームにコンポーネントを配置した後は、コンポーネントのプロパティを設定し、イベントハンドラをコーディングします。コンポーネントのプロパティを設定することにより、アプリケーションでのコンポーネントの表示や動作を変更することができます。フォーム上でコンポーネントを選択すると、そのプロパティとイベントがオブジェクトインスペクタに表示されます。

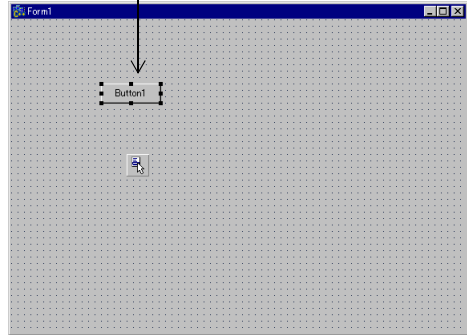


このドロップダウンリストを使ってオブジェクトを選択することもできます。ここでは、Button1 が選択され、そのプロパティが表示されています。

フォーム上のコンポーネント（オブジェクト）をクリックすれば、コンポーネントを選択できます。

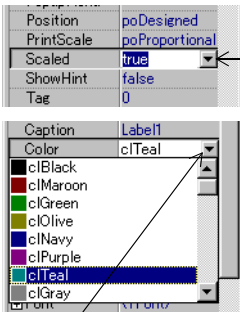
プロパティを選択し、右側の列の値を変更します。

[...] をクリックしてダイアログボックスを開くと、ヘルパーオブジェクトのプロパティを変更できます。



正符号 (+) をクリックすると、詳細リストを開くこともできます。

プロパティの多くは、色の名前や、true または false、整数などの単一の値を持ちます。論理型のプロパティは、ダブルクリックすることにより true と false を切り替えられます。プロパティによっては、関連付けられたプロパティエディタを使ってより複雑な値を設定できます。そのようなプロパティ値をクリックすると、省略記号 [...] が表示されます。サイズなどのプロパティには、値を入力します。



ここをダブルクリックして値を true から false に切り替えます。

[...] をクリックして、このプロパティのプロパティエディタを表示します。



下矢印をクリックすると有効な値のリストが表示されます。

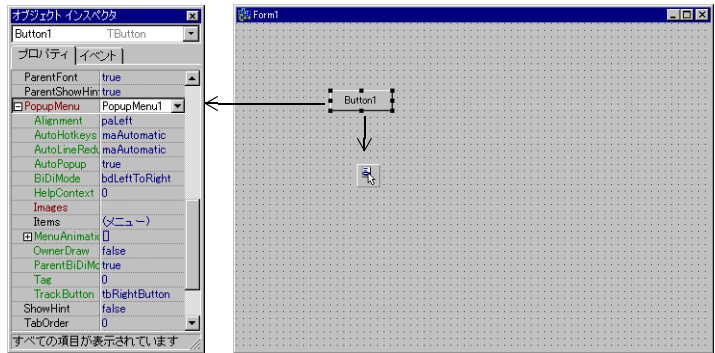


フォームで複数のコンポーネントが選択されているときには、オブジェクトインスペクタは選択されているコンポーネント間で共有されるすべてのプロパティを表示します。

オブジェクトインスペクタは、インラインコンポーネント参照もサポートしています。これによって、参照されるコンポーネント自体を選択しなくても、参照先コンポーネントのプロパティとイベント

トにアクセスできます。たとえば、フォームにボタンとポップアップメニューを追加したとします。ボタンコンポーネントを選択すると、オブジェクトインスペクタで PopupMenu プロパティに PopupMenu1 を設定でき、設定したポップアップメニューのすべてのプロパティが表示されます。

ボタンコンポーネントの PopupMenu プロパティに PopupMenu1 を設定すると、[+] をクリックしたときにポップアップメニューのすべてのプロパティが表示されます。インラインコンポーネント参照は赤で表示され、そのサブプロパティは緑で表示されます。



詳細については、ヘルプのキーワードで「オブジェクトインスペクタ」を検索してください。

コードの記述

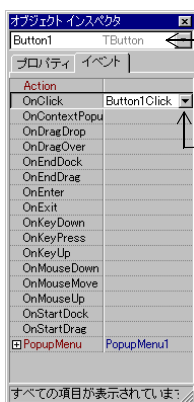
アプリケーションの肝心な部分は、各コンポーネントの背後にあるコードです。C++Builder の RAD 環境は、インストール済みのビジュアルおよび非ビジュアルコンポーネントなど、アプリケーションの構成要素のほとんどを提供していますが、通常はイベントハンドラやメソッドは開発者が作成する必要があります。この作業を容易にするために、C++Builder の VCL および CLX クラスライブラリの数千のオブジェクトからコンポーネントを使用することができます。ソースコードの操作については、2-6 ページの「コードエディタ」を参照してください。

イベントハンドラの作成

コードでは、アプリケーションの実行時にコンポーネントに対して発生するイベントにตอบสนองする必要があります。イベントは、ボタンのクリックなどシステム内で発生することと、発生したことにตอบสนองするコードの間のリンクとなります。この応答コードがイベントハンドラです。このコードでは、プロパティ値を変更したり、メソッドを呼び出したりします。

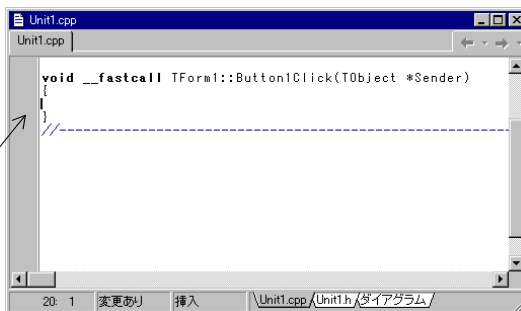
コードの記述

フォーム上のコンポーネントの定義済みのイベントハンドラを見るには、そのコンポーネントを選択して、オブジェクトインスペクタの [イベント] タブをクリックします。



ここでは、Button1 が選択され、その型 TButton が表示されています。オブジェクトインスペクタの [イベント] タブをクリックすると、Button コンポーネントで処理できるイベントを見ることができます。

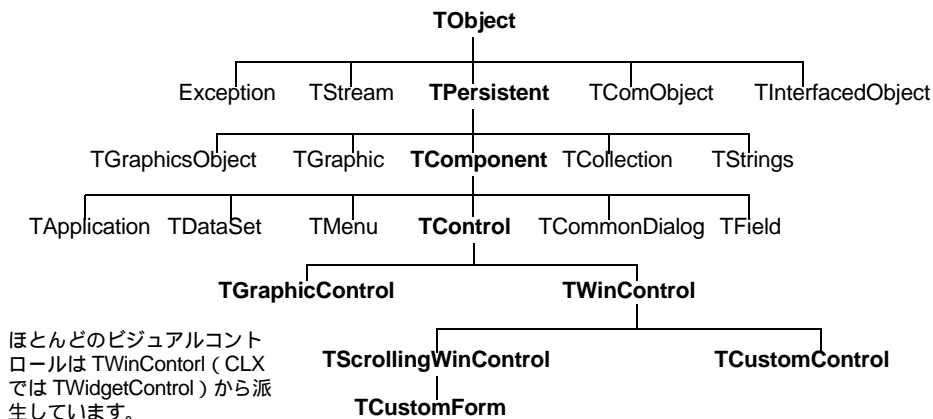
ドロップダウンリストから既存のイベントハンドラを選択します。また、値列をダブルクリックすると、C++Builder は新しいイベントハンドラのスケルトン (骨組) コードを生成します。



詳細については、ヘルプのキーワードで「イベント」を検索してください。

VCL/CLX ライブラリを使う

C++Builder には、コードを書くときに使用するオブジェクトで構成されるクラスライブラリが付属しています。このオブジェクトには、コンポーネントやコントロールも含まれます。Windows アプリケーションには VCL (ビジュアルコンポーネントライブラリ) を、Windows および Linux アプリケーションには CLX (クロスプラットフォーム用コンポーネントライブラリ) を使用できます。これらのライブラリには、実行時に目に見えるオブジェクト (編集コントロールやボタンなどのユーザーインターフェース要素) とともに、目に見えない非ビジュアルコントロール (データセットやタイマーなど) が含まれています。次の階層図は、VCL を構成する主要なクラスです。CLX の階層も似たような構造を持っています。



TComponent から派生したオブジェクトは、コンポーネントパレットにインストールして C++Builder のフォームおよびデータモジュールへの追加を可能にするプロパティとメソッドを持っています。VCL と CLX のコンポーネントは IDE に組み込まれているので、フォームデザイナーなどのツールを使ってアプリケーションをすばやく開発することができます。

コンポーネントは、高度にカプセル化されています。たとえばボタンは、OnClick イベントの発生によってマウスクリックにตอบสนองするようにあらかじめプログラムされています。VCL または CLX のボタンコントロールを使えば、ボタンがクリックされたときに生成されるイベントを処理するコードを自分で書く必要はありません。プログラマの責任となるのは、そのクリックへの応答において実行されるアプリケーションロジックについてだけです。

C++Builder のほとんどの版には、VCL および CLX のソースコードが付属しています。

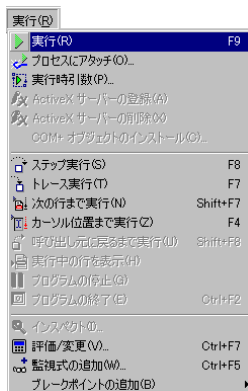
詳細については、ヘルプの目次から「VCL リファレンス」および「CLX リファレンス」を参照してください。また、ヘルプのキーワードで「VCL」および「CLX」を検索してください。

プロジェクトのコンパイルとデバッグ

コードの記述が終わったら、プロジェクトをコンパイルしてデバッグします。C++Builder では、先にプロジェクトをコンパイルしてデバッグは別に行うことも、統合デバッガを使ってコンパイルとデバッグを 1 ステップで行うこともできます。デバッグ情報付きでプログラムをコンパイルするには、[プロジェクト | オプション] を選択し、[コンパイラ] ページをクリックして、[デバッグ情報] をチェックします。

C++Builder の統合デバッガを使うと、プログラムの実行の制御、変数の監視、およびデータ値の変更ができます。また、コードを 1 度に 1 行ずつ実行しながら、あるいはブレークポイントごとに停止してプログラムの状態を検査することができます。統合デバッガを使うには、[ツール | デバッガオプション] を選択し、[一般] ページをクリックして、[統合開発環境を使う] をチェックします。

IDE でデバッグセッションを開始するには、デバッグツールバーの [実行] ボタンをクリックするか、[実行 | 実行] を選択するか、あるいは [F9] を押します。



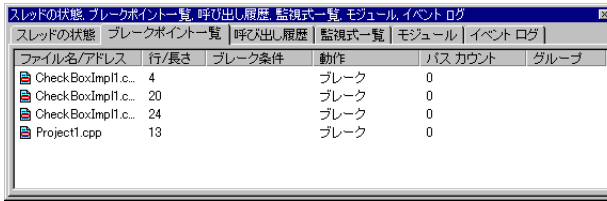
デバッグコマンドは [実行] メニューから選択します。コマンドのいくつかはツールバーからも使用できます。



実行ボタン

アプリケーションの配布

統合デバッガでは、デバッグ用のウィンドウとして、ブレークポイント一覧、呼び出し履歴、監視式一覧、ローカル変数、スレッドの状態、モジュール、CPU ビュー、イベントログなどが使用できます。これらは [表示 | デバッグ] を選択して表示します。C++Builder の版によっては、使用できないデバッガウィンドウもあります。



デバッグ用のウィンドウは、使いやすいようにドッキングできます。

デバッグ用のウィンドウは、使いやすいようにドッキングすることができます。ドッキングについては 6-2 ページの「ツールウィンドウのドッキング」を参照してください。

デバッグ用にデスクトップを配置したら、その設定をデバッグ用あるいは実行時用のデスクトップとして保存することができます。このデスクトップ配置は、どんなアプリケーションをデバッグするときにも使用できます。詳細については、6-5 ページの「デスクトップ配置の保存」を参照してください。

詳細については、ヘルプのキーワードで「デバッグ」および「統合デバッガ」を検索してください。

アプリケーションの配布

作成したアプリケーションを配布して、他の人がインストールして実行できるようにできます。アプリケーションを配布するときには、実行形式ファイル、DLL、パッケージファイル、ヘルパーアプリケーションなど、必須のファイルとその他のサポートファイルがすべて必要です。C++Builder には InstallShield Express というセットアップツールキットがバンドルされており、これらのファイルを含むアプリケーションのインストールプログラムの作成が容易に行えます。InstallShield Express をインストールするには、C++Builder のインストール画面で [InstallShield Express Custom Edition for C++Builder] を選択します。

詳細については、ヘルプのキーワードで「配布、アプリケーション」を検索してください。

アプリケーションの国際化対応

C++Builder には、アプリケーションの国際化とローカライズのための機能がいくつかあります。IDE と VCL は、IME (input method editor) と拡張文字セットをサポートしており、プロジェクトの各国語対応が可能です。C++Builder の一部のバージョン (版) には、ソフトウェアのローカライズと複数ロケールの同時開発のためのトランスレーションーツが含まれています。トランスレーション

スーツを使用すると、アプリケーションの複数のローカライズバージョンを1つのプロジェクトの一部として管理することができます。

トランスレーションスーツには、3つの統合ツールが含まれています。

- リソース DLL ウィザード：リソース DLL の生成と管理を行う DLL ウィザード
- トランスレーションマネージャ：翻訳されたリソースの表示と編集を行うテーブル
- トランスレーションリポジトリ：トランスレーションを格納する共有データベース

リソース DLL ウィザードを開くには、[ファイル | 新規作成 | その他] を選択して、[リソース DLL ウィザード] アイコンをダブルクリックします。トランスレーションツールについての設定は、[ツール | トランスレーションツールオプション] で行います。

詳細については、ヘルプのキーワードで「国際化、アプリケーション」を検索してください。

プロジェクトの種類

C++Builder のすべてのバージョン（版）は、32 ビット Windows プログラミング、DLL、パッケージ、カスタムコンポーネント、マルチスレッド、COM（Component Object Model）とオートメーションコントローラ、およびマルチプロセスデバッグをサポートしています。また、一部の版では Web サーバーアプリケーション、データベースアプリケーション、COM サーバー、多層アプリケーション、CORBA、および意志決定支援システムをサポートしています。

詳細については、使用しているバージョン（版）がどのツールをサポートしているかは次の URL から機能一覧を参照してください。 www.borland.co.jp/cppbuilder/

CLX アプリケーション

C++Builder を使うと、Windows と Linux の両方のオペレーティングシステムで動作するクロスプラットフォームの 32 ビットアプリケーションを開発できます。Linux については、Borland の C++ ソリューションは利用できませんが、C++Builder でアプリケーションを開発することにより、前もって準備を進めることができます。CLX アプリケーションを開発するには、[ファイル | 新規作成 | CLX アプリケーション] を選択します。IDE の操作は通常の C++Builder アプリケーションの場合と同様ですが、コンポーネントパレットとオブジェクトリポジトリには、CLX アプリケーションで使用できるコンポーネントと項目だけが表示されます。C++Builder でサポートされる Windows 固有の機能は、Linux 環境に直接移植されません。

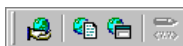
詳細については、クロスプラットフォームアプリケーションの開発に使用できるコンポーネントについては、ヘルプの目次から「CLX リファレンス」を参照してください。

Web サーバーアプリケーション

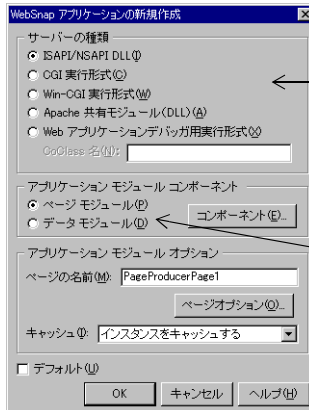
Web サーバーアプリケーションは、クライアントのリクエストを処理して、Web ページの形式で HTTP メッセージを返すことにより、Web サーバーとともに動作します。Web 用データの作成のために、C++Builder の一部のバージョン（版）には 2 種類のテクノロジーが組み込まれています。

C++Builder の古い Web サーバーアプリケーションテクノロジーは WebBroker と呼ばれるものです。WebBroker アプリケーションは、リクエストのディスパッチ、アクションの実行、およびユーザーに Web ページを返すことができます。アプリケーションのビジネスロジックのほとんどは、アプリケーション開発者が記述するイベントハンドラの中で定義されます。WebBroker Web アプリケーションを作成するには、[ファイル | 新規作成 | その他] を選択して、[Web サーバーアプリケーション] アイコンをダブルクリックします。Web モジュールには、コンポーネントパレットの [Internet] ページと [InternetExpress] ページからコンポーネントを追加できます。

WebSnap は、この機能にアダプタ、追加ディスパッチャ、追加ページプロデューサ、セッションサポート、および Web ページモジュールを付け加えます。これらの追加機能は、一般的な Web サーバーアプリケーションタスクを自動的に処理する目的で設計されています。WebSnap による開発は、WebBroker での開発に比べて、よりビジュアルであり容易です。WebSnap アプリケーションを開発する場合、アプリケーションのビジネスロジックの設計により多くの時間を使うことができ、一般的なページ転送タスクを行うイベントハンドラを短時間で作成することができます。WebSnap サーバーアプリケーションを作成するには、[ファイル | 新規作成 | その他] を選択し、[WebSnap] ページをクリックして、[Web サーバーアプリケーション] アイコンをダブルクリックします。WebSnap のコンポーネントは、コンポーネントパレットの [WebSnap] ページから追加できます。



[表示 | ツールバー | インターネット] を選択して [WebSnap アプリケーション 新規作成] アイコンをクリックすれば、WebSnap アプリケーションのデータモジュールにアクセスすることもできます。



さまざまな Web サーバーアプリケーションで実行できるアプリケーションを作成できます。たとえば、Web サーバーアプリケーションをデバッグするテストサーバーなども含まれます。

データモジュールを作成するか、HTML ページを表示するページモジュールを作成するかを選択します。

詳細については、ヘルプのキーワードで「Web アプリケーション」を検索してください。

データベースアプリケーション

C++Builder が提供される多様なデータベース接続ツールによって、データベースアプリケーションの開発が簡略化されます。

データベースアプリケーションを作成する場合は、まずコンポーネントパレットの [Data Controls] ページのコンポーネントを使ってフォームのインターフェースを設計します。次に、[Data Access] ページからデータモジュールにデータソースを追加します。そのあと、各種のデータベースサーバーに接続するために、下記の接続ツールに対応したパレットページからデータセットコンポーネントと接続コンポーネントをデータモジュールに追加します。

- dbExpress は、SQL データベースサーバー (DB2, InterBase, MySQL, Oracle) への高速なアクセスを提供する、クロスプラットフォームアプリケーション用のデータベースドライバのコレクションです。dbExpress ドライバによって、単方向データセットを使ってデータベースにアクセスできます。
- BDE (ボーランドデータベースエンジン) は、dBASE, Paradox, Microsoft Access, および ODBC データソースを含め、広く使われているデータベースフォーマットをサポートするドライバのコレクションです。SQL Link ドライバは C++Builder の Enterprise 版に付属し、Oracle, Sybase, Informix, DB2, SQL Server, InterBase などのサーバーをサポートします。
- ADO (ActiveX Data Objects) は Microsoft の高レベルインターフェースで、リレーショナル/非リレーショナルデータベース、電子メールとファイルシステム、テキストとグラフィックス、およびカスタムビジネスオブジェクトなど、任意のデータソースにアクセスできます。
- InterBase Express (IBX) のコンポーネントは、C++Builder のカスタムデータアクセスコンポーネントのアーキテクチャに基づいています。IBX アプリケーションでは、InterBase の高度な機能にアクセスでき、バージョン 5.5 以降の InterBase に対してパフォーマンスが最も高いコンポーネントインターフェースを使用できます。IBX は、C++Builder のデータベース対応コンポーネントのライブラリと互換性があります。

一部のデータベース接続ツールは、C++Builder のバージョン (版) によっては使用できません。

詳細については、ヘルプのキーワードで「データベースアプリケーション」を検索してください。

BDE Administrator

BDE Administrator (BDEAdmin.exe) は、BDE ドライバの設定や、VCL のデータベース対応コントロールでデータベースに接続するために使用するエリアスの設定を行うために使用します。

詳細については、Windows の [スタート] メニューから [プログラム | Borland C++Builder 6 | BDE Administrator] を選択し、[ヘルプ | 目次] を選択してください。

SQL エクスプローラ (データベースエクスプローラ)

SQL エクスプローラ (DBExplor.exe) では、データベースのブラウズと編集が行えます。これを使って、データベースエリアスの作成、スキーマ情報の表示、SQL 問い合わせの実行、データディクショナリと属性セットの管理ができます。

詳細については、

C++Builder のメインメニューから [データベース | エクスプローラ] を選択し、[ヘルプ | 目次] を選択してください。または、ヘルプのキーワードで「データベースエクスプローラ」を検索してください。

Database Desktop

Database Desktop (DBD32.exe) では、Paradox および dBASE のデータベーステーブルをさまざまな形式で作成、表示、編集を行うことができます。

詳細については、

Windows の [スタート] メニューから [プログラム | Borland C++Builder 6 | データベースデスクトップ] を選択し、[ヘルプ | 目次] を選択してください。

データディクショナリ

BDE を使用する場合、データディクショナリは、データの内容と外観を記述する拡張フィールド属性セットを作成する場所として、アプリケーションから独立した、カスタマイズ可能な領域を提供します。データディクショナリは、リモートサーバーに常駐させてさらに情報を共有できます。

詳細については、

[ヘルプ | C++Builder ツール] を選択して、[データディクショナリ] を開いてください。

カスタムコンポーネント

C++Builder に付属するコンポーネントはあらかじめコンポーネントパレットにインストールされており、ほとんどの開発ニーズに応えられる範囲の機能が用意されています。何年も新しいコンポーネントをインストールせずに C++Builder でプログラミングをしてきたプログラマでも、特別な問題の解決や特定の動作を表すためにカスタムコンポーネントを必要とする場合があるでしょう。カスタムコンポーネントによって、コードの再使用とアプリケーション間の一貫性が高まります。

カスタムコンポーネントは、サードパーティ製のもの、独自に作成したもの、どちらもインストールすることができます。新しいコンポーネントを作成するには、[コンポーネント | 新規コンポーネント] を選択して、[新規コンポーネント] ウィザードを表示します。サードパーティ製コンポーネントのインストールについては、6-7 ページの「コンポーネントパッケージのインストール」を参照してください。

詳細については、

『開発者ガイド』の第 V 部「カスタムコンポーネントの作成」を参照してください。また、ヘルプのキーワードで「コンポーネント、作成」を検索してください。

DLL

ダイナミックリンクライブラリ (DLL) は、アプリケーションおよび他の DLL から呼び出せるルーチンの入ったコンパイル済みのモジュールです。DLL には、通常は複数のアプリケーションで使用

されるコードまたはリソースが含まれます。[ファイル | 新規作成 | その他] を選択して [DLL ウィザード] アイコンをダブルクリックすると、DLL 用のテンプレートを作成できます。

詳細については、ヘルプのキーワードで「DLL」を検索してください。

COM と ActiveX

C++Builder は、Microsoft の COM 規格をサポートしており、ActiveX コントロールを容易に作成できるウィザードを備えています。[ファイル | 新規作成 | その他] を選択して [ActiveX] タブをクリックすると、ウィザードにアクセスできます。ActiveX コントロールのサンプルがコンポーネントパレットの [ActiveX] ページにインストールされています。コンポーネントパレットの [Servers] タブには、多数の COM サーバーコンポーネントがあります。これらのコンポーネントは、VCL コンポーネントと同じように使うことができます。たとえば、Microsoft Word コンポーネントの 1 つをフォームに配置して、アプリケーションインターフェースの中から Microsoft Word のインスタンスを起動することができます。

詳細については、ヘルプのキーワードで「COM」および「ActiveX」を検索してください。

タイプライブラリ

タイプライブラリは、ActiveX コントロールまたはサーバーによって提供されるデータ型、インターフェース、メンバー関数、およびオブジェクトクラスに関する情報を含むファイルです。COM アプリケーションまたは ActiveX ライブラリとともにタイプライブラリを含むことによって、これらの構成要素に関する情報が他のアプリケーションやプログラミングツールから利用できるようになります。C++Builder には、タイプライブラリの作成と管理のためのタイプライブラリエディタがあります。

詳細については、ヘルプのキーワードで「タイプライブラリ」を検索してください。

第4章

チュートリアル： テキストエディタの作成

このチュートリアルでは、メニュー、ツールバー、およびステータスバーを備えたテキストエディタを作成します。

メモ このチュートリアルは、C++Builder のすべての版に対応しています。

新規アプリケーションの開始

新しいアプリケーションに取り掛かる前に、ソースファイルを格納するためのディレクトリを作成しておきます。

1. C:\Program Files\Borland\CBuilder6\Projects ディレクトリの中に TextEditor というディレクトリを作成してください。
2. 新規プロジェクトを開始するには、[ファイル | 新規作成 | アプリケーション] を選択するか、C++Builder を起動したときに作成されるデフォルトのプロジェクトを使用します。

各アプリケーションは「プロジェクト」として表されます。C++Builder を起動すると、デフォルトで空のプロジェクトが作成され、自動的に次のファイルが作成されます。

- Project1.cpp：プロジェクトに関連付けられたソースコードファイル。これは**プロジェクトファイル**と呼ばれます。
- Unit1.cpp：プロジェクトのメインフォームに関連付けられたソースコードファイル。これは**ユニットファイル**と呼ばれます。
- Unit1.h：プロジェクトのメインフォームに関連付けられたヘッダーファイル。これは**ユニットヘッダーファイル**と呼ばれます。
- Unit1.dfm：プロジェクトのメインフォームに関する情報を格納するリソースファイル。これは**フォームファイル**と呼ばれます。

プロパティ値の設定

各フォームには、それぞれ対応するユニットファイル (Unit1.cpp), ヘッダーファイル (Unit1.h), およびフォームファイル (Unit1.dfm) があります。フォームをもう 1 つ作成すると, 2 番目のユニットファイル (Unit2.cpp), ヘッダーファイル (Unit2.h), およびフォームファイル (Unit2.dfm) が自動的に作成されます。

3. [ファイル | すべて保存] を選択して, ファイルをディスクに保存します。[Unit1 に名前を付けて保存] ダイアログボックスが表示されたら,

- TextEditor フォルダに移動します。
- Unit1 をデフォルトの名前 (Unit1.cpp) で保存します。
- プロジェクトを TextEditor.bpr という名前で保存します (実行形式ファイルの名前は, プロジェクト名に拡張子 .exe を付加したものになります)。

[ファイル | すべて保存] を選択すれば, いつでも作業を保存することができます。

プロジェクトを保存すると, C++Builder はプロジェクトファイルなどの追加ファイルをプロジェクトディレクトリに作成します。これらのファイルは削除しないでください。



デフォルトのフォームには, 最大化ボタン, 最小化ボタン, 閉じるボタン, システムメニューがあります。

ここで [F9] を押してフォームを実行すると, これらのボタンはすべて機能します。

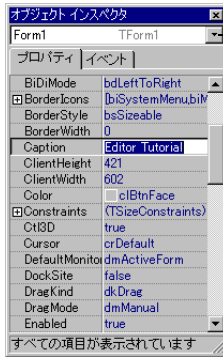
設計モードに戻るには, [x] をクリックしてフォームを閉じます。

プロパティ値の設定

新規プロジェクトを開くと, C++Builder ではデフォルトでプロジェクトのメインフォームとして Form1 という名前のフォームが表示されます。このフォームにコンポーネントを配置して, ユーザーインターフェースや他の部分を作成します。

フォームの横に表示されているオブジェクトインスペクタを使って, フォームおよびフォームに配置したコンポーネントのプロパティの値を設定できます。プロパティを設定すると, それに応じて C++Builder は対応するソースコードを保守します。オブジェクトインスペクタで設定した値を設計時設定と呼びます。

1. オブジェクトインスペクタ内で, フォームのデフォルトのキャプション「Form1」を表示している Caption プロパティに「Text Editor Tutorial」と入力します。フォームのタイトルバーに表示されるキャプションが変更されることに注目してください。



オブジェクトインспекタの一番上のドロップダウンリストには、現在選択されているコンポーネントが表示されます。ここでは、コンポーネントは Form1 で、その型は TForm です。

コンポーネントが選択されているときは、オブジェクトインспекタはそのプロパティを表示します。

1. フォームにコンポーネントが何もなくても、[F9] を押せばフォームを実行できます。
2. Form1 の設計時表示に戻るには、次のいずれかの操作を行います。
 - アプリケーション（実行時のフォーム）のタイトルバーの右上隅にある [×] をクリックします。
 - タイトルバーの左上隅のアプリケーション終了ボタンをクリックし、[閉じる] をクリックします。
 - [実行 | プログラムの終了] を選択します。

フォームへのコンポーネントの追加

フォームにコンポーネントを追加する前に、アプリケーションのユーザーインターフェース (UI) を作成するための最適な方法について考える必要があります。UI はアプリケーションユーザーがアプリケーションと対話するのを可能にするものなので、使いやすさを考慮して設計しなければなりません。

C++Builder には、アプリケーションの部品を表す多くのコンポーネントが用意されています。たとえば、メニュー、ツールバー、ダイアログボックス、その他の数多くのビジュアル（および非ビジュアル）プログラム要素のプログラミングを容易にするコンポーネント（オブジェクトとも呼ばれます）がコンポーネントパレットにあります。

テキストエディタアプリケーションには、編集領域、編集中のファイルの名前などを表示するステータスバー、メニュー、あるいはコマンドへ簡単にアクセスできるようにするためのボタンを配置したツールバーなどが必要になります。C++Builder を使用してインターフェースを設計する場合、いろいろなコンポーネントを試してその結果をすぐに見ることができるという利点があります。したがって、アプリケーションのインターフェースのプロトタイプをすばやく作成することができます。

テキストエディタを設計するうえでの最初の作業として、テキスト領域とステータスバーをフォームに追加します。

フォームへのコンポーネントの追加

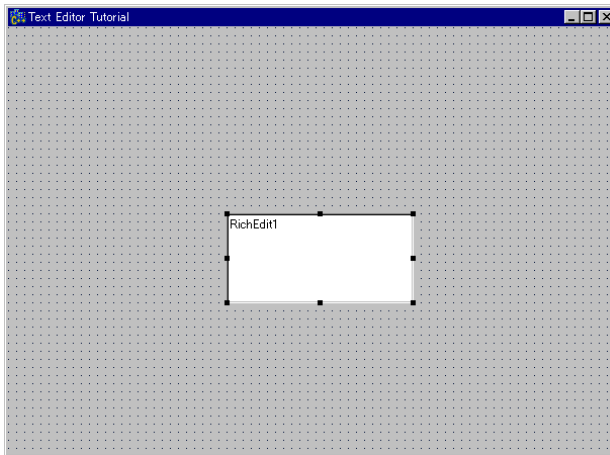


1. テキスト領域を作成するために、まず RichEdit コンポーネントを追加します。RichEdit コンポーネントを見つけるには、コンポーネントパレットの [Win32] ページのアイコンをマウスでポイントします。少し待つと、C++Builder はヘルプヒントとしてコンポーネントの名前を表示します。



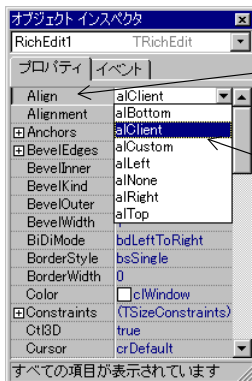
RichEdit コンポーネントが見つかったら、次のどちらかの操作をします。

- パレットでそのコンポーネントを選択し、フォーム上の配置する位置をクリックします。
- パレットでそのコンポーネントをダブルクリックして、フォームの中央に配置します。



C++Builder のコンポーネントはそれぞれがクラスであり、フォーム上にコンポーネントを配置するとそのクラスのインスタンスが作成されます。フォームにコンポーネントを配置すると、C++Builder は、アプリケーションを実行したときにインスタンスオブジェクトを構築するために必要なコードを作成します。

2. RichEdit コンポーネントを選択した状態で、オブジェクトインスペクタで Align プロパティのドロップダウン矢印をクリックして、alClient を設定します。



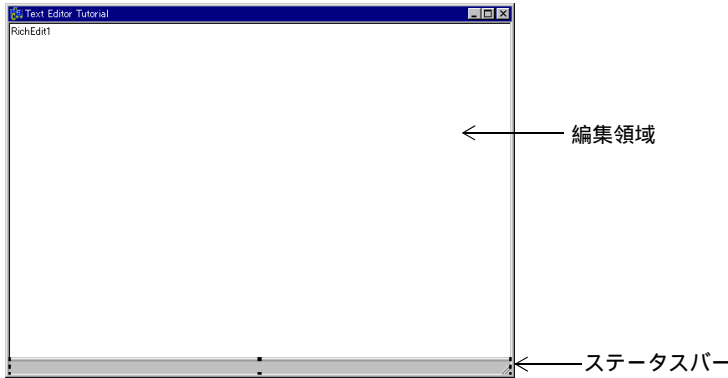
フォームの RichEdit1 コンポーネントを選択します。

オブジェクトインスペクタで Align プロパティを探します。下矢印をクリックしてプロパティのドロップダウンリストを表示します。

alClient を選択します。


RichEdit コンポーネントがフォーム全体に広がって、大きなテキスト編集領域が確保されます。Align プロパティに alClient 値を設定すると、フォームのサイズを変更しても RichEdit コントロールのサイズがウィンドウのサイズに合わせて変化します。

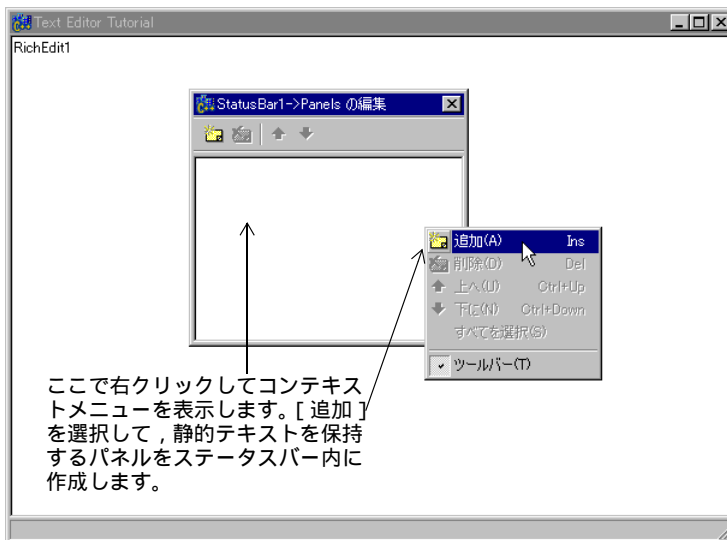
3. コンポーネントパレットの [Win32] ページにある StatusBar コンポーネントをダブルクリックして、フォームの一番下にステータスバーを追加します。



4. ステータスバーをダブルクリックして [StatusBar1->Panels の編集] ダイアログボックスを表示します。

ヒント [StatusBar1->Panels の編集] ダイアログボックスは、オブジェクトインスペクタでステータスバーの Panels プロパティの (TStatusPanel) の横の省略記号ボタンをクリックして表示することもできます。

5. ダイアログのツールバーにある [新規追加] ボタン  をクリックするか、右クリックして [追加] を選択し、ステータスバーにパネルを追加します。このパネルに、テキストエディタで編集中のファイルのパスと名前を表示することにします。



メニューとツールバーのサポートを追加する

- オブジェクトインスペクタで、Text プロパティを `untitled.txt` に変更します。このテキストエディタを使うときに、編集中のファイルがまだ保存されていない場合、その名前は `untitled.txt` になります。
 - [×] をクリックして、[StatusBar]->Panels の編集] ダイアログボックスを閉じます。
- 以上で、テキストエディタのユーザーインターフェースのメイン編集領域の設定はおしまいです。

メニューとツールバーのサポートを追加する

アプリケーションで何かの操作を行うためには、メニューやコマンド、ツールバーなどが必要になります。そのコマンドを直接コーディングすることもできますが、C++Builder のアクションリストまたはアクションマネージャを使えば、メニューコマンドとツールバーボタンに関するアクション、イメージ、およびコードを一元的に管理することができます。

ここでは慣例に従って、メニューコマンドに結び付けられるアクションに、トップレベルのメニュー名とそのコマンド名を組み合わせた名前を付けます。たとえば、FileExit アクションは [ファイル] メニューの [終了] コマンドを意味します。次の表に、テキストエディタアプリケーションに必要なメニューコマンドの種類と、そのアクションをツールバーボタンに関連付けるかどうかを示します。

表 4.1 テキストエディタに搭載予定のコマンド

メニュー	コマンド	ツールバーに配置	説明
ファイル	新規作成	はい	ファイルを新規作成する。
ファイル	開く	はい	既存のファイルを開く。
ファイル	保存	はい	現在のファイルをディスクに保存する。
ファイル	名前を付けて保存	いいえ	ファイルに新しい名前を付けて保存する（および新しいファイルに名前を付けて保存する）。
ファイル	終了	はい	エディタプログラムを終了する。
編集	切り取り	はい	テキストを削除してクリップボードに保存する。
編集	コピー	はい	テキストをコピーしてクリップボードに保存する。
編集	貼り付け	はい	クリップボードからテキストを挿入する。
ヘルプ	目次	いいえ	ヘルプトピックにアクセスするためのヘルプ目次を表示する。
ヘルプ	キーワード	いいえ	ヘルプのキーワード画面を表示する。
ヘルプ	バージョン情報	いいえ	アプリケーションについての情報をダイアログボックスに表示する。

アクションマネージャエディタとアクションリストエディタの違い

C++Builder の版によって、メニューとツールバーのアクションとイメージを管理する方法は 2 つあります。アクションリストエディタは C++Builder のすべての版で使用でき、ユーザーコマンドへの応答を一元的に管理することができます。アクションリストエディタは、CLX (クロスプラットフォーム用コンポーネントライブラリ) の一部であり、将来ほかのプラットフォームに移植する可能性がある場合には、アクションマネージャエディタではなく、こちらを使用する必要があります。

アクションマネージャエディタは、いくつか特別な機能を持っていますが、Windows プラットフォームに固有の VCL (ビジュアルコンポーネントライブラリ) の一部としてしか使用できません。アクションマネージャエディタのカスタマイズダイアログを使うと、エンドユーザーがカスタマイズ可能なメニューアクションや、Microsoft Office の一部の機能 (使用頻度の低いメニュー項目を非表示にするなど) を持つメニューアクションを提供することができます。またアクションマネージャでは、カスタマイズダイアログからフォーム上のメニューコンポーネントにアクションを直接ドラッグできるので、開発プロセスを簡略化することができます。

注意 このチュートリアルでは、C++Builder の Enterprise 版と Professional 版に含まれるアクションマネージャのカスタマイズダイアログを使用します。Enterprise 版または Professional 版を使用している場合は、次の「メニューとツールバーのイメージの追加 (Enterprise/Professional)」に進んでください。

Personal 版を使用している場合、またはアクションリストエディタを使用したい場合は、4-14 ページの「イメージリストとイメージの追加 (Personal 版)」に進んでください。

メニューとツールバーのイメージの追加 (Enterprise/Professional)

このセクションでは、アクションバンドとともに使用するイメージを追加します。

多くの場合、フォームに ImageList コンポーネントを追加して、独自のイメージをインポートします。このチュートリアルでは、手間を省くために C++Builder の IDE に使用されているイメージリストをインポートすることにします。独自のグラフィックが追加されていないければ、コンポーネントレット上の ImageList は標準アクションにはデフォルトイメージを使用します。

既存のイメージリストを追加するには、

1. C++Builder をデフォルトがディレクトリにインストールされている場合は、[ファイル | 開く] を選択して `C:\Program Files\Borland\CBuilder6\Source\vcl\actnres.pas` を選択します。[開く] ダイアログでこのファイルを見るには、[ファイルの種類] を [すべてのファイル (*.*)] に設定します。
2. ImageList1 コンポーネントを選択し、コピーしてフォームに貼り付けます。これは非ビジュアルコンポーネントなので、どこに貼り付けてもかまいません。

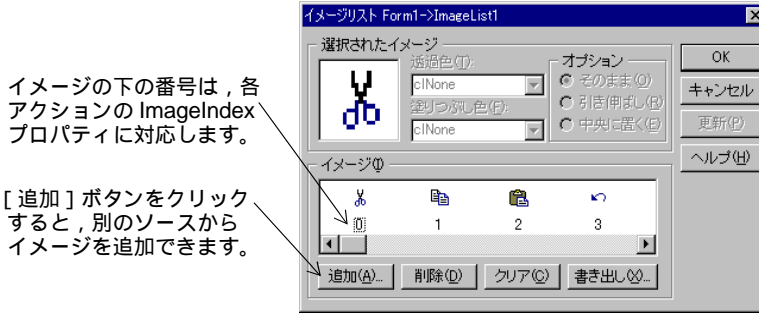


メモ ImageList1 をコピーするには、そのコンポーネントを右クリックして、[編集 | コピー] をクリックします。フォーム上で右クリックし、[編集 | 貼り付け] を選択します。

3. [StandardActions] ウィンドウを閉じます。

メニューとツールバーのサポートを追加する

4. ImageList1 コンポーネントをダブルクリックして、使用できるすべてのイメージを表示します。



追加した標準アクションと ImageList1 の ImageIndex プロパティは次のとおりです。

コマンド	ImageIndex プロパティ
[編集 切り取り]	0
[編集 コピー]	1
[編集 貼り付け]	2
[ファイル 新規作成]	6
[ファイル 開く]	7
[ファイル 上書き保存]	8
[ファイル 名前を付けて保存]	30
[ファイル 終了]	43
[ヘルプ 目次]	40

メモ まったく別のリストからイメージを追加できます。4-14 ページの「イメージリストとイメージの追加 (Personal 版)」を参照してください。

5. [OK] をクリックして [ImageList1] ダイアログを閉じます。

アクションマネージャにアクションを追加する (Enterprise/Professional)

Enterprise 版と Professional 版には、メニューとツールバーにアクションを簡単に追加できるアクションマネージャが含まれています。まずアクションマネージャを追加してからアクションを追加していきます。

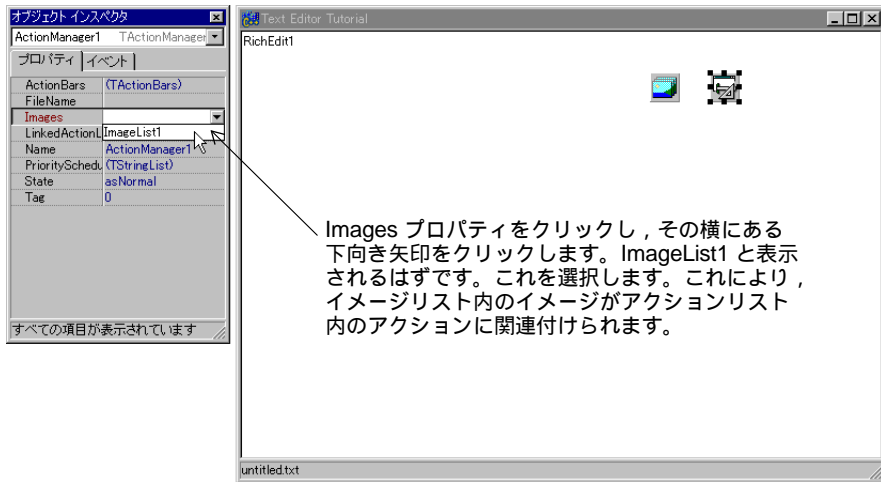


1. コンポーネントパレットの [Additional] ページにある ActionManager コンポーネントをダブルクリックして、フォームに配置します。このコンポーネントは非ビジュアルなので、フォームのどこに配置してもかまいません。

ヒント フォームに配置した非ビジュアルコンポーネントのキャプションを表示するには、[ツール | 環境オプション] を選択して [デザイナ] ページをクリックし、[コンポーネントのキャプションを表

示]を選択して[OK]をクリックします。また、コンポーネントの位置にマウスを合わせると、その名前が表示されます。

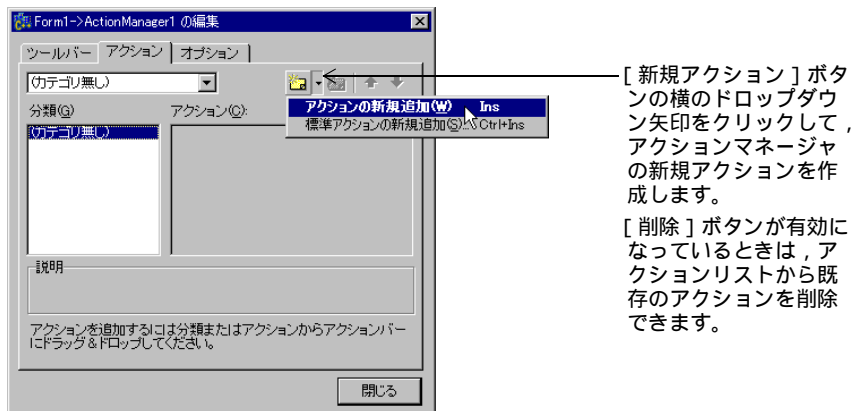
2. フォーム上の ActionManager1 を選択し、オブジェクトインスペクタで Images プロパティに ImageList1 を設定します。



次は、アクションマネージャにアクションを追加して、そのプロパティを設定します。すべてのプロパティを開発者が設定する非標準アクションと、自動的にプロパティが設定される標準アクションの両方を追加することになります。

3. ActionManager コンポーネントをダブルクリックして、アクションマネージャエディタを開きます。
[Form1->ActionManager1 の編集] ダイアログボックス (アクションマネージャエディタ) が表示されます。
4. [アクション] タブを表示します。[新規アクション] ボタンの横のドロップダウン矢印をクリックして、[アクションの新規追加] をクリックします。

ヒント アクションマネージャエディタを右クリックして [アクションの新規追加] を選択しても同じです。

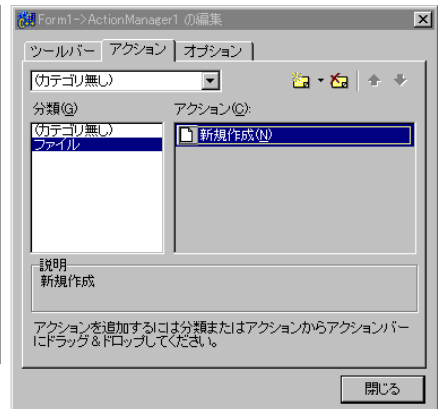


メニューとツールバーのサポートを追加する

5. [(カテゴリ無し)] を選択し、アクションリストの [アクション] の下の Action1 を選択しておきます。オブジェクトインスペクタで、プロパティを以下のように設定します。
 - Caption の右の列に、「新規作成 (&N)」と入力します。文字の前にアンド記号を付けると、その文字がコマンドにアクセスするためのショートカットとなります。
 - Category の右の列に「ファイル」と入力します（ファイルメニューのコマンドが1カ所にまとめられます）。
 - Hint の右の列に、「新規作成」と入力します（このテキストがヘルプヒントになります）。
 - ImageIndex を 6 に設定します（これで、インポートしたイメージリストに一致します。下矢印をクリックして、適切なイメージを選択することもできます）。
 - Name の右の列に、「FileNew」([ファイル | 新規作成] コマンドの意味) を入力し、[Enter] を押して変更を保存します。

アクションマネージャエディタで Action1 を選択し、オブジェクトインスペクタでそのプロパティを変更します。

Caption はアクションの名前、Category はアクションのタイプ、Hint はヘルプヒントです。ImageIndex は、イメージリスト内のイメージを参照します。Name は、コード内で使用されるアクションの名称です。



6. [Form1->ActionManager1 の編集] ウィンドウで [ファイル] が選択されていることを確認します。[新規アクション] ボタンの横のドロップダウン矢印をクリックして、[アクションの新規追加] をクリックします。
7. オブジェクトインスペクタで、プロパティを以下のように設定します。
 - Caption の右の列に、「上書き保存 (&S)」と入力します。
 - Category が「ファイル」になっていることを確認します。
 - Hint の右の列に、「上書き保存」と入力します。
 - ImageIndex の右の列で、「8」を選択します。
 - Name の右の列に、「FileSave」([ファイル | 上書き保存] コマンドの意味) と入力します。
8. [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[アクションの新規追加] をクリックします。
9. オブジェクトインスペクタで、プロパティを以下のように設定します。
 - Caption の右の列に、「キーワード (&I)」と入力します。
 - Category の右の列に、「ヘルプ」と入力します。
 - ImageIndex を指定する必要はありません。デフォルト値をそのまま使用します。
 - Name の右の列に、「HelpIndex」([ヘルプ | キーワード] コマンドの意味) と入力します。

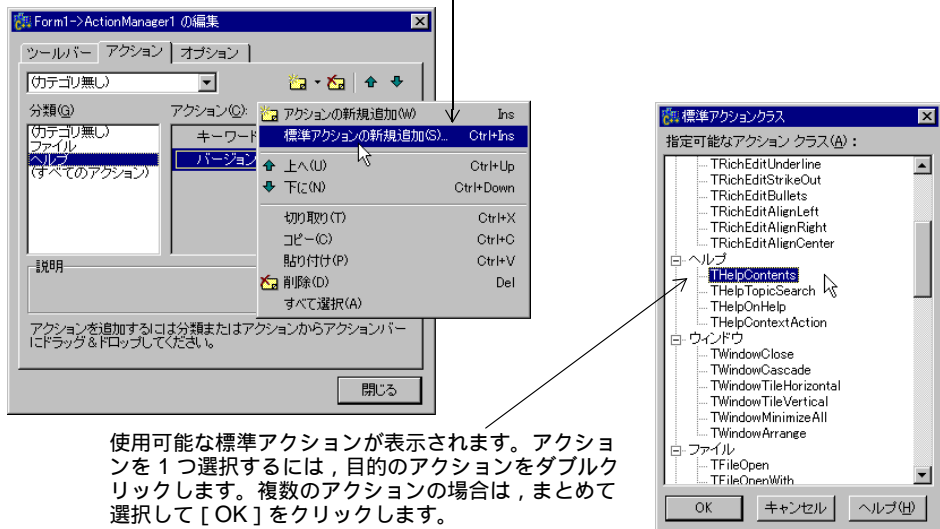
10. [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[アクションの新規追加] をクリックします。
11. オブジェクトインスペクタで、プロパティを以下のように設定します。
 - Caption の右の列に、「バージョン情報 (&A)」と入力します。
 - Category が「ヘルプ」になっていることを確認します。
 - ImageIndex を指定する必要はありません。デフォルト値をそのまま使用します。
 - Name の右の列に、「HelpAbout」([ヘルプ | バージョン情報] コマンドの意味)と入力します。
12. アクションマネージャのカスタマイズダイアログを表示したままにしておきます。
13. [ファイル | すべて保存] をクリックして作業を保存します。

標準アクションの追加 (Enterprise/Professional)

次は、アクションマネージャに標準アクション（開く、名前を付けて保存、終了、切り取り、コピー、貼り付け、およびヘルプ目次）を追加します。

1. アクションマネージャエディタがまだ表示されているはずですが。表示されていない場合は、ActionManager コンポーネントをダブルクリックして表示します。
 2. [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[標準アクションの新規追加] をクリックします。
- [標準アクションクラス] ダイアログボックスが表示されます。
3. [編集] カテゴリから、[Ctrl] キーを使って、TEditCut, TEditCopy, TEditPase を選択します。
- [OK] をクリックすると、この 3 つのアクションが、[Form1->ActionManager1 の編集] ダイアログボックスの [カテゴリ] リストの新しい [編集] カテゴリに追加されます。

アクションマネージャエディタ上で右クリックし、[標準アクションの新規追加] を選択します。



メニューとツールバーのサポートを追加する

4. [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[標準アクションの新規追加] をクリックします。
5. [ファイル] カテゴリまでスクロールして、TFileOpen, TFileSaveAs, TFileExit の 3 つを選択します。[OK] をクリックして、3 つのアクションを [ファイル] カテゴリに追加します。
6. [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[標準アクションの新規追加] をクリックします。
7. [ヘルプ] カテゴリまでスクロールして、THelpContents を選択します。[OK] をクリックして、このアクションを [ヘルプ] カテゴリに追加します。

メモ カスタムの [ヘルプ | 目次] コマンドでは、ヘルプファイルに [目次] タブが表示されるようになります。標準の [ヘルプ | 目次] コマンドでヘルプを呼び出すと、前回表示されていたタブページ ([目次], [キーワード], または [テキスト検索]) が表示されます。

これで、アプリケーションに必要な標準アクションがすべて追加されました。標準アクションには、イメージインデックスも含めて、プロパティがすべて自動的に設定されています。イメージインデックスを変更すれば、別のイメージを表示できます。

8. 必要であれば、標準アクションのイメージを変更することもできます。たとえば、[Form1->ActionManager1 の編集] ウィンドウで [ファイル | 開く] アクションを選択します。そして、デフォルトイメージをイメージ 7 に変更します。
9. [閉じる] ボタンを押して、アクションマネージャエディタを閉じます。
10. [ファイル | すべて保存] を選択して、変更をすべて保存します。

メニューの追加 (Enterprise/Professional)

次の 2 つのセクションでは、カスタマイズ可能なメニューバーとツールアクションバンドを追加します。テキストエディタのメニューバーには、3 つのドロップダウンメニュー (ファイル、編集、ヘルプ) とそのメニューコマンドがあります。アクションマネージャのカスタマイズダイアログを使うと、各メニューカテゴリとそのコマンドを 1 回の操作でメニューバーにドラッグすることができます。



1. コンポーネントパレットの [Additional] ページにある ActionMainMenuBar コンポーネントをダブルクリックしてフォームに追加します。
フォームの一番上にブランクのメニューバーが表示されます。
2. アクションマネージャのカスタマイズダイアログで、[カテゴリ] リストから [ファイル] を選択します。サブメニューのコマンドは希望の順序には並んでいませんが、[上へ] ボタンと [下へ] ボタン (または [Ctrl] + [↑] と [Ctrl] + [↓]) を使って簡単に順序を変更できます。
3. [開く] アクションを選択して、アクションマネージャカスタマイズダイアログのツールバーの [上へ] ボタンをクリックすると、[ファイル] メニューのコマンドは、[新規作成], [開く], [上書き保存], [名前を付けて保存], 「終了」の順序になります。
4. [ファイル] をメニューバーまでドラッグします。[ファイル] メニューとそのサブメニューのコマンドがメニューバーに表示されます。

- ヒント メニューカテゴリをメニューバーにドラッグした後も、メニューコマンドを位置を変更することができます。たとえば、メニューバーの [ファイル] をクリックするとそのメニューコマンドが表示され、[開く] を [新規作成] より上にドラッグすれば、元の位置に戻せます。
5. アクションマネージャカスタマイズダイアログの [カテゴリ] リストから、[編集] をメニューバーの [ファイル] の右までドラッグします。
 6. アクションマネージャカスタマイズダイアログの [カテゴリ] リストから、[ヘルプ] をメニューバーの [編集] の右までドラッグします。
 7. [ヘルプ] メニューをクリックして、サブメニューコマンドを表示します。[目次] コマンドを [キーワード] コマンドの上にドラッグします。
 8. [Esc] を押すか [ヘルプ] メニューをもう一度クリックして閉じます。
 9. [ファイル | すべて保存] を選択して、作業の内容を保存します。
- 次は、簡単にコマンドにアクセスできるようにツールバーを追加します。

ツールバーの追加 (Enterprise/Professional)

アクションマネージャカスタマイズダイアログでアクションを設定したので、メニューで使用されたのと同じアクションの一部をアクションバンドツールバーに追加することができます。アクションバンドツールバーは、完成すると Microsoft Office 2000 のツールバーに似たものになります。



1. コンポーネントパレットの [Additional] ページにある ActionToolBar コンポーネントをダブルクリックしてフォームに追加します。
- ブランクのアクションバンドツールバーがメニューバーの下に表示されます。

- ヒント アクションバンドツールバーは、アクションマネージャカスタマイズダイアログを開いて [ツールバー] タブをクリックし、[新規作成] ボタンをクリックして追加することもできます。
2. アクションマネージャカスタマイズダイアログで、[カテゴリ] リストから [ファイル] を選択します。アクションリストで、[新規作成]、[開く]、[終了] を選択して、ツールバーにドラッグします。これらについては、それぞれ割り当てられているイメージが自動的にボタンとして表示されます。
 3. アクションマネージャカスタマイズダイアログで、[編集] カテゴリをツールバーにドラッグします。すべての編集コマンドがツールバーに表示されるはずですが。
- メモ 間違ったコマンドをツールバーにドラッグした場合は、ツールバーの外にドラッグして削除できます。あるいは、オブジェクトツリービューでその項目を選択して [Delete] キーを押して削除することもできます。ボタンの位置は、それぞれを左か右にドラッグするだけで変更できます。
4. [ファイル | すべて保存] を選択して、作業した内容を保存します。
 5. [F9] を押して、プロジェクトをコンパイルし実行します

- ヒント プロジェクトの実行は、デバッグツールバーの [実行] ボタンを押すか、[実行 | 実行] を選択しても行えます。プロジェクトを実行すると、C++Builder はフォーム上で設計したときと同じような実行時ウィンドウでプログラムを実行します。

メニューとツールバーのサポートを追加する

これで予定した機能の多くがテキストエディタに追加されました。テキスト領域内でテキストを選択すると、[切り取り], [コピー], および [貼り付け] の各ボタンが動作するはずですが、コマンドのいくつかは淡色表示されますが、メニューとツールバーボタンはすべて機能します。一部のコマンドは、機能させるためにはイベントハンドラを書く必要があります。

6. 設計モードに戻るには、アプリケーションの右上隅の [×] をクリックします。

チュートリアルを続けるには、4-23 ページページの「テキスト領域のクリア (すべての版)」に進んでください。

イメージリストとイメージの追加 (Personal 版)

このセクションでは、フォームに ImageList コンポーネントを追加し、そのリストにイメージを追加します。コンポーネントパレットの [Win32] ページで ImageList コンポーネントを選択し、フォームに ImageList1 コンポーネントを追加します。

各コマンドで使用するイメージを次に示します。

コマンド	イメージ名	ImageIndex プロパティ
[ファイル 新規作成]	Filenew.bmp	0
[ファイル 開く]	Fileopen.bmp	1
[ファイル 上書き保存]	Filesave.bmp	2
[ファイル 終了]	Doorshut.bmp	3
[編集 切り取り]	Cut.bmp	4
[編集 コピー]	Copy.bmp	5
[編集 貼り付け]	Paste.bmp	6
[ヘルプ 目次]	Help.bmp	7

メモ グラフィックをインポートしない場合、ImageList は標準アクションにはデフォルトイメージを使用します。

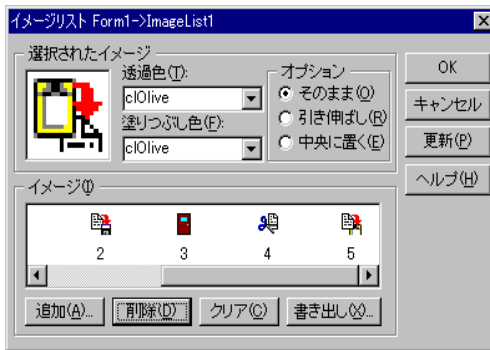
イメージリストにイメージを追加するには、

1. フォーム上の ImageList コンポーネントをダブルクリックしてイメージリストエディタを表示します。
2. [追加] ボタンをクリックします。
3. [イメージの追加] ダイアログボックスで、製品に付属するボタンイメージが格納されているディレクトリに移動します。デフォルトの位置は、C: ¥ Program Files ¥ Common Files ¥ Borland Shared ¥ Images ¥ Buttons です。
4. filenew.bmp をダブルクリックします。

ビットマップを 2 個に分割するかどうかを尋ねるメッセージが表示された場合は、[はい] をクリックします。それぞれのアイコンには、アクティブ表示のイメージと淡色表示のイメージが含まれています。両方のイメージが表示されるはずですが、淡色表示 (2 番目) のイメージを削除します。

5. ほかのイメージを追加するには、

- [追加] をクリックし、fileopen.bmp をダブルクリックします。淡色表示のイメージを削除します。
- [追加] をクリックし、filesave.bmp をダブルクリックします。淡色表示のイメージを削除します。
- [追加] をクリックし、doorshut.bmp をダブルクリックします。淡色表示のイメージを削除します。
- [追加] をクリックし、cut.bmp をダブルクリックします。淡色表示のイメージを削除します。
- [追加] をクリックし、copy.bmp をダブルクリックします。淡色表示のイメージを削除します。
- [追加] をクリックし、paste.bmp をダブルクリックします。淡色表示のイメージを削除します。



ヒント イメージをクリックするときに [Ctrl] キーを使うと、複数のイメージをまとめて選択できます。そして、淡色表示のイメージを削除します。

[OK] をクリックしてイメージリストエディタを閉じます。

これで、各アクションの ImageIndex プロパティに対応する 0 ~ 6 の番号が付けられた 7 個のイメージをイメージリストに追加したことになります。

メモ 上記とは異なる順序でイメージを追加した場合は、イメージリストエディタ内でドラッグアンドドロップすることによってイメージを並べ替えることができます。

メニューやツールバーを追加する準備が整いました。

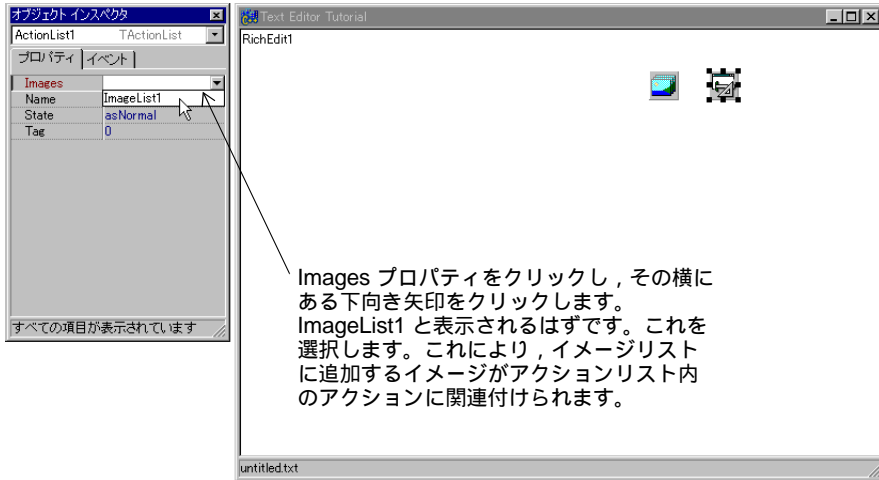
アクションリストにアクションを追加する (Personal 版)

ここまでで、フォームに ImageList コンポーネントを追加し、その ImageList にイメージを追加しました。このセクションでは、アクションリストとアクションを追加します。

1. コンポーネントパレットの [Standard] タブにある ActionList コンポーネントをダブルクリックします。フォームに追加される ActionList コンポーネントは非ビジュアルコンポーネントなので、フォーム上のどこに配置してもかまいません。

メニューとツールバーのサポートを追加する

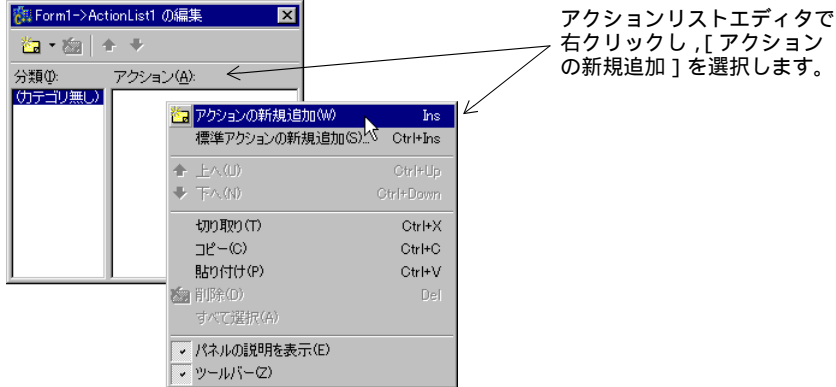
2. フォーム上の ActionList コンポーネントが選択されている状態で、Images プロパティに ImageList1 を設定します。



3. ActionList コンポーネントをダブルクリックして開きます。

[Form1->ActionList1 の編集] 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はアクションリストエディタとも呼ばれます。

4. アクションリストエディタで右クリックして [アクションの新規追加] を選択します。



ヒント [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[アクションの新規追加] をクリックしても同じです。

5. オブジェクトインスペクタで、Action1 のプロパティを以下のように設定します。
 - Caption の右の列に、「新規作成 (&N)」と入力します。文字の前にアンド記号を付けると、その文字がコマンドにアクセスするためのショートカットとなります。
 - Category の右の列に「ファイル」と入力します（ファイルメニューのコマンドが 1 カ所にまとめられます）。
 - Hint の右の列に、「新規作成」と入力します（このテキストがヘルプヒントになります）。

- ImageIndex の右の列で、対応するイメージを選択します（ここまでの手順通りにイメージリストを追加している場合はイメージ0になります）。
 - Name の右の列に、「FileNew」（[ファイル | 新規作成] コマンドの意味）を入力し、[Enter] を押して変更を保存します。
6. アクションリストエディタで右クリックし、[アクションの新規追加] を選択します。
 7. オブジェクトインスペクタで、Action1 のプロパティを以下のように設定します。
 - Caption の右の列に、「上書き保存 (&S)」と入力します。
 - Category を「ファイル」に設定します。
 - Hint の右の列に、「上書き保存」と入力します。
 - ImageIndex の右の列で、対応するイメージを選択します（ここまでの手順通りにイメージを追加している場合はイメージ2になります）。
 - Name の右の列に、「FileSave」（[ファイル | 上書き保存] コマンドの意味）と入力します。
 8. アクションリストエディタ上で右クリックし、[アクションの新規追加] を選択します。
 9. オブジェクトインスペクタで、Action1 のプロパティを以下のように設定します。
 - Caption の右の列に、「キーワード (&I)」と入力します。
 - Category の右の列に、「ヘルプ」と入力します。
 - ImageIndex を指定する必要はありません。デフォルト値をそのまま使用します。
 - Name の右の列に、「HelpIndex」（[ヘルプ | キーワード] コマンドの意味）と入力します。
 10. アクションリストエディタ上で右クリックし、[アクションの新規追加] を選択します。
 11. オブジェクトインスペクタで、Action1 のプロパティを以下のように設定します。
 - Caption の右の列に、「バージョン情報 (&A)」と入力します。
 - Category の右の列に、「ヘルプ」と入力します。
 - ImageIndex を指定する必要はありません。デフォルト値をそのまま使用します。
 - Name の右の列に、「HelpAbout」（[ヘルプ | バージョン情報] コマンドの意味）と入力します。

アクションリストエディタを表示したままにしておきます。

アクションリストに標準アクションを追加する (Personal 版)

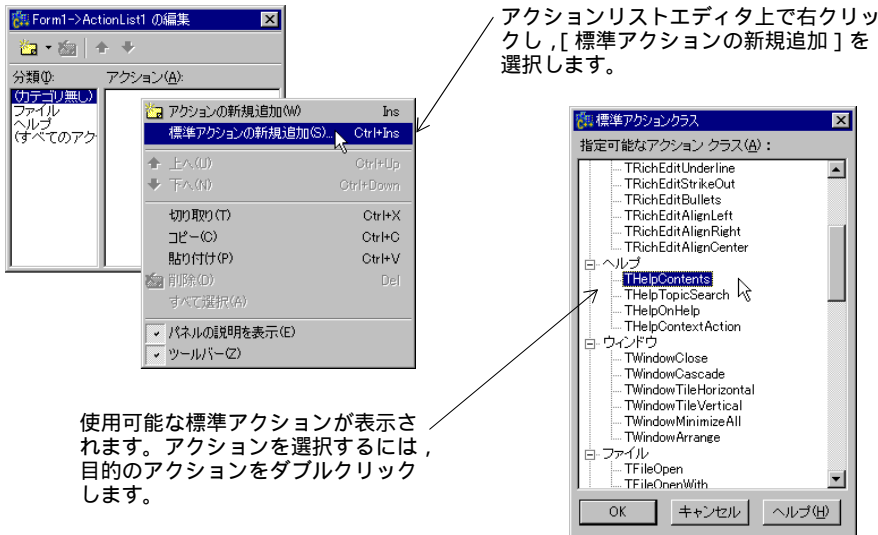
C++Builder では、アプリケーションの開発において頻繁に使用されるいくつかの標準アクションが提供されています。ここでは、標準アクション（切り取り、コピー、貼り付けなど）をアクションリストに追加します。

1. 画面上にはアクションリストエディタがまだ表示されているはずですが、表示されていない場合は、フォーム上の ActionList コンポーネントをダブルクリックします。
2. アクションリストエディタを右クリックして、[標準アクションの新規追加] をクリックします。

ヒント [新規アクション] ボタンの横のドロップダウン矢印をクリックして、[標準アクションの新規追加] をクリックしても同じです。

メニューとツールバーのサポートを追加する

3. [標準アクションクラス] ダイアログボックスで,[編集] カテゴリまでスクロールし,[Ctrl] キーを使って TEditCut, TEditCopy, および TEditPaste を選択します。[OK] をクリックして,3つのアクションをアクションリストエディタの[編集] カテゴリに追加します。



4. アクションリストエディタを右クリックして,[標準アクションの新規追加]をクリックします。
5. [ファイル] カテゴリまでスクロールして,TFileOpen,TFileSaveAs,TFileExitの3つのアクションを選択します。[OK] をクリックして,3つのアクションを[ファイル]カテゴリに追加します。
6. アクションリストエディタを右クリックして,[標準アクションの新規追加]をクリックします。
7. [ヘルプ] カテゴリまでスクロールして,THelpContentsを選択します。[OK] をクリックして,このアクションを[ヘルプ]カテゴリに追加します。

メモ カスタムの[ヘルプ | 目次] コマンドは,ヘルプファイルに[目次]タブが表示されるようになります。標準の[ヘルプ | 目次] コマンドでヘルプを呼び出すと,前回表示されていたタブページ([目次],[キーワード],または[テキスト検索])が表示されます。

標準アクションには,プロパティが自動的に設定されています。ただしイメージインデックスプロパティは,Personal版に付属する適切なイメージに関連付けるために変更する必要があります。

8. アクションリストエディタの[カテゴリ]リストで[(すべてのアクション)]を選択します。
9. 標準アクションにはデフォルトのイメージが含まれています。ここでは,デフォルトイメージを,前に追加したイメージに変更します。アクションリストエディタで以下のアクションを1つずつ選択し,オブジェクトインスペクタでそれぞれの ImageIndex プロパティを変更します。
 - EditCut1 を選択し, ImageIndex プロパティに 4 を設定する。
 - EditCopy1 を選択し, ImageIndex プロパティに 5 を設定する。
 - EditPaste1 を選択し, ImageIndex プロパティに 6 を設定する。
 - FileOpen1 を選択し, ImageIndex プロパティに 1 を設定する。
 - FileExit1 を選択し, ImageIndex プロパティに 3 を設定する。

10. [×] をクリックしてアクションリストエディタを閉じます。

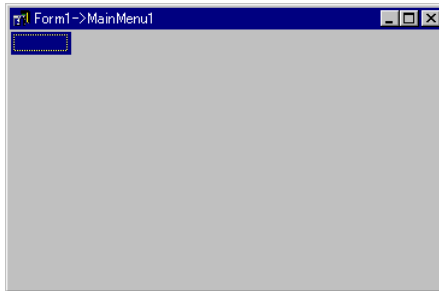
11. [ファイル | すべて保存] を選択して、作業した内容を保存します。

メニューの追加 (Personal 版)

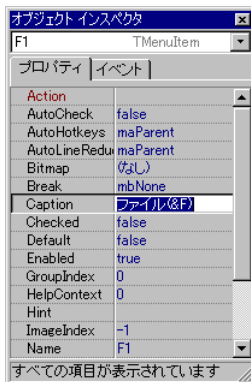
このセクションでは、3つのドロップダウンメニュー ([ファイル], [編集], [ヘルプ]) を持つメインメニューバーを追加し、アクションリスト内の標準アクションを使って各メニューにメニュー項目を追加します。



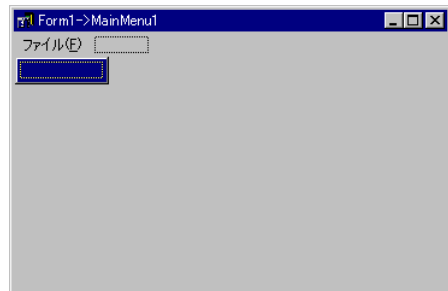
1. コンポーネントパレットの [Standard] タブから、MainMenu コンポーネントをフォームに配置します。これは、どこに配置してもかまいません。
2. メインメニューの Images プロパティに ImageList1 を設定します。これにより、メニューコマンドにイメージを表示できるようになります。
3. MainMenu コンポーネントをダブルクリックしてメニューデザイナーを表示します。



4. メニューデザイナーで最初の最上位メニュー項目を設定するために、オブジェクトインスペクタで Caption プロパティに「ファイル (&F)」と入力し、[Enter] を押します。



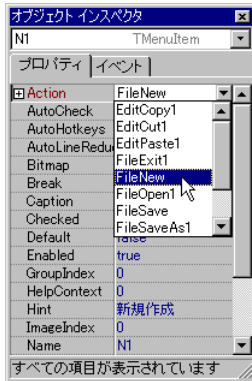
「ファイル (&F)」と入力してからメニューデザイナーにフォーカスを移すと最上位コマンド [ファイル (F)] が表示され、最初のメニュー項目を追加できます。



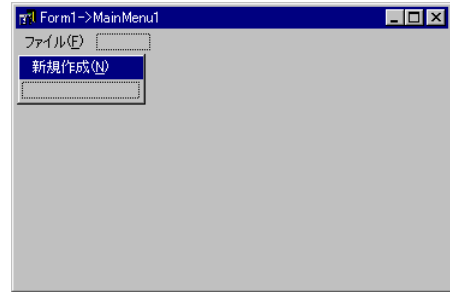
5. 作成した [ファイル] コマンドの下の空白の項目を選択します。

メニューとツールバーのサポートを追加する

6. オブジェクトインスペクタで、Action プロパティを FileNew に設定します。ここでは、アクションリストのすべてのアクションが表示されます。



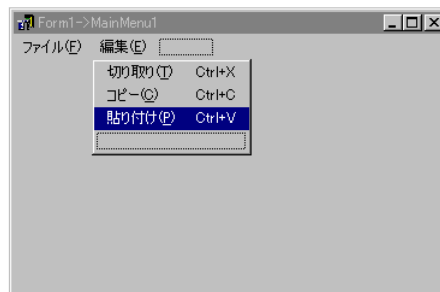
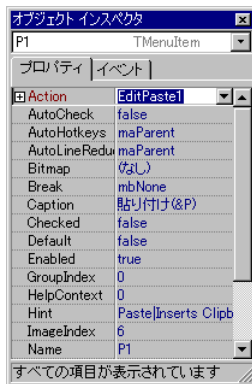
Action プロパティリストから FileNew を選択すると、適切な Caption と ImageIndex が設定された [新規作成 (N)] コマンドが表示されます。



- [新規作成 (N)] の下の項目を選択し、Action プロパティを FileOpen1 に設定します。
- [開く (O)] の下の項目を選択し、Action プロパティを FileSave に設定します。
- [上書き保存 (S)] の下の項目を選択し、Action プロパティを FileSaveAs1 に設定します。
- [名前を付けて保存 (A)] の下の項目を選択します。Caption の右側の列にハイフンを入力して、[Enter] を押します。これにより、メニューの分割線が作成されます。
- 分割線の下項目を選択し、Action プロパティを FileExit1 に設定します。

7. 次に、[編集] メニューを作成します。

- [ファイル (F)] コマンドの右側の項目を選択し、Caption プロパティに「編集 (&E)」と入力して [Enter] を押します。
- フォーカスは [編集 (&E)] の下をクリックして、空白の項目を選択します。Action プロパティを EditCut1 に設定します。
- [切り取り (T)] の下の項目を選択し、Action プロパティを EditCopy1 に設定します。
- [コピー (C)] の下の項目を選択し、Action プロパティを EditPaste1 に設定します。



8. [ヘルプ] メニューを作成します。

- [編集 (E)] コマンドの右側の項目を選択し、Caption プロパティに「ヘルプ (&H)」と入力して [Enter] を押します。

- [ヘルプ(H)] の下の項目を選択し、Action プロパティを HelpContents1 に設定します。
- [目次(&C)] の下の項目を選択し、Action プロパティを HelpIndex に設定します。
- [キーワード] の下の項目を選択します。Caption の右側の列にハイフンを入力し、[Enter] を押して [ヘルプ(H)] メニューの分割線を作成します。
- 分割線の下項目を選択し、Action プロパティを HelpAbout に設定します。

9. [×] をクリックしてメニューデザイナーを閉じます。

10.[ファイル | すべて保存] を選択します。

11.[F9] を押して、プロジェクトをコンパイルし実行します。

12.設計モードに戻るには、アプリケーションの右上隅の [×] をクリックします。

メモ フォームが見えなくなった場合は、[表示 | フォーム] を選択して Form1 を選択し、[OK] をクリックします。

ツールバーの追加 (Personal 版)

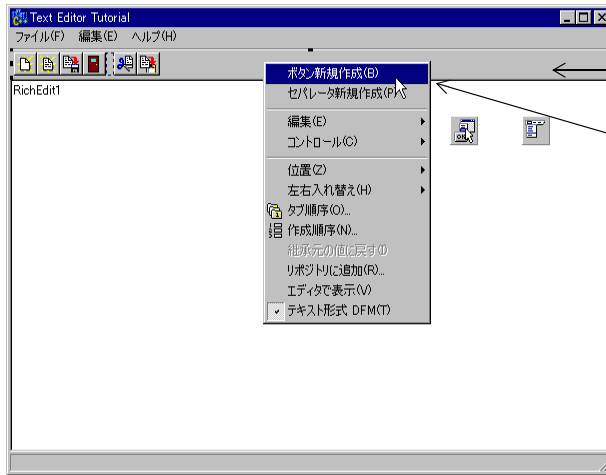
アクションリストにはすでにアクションを設定してあるので、メニューに使用したのと同じアクションのいくつかをツールバーに設定できます。



1. コンポーネントパレットの [Win32] タブにある ToolBar をダブルクリックしてフォームにツールバーを追加します。メインメニューの下に空白のツールバーが追加されます。
2. ツールバーが選択された状態で、オブジェクトインスペクタを使ってプロパティを以下のように変更します。
 - Images プロパティに ImageList1 を設定する。
 - ツールバーの Indent プロパティを 4 に設定する (これにより、アイコンがツールバーの左端から 4 ピクセル分右にインデントされます)。
 - ShowHint を true に設定する (ヒント: false をダブルクリックすると true に切り替わります)。
3. ツールバーにボタンを追加するために、右クリックして [ボタン新規作成] を 4 回選択する。
4. ツールバーにセパレータ (分割線) を追加するために、右クリックして [セパレータ新規作成] を選択する。
5. 右クリックして [ボタン新規作成] をさらに 3 回選択する。

メモ この時点では、正しいアイコンが表示されなくてもかまいません。ボタンにアクションを割り当てれば正しいアイコンが選択されます。

メニューとツールバーのサポートを追加する

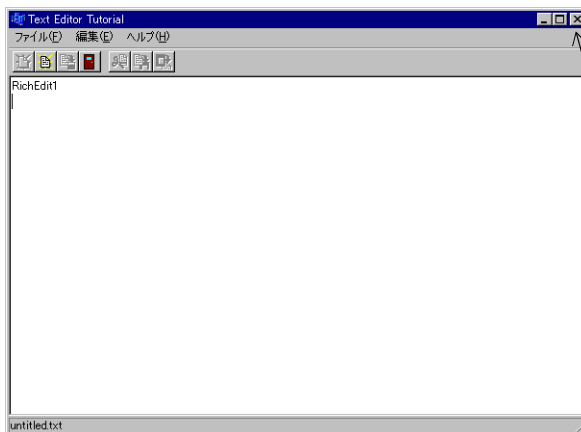


ツールバーオブジェクトはデフォルトでメニューの下に追加されます。

ボタンまたはセパレータを追加するには、ツールバーを選択し、右クリックして [ボタン新規作成] または [セパレータ新規作成] を選択します。そのあとでアクションリストからアクションを割り当てます。

1. 最初のボタン群にアクションリストからアクションを割り当てます。
 - 最初のボタンを選択し、Action プロパティに FileNew を設定する。
 - 2 番目のボタンを選択し、Action プロパティに FileOpen1 を設定する。
 - 3 番目のボタンを選択し、Action プロパティに FileSave を設定する。
 - 4 番目のボタンを選択し、Action プロパティに FileExit1 を設定する。
2. 次のボタン群にアクションを割り当てます。
 - 5 番目のボタンを選択し、Action プロパティに EditCut1 を設定する。
 - 6 番目のボタンを選択し、Action プロパティに EditCopy1 を設定する。
 - 最後のボタンを選択し、Action プロパティに EditPaste1 を設定する。
3. [ファイル | すべて保存] を選択します。
4. [F9] を押して、プロジェクトをコンパイルし実行します

メモ プロジェクトの実行は、デバッグツールバーの [実行] ボタンを押すか、[実行 | 実行] を選択しても行えます。



[F9] を押してプロジェクトを実行すると、アプリケーションのインターフェースが表示されます。メニュー、テキスト領域、およびステータスバーがすべてフォーム上に表示されます。設計モードに戻るには、[×] をクリックしてフォームを閉じます。

プロジェクトを実行すると、C++Builder はフォーム上で設計したときと同じような実行時ウィンドウでプログラムを実行します。ほとんどのコマンドは淡色表示されていますが、メニューはすべて機能します。メニュー項目の横には、イメージインデックスを使って関連付けたイメージが表示されます。

これで予定した機能の多くがテキストエディタに追加されました。テキスト領域に文字を入力してみてください。テキスト領域内でテキストを選択すると、[切り取り]、[コピー]、および [貼り付け] の各ボタンが動作するはずですが、

5. 右上隅の [×] をクリックしてアプリケーションを閉じ、設計時のフォームの表示に戻ります。

テキスト領域のクリア (すべての版)

重要 チュートリアルでの以降の部分は、すべての版に共通です。

プログラムを実行したとき、テキスト領域に RichEdit1 という名前が表示されました。このテキストは、文字列リストエディタを使って削除できます。ここでテキストをクリアしない場合は、最後のステップでメインフォームを初期化するときにテキストを削除する必要があります。

テキスト領域をクリアするには、

1. メインフォーム上の RichEdit1 コンポーネントをクリックします。
2. オブジェクトインスペクタで、Lines プロパティの横の値「(TStrings)」をダブルクリックして文字列リストエディタを表示します。
3. 文字列リストエディタで、テキスト (RichEdit1) を選択して削除し、[OK] をクリックします。
4. 変更内容を保存してからプログラムを実行します。

メインフォームが表示されたときにテキスト編集領域がクリアされているはずですが、

イベントハンドラの作成

ここまででは、コードを 1 行も書かずにアプリケーションを開発してきました。設計時にオブジェクトインスペクタを使ってプロパティの値を設定することで、C++Builder の RAD 環境の利点をフル活用してきたわけです。このセクションでは、アプリケーションの実行中にユーザー入力にตอบสนองする「イベントハンドラ」と呼ばれる関数のコードを書きます。イベントハンドラをメニューおよびツールバーの項目に結び付けることによって、項目が選択されたときにアプリケーションがイベントハンドラのコードを実行するようにします。

標準でないアクションについては、開発者がイベントハンドラを作成しなければなりません。[ファイル | 終了] や [編集 | 貼り付け] などの標準アクションのイベントは、生成されたコードに含まれています。しかし、[ファイル | 名前を付けて保存] などの一部のコマンドについては、独自のイベントハンドラを書いてコマンドをカスタマイズする場合があります。

メニュー項目とツールバーのアクションはすべてアクションマネージャまたはアクションリストエディタに集約されているので、アクションリストを使用してイベントハンドラを作成します。

重要 C++Builder の Personal 版を使用している場合は、以下の手順では ActionManager コンポーネントの代わりに ActionList コンポーネントを使ってください。

新規作成コマンドのイベントハンドラの作成

[新規作成] コマンドのイベントハンドラを作成するには、

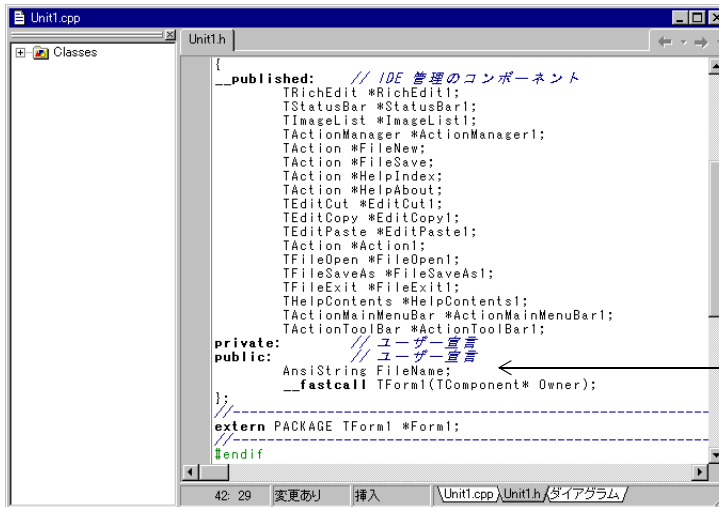
1. [表示 | ユニットの表示] を選択し、Unit1 を選択して Form1 に関連付けられたコードを表示します。
2. イベントハンドラで使用するファイル名を宣言する必要があります。これには、ファイル名用のカスタムプロパティを追加して、他のメソッドからグローバルにアクセスできるようにします。コードエディタで Unit1.cpp ファイルを右クリックして [ソース / ヘッダーファイルを開く] を選択し、Unit1.h ファイルを開きます。ヘッダーファイル内で、TForm1 クラスのパブリック宣言部を探し、次の行の下に

```
public:    // ユーザー宣言
```

次のように入力します。

```
AnsiString FileName;
```

画面は次のようになるはずですが。



この行では、FileName をほかのメソッドからグローバルにアクセスできる文字列として定義しています。

3. [F12] を押してメインフォームに戻ります。

ヒント [F12] を使用すると、フォームとそれに関連付けられているコードの表示を切り替えることができます。また、[表示 | フォーム] を選択し、Form1 を選択しても同じです。

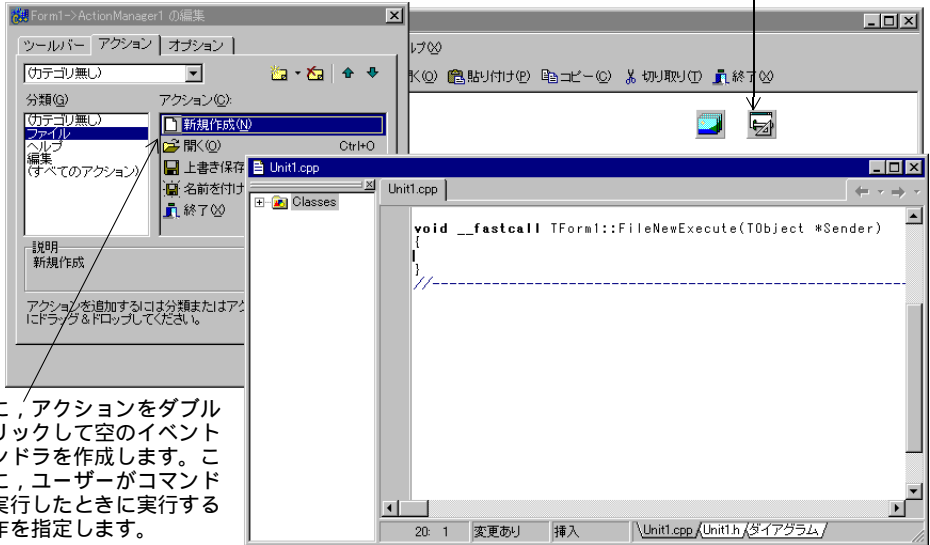
4. ActionManager または ActionList コンポーネントをダブルクリックして開きます。

5. [ファイル | 新規作成] (FileNew) アクションをダブルクリックします。

ヒント オブジェクトツリーで [ファイル | 新規作成] をダブルクリックしても同じです。

コードエディタが開かれ、イベントハンドラ内にカーソルが位置付けられます。

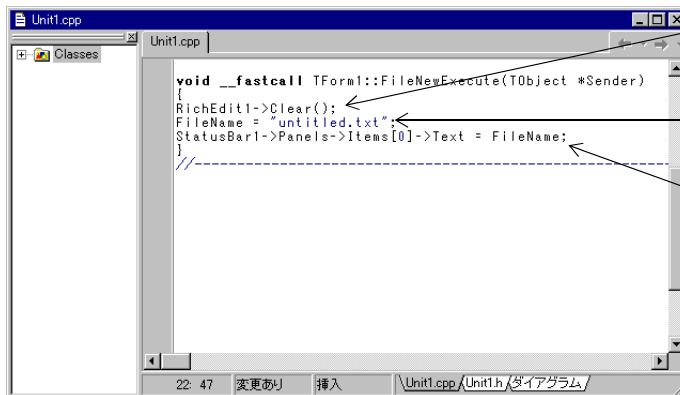
まず、ActionManager または ActionList オブジェクトをダブルクリックしてそのエディタを表示します。



6. コードエディタ内のカーソルが位置付けられたところ（{と}の間）に次のように入力します。

```
RichEdit1->Clear();
FileName = "untitled.txt";
StatusBar1->Panels->Items[0]->Text = FileName;
```

イベントハンドラは以下になるはずですが。



この行は、新しいファイルを作成するときにテキスト領域をクリアします。

この行では、新しいファイルの名前を Untitled.txt として定義しています。

この行は、ファイル名をステータスバーに表示します。

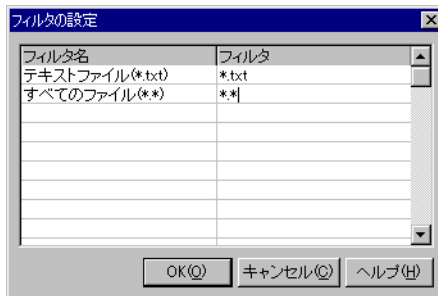
7. [ファイル | すべて保存] を選択します。

メモ コードを表示しているウィンドウのサイズを変更すれば、水平方向にスクロールする必要がなくなりコードが見やすくなります。

開くコマンドのイベントハンドラの作成

テキストエディタでファイルを開くために、Windows 標準のファイルを開くダイアログボックスを使用します。すでにアクションマネージャまたはアクションリストのエディタに標準の [ファイル | 開く] コマンドを追加してありますが、これによって自動的にこのダイアログボックスが含まれます。しかし、このコマンドのイベントハンドラをカスタマイズする必要があります。

1. [F12] を押してメインフォームに移動し、ActionManager または ActionList コンポーネントをダブルクリックして前面に表示します。
2. [ファイル | 開く] (FileOpen1) アクションを選択します。
3. オブジェクトインスペクタで、Dialog プロパティの [+] をクリックして、サブプロパティを開きます。Dialog は、[開く] ダイアログボックスを作成する被参照コンポーネントです。C++Builder は、このダイアログボックスにデフォルトで FileOpen1->OpenDialog という名前を付けます。(OpenDialog1 の Execute メソッドを呼び出すと、ファイルを開くための標準ダイアログボックスが呼び出されます)。
4. DefaultExt に txt を設定します。
5. Filter の横のテキスト領域をダブルクリックしてフィルタエディタを表示します。
 - [フィルタ名] 列の下の最初の行に「テキストファイル (*.txt)」と入力します。[フィルタ] 列に「*.txt」と入力します。
 - [フィルタ名] 列の2番目の行に「すべてのファイル (*.*)」と入力し、[フィルタ] 列に「*. *」と入力します。
 - [OK] をクリックします。



フィルタエディタを使って、FileOpen1.Dialog と FileSaveAs1.Dialog のアクション用のフィルタを定義します。

6. Title を「ファイルを開く」に設定します。このテキストは、ファイルを開くダイアログボックスの一番上に表示されます。
7. [イベント] タブをクリックします。OnAccept イベントの横の領域をダブルクリックすると、FileOpen1Accept が表示されます。

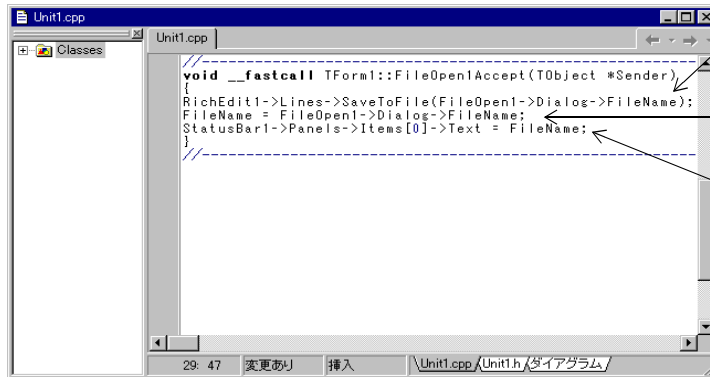
コードエディタが開かれ、イベントハンドラ内にカーソルが位置付けられます。
8. カーソルが位置付けられたところ ({ と } の間) に次のように入力します。

```
RichEdit1->Lines->SaveToFile(FileOpen1->Dialog->FileName);
FileName = FileOpen1->Dialog->FileName;
StatusBar1->Panels->Items[0]->Text = FileName;
```

ヒント

2-6 ページで説明している支援機能ツールを使うと、コーディングを短時間で進めることができます。たとえば、RichEdit1 に続いて矢印 (->) を入力すると、コード補完のダイアログボックスが表示されます。「1」(エル) を入力すると、ダイアログボックスの先頭に Lines : TStrings; が表示されます。[Enter] を押すか、その項目をダブルクリックすると、それがコードに追加されます。

FileOpen のイベントハンドラは以下のようになるはずです。



この行は、指定されたファイルからテキストを挿入します。

この行は、ファイル名を [ファイルを開く] ダイアログ内の名前に設定します。

この行は、ファイル名をステータスバーに表示します。

これで [ファイル | 開く] コマンドと [ファイルを開く] ダイアログの処理が完成しました。

上書き保存コマンドのイベントハンドラの作成

[上書き保存] コマンドのイベントハンドラを作成するには、

1. [F12] を押してメインフォームを表示し、ActionManager または ActionList コンポーネントをダブルクリックします。
2. [ファイル | 上書き保存] (FileSave) アクションをダブルクリックします。

コードエディタが開かれ、イベントハンドラ内にカーソルが位置付けられます。

ヒント

オブジェクトツリーで [ファイル | 上書き保存] をダブルクリックしても同じです。

3. カーソルが位置付けられたところ ({ と } の間) に次のように入力します。

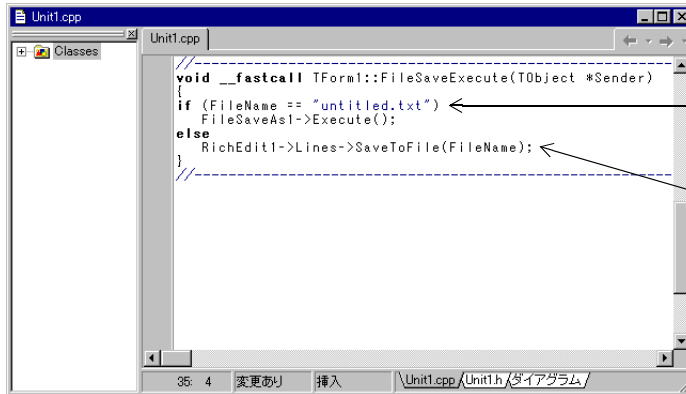
```

if (FileName == "untitled.txt")
    FileSaveAs1->Execute();
else
    RichEdit1->Lines->SaveToFile(FileName);

```

このコードは、ファイルに名前が付けられていない場合に、ユーザーがファイル名を指定できるように、テキストエディタで [名前を付けて保存] ダイアログボックスを表示することを定義しています。ファイルに名前が付けられている場合は、その現在の名前でファイルを保存します。[名前を付けて保存] ダイアログボックスは、[名前を付けて保存] コマンドのイベントハンドラ内で定義します。FileSaveAs1BeforeExecute は、上書き保存コマンド用に自動的に生成された名前です。

イベントハンドラは以下のようになるはずですが。



ファイルに名前が付けられていない場合は、[名前を付けて保存]ダイアログを表示します。

ファイルに名前が付けられている場合は、そのファイルに保存します。

これで [ファイル | 上書き保存] コマンドの処理が完成しました。

名前を付けて保存コマンドのイベントハンドラの作成

SaveDialog の Execute メソッドを呼び出すと、ファイルに名前を付けて保存する Windows の標準ダイアログボックスが呼び出されます。[名前を付けて保存] コマンドのイベントハンドラを作成するには、

1. [F12]を押してメインフォームを表示し、ActionManager または ActionList コンポーネントをダブルクリックします。
2. [ファイル | 名前を付けて保存] (FileSaveAs1) アクションを選択します。
3. オブジェクトインスペクタで [プロパティ] タブをクリックします。Dialog プロパティの [+] をクリックして、サブプロパティを開きます。Dialog は、名前を付けて保存ダイアログのコンポーネントを参照し、そのダイアログのプロパティを表示します。
4. DefaultExt に txt を設定します。
5. Filter の横のテキスト領域をダブルクリックしてフィルタエディタを表示します。フィルタエディタで、[開く] ダイアログボックスで使用するファイルの種類フィルタを指定します。
 - [フィルタ名] 列の下の最初の行に「テキストファイル (*.txt)」と入力します。[フィルタ] 列に「*.txt」と入力します。
 - [フィルタ名] 列の2番目の行に「すべてのファイル (*.*)」と入力し、[フィルタ] 列に「*. *」と入力します。
 - [OK] をクリックします。
6. Title を「名前を付けて保存」に設定します。
7. [イベント] タブをクリックします。BeforeExecute の横のテキスト領域をダブルクリックすると、コードエディタが開いて、FileSaveAs1BeforeExecute イベントハンドラ内にカーソルが位置付けられます。
8. コードエディタ内のカーソルが位置付けられたところに、次のように入力します。

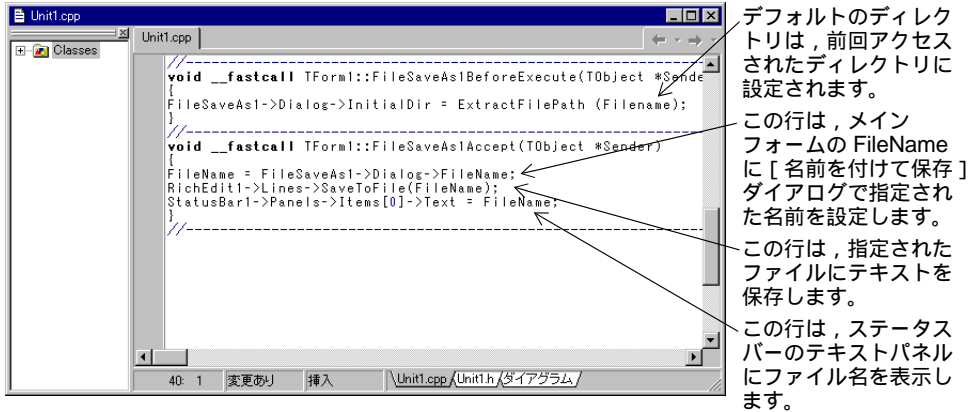
```
FileSaveAs1->Dialog->InitialDir = ExtractFilePath (FileName);
```

9. [イベント] タブがまだ表示されているはずですが。OnAccept イベントの横のテキスト領域をダブルクリックすると、コードエディタに FileSaveAs1Accept が表示されます。

- 10.カーソルが位置付けられたところに次のように入力します。

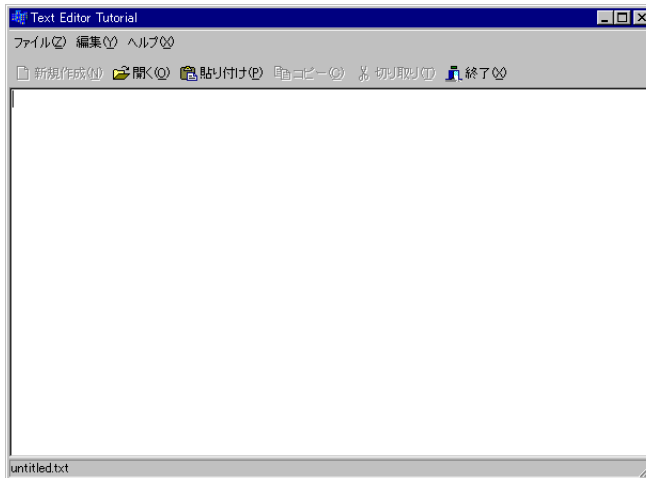
```
FileName = FileSaveAs1->Dialog->FileName;
RichEdit1->Lines->SaveToFile(FileName);
StatusBar1->Panels->Items[0]->Text = FileName;
```

FileSaveAs のイベントハンドラは以下のようになるはずですが。



- 11.[ファイル | すべて保存] を選択して、作業の内容を保存します。

- 12.アプリケーションがどのように動作するかを確認するために、[F9] を押して実行します。



実行中のアプリケーションは、設計モードのメインフォームとほぼ同じ外観を持ちます。非ビジュアルオブジェクトが表示されていないことに注意してください。

アプリケーションを閉じるには、次の 3 つの方法があります。

[×] をクリックする。

[ファイル | 終了] を選択する。

ツールバーの [アプリケーションの終了] ボタンをクリックする。

コードエディタの一番下にエラーメッセージが出た場合は、そこをダブルクリックするとエラーが発生したコードの場所にジャンプします。これまでの説明に従って操作を行っていることを確認してください。

- 13.設計モードに戻るには、アプリケーションの右上隅の [×] をクリックします。

ヘルプファイルの作成

作成したアプリケーションの使い方を説明するヘルプファイルを作成するのは良い考えです。

C++Builder には、Microsoft Help Workshop が付属しています。このプログラムは、Windows ヘルプファイルの設計やコンパイルに関する情報とともに C:\Project Files\Borland\CBuilder6\Help\Tools ディレクトリに格納されています。サンプルのテキストエディタアプリケーションでは、ユーザーは [ヘルプ | 目次] または [ヘルプ | キーワード] を選択することによって、ヘルプファイルにアクセスするときに目次またはキーワードを表示できます。

前の手順では、アクションマネージャまたはアクションリストエディタで、コンパイルしたヘルプファイルの [目次] または [キーワード] タブを表示するための HelpContents および HelpIndex アクションを作成しました。次に必要な作業は、定数値をヘルプのパラメータに割り当て、必要なものを表示するイベントハンドラを作成することです。

[ヘルプ] コマンドを使用するためには、Windows ヘルプファイルを作成してコンパイルする必要があります。このチュートリアルでは、ヘルプファイルの作成法については省略します。なお、サンプルの rtf ファイル (TextEditor.rtf)、ヘルプファイル (TextEditor.hlp)、および目次ファイル (TextEditor.cnt) は Borland の Web サイトからダウンロードできます。

1. Windows のエクスプローラで、C:\Project Files\Borland\CBuilder6\Help ディレクトリにある B6X1.zip を開きます。
2. .hlp ファイルと .cnt ファイルを TextEditor ディレクトリ(デフォルトでは C:\Project Files\Borland\CBuilder6\Projects\TextEditor) に展開します。

メモ プロジェクトで任意の .hlp または .cnt ファイル (たとえば C++Builder のヘルプファイルおよび対応する .cnt ファイル) を使用できます。その場合は、アプリケーションがファイルを見つけられるように、プロジェクトディレクトリにファイルをコピーして、ファイル名を TextEditor.hlp と TextEditor.cnt に変更する必要があります。

ヘルプ | 目次コマンドのイベントハンドラの作成

[ヘルプ | 目次] コマンドのイベントハンドラを作成するには、

1. ActionManager または ActionList コンポーネントをダブルクリックします。
2. [ヘルプ | 目次] (HelpContents1) アクションをダブルクリックします。

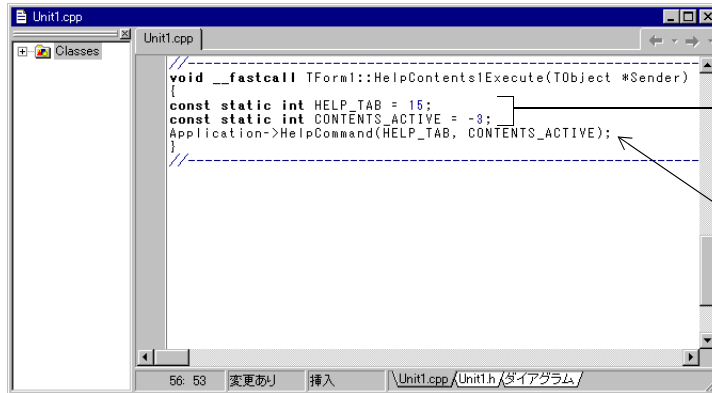
コードエディタが開かれ、イベントハンドラ内にカーソルが位置付けられます。

3. カーソルが位置付けられたところに次のように入力します。

```
const static int HELP_TAB = 15;  
const static int CONTENTS_ACTIVE = -3;  
Application->HelpCommand(HELP_TAB, CONTENTS_ACTIVE);
```

このコードは、定数値を HelpCommand パラメータに割り当てます。HELP_TAB に 15 を設定すると、ヘルプダイアログが表示されます。CONTENTS_ACTIVE に -3 を設定すると、[目次] タブが表示されます。

イベントハンドラは以下のようになるはずですが。



これらの行では、TApplication の HelpCommand メソッドのコマンドおよびデータパラメータを定義しています。
この行は、[目次] タブを表示した状態でヘルプダイアログを表示することを定義しています。

メモ HelpCommand イベントについてのヘルプを表示するには、エディタ内で HelpCommand の横にカーソルを置いて [F1] を押します。

これで [ヘルプ | 目次] コマンドの処理が完成しました。

ヘルプ | キーワードコマンドのイベントハンドラの作成

[ヘルプ | キーワード] コマンドのイベントハンドラを作成するには

1. 画面上にはアクションマネージャまたはアクションリストエディタがまだ表示されているはずですが。表示されていない場合は、フォーム上の ActionManager または ActionList コンポーネントをダブルクリックします。

2. [ヘルプ | キーワード] (HelpIndex1) アクションをダブルクリックします。

コードエディタが開かれ、イベントハンドラ内にカーソルが位置付けられます。

3. テキストエディタ内のカーソルが位置付けられたところに次のように入力します。

```

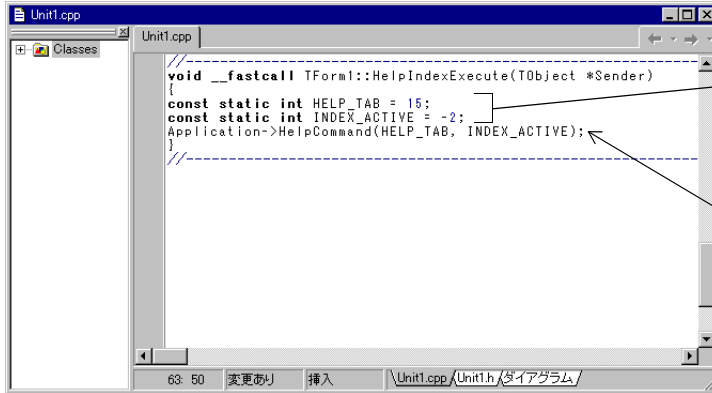
const static int HELP_TAB = 15;
const static int INDEX_ACTIVE = -2;
Application->HelpCommand(HELP_TAB, INDEX_ACTIVE);

```

このコードは、定数値を HelpCommand パラメータに割り当てます。HELP_TAB に 15 を設定すると、ヘルプダイアログボックスが表示されます。INDEX_ACTIVE に -2 を設定すると、[キーワード] タブが表示されます。

バージョン情報ダイアログボックスの作成

イベントハンドラは以下のようにするはずですが。



これらの行では、TApplication の HelpCommand メソッドのコマンドおよびデータパラメータを定義しています。
この行は、[キーワード] タブを表示した状態でヘルプダイアログを表示することを定義しています。

これで [ヘルプ | キーワード] コマンドの処理が完成しました。

バージョン情報ダイアログボックスの作成

多くのアプリケーションでは、名前やバージョン、ロゴといった製品に関する情報や、著作権などの法律に関する情報を表示する [バージョン情報] ダイアログボックスが装備されています。

[ヘルプ | バージョン情報] コマンドは、前の手順ですでにアクションマネージャまたはアクションリストエディタで定義してあります。

[バージョン情報] ダイアログボックスを追加するには、

1. [ファイル | 新規作成 | その他] を選択して [新規作成] ダイアログボックスを表示します。
2. [フォーム] タブをクリックし、[バージョン情報] アイコンをダブルクリックします。



[新規作成] ダイアログボックスは、オブジェクトリポジトリとも呼ばれます。
オブジェクトリポジトリに含まれている項目を基にして項目を作成する場合、項目をコピー、継承、または直接使用することができます。
[コピー] (デフォルト) を選択すると、項目のコピーがプロジェクト内に作成されます。
[継承] を選択すると、プロジェクト内に作成された項目は、リポジトリ内の項目に対して加えられた変更を継承します。
[直接使用] を選択すると、リポジトリ内のオブジェクトは、プロジェクト内に作成されたオブジェクトに対して加えられた変更を継承します。

[バージョン情報] ダイアログボックス用の定義済みのフォームが表示されます。

3. (グリッド部分をクリックして) フォームを選択し、オブジェクトインスペクタで [プロパティ] タブをクリックし、Caption プロパティを「Text Editor について」に変更します。
4. フォームをクリックして戻ります (フォームのキャプションは「Text Editor について」に変わっています)。フォーム上の各値を変更するために、それぞれをクリックして選択し、新しい値を入力します。
 - 「Product Name」を「Text Editor」に変更する。
 - 「Verion」を「Verion 1.0」に変更する。
 - 「Copyright」を「Copyright 2002」に変更する。



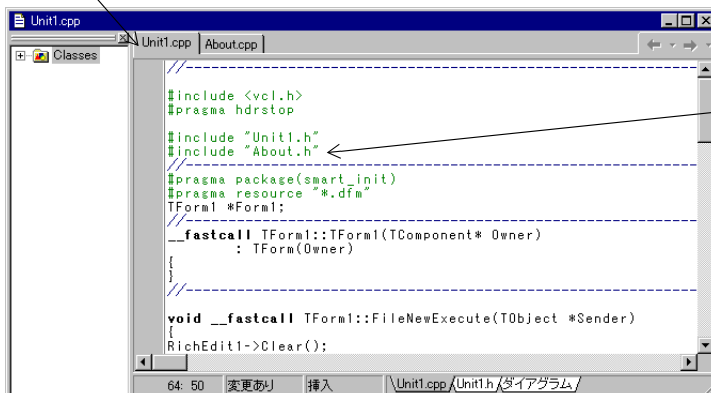
オブジェクトリポジトリには、アプリケーションに合わせて情報を変更できる標準のバージョン情報ボックスが含まれています。

5. [ファイル | 名前を付けて保存] を選択し、About.cpp という名前でフォームを保存します。

C++Builder のエディタには、Unit1.cpp, Unit1.h, About.cpp, および ActnRes (Enterprise または Professional 版でアクションマネージャを使用した場合のみ) の各ファイルが表示されます。ActnRes ユニットの必要ありませんが、そのままにしておかまいません。

6. [Unit1.cpp] タブをクリックして、コードエディタの先頭までスクロールします。Unit1 の include 文に About ユニットを追加します。これを行うには、[ファイル | ユニットヘッダーファイルの追加] を選択し、About を選択して [OK] をクリックします。.cpp ファイルの先頭部分に #include "About.h" という行が追加されます。

タブをクリックしてユニットに関連付けられているファイルを表示します。プロジェクトの作業中に別のファイルを開くと、そのファイルを示すタブがエディタに追加されます。



アプリケーションに新しいフォームを作成した場合は、それをメインフォームに追加する必要があります。[ファイル | ユニットヘッダーファイルの追加] を選択し、追加するヘッダーを選択します。

アプリケーションを完成させる

7. [F12] を押して、設計モードに戻ります。ActionManager または ActionList コンポーネントをダブルクリックして開きます。
8. [ヘルプ | バージョン情報] (HelpAbout) アクションをダブルクリックして、イベントハンドラを作成します。コードエディタ内のカーソルが位置付けられたところに、次のように入力します。

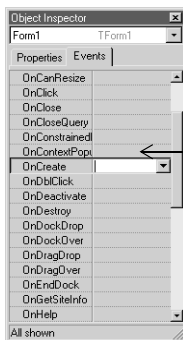
```
AboutBox->ShowModal();
```

このコードは、ユーザーが [ヘルプ | バージョン情報] をクリックしたときに [バージョン情報] ダイアログボックスを開くことを指定します。ShowModal は、フォームをモード付き状態で開きます。その場合、実行時には、フォームを閉じるまでユーザーは他のことはできません。
9. [ファイル | すべて保存] を選択します。

アプリケーションを完成させる

アプリケーションはほとんど完成した状態です。ただし、メインフォームに対して指定しなければならない項目がまだ残っています。アプリケーションを完成させるには、

1. [F12] を押してメインフォームに移ります。
2. フォームを選択します。フォーカスがコンポーネントではなくフォームにあることを確認します。そうでない場合は、オブジェクトインスペクタの一番上のドロップダウンリストで Form1: TForm1 を選択します。
3. [イベント] タブをクリックし、OnCreate の横の領域をダブルクリックします。



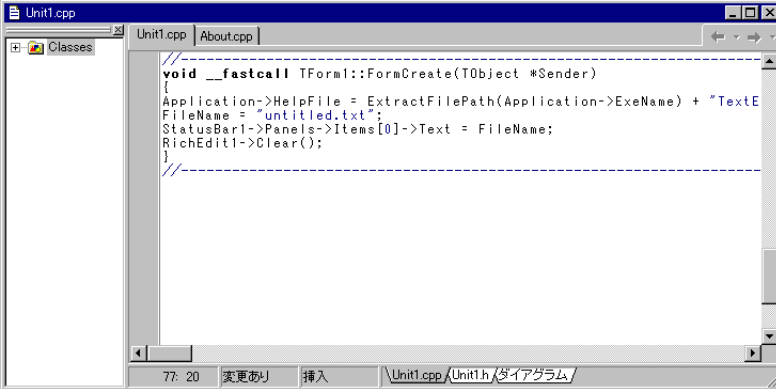
ここをチェックしてフォーカスがメインフォームにあることを確認します。フォーカスがメインフォームにない場合は、ドロップダウンリストから Form1 を選択します。

ここをダブルクリックしてフォームの OnCreate イベントのイベントハンドラを作成します。

4. コードエディタ内のカーソルが位置付けられたところに、次のように入力します。

```
Application->HelpFile = ExtractFilePath(Application->ExeName) + "TextEditor.hlp";  
FileName = "untitled.txt";  
StatusBar1->Panels->Items[0]->Text = FileName;  
RichEdit1->Clear();
```

このコードはアプリケーションを初期化します。具体的には、ヘルプファイルに関連付け、FileName の値を untitled.txt に設定し、ファイル名をステータスバーに表示し、テキスト編集領域をクリアします。



```
Unit1.cpp | Classes | Unit1.cpp | About.cpp |
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Application->HelpFile = ExtractFilePath(Application->ExeName) + "TextE
    FileName = "untitled.txt";
    StatusBar1->Panels->Items[0]->Text = FileName;
    RichEdit1->Clear();
}
//-----
77: 20 | 変更あり | 挿入 | \Unit1.cpp\Unit1.h | ダイアグラム |
```

5. [ファイル | すべて保存] を選択して、作業内容を保存します。
 6. [F9] を押してアプリケーションを実行します。
- おつかれさまでした。これでチュートリアルは終了です。

第5章

チュートリアル: CLX データベースアプリケーションの作成

このチュートリアルでは、サンプルの従業員データベースの表示と更新を行うクロスプラットフォームアプリケーションの作成を順を追って説明していきます。クロスプラットフォームアプリケーションでは、CLX（クロスプラットフォーム用コンポーネントライブラリ）を使用します。CLXは、異なるプラットフォームでコンパイルして実行できるように設計されているので、CLXアプリケーションはWindows/Linux間で移植するために最小限の変更しか必要としません（クロスプラットフォームコンパイラのサポートに関するポーランドの最新の製品情報をご覧ください）。

メモ このチュートリアルは、データベースコンポーネントが含まれる製品バージョン（版）用に記述されています。ここでは、Personal版では使用できない機能を必要とするデータベースアクセスを使用します。また、このチュートリアルをすべて実行するには、InterBaseがインストールされている必要があります。

データベースアーキテクチャの概要

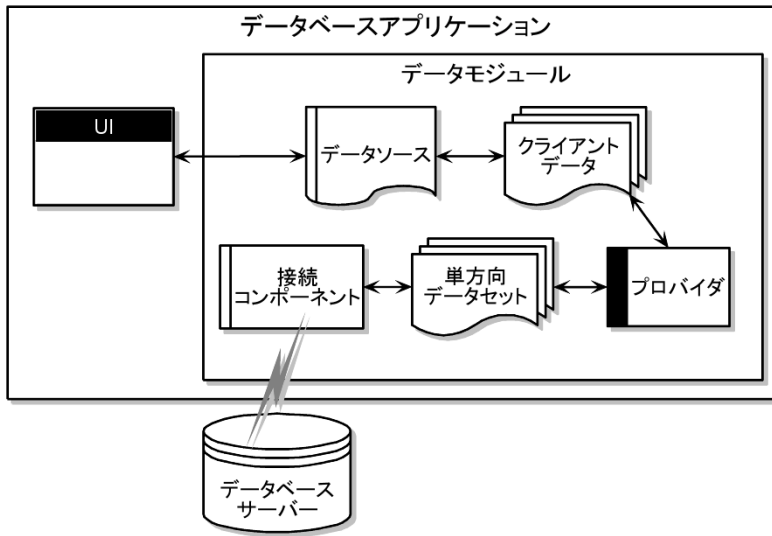
データベースアプリケーションのアーキテクチャは、複数のコンポーネントを使うことによって、実際のデータベースアプリケーションの開発と保守が容易になることが分かってきます。

データベースアプリケーションを構成する主要な部分は、ユーザーインターフェース、データアクセスコンポーネントのセット、そしてデータベース自体の3つです。このチュートリアルでは、dbExpressデータベースアプリケーションを作成します。他のデータベースアプリケーションも似たようなアーキテクチャを持っています。

ユーザーインターフェースには、ユーザーがデータベースのデータを編集したり登録するグリッドなどのデータベース対応コントロールが含まれます。データアクセスコンポーネントには、データソース、クライアントデータセット、データプロバイダ、単方向データセット、接続コンポーネントがあります。データソースは、ユーザーインターフェースとクライアントデータセットを繋ぐパイプとして機能します。クライアントデータセットはアプリケーションの心臓部であり、基になるデータベ

新しい CLX アプリケーションを作成する

スから取り出されるレコードのセットを保持し、メモリ上でバッファリングします。プロバイダは、クライアントデータセットと単方向データセットの間でデータを転送します。単方向データセットは、データベースからデータを直接取り出します。接続コンポーネントは、データベースへの接続を確立します。単方向データセットは、その種類ごとに使用する接続コンポーネントの種類が異なります。



データベース開発の詳細については、『開発者ガイド』またはオンラインヘルプで「データベースアプリケーションの設計」を参照してください。

新しい CLX アプリケーションを作成する

チュートリアルを開始する前に、ソースファイルを格納するフォルダを作成します。そして、新規プロジェクトを作成して保存します。

1. Tutorial というフォルダを作成します。ここに、このチュートリアルで作成するプロジェクトファイルを格納します。
2. 新規の CLX プロジェクトを開始します。[ファイル | 新規作成 | CLX アプリケーション] を選択して、新規のクロスプラットフォームプロジェクトを作成します。
3. [ファイル | すべて保存] を選択して、ファイルをディスクに保存します。保存ダイアログが表示されたら、Tutorial フォルダに移動して、各ファイルをデフォルトの名前で保存します。

以降は、どの時点でも [ファイル | すべて保存] を選択して作業を保存することができます。チュートリアルを途中で中断した場合も、[ファイル | 開き直す] を選択してリストから Tutorial を選べば、保存しておいたバージョンを開くことができます。

データアクセスコンポーネントのセットアップ

データアクセスコンポーネントは、データ（データセット）と、そのデータセットをアプリケーションの他の部分と結び付けるコンポーネントの両方を表します。各データアクセスコンポーネントは、それぞれがその下位のコンポーネントを指し示します。たとえば、データソースはクライアントデータセットを指し、クライアントデータセットはプロバイダを指し、プロバイダは・・・、というようになります。したがって、データアクセスコンポーネントをセットアップするときには、適切な順序でコンポーネントを追加していくことになります。

以降のセクションでは、データベースコンポーネントを追加して、データベース接続、単方向データセット、プロバイダ、クライアントデータセット、およびデータソースを作成していきます。そのあと、アプリケーションのユーザーインターフェースを作成します。これらのコンポーネントは、コンポーネントパレットの [dbExpress], [Data Access], および [Data Control] ページにあります。

ヒント ユーザーインターフェースはフォームに配置し、データアクセスコンポーネントはそれとは別にデータモジュールで管理するのが望ましいのですが、このチュートリアルでは、簡単にするために、ユーザーインターフェースと他のすべてのコンポーネントを1つのフォームに配置することにします。

データベース接続のセットアップ

[dbExpress] ページには、SQL データベースサーバーへの高速アクセスを提供するコンポーネントがあります。

接続コンポーネントを追加して、データベースに接続できるようにする必要があります。使用する接続コンポーネントの種類は、使用するデータセットコンポーネントの種類によって異なります。このチュートリアルでは、TSQLConnection コンポーネントと TSQLDataSet コンポーネントを使うことにします。

dbExpress 接続コンポーネントを追加するには、



1. コンポーネントパレットの [dbExpress] ページをクリックし、TSQLConnection をダブルクリックしてフォームに配置します。TSQLConnection コンポーネントを見つけるには、パレット上のアイコンを少しの間マウスでポイントし、ヒントとして表示されるコンポーネントの名前を確認します。配置したコンポーネントは、デフォルトで TSQLConnection1 という名前になります

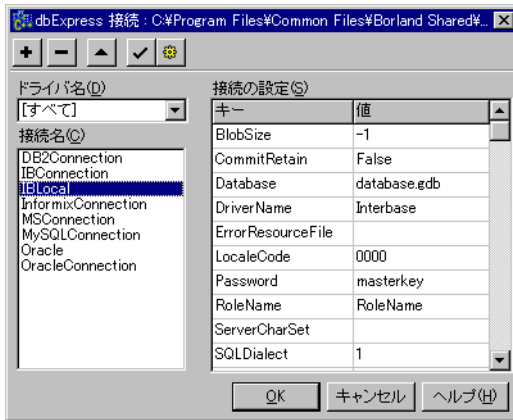
TSQLConnection コンポーネントは非ビジュアルコンポーネントなので、フォーム上のどこに配置してもかまいません。ただし、このチュートリアルでは、非ビジュアルコンポーネントはフォームの上の方に並べています。

ヒント フォームに配置したコンポーネントのキャプションを表示するには、[ツール | 環境オプション | デザイン] を選択して [コンポーネントのキャプションを表示] をクリックします。

2. オブジェクトインスペクタで、TSQLConnection1 の ConnectionName プロパティを IBLocal に設定します（ドロップダウンリストから選択できます）。
3. LoginPrompt プロパティを **false** に設定します（このプロパティを **false** に設定すると、データベースにアクセスするときにログインプロンプトが表示されなくなります）。

データアクセスコンポーネントのセットアップ

4. フォーム上の TSQLConnection コンポーネントをダブルクリックして、接続エディタ ([dbExpress 接続] ダイアログ) を表示します。



接続エディタを使って、TSQLConnection コンポーネントの接続設定の選択や、dbxconnections ファイルに格納されている接続設定の編集を行います。ダイアログで行った変更は、[OK] をクリックしたときにこのファイルに書き込まれます。また、[OK] をクリックすると、選択された接続が SQLConnection コンポーネントの ConnectionName プロパティの値として割り当てられます。

5. 接続エディタで、システム上のデータベースファイル employee.gdb のパス名を指定します。このチュートリアルでは、C++Builder に付属している InterBase のサンプルデータベースに接続します。デフォルトでは、InterBase のインストール時に employee.gdb は C:\Program Files\Common Files\Borland Shared\Data に配置されています。
6. [User_Name] および [Password] フィールドが適切な値かどうか確認します。デフォルト値を変更していない場合は、これらのフィールドを変更する必要はありません。別の人データベースアクセスを管理している場合は、データベースアクセス用のユーザー名とパスワードを取得する必要があるかもしれません。
7. 2つのフィールドのチェックと編集が終わったら、[OK] をクリックして変更を保存し、接続エディタを閉じます。

これらの変更は dbxconnections.ini ファイルに書き込まれ、選択された接続が SQL 接続コンポーネントの ConnectionName プロパティの値として割り当てられます。

ヒント 接続エディタの使い方で分からないことがある場合は、[ヘルプ] ボタンをクリックしてください。

8. [ファイル | すべて保存] を選択してプロジェクトを保存します。

データセットのセットアップ

基本的なデータベースアプリケーションでは、データセットを使ってデータベース内の情報にアクセスします。dbExpress アプリケーションでは、単方向データセットを使用します。単方向データセットは、データベースから情報を読み込みますが、データの更新は行いません。

単方向データセットを追加するには、



1. コンポーネントパレットの [dbExpress] タブから、TSQLDataSet をフォームの上の方にドロップします。
2. オブジェクトインスペクタで、SQLConnection プロパティが SQLConnection1 (すでに作成してあるデータベース接続) を指すように設定します。
3. CommandText に「Select * from SALES」を設定して、データセットが実行するコマンドを指定します。オブジェクトインスペクタで Select 文を直接入力することができます。また、CommandText の右の省略記号ボタンをクリックして CommandText エディタ ([CommandText の編集] ダイアログ) を表示し、独自の問い合わせ文を作成することもできます。

ヒント

CommandText エディタの使い方で分からないことがある場合は、[ヘルプ] ボタンをクリックしてください。

4. Active を true に設定して、データセットを開きます。
5. [ファイル | すべて保存] を選択してプロジェクトを保存します。

プロバイダ、クライアントデータセット、データソースのセットアップ

[Data Access] ページには、dbExpress 以外にも、さまざまなデータアクセスメカニズムに対して使用できるコンポーネントがあります。

プロバイダコンポーネントは、クライアントデータセットが他のデータセットからデータを取得する手段となります。プロバイダは、クライアントデータセットからデータ要求を受け取ると、データを取り出し、バッファージ化して、クライアントデータセットに返します。dbExpress を使う場合、プロバイダは、クライアントデータセットから更新を受け取り、その更新をデータベースサーバーに適用します。

プロバイダを追加するには、



1. コンポーネントパレットの [Data Access] ページから、TDataSetProvider コンポーネントをフォームの上の方にドロップします。
2. オブジェクトインスペクタで、プロバイダの DataSet プロパティを SQLDataSet1 に設定します。

クライアントデータセットは、メモリ上でデータをバッファリングします。また、データベースに送られる更新をキャッシュします。クライアントデータセットを使って、ユーザーインターフェース上のデータソースコンポーネントを使用するデータベース対応コントロールにデータを供給することができます。

クライアントデータセットを追加するには、



1. コンポーネントパレットの [Data Access] ページから TClientDataSet コンポーネントを、TDataSetProvider コンポーネントの右側にドロップします。
2. ProviderName プロパティを DataSetProvider1 に設定します。
3. Active プロパティを true に設定して、データがアプリケーションに渡されるようにします。

データソースは、クライアントデータセットをデータベース対応コントロールに接続します。各データベース対応コントロールは、それぞれデータソースコンポーネントに関連付けないと、データの表示や操作が行えません。同様に、データセットもすべてデータソースコンポーネントに関連付けないと、フォーム上のデータベース対応コントロールでデータの表示や操作が行えません。

データソースを追加するには、



1. コンポーネントパレットの [Data Access] ページから TDataSource コンポーネントを、TClientDataSet コンポーネントの右側にドロップします。
2. データソースの DataSet プロパティを ClientDataSet1 に設定します。
3. [ファイル | すべて保存] を選択してプロジェクトを保存します。

以上で、非ビジュアルのデータベース基盤をアプリケーションに追加できました。次は、ユーザーインターフェースを設計します。

ユーザーインターフェースの設計

こんどは、アプリケーションにビジュアルコントロールを追加して、アプリケーションのユーザーがデータの表示や編集、保存ができるようにする必要があります。[Data Controls] ページには、データベース内のデータを扱うユーザーインターフェースを作成するデータベース対応コントロールが揃っています。これから、グリッドにデータベースを表示して、ナビゲータバーにコマンドをいくつか追加します。

グリッドとナビゲーションバーの作成

アプリケーションのインターフェースを作成するには、



1. まず、フォームにグリッドを追加することから始めます。[Data Controls] ページから、TDBGrid コンポーネントをフォームにドロップします。
2. DBGrid のプロパティを、グリッドに合わせるように設定します。オブジェクトインスペクタで、Anchors の横の [+] をクリックして akLeft, akTop, akRight, akBottom を表示し、それを全部 true に設定します。これを簡単に行うには、オブジェクトインスペクタで各プロパティの右の false をダブルクリックします。
3. Align プロパティを alBottom に設定して、グリッドをフォームの一番下に合わせて配置します。グリッドのサイズは、ドラッグするか、あるいは Height プロパティを（たとえば 400 に）設定して大きくすることもできます。
4. グリッドの DataSource プロパティを DataSource1 に設定します。設定すると、グリッドに employee データベースのデータが入ります。グリッドにデータが表示されない場合は、フォーム上のオブジェクトのプロパティが、ここまでの手順に示した通りに正しく設定されているか確認してください。

これでアプリケーションは次のようになるはずです。

DBGrid コントロールは、IDE で作業している設計時にデータを表示します。これによって、データベースに正しく接続していることを検証できます。ただし、設計時にデータを編集することはできません。データを編集するには、アプリケーションを実行する必要があります。



1. [Data Controls] ページから、TDBNavigator コントロールをフォームにドロップします。データベースナビゲータは、(たとえば「次」や「前」の矢印を使って) データセット内のデータ間の移動や、データの操作を行うためのツールです。
2. ナビゲータバーの DataSource プロパティを DataSource1 に設定して、ナビゲータからクライアントデータセットのデータが見えるようにします。
3. ナビゲータバーの ShowHint プロパティを true に設定します (ShowHint を true に設定すると、実行時にナビゲータバーの各項目にマウスカーソルを置いたときに、ヘルプヒントが表示されるようになります)。
4. [ファイル | すべて保存] を選択してプロジェクトを保存します。



5. [F9] を押して、プロジェクトをコンパイルし実行します。プロジェクトの実行は、デバッグツールバーの [実行] ボタンを押すか、[実行]メニューから [実行] を選択して行うこともできます。

PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE	SHIP_DATE
V91 E0210	1004	11	shipped	1991/03/04	1991/03/05
V92E0340	1004	11	shipped	1992/10/15	1992/10/16
V92J1003	1010	61	shipped	1992/07/26	1992/08/04
V93 J2004	1010	118	shipped	1993/10/30	1993/12/02
V93 J3100	1010	118	shipped	1993/08/20	1993/08/20
V92F3004	1012	11	shipped	1992/10/15	1993/01/16
V93F3088	1012	134	shipped	1993/08/27	1993/09/08
V93F2030	1012	134	open	1993/12/12	
V93F2051	1012	134	waiting	1993/12/18	
V93H0030	1005	118	open	1993/12/12	
V94H0079	1005	61	open	1994/02/13	
V9324200	1001	72	shipped	1993/08/09	1993/08/09
V9324320	1001	127	shipped	1993/08/16	1993/08/16
V9320630	1001	127	open	1993/12/12	
V9420099	1001	127	open	1994/01/17	
V9427029	1001	127	shipped	1994/02/07	1994/02/10
V9333005	1002	11	shipped	1993/02/03	1993/03/03
V9333006	1002	11	shipped	1993/04/27	1993/05/02
V9336100	1002	11	waiting	1993/12/27	1994/01/01

プロジェクトを実行すると、フォーム上で設計したときと似たウィンドウにプログラムが開きます。employee データベースに対してナビゲーションバーをテストすることができます。たとえば、矢印コマンドでレコード間を移動したり、[+] コマンドでレコードを追加したり、[-] コマンドでレコードを削除したりできます。

ヒント 初期バージョンのアプリケーションのテスト中にエラーが起きた場合は、[実行 | プログラムの終了] を選択して設計時の表示に戻ってください。

メニューのサポートの追加

プログラムはすでにかなりの機能を持っていますが、通常の GUI アプリケーションにあるような機能の多くが備わっていません。たとえば、ほとんどのアプリケーションには使いやすくするためにメニューやボタンが実装されています。

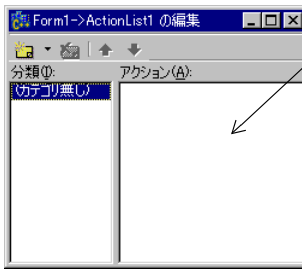
このセクションでは、アクションリストを追加します。アクションリストを使わなくてもメニューやツールバー、ボタンを作成することはできますが、アクションリストを使うと、ユーザーコマンドへの応答を一元管理できるため、開発と保守を簡略化することができます (Windows 専用の開発では、アクションバンドを使用して、ツールバーとメニューの開発を簡略化することもできます)。

1. アプリケーションがまだ実行中の場合は、右上隅の [×] をクリックしてアプリケーションを閉じ、設計時表示のフォームに戻ります。
2. コンポーネントパレットの [Common Controls] ページから、ImageList をフォームにドロップします。これを、他の非ビジュアルコンポーネントの横に並べます。ImageList には、[切り取り] や [貼り付け] などの標準アクションを表すアイコンが入っています。





- コンポーネントパレットの [Standard] ページから、ActionList コンポーネントをフォームにドロップします。アクションリストの Images プロパティに ImageList1 を設定します。
- ActionList1 オブジェクトをダブルクリックして、アクションリストエディタを表示します。

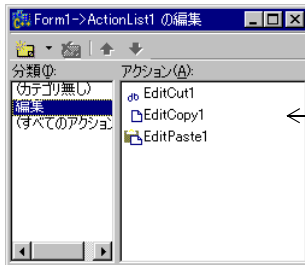


アクションリストエディタで右クリックし、[標準アクションの新規追加] を選択して [標準アクション] リストボックスを表示します。

使いたいアクションを選択して [OK] をクリックします。複数のアクションを選択するには、[Ctrl] を押しながら選択します。



- アクションリストエディタを右クリックして、[標準アクションの新規追加] を選択します。[標準アクション] リストボックスが表示されます。
- TEditCopy, TEditCut, および TEditPaste の 3 つのアクションを選択します (複数のアクションを選択するには、[Ctrl] を押しながら選択します)。そして、[OK] をクリックします。



これで、製品に含まれている標準のアクションが追加されました。

これらのアクションはメニュー上で使用できます。

- アクションリストエディタを右クリックして [アクションの新規追加] を選択し、(デフォルトでは提供されない) アクションをもう 1 つ追加します。デフォルトでは Action1 が追加されます。オブジェクトインスペクタで、その Caption プロパティに「今すぐ更新」を設定します。

この同じアクションは、メニューとボタンで使用します。あとで、イベントハンドラを追加して、データベースを更新できるようにします。

- [(カテゴリ無し)] をクリックし、右クリックして [アクションの新規追加] を選択し、さらにアクションを 1 つ追加します。Action2 が追加されます。その Caption プロパティに「終了 (&X)」を設定します。
- 右上隅の [×] をクリックして、アクションリストエディタを閉じます。

これで、標準アクション 3 つと、後でイベントハンドラに接続する他のアクションを 2 つ追加しました。

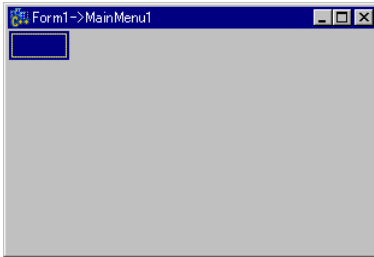
- [ファイル | すべて保存] を選択してプロジェクトを保存します。

メニューの追加

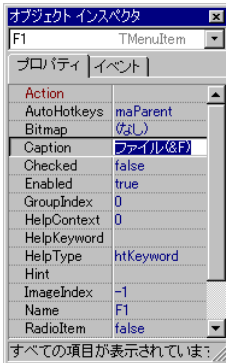
このセクションでは、3つのドロップダウンメニュー（[ファイル]、[編集]、[ヘルプ]）を持つメインメニューバーを追加し、アクションリスト内の標準アクションを使って各メニューにメニュー項目を追加します。



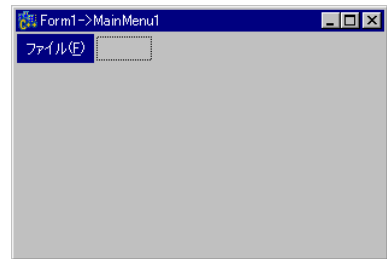
1. コンポーネントパレットの [Standard] ページから、TMainMenu コンポーネントをフォームに配置します。これをドラッグして、他の非ビジュアルコンポーネントの横に並べます。
2. メインメニューの Images プロパティに ImageList1 を設定して、イメージリストをメニュー項目に関連付けます。
3. メニューコンポーネントをダブルクリックしてメニューデザイナーを表示します。



4. 最初の最上位メニュー項目の Caption プロパティに「ファイル (&F)」と入力して、[Enter] を押します。

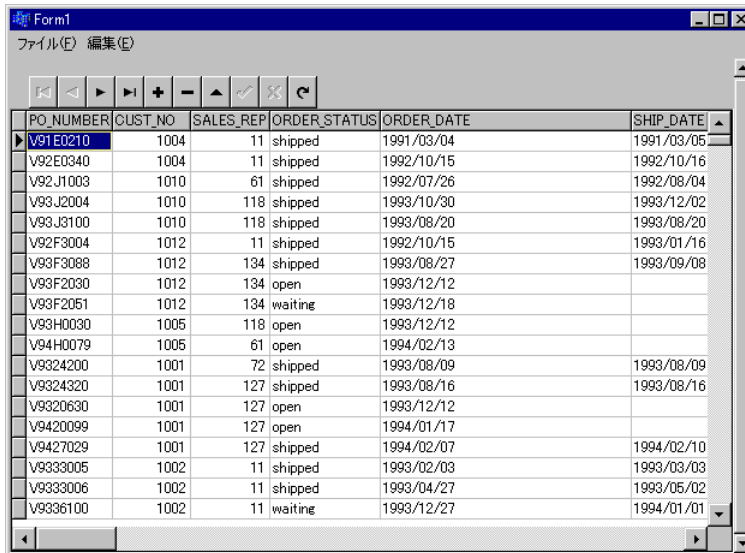


「ファイル (&F)」と入力して [Enter] を押すと、最上位レベルの [ファイル (F)] コマンドが表示され、最初のメニュー項目を追加できます。
英字の前のアンド記号によって、アクセラレータキーが有効になります。



5. [ファイル (F)] メニューの下の空白のメニュー項目を選択します。そのメニュー項目の Action プロパティを Action2 に設定します。[ファイル (F)] メニューの下に [終了 (X)] メニュー項目が表示されます。
6. 2番目の最上位メニュー項目 ([ファイル (F)] の右側) をクリックします。その Caption プロパティに「編集 (&E)」を設定して [Enter] を押します。[編集 (E)] メニューの下の空白のメニュー項目を選択します。
7. オブジェクトインスペクタで Action プロパティに EditCut1 を設定し、[Enter] を押します。項目のキャプションが自動的に [切り取り (T)] に設定され、メニューに切り取りコマンドのデフォルトのビットマップが表示されます。

8. [切り取り (T)] の下の空白のメニュー項目を選択し、Action プロパティに EditCopy1 を設定します (コピーコマンドのデフォルトのビットマップがメニューに表示されます)。
9. [コピー (C)] の下の空白のメニュー項目を選択し、Action プロパティに EditPaste1 を設定します (貼り付けコマンドのデフォルトのビットマップがメニューに表示されます)。
10. [貼り付け (P)] の下の空白のメニュー項目を選択し、Caption プロパティに 「-」 (ハイフン) を設定して、メニューに分割線を追加します。[Enter] を押します。
11. 分割線の下側の空白のメニュー項目を選択し、Action プロパティに Action1 を設定します。メニュー項目として [今すぐ更新] が表示されます。
12. [×] をクリックしてメニューデザイナーを閉じます。
13. [ファイル | すべて保存] を選択してプロジェクトを保存します。
14. [F9] を押すか、ツールバーの [実行] ボタン (緑の右向き矢印) をクリックして、プログラムを実行し、どのように動作するか確かめます。



PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE	SHIP_DATE
V91E0210	1004	11	shipped	1991/03/04	1991/03/05
V92E0340	1004	11	shipped	1992/10/15	1992/10/16
V92J1003	1010	61	shipped	1992/07/26	1992/08/04
V93J2004	1010	118	shipped	1993/10/30	1993/12/02
V93J3100	1010	118	shipped	1993/08/20	1993/08/20
V92F3004	1012	11	shipped	1992/10/15	1993/01/16
V93F3088	1012	134	shipped	1993/08/27	1993/09/08
V93F2030	1012	134	open	1993/12/12	
V93F2051	1012	134	waiting	1993/12/18	
V93H0030	1005	118	open	1993/12/12	
V94H0079	1005	61	open	1994/02/13	
V9324200	1001	72	shipped	1993/08/09	1993/08/09
V9324320	1001	127	shipped	1993/08/16	1993/08/16
V9320630	1001	127	open	1993/12/12	
V9420099	1001	127	open	1994/01/17	
V9427029	1001	127	shipped	1994/02/07	1994/02/10
V9333005	1002	11	shipped	1993/02/03	1993/03/03
V9333006	1002	11	shipped	1993/04/27	1993/05/02
V9336100	1002	11	waiting	1993/12/27	1994/01/01

編集メニューとナビゲーションバーのたくさんのコマンドは、この時点で操作できます。[編集] メニューの [コピー] と [切り取り] は、データベース内のテキストを選択するまでは淡色表示になっています。ナビゲーションバーを使うと、データベース内のレコード間の移動や、レコードの挿入、レコードの削除が行えます。[今すぐ更新] コマンドはまだ機能しません。

次に進む用意ができたならアプリケーションを閉じます。

ボタンを追加する

このセクションでは、アプリケーションに [今すぐ更新] ボタンを追加する手順を説明します。このボタンは、ユーザーが行った変更 (レコードの編集、新規レコードの追加、レコードの削除など) をデータベースに適用します。

タイトルとイメージの表示

ボタンを追加するには、



1. コンポーネントパレットの [Standard] ページから、TButton コンポーネントをフォームに配置します (パレットでコンポーネントを選択し、フォーム上のナビゲーションバーの横をクリックします)。
2. そのボタンの Action プロパティを Action1 に設定します。

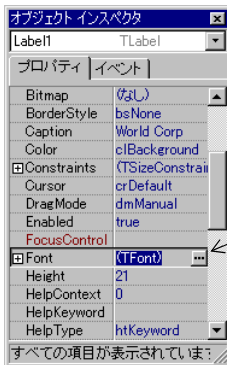
ボタンのキャプションが「今すぐ更新」に変わります。アプリケーションを実行したときは、イベントハンドラを追加してボタンが機能するようになるまでは、ボタンは淡色表示になっています。

タイトルとイメージの表示

会社の名前とイメージを追加すれば、アプリケーションを実際のものらしくすることができます。



1. コンポーネントパレットの [Standard] ページから、TLabel コンポーネントをフォームにドロップします (デフォルトで Label1 という名前が付けられます)。
2. オブジェクトインスペクタで、ラベルの Caption プロパティを会社名 (たとえば「World Corp」) に変更します。
3. Font プロパティをクリックして、会社名のフォントを変更します。右側の省略記号をクリックし、[フォント] ダイアログボックスでフォントを MS P ゴシック、太字、16 ポイントに変更します。[OK] をクリックします。



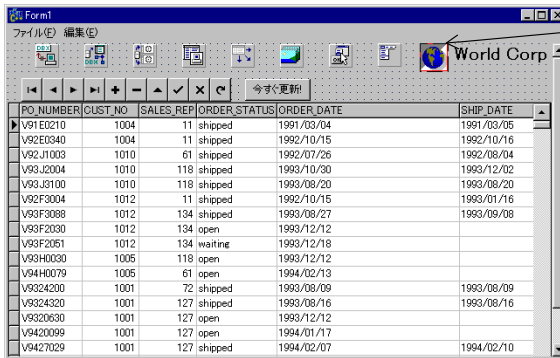
オブジェクトインスペクタで Font プロパティを使うと、ラベルのフォントを変更できます。省略記号ボタンをクリックして標準のフォント設定ダイアログを表示します。



4. このラベルをフォームの右上の位置に配置します。
5. コンポーネントパレットの [Additional] ページから、TImage コンポーネントをラベルの横にドロップします (デフォルトで Image1 という名前が付けられます)。
6. Image1 にイメージを追加するために、Picture プロパティをクリックします。省略記号ボタンをクリックして画像エディタ ([画像の設定] ダイアログ) を表示します。
7. 画像エディタで、[読み込み] を選択して earth.ico を読み込みます (このファイルは、C++Builder では Program Files ¥ Common Files ¥ Borland Shared ¥ Images ¥ lcons ¥ earth.ico です)。



8. earth.ico をダブルクリックします。[OK] をクリックして画像を読み込み、画像エディタを閉じます。
9. デフォルトのイメージ領域のサイズを、画像に合わせて変更します。イメージをラベルの横に配置します。



イメージの端をドラッグして Image の幅を設定できますが、オブジェクトインスペクタで Width プロパティを設定することによって幅を変更できます。

10. テキストとイメージの位置を揃えるには、両方のオブジェクトを選択し、右クリックして [位置合わせ] を選択します。[位置合わせ] ダイアログボックスで、[垂直位置合わせ] の下の [下に寄せる] を選択します。
 11. [ファイル | すべて保存] を選択してプロジェクトを保存します。
 12. [F9] を押して、アプリケーションをコンパイルして実行します。
- 次に進む用意ができたならアプリケーションを閉じます。

イベントハンドラの作成

コンポーネントパレット上のほとんどのコンポーネントにはイベントがあり、コンポーネントのほとんどにはデフォルトイベントがあります。共通のデフォルトイベントは OnClick で、これはコンポーネント（たとえば TButton）がクリックされたときに呼び出されます。フォーム上のオブジェクトを選択してオブジェクトインスペクタの [イベント] タブをクリックすると、コンポーネントのイベントのリストが表示されます。

イベントおよびイベントハンドラの詳細については、『開発者ガイド』の第 5 章またはオンラインヘルプの目次から「アプリケーションユーザーインターフェースの開発」を参照してください。

[今すぐ更新] コマンドのイベントハンドラの作成

まずはじめに、[今すぐ更新] のコマンドとボタンのイベントハンドラを作成します。

1. ActionList コンポーネントをダブルクリックして、アクションリストエディタを表示します。
2. [(カテゴリ無し)] を選択して、Action1 と Action2 を見つけます。

イベントハンドラの作成

3. Action1 をダブルクリックします。コードエディタに次のスケルトンイベントハンドラが表示されます。

```
void __fastcall TForm1::Action1Execute(TObject *Sender)
{
}

```

カーソルが位置付けられた場所 ({ と } の間) に次のように入力します。

```
if(ClientDataSet1->State == dsEdit || ClientDataSet1->State == dsInsert)
    ClientDataSet1->Post();
    ClientDataSet1->ApplyUpdates(-1);

```

このイベントハンドラは、最初にデータベースがどの状態にあるかをチェックします。変更したレコードから移動すると、その変更は自動的にポストされます。しかし、変更したレコードから移動していない間は、データベースは編集モードか挿入モードのままになっています。この if 文では、変更された可能性があって、まだクライアントデータセットに渡されていないデータをポストしています。2 つ目の文では、クライアントデータセットで保持されていた更新をデータベースに適用しています。

メモ dbExpress を使っている場合、データに対する変更は自動的にデータベースにポストされません。クライアントデータセットで更新、挿入、削除したすべてのレコードは、ApplyUpdates メソッドを呼び出してデータベースに書き込む必要があります。

終了コマンドのイベントハンドラの作成

次は、[終了] コマンドのイベントハンドラを作成します。

1. (表示されていない場合は) ActionList コンポーネントをダブルクリックして、アクションリストエディタを表示します。
2. [(カテゴリ無し)] を選択して Action2 を見つけます。
3. Action2 をダブルクリックします。コードエディタに次のスケルトンイベントハンドラが表示されます。

```
void __fastcall TForm1::Action2Execute(TObject *Sender)
{
}

```

カーソルが位置付けられた場所 ({ と } の間) に次のように入力します。

```
Close();

```

このイベントハンドラは、メニューの [ファイル | 終了] コマンドが使用されたときに、アプリケーションを閉じます。

4. アクションリストエディタを閉じます。
5. [ファイル | すべて保存] を選択してプロジェクトを保存します。

FormClose イベントハンドラの作成

最後に、アプリケーションが閉じられるときに呼び出されるイベントハンドラを作成します。アプリケーションは、[ファイル | 終了] を選択するか、右上隅の [×] をクリックすることによって終了します。どちらの場合も、プログラムは、データベースに対する保留中の更新がないかどうかを確認し、変更が保留状態の場合は、ユーザーに処理を尋ねるメッセージウィンドウを表示します。

このコードを終了コマンドのイベントハンドラに配置することもできますが、ユーザーが [×] をクリックしてアプリケーションを終了した場合には、保留状態のデータベースの変更は失われてしまいます。

1. メインフォームをクリックして選択します (フォーム上の、他のオブジェクトが配置されていない部分をクリックします)。
2. オブジェクトインスペクタで [イベント] タブを選択して、フォームのイベントを表示します。
3. OnClose をダブルクリックします (または、OnClose イベントの右側に FormClose と入力して、そこをクリックします)。スケルトンの FormClose イベントハンドラが作成され、コードエディタに他のイベントハンドラの後に表示されます。

```
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
}

```

カーソルが位置付けられた場所 ({ と } の間) に次のように入力します。

```
TMessageButton Option;
TMessageButtons msgButtons;

msgButtons << smbYes << smbNo << smbCancel;

Action = caFree;
if(ClientDataSet1->State == dsEdit || ClientDataSet1->State == dsInsert)
    ClientDataSet1->Post();
if(ClientDataSet1->ChangeCount > 0) {
    Option = Application->MessageBox("You have pending updates. Do you want to write
them to the database?", "Pending Updates", msgButtons, smsWarning, smbYes, smbNo);
    if(Option == smbYes)
        ClientDataSet1->ApplyUpdates(-1);
    else
        if(Option == smbCancel)
            Action = caNone;
}

```

このイベントハンドラは、データベースの状態をチェックします。変更が保留中の場合、変更カウントが増加しているクライアントデータセットにポストされます。そして、アプリケーションを閉じる前に、メッセージボックスを表示し、変更をどのように処理するかを問い合わせます。その応答オプションは、[はい], [いいえ], [キャンセル] の3つです。[はい] と答えると、更新をデータベースに適用します。[いいえ] と答えると、データベースを変更せずにアプリケーションを閉じます。[キャンセル] を選択すると、アプリケーションの終了はキャンセルされますが、データベースへの変更はキャンセルされず、アプリケーションは実行中のままになります。

4. 手続き全体が次のようになっているか確認してください。

イベントハンドラの作成

```
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    TMessageButton Option;
    TMessageButtons msgButtons;

    msgButtons << smbYes << smbNo << smbCancel;

    Action = caFree;
    if(ClientDataSet1->State == dsEdit || ClientDataSet1->State == dsInsert)
        ClientDataSet1->Post();
    if(ClientDataSet1->ChangeCount > 0) {
        Option = Application->MessageBox("You have pending updates. Do you want to write
them to the database?", "Pending Updates", msgButtons, smsWarning, smbYes, smbNo);
        if(Option == smbYes)
            ClientDataSet1->ApplyUpdates(-1);
        else
            if(Option == smbCancel)
                Action = caNone;
    }
}
```

- 完了させるために、[ファイル | すべて保存] を選択してプロジェクトを保存します。[F9] を押してアプリケーションを実行します。

ヒント エラーが出たら、そのエラーメッセージをダブルクリックして問題のコードに移動して修正するか、メッセージで [F1] を押してヘルプを見ます。

以上です。アプリケーションがどのように動作するか試してみてください。プログラムを終了するときには、完全に機能する [ファイル | 終了] コマンドを使うことができます。

第6章

デスクトップのカスタマイズ

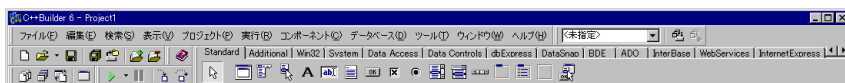
この章では、C++Builder の IDE をカスタマイズする方法について説明します。

作業領域の整理

IDE は開発をサポートするツールを多数提供していますが、メニューやツールバーの並べ替え、ツールウィンドウの結合、デスクトップの新しい外観の保存などにより、開発者が最も使いやすいように作業環境を整理することができます。

メニューとツールバーの配置

メインウィンドウでは、メニュー、ツールバー、およびコンポーネントパレットを、それぞれの左端にあるグラバーをクリックしてドラッグすることで、その位置を変えることができます。



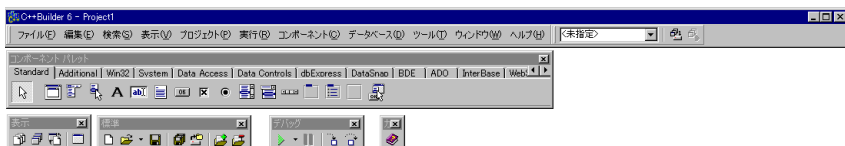
デフォルト構成のメインウィンドウ

ツールバーとメニューは、メインウィンドウ内で移動できます。グラバー（左端にある二重線）をクリックして、好きな位置までドラッグします。



構成を変更したメインウィンドウ

各部分は、メインウィンドウから切り離して別の場所に配置したり、デスクトップから消去することができます。これは、デュアルモニターを使う場合に便利です。



構成を変更したメインウィンドウ

[表示 | ツールバー | カスタマイズ] を選択すれば、ツールバーのツールを追加または削除することができます。[コマンド] ページをクリックし、カテゴリを選択してコマンドを選択し、コマンドを配置するツールバーまでドラッグします。



[コマンド] ページでコマンドを選択し、ツールバーにドラッグします。

[オプション] ページで [ツールチップヒントを表示する] をチェックすると、コンポーネントのヒントが表示されます。

詳細については、ヘルプのキーワードで「ツールバー、カスタマイズ」を検索してください。

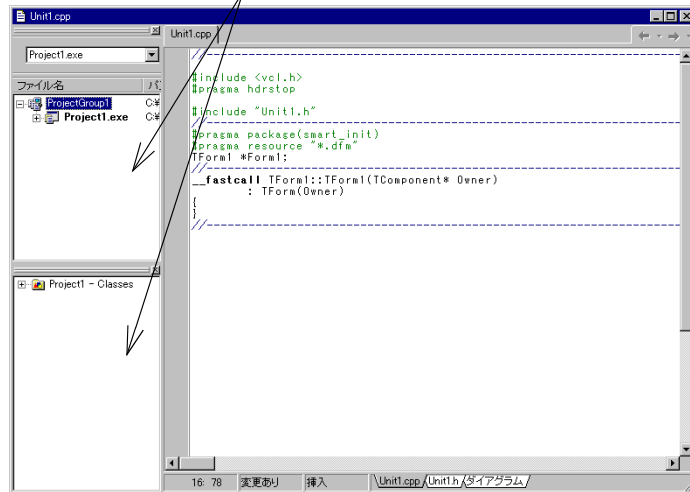
ツールウィンドウのドッキング

各ツールウィンドウを、個別に開いたり閉じたりすることができ、デスクトップ上に好きなように配置できます。また、多くのウィンドウは管理を容易にするために **ドッキング** することもできます。ドッキングとは、複数のウィンドウを結合して一緒に移動できるようにすることで、画面領域を効率よく使用でき、ツールへもアクセスしやすくなります。

[表示] メニューから任意のツールウィンドウを表示して、別のウィンドウに直接ドッキングすることができます。たとえば、C++Builder をデフォルトの設定で開くと、コードエディタの左にクラスエクスプローラがドッキングされています。必要に応じて、このウィンドウにプロジェクトマネージャを追加して、3つのウィンドウをドッキングすることもできます。

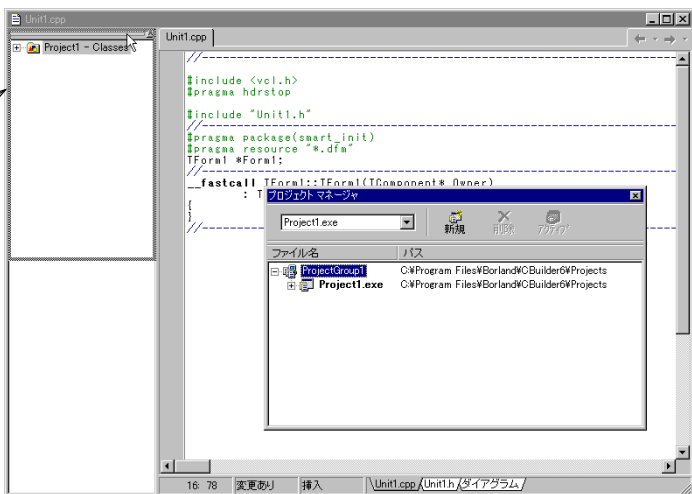
ここでは、プロジェクトマネージャとクラスエクスプローラをコードエディタにドッキングしています。

ウィンドウは、左側にあるグラバー(タブ)を使って結合またはドッキングできます。

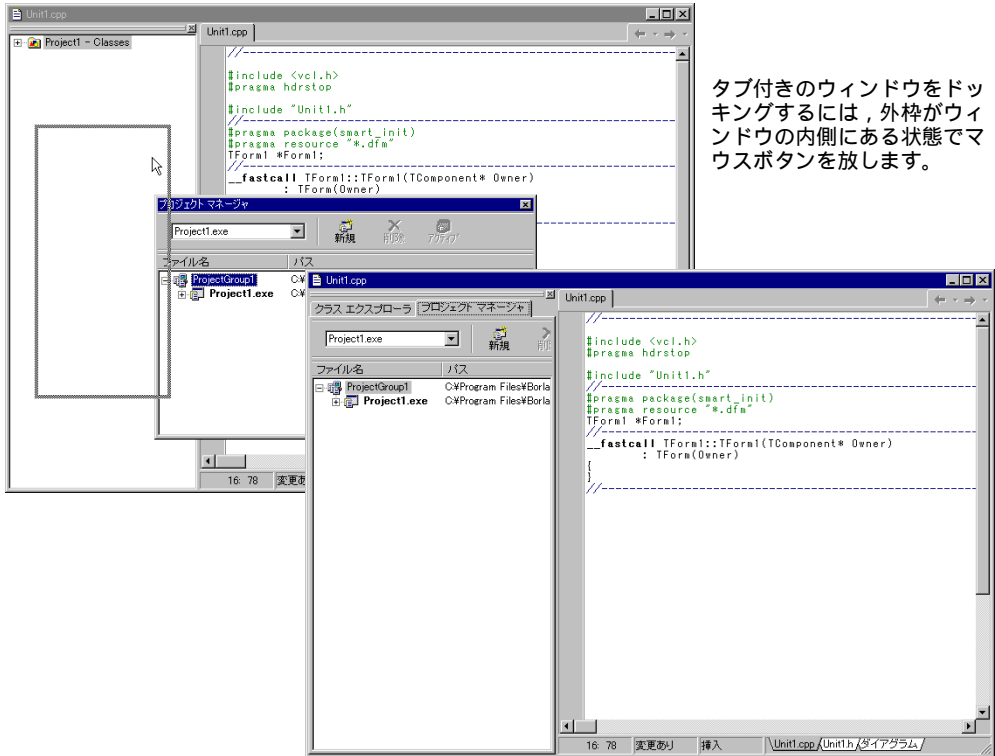


ウィンドウをドッキングするには、タイトルバーをクリックして、他のウィンドウまでドラッグします。ドラッグしている外枠が小さな長方形になり、位置がウィンドウの隅に合ったら、マウスを放します。これで、2つのウィンドウがドッキングされます。

グラバーを使ってウィンドウをドッキングするには、外枠がウィンドウの隅に合ったときにマウスを放します。



ツールをドッキングしてタブ付きウィンドウにすることもできます。



ウィンドウのドッキングを切り離すには、グラバーまたはタブをダブルクリックするか、タブをクリックしてからタブをドッキング領域の外へドラッグします。

自動ドッキングをオフにするには、[Ctrl] を押しながらウィンドウをドラッグするか、[ツール | 環境オプション] を選択し、[設定] ページをクリックして、[自動ドッキング] チェックボックスのチェックを外します。

詳細については、ヘルプのキーワードで「ドッキング」を検索してください。

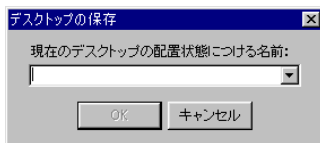
デスクトップ配置の保存

デスクトップの配置をカスタマイズして保存することができます。IDE のデスクトップツールバーには、使用できるデスクトップ配置の選択リストと、デスクトップのカスタマイズを簡単に行うための 2 つのアイコンがあります。



目的に応じて、各ウィンドウの表示 / 非表示、サイズ、およびドッキングの設定を調整し、デスクトップの配置を決めます。

デスクトップツールバーの [現在のデスクトップを保存] アイコンをクリックするか、[表示 | デスクトップ | デスクトップを保存] を選択し、新しい配置に名前を付けます。



保存するデスクトップ配置の名前を入力して、[OK] をクリックします。

詳細については、ヘルプのキーワードで「デスクトップ、配置」を検索してください。

コンポーネントパレットのカスタマイズ

デフォルトの設定では、コンポーネントパレットには多数の VCL または CLX オブジェクトがタブ付きページに機能別に分類して整理されています。コンポーネントパレットは次の点をカスタマイズできます。

- コンポーネントの非表示、再構成
- ページの追加、削除、再構成、名前の変更
- コンポーネントテンプレートの作成とパレットへの追加
- 新しいコンポーネントのインストール

コンポーネントパレットの再構成

コンポーネントパレットのページの追加、削除、再構成、名前の変更、コンポーネントの非表示や再構成を行うには、[コンポーネント | パレットの設定] ダイアログボックスを使用します。このダイアログボックスは次の方法で開くことができます。

コンポーネントパレットのカスタマイズ

- [コンポーネント | パレットの設定] を選択する
- [ツール | 環境オプション] を選択し, [パレット] タブをクリックする
- コンポーネントパレットを右クリックし, [プロパティ] を選択する

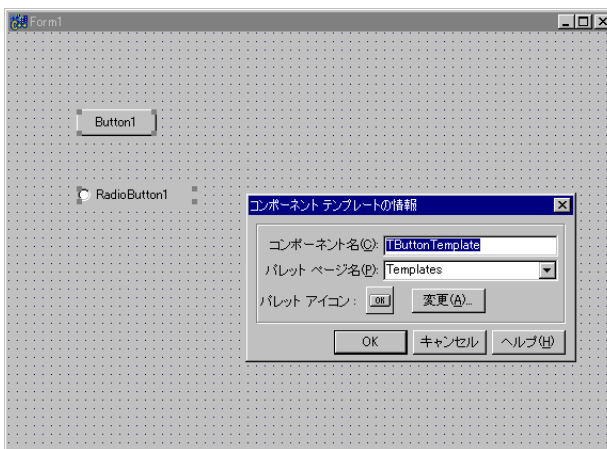
詳細については,
[パレットの設定] ダイアログボックスで [ヘルプ] ボタンをクリックしてください。

コンポーネントテンプレートの作成

コンポーネントテンプレートは, 1 回の操作でフォームに追加できるコンポーネントのグループです。テンプレートを使うと, フォームに複数のコンポーネントを設定してから, その配置, デフォルトプロパティ, およびイベントハンドラをコンポーネントパレット上に保存し, 他のフォームで再利用することができます。

コンポーネントテンプレートを作成するには, フォームにコンポーネントをいくつか配置し, オブジェクトインスペクタでプロパティを設定してから, マウスでドラッグしてそのすべてのコンポーネントを選択します。次に, [コンポーネント | コンポーネントテンプレートの作成] を選択します。表示される [コンポーネントテンプレートの情報] ダイアログボックスでは, テンプレートの名前, テンプレートを表示するパレットページ, およびパレット上でテンプレートを表すアイコンを選択します。

テンプレートをフォームに配置した後は, 各コンポーネントを 1 つずつ配置した場合と同じように, 個別に位置を変えたり, プロパティを再設定したり, イベントハンドラを作成または変更したりすることができます。



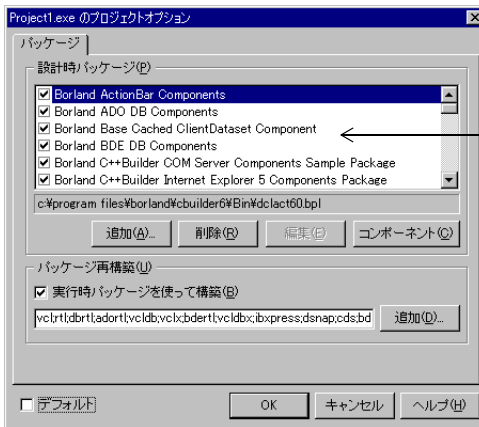
詳細については,
ヘルプのキーワードで「コンポーネントテンプレート」を検索してください。

コンポーネントパッケージのインストール

カスタムコンポーネントは、自分で作成する場合でもベンダーから入手した場合でも、パッケージとしてコンパイルされていないと、コンポーネントパレットにインストールすることはできません。

パッケージは、C++Builder アプリケーション間、IDE、またはその両方で共有できるコードを含む特別な DLL です。実行時パッケージは、ユーザーがアプリケーションを実行するときに必要な機能を提供します。設計時パッケージは、IDE にコンポーネントをインストールするために使用します。C++Builder のパッケージには、拡張子 .bpl が付きます。

サードパーティ製のコンポーネントがすでにパッケージとしてコンパイルされている場合は、そのベンダーの指示に従うか、[コンポーネント | パッケージのインストール] を選択します。



これらのコンポーネントは、あらかじめ C++Builder にインストールされています。サードパーティ製の新しいコンポーネントをインストールすると、そのパッケージがこのリストに表示されます。

[コンポーネント] をクリックすると、パッケージに入っているコンポーネントを確認できます。

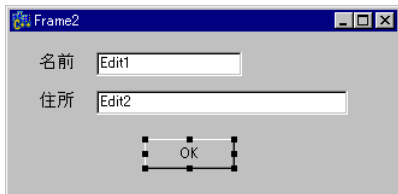
詳細については、ヘルプのキーワードで「インストール、コンポーネント」および「パッケージ」を検索してください。

フレームを使う

フレーム (TFrame) は、フォームと似ており、再使用したい他のコンポーネントのコンテナとなります。フレームは、フォームよりもカスタマイズされたコンポーネントに近いと言えます。フレームは、再利用を容易にするためにコンポーネントパレットに保存でき、フォームや他のフレーム、あるいは他のコンテナオブジェクトの中にネストすることができます。作成して保存したフレームはユニットとして機能し、中に含まれるコンポーネント (他のフレームも含む) からの変更を継承します。別のフレームまたはフォームに埋め込まれたフレームは、そこから派生したフレームに対する変更を継承します。

プロジェクトオプションの設定

新しいフレームを開くには、[ファイル | 新規作成 | フレーム] を選択します。



フレームには、どのビジュアルおよび非ビジュアルコンポーネントでも追加できます。新しいユニットが自動的にコードエディタに追加されます。

詳細については、ヘルプのキーワードで「フレーム」および「TFrame」を検索してください。

ActiveX コントロールの追加

ActiveX コントロールをコンポーネントパレットに追加して、C++Builder のプロジェクトで使用することができます。[コンポーネント | ActiveX コントロールの取り込み] を選択して [ActiveX の取り込み] ダイアログボックスを表示します。このダイアログで、新しい ActiveX コントロールを登録するか、すでに登録済みのコントロールを IDE にインストールするために選択します。ActiveX コントロールをインストールするときに、C++Builder はそのための「ラッパー」ユニットファイル作成してコンパイルします。

詳細については、[コンポーネント | ActiveX コントロールの取り込み] を選択して、[ヘルプ] ボタンをクリックしてください。

プロジェクトオプションの設定

プロジェクトディレクトリの管理や、プロジェクトのフォーム、アプリケーション、コンパイラ、リンクのオプションを指定する必要がある場合は、[プロジェクト | オプション] を選択します [プロジェクトオプション] ダイアログボックスで設定を変更した場合、その変更は現在のプロジェクトについてだけ有効ですが、その設定を新しいプロジェクトで使用されるデフォルト設定として保存することもできます。

デフォルトプロジェクトオプションの設定

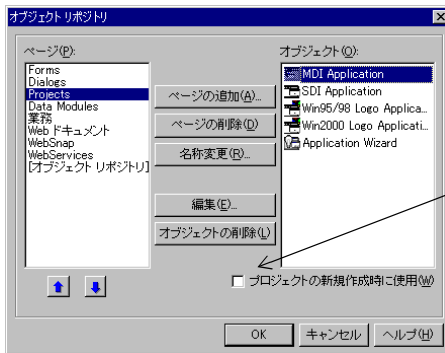
新しいプロジェクトで使われるデフォルト設定として保存するには、[プロジェクトオプション] ダイアログボックスの左下にある [デフォルト] をチェックします。[デフォルト] をチェックしておくと、ダイアログボックスの現在の設定内容が、Cbuilder6¥Bin ディレクトリにあるオプションファイル default.bpr に書き込まれます。C++Builder の元のデフォルト設定に戻すには、default.bpr ファイルを削除するか、ファイル名を変更します。

詳細については、ヘルプのキーワードで「プロジェクトオプションダイアログボックス」を検索してください。

プロジェクトおよびフォームテンプレートをデフォルトに指定する

[ファイル | 新規作成 | アプリケーション] を選択すると、プロジェクトテンプレートをデフォルトプロジェクトとして指定していない場合には、C++Builder は空のフォームを持つ標準の新規アプリケーションを作成します。[プロジェクト | リポジトリに追加] を選択すると、独自のプロジェクトを、オブジェクトリポジトリの [プロジェクト] ページにテンプレートとして保存することができます (6-10 ページの「オブジェクトリポジトリへのテンプレートの追加」を参照してください)。また、オブジェクトリポジトリから C++Builder の既存のプロジェクトテンプレートを選ぶことができます (2-5 ページの「オブジェクトリポジトリ」を参照してください)。

プロジェクトテンプレートをデフォルトとして指定するには、[ツール | リポジトリ] を選択します。[オブジェクトリポジトリ] ダイアログボックスで、[ページ] の下の [プロジェクト] を選択します。[プロジェクト] ページでプロジェクトをテンプレートとして保存すると、そのテンプレートが [オブジェクト] リストに表示されます。テンプレートの名前を選択し、[プロジェクトの新規作成時に使用] をチェックして、[OK] をクリックします。



オブジェクトリポジトリのページには、プロジェクトテンプレートのみ、フォームテンプレートのみ、または両方のテンプレートが含まれます。

プロジェクトテンプレートをデフォルトに設定するには、[オブジェクト] リストで項目を選択し、[プロジェクトの新規作成時に使用] をチェックします。フォームテンプレートをデフォルトに設定するには、[オブジェクト] リストで項目を選択し、[プロジェクトの新規作成時に使用] または [メインフォームとして使用] をチェックします。

プロジェクトテンプレートをデフォルトとして指定すると、C++Builder は [ファイル | 新規作成 | アプリケーション] が選択されたときに自動的にそのテンプレートを開きます。

デフォルトプロジェクトを指定するのと同じ方法で、オブジェクトリポジトリの既存のテンプレートのリストからデフォルトの新規フォームとデフォルトのメインフォームを指定することができます。デフォルト新規フォームは、[ファイル | 新規作成 | フォーム] を選択してプロジェクトにフォームを追加するときに作成されるフォームです。デフォルトメインフォームは、新規アプリケーションを開くときに作成されるフォームです。デフォルトのフォームが指定されていない場合、C++Builder は空白のフォームを使用します。

[ファイル | 新規作成 | その他] を選択して [新規作成] ダイアログボックスで別のテンプレートを選択すれば、デフォルトプロジェクトまたはデフォルトフォームを一時的に無効にできます。

詳細については、

ヘルプのキーワードで「テンプレート、オブジェクトリポジトリへの追加」、「プロジェクト、デフォルトの指定」、および「フォーム、デフォルトの指定」を検索してください。

オブジェクトリポジトリへのテンプレートの追加

作成したオブジェクトをテンプレートとしてオブジェクトリポジトリに追加して、再利用したり、ネットワークを通して他の開発者と共有することもできます。オブジェクトを再利用することによって、共通のインターフェースと機能を持ったアプリケーションのファミリーを構築でき、開発時間と品質を高めることができます。

たとえば、リポジトリにプロジェクトをテンプレートとして追加するには、プロジェクトを保存してから、[プロジェクト | リポジトリに追加] を選択します。[リポジトリに追加] ダイアログボックスで必要な設定を行います。



タイトル、説明、および作成者を入力します。[ページ] リストボックスで [プロジェクト] を選択して、このプロジェクトがリポジトリの [プロジェクト] ページに表示されるようになります。

これで、[新規作成] ダイアログボックスを開いたときに、このプロジェクトテンプレートが [プロジェクト] ページ（またはテンプレートを追加したページ）に表示されます。作成したテンプレートを、C++Builder 起動時のデフォルトにする方法については、6-9 ページの「プロジェクトおよびフォームテンプレートをデフォルトに指定する」を参照してください。

詳細については、ヘルプのキーワードで「テンプレート、オブジェクトリポジトリへの追加」を検索してください。

ツールの環境設定

フォームデザイナーやオブジェクトインスペクタ、コードエクスプローラなど、IDE の外観と動作のさまざまな面を制御することができます。これらの設定は、現在のプロジェクトだけではなく、それ以降に開いてコンパイルするプロジェクトにも有効です。プロジェクトに関する IDE のグローバルな設定を変更するには、[ツール | 環境オプション] を選択します。

詳細については、ヘルプのキーワードで「環境オプションダイアログボックス」を検索するか、[環境オプション] ダイアログボックスの各ページの [ヘルプ] ボタンをクリックしてください。

フォームデザイナーのカスタマイズ

[ツール | 環境オプション] ダイアログボックスの [デザイナー] ページには、フォームデザイナーに関する設定があります。たとえば、コンポーネントを最も近いグリッドの位置に合わせて配置する

「グリッドに合わせる」機能を有効にするかどうか、またフォームに配置した非ビジュアルコンポーネントの名前（キャプション）を表示するかどうかを設定できます。

詳細については、
[環境オプション] ダイアログボックスの [デザイナ] ページで [ヘルプ] ボタンをクリックしてください。

コードエディタのカスタマイズ

エディタは、すぐにもカスタマイズしたいツールでしょう。[ツール | エディタオプション] ダイアログボックスの各ページでは、コードの編集に関する設定を行います。たとえば、キーストロークの設定、フォント、余白の幅、色、構文の強調表示、タブ、インデントなどの形式を選択できます。

エディタ内で使用できる支援機能ツールについても、[エディタオプション] ダイアログの [支援機能] ページで設定を変更できます。これらのツールの詳細については、2-6 ページの「支援機能」を参照してください。

詳細については、
[エディタオプション] ダイアログボックスの [一般]、[表示]、[キーの割り当て]、[色]、[支援機能] の各ページで [ヘルプ] ボタンをクリックしてください。

索引

A

ActiveX

~コントロールのインストール 6-8

コンポーネントパレットページ 3-13

ADO 3-11

B

BDE 3-11

BDE Administrator 3-11

.BPR ファイル 4-2

C

C++Builder

カスタマイズ 6-1 ~ 6-11

プログラミング 3-1

CLX

アプリケーションの作成 3-9

概要 3-6

コンポーネントの追加 2-3

CLX (クロスプラットフォーム用コンポーネントライブラリ) 3-6

D

Database Desktop 3-12

dbExpress 3-11

.dfm ファイル 2-8, 4-1

DLL

作成 2-5

~とは 3-12

配布 3-8

G

GUI の作成 4-2

I

IDE

IDE とは 1-1

概要 2-1

カスタマイズ 6-1 ~ 6-11

作業環境の整理 6-1

IME 3-8

InterBase 3-11

M

MainMenu コンポーネント 5-10

O

ODBC 3-11

P

Panel コンポーネント 5-12

Paradox 3-11

.pas ファイル 4-1

S

SQL Link 3-11

SQL Server 3-11

SQL エクスプローラ 3-11

SQL データベースサーバ 3-11

[StatusBar1.Panels の編集] ダイアログボックス 4-5

T

To-Do リスト 2-10

V

VCL (ビジュアルコンポーネントライブラリ)

コンポーネントの追加 2-3

使い方 3-6

W

Web サーバーアプリケーション

作成 3-10

Web サイト (Borland) 1-6

WebSnap (概要) 3-10

X

.xfrm ファイル 2-8

あ

アクション

アプリケーションへの追加 4-9, 4-11, 4-17

アクションマネージャエディタ 4-9 ~ 4-12

新しい機能 1-3

アプリケーション

Web サーバー ~ 3-10

国際化 3-8

コンパイルとデバッグ 3-7, 4-13, 4-22

作成 3-1, 3-9

データベース ~ 3-10

配布 3-8

い

イベント 5-13 ~ 5-16

イベントハンドラ 5-13 ~ 5-16

作成 4-23 ~ 4-29

~とは 3-5

イメージ
アプリケーションへの追加 4-7, 4-14
表示する 5-12
インストール
コンポーネント 6-7

う

ウィザード
オブジェクトリポジトリ 2-5
ウィンドウ
ドッキング 6-2

え

[エディタオプション] ダイアログボックス 2-7, 6-11
エラーメッセージ 4-29

お

オブジェクト
オブジェクトとは 3-6
オブジェクトインスペクタ
概要 2-4
使い方 3-4, 4-2
~内でのコンポーネントの参照 3-4
オブジェクトツリービュー 2-4
オブジェクトリポジトリ
オブジェクトリポジトリとは 2-5, 3-1
使い方 2-5
テンプレートの追加 6-9, 6-10
オプション
プロジェクト~の設定 6-8
親子関係 2-4
オンラインヘルプファイル 1-3

か

開発者サポート 1-6
カスタマイズ
IDE 6-1 ~ 6-11
コードエディタ 6-11
コンポーネントパレット 2-2
フォームデザイナー 6-10

画像
表示する 5-12

[環境オプション] ダイアログボックス 6-10

き

キーストローク設定 6-11
起動 (C++Builder) 2-1

く

クラス
~とは 4-4
クラスエクスプローラ 2-9

クラスライブラリ 3-6
グラフィック
表示する 5-12
クロスプラットフォーム
~アプリケーションの開発 3-9

こ

項目の追加
オブジェクトリポジトリ 2-5
コード
イベントハンドラ 3-5
記述する 3-5
コーディング時の支援機能 2-6 ~ 2-7
表示と編集 2-6 ~ 2-9
コードエクスプローラ
使い方 2-9
コードエディタ
カスタマイズ 6-11
他のウィンドウとのドッキング 6-2
使い方 2-6 ~ 2-7
コードテンプレート 2-6
コードパラメータ 2-6
コード補完 2-6
国際化 (アプリケーションの~) 3-8
コンテキストメニュー
表示する 2-3
コントロール
フォームへの追加 3-2, 4-3
コンパイル (アプリケーションの~) 3-7
コンパイル (プログラムの~) 4-22, 5-8
コンポーネント
インストール 3-12, 6-7
カスタマイズ 3-12, 6-6
カスタム~の作成 3-12
コンポーネントパレット上での配置 6-5
コンポーネントパレットへの追加 6-5
~とは 4-3
フォームへの追加 3-2, 4-4
プロパティの設定 3-4, 4-2
コンポーネントテンプレート
作成 6-6
コンポーネントパレット
カスタマイズ 6-5 ~ 6-7
カスタムコンポーネントの追加 3-12
コンポーネントパレットとは 2-3
使い方 3-2
ページの追加 6-5

さ

サポートサービス 1-6
サンプルプログラム 4-1 ~ 4-35, 5-1 ~ 5-16

し

式評価 2-6
識別子定義の表示 2-6
実行 (アプリケーション) 4-22, 5-8
実行 (アプリケーションの～) 3-7, 4-13
実行形式ファイル
 配布 3-8
実行ボタン 5-8
情報の検索 1-3
書体と表記 1-6
[新規作成] ダイアログボックス
 使い方 2-5, 4-32
 テンプレートの保存 6-9, 6-10
新規フォーム 6-9

せ

設計時表示 (フォームを閉じる) 4-3
設定
 プロパティ 3-4, 4-2, 4-10, 4-12, 4-16

そ

ソースコード
 コーディング時の支援機能 2-6～2-7

た

[ダイアグラム] ページ 2-7
ダイアログボックス
 オブジェクトリポジトリ 2-5
タイプライブラリ 3-13
タブ付きウィンドウ
 ドッキング 6-3

ち

チュートリアル 4-1～4-35, 5-1～5-16

つ

追加
 コンポーネントをフォームに～ 4-3, 4-19
ツールウィンドウ
 ドッキング 6-2
ツールバー 2-2
 アプリケーションへの追加 4-13, 4-21
 ツールの追加と削除 6-2
 配置 6-1

て

データディクショナリ 3-12
データベース
 サンプルプログラム 5-1～5-16
データベースアプリケーション
 アクセス 5-3～5-4
 作成 3-10

データベースエクスプローラ 3-11
データモジュール
 作成 2-5
 追加 3-2
テキストエディタ (チュートリアル) 4-1～4-35
テクニカルサポート 1-6
デスクトップ
 作業環境の整理 6-1～6-5
 配置の保存 6-5
デバッグ (プログラムの～) 3-7～3-8, 4-13
デフォルト
 プロジェクトオプション 6-8
 プロジェクト/フォームテンプレート 6-9
テンプレート
 オブジェクトリポジトリへの追加 6-10
 デフォルトとして指定する 6-9

と

統合開発環境 (IDE)
 概要 2-1
 カスタマイズ 6-1～6-11
統合デバッグ 3-7
閉じる
 フォーム 4-3
ドッキング (ウィンドウ) 6-2～6-4
トランスレーションツール 3-8

に

入力メソッドエディタ (IME) 3-8

は

バージョン (C++Builder) 3-9
[バージョン情報] ダイアログボックス 4-32
配布 (アプリケーションの～) 3-8
パッケージ 6-7

ひ

ビットマップ
 アプリケーションへの追加 4-7, 4-14
標準アクション
 アプリケーションへの追加 4-17
ヒント 4-4

ふ

ファイル
 フォーム～ 2-8, 4-1
 プロジェクト～ 4-1
 保存 4-2
 ユニット～ 4-1
フォーム
 オブジェクトリポジトリ 2-5
 コンポーネントの追加 3-2, 4-3

- デフォルトとして指定する 6-9
- 閉じる 4-3
- メイン ~ 4-2, 6-9
- フォームデザイン
 - 概要 2-4
 - カスタマイズ 6-10
- フォームファイル
 - コードの表示 2-8
 - フォームファイルとは 4-1
- フレーム 6-7
- プログラミング (C++Builder) 3-1
- プログラム
 - CLX アプリケーション 3-9
 - Web サーバーアプリケーション 3-10
 - 国際化 3-8
 - コンパイルとデバッグ 3-7, 4-13, 4-22
 - 配布 3-8
- プロジェクト
 - 管理 2-9 ~ 2-10
 - 項目の追加 2-5
 - 作成 3-1
 - 種類 3-9 ~ 3-12
 - デフォルトとして指定する 6-9
 - 保存 4-2
 - オプションをデフォルトとして設定する 6-8
- [プロジェクトオプション] ダイアログボックス 6-8
- プロジェクトグループ 2-10
- プロジェクトテンプレート 6-10
- プロジェクトファイル 4-2
 - デフォルトの名前 4-1
- プロジェクトマネージャ 2-9 ~ 2-10
- プロパティ
 - 設定 3-4, 4-2, 4-10, 4-12, 4-16

へ

- ヘッダーファイル 4-1
- ヘルプ (F1) 1-3

- ヘルプヒント 4-4
- ヘルプファイル
 - アプリケーションへの追加 4-30

ほ

- 保存
 - デスクトップの配置 6-5
 - プロジェクト 4-2

み

- 右クリックメニュー 2-3

め

- メイクファイル 4-2
- メインフォーム 6-9
- メッセージ
 - エラー ~ 4-29
- メニュー
 - C++Builder の ~ 2-3
 - アプリケーションへの追加 4-19
 - コンテキスト ~ 2-3
 - 配置 2-2, 6-1

も

- 文字セット (拡張 ~) 3-8

ゆ

- ユーザーインターフェースの作成 3-2, 4-2, 4-3
- ユニットファイル 4-1
- ユニットヘッダーファイル 4-1

り

- リソース DLL ウィザード 3-9

ろ

- ローカライズ (アプリケーションの ~) 3-8