

ゲームプログラミング

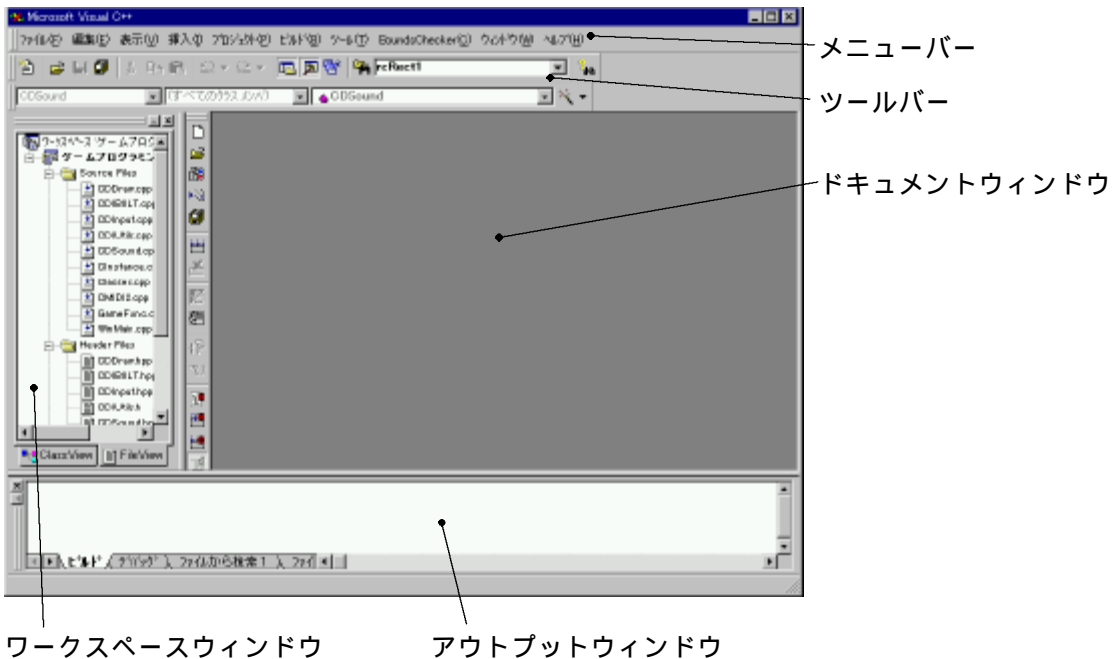
基礎編 - 第1回 VisualC++マニュアル

VisualC++の基本的な操作方法や使い方を解説します。

キー操作一覧

,	カーソルの1文字移動	Ctrl + N	新規作成
,	カーソルの1行移動	Ctrl + O	ファイルを開く
Ctrl + ,	カーソルの1ワード移動	Ctrl + S	ファイルの保存
Ctrl + ;	1行スクロール		
Enter	改行を挿入	Ctrl + F7	コンパイル
Tab	タブを挿入	F7	ビルド
Insert	上書き/挿入モード切替	Ctrl + Break	ビルドの中止
Delete	削除	Ctrl + F5	プログラムの実行
Backspace	後方の1文字を削除	F4	エラーがある行を画面に表示
Home, End	行の先頭, 末尾に移動	F5	デバッグ起動
Page Up, Page Down	1画面スクロール	F9	ブレイクポイントの挿入/削除
Ctrl + Home, End	ドキュメントの先頭, 末尾に移動	Ctrl + Shift + F9	すべてのブレイクポイントを削除
		F10	1行実行(ステップオーバー)
Shift +	選択範囲の指定	F11	1行実行(ステップイン)
Ctrl + A	すべて選択	Ctrl + F10	カーソルの直前まで実行
Ctrl + C	コピー	Shift + F11	ステップアウト(関数から抜ける)
Ctrl + V	貼り付け	Shift + F5	デバッグの中止
Ctrl + X	切り取り		
Ctrl + Y	やり直し	選択範囲指定 + Tab	選択範囲すべての行にタブを挿入
Ctrl + Z	元に戻す	選択範囲指定 + Shift + Tab	選択範囲すべての行からタブを1つ削除
Ctrl + F	文字列の検索	Alt + F8	選択範囲の書식을自動整形

各部の名称



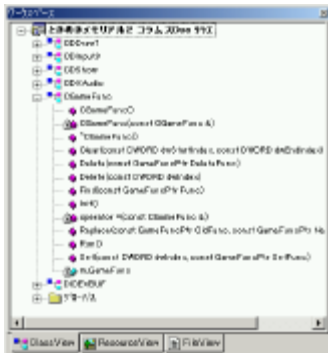
ワークスペースウィンドウ

ワークスペースウィンドウは、"ClassView", "ResourceView", "FileView"といったペインから構成され、作成されたプロジェクト単位に構成要素を表示します。
各ペインが表示する構成要素は、次のようになっています。

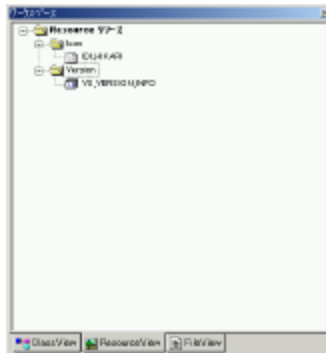
ClassView：構造体とクラスのメンバ、グローバル変数と関数を表示します。

ResourceView：リソースを表示します。

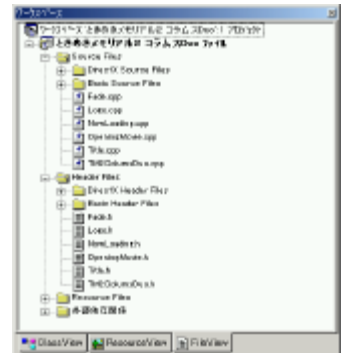
FileView：プロジェクトを構成するファイル(ソースファイル、ヘッダファイル、リソースなど)を表示します。



ClassView



ResourceView

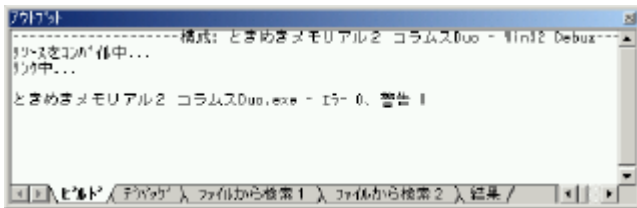


FileView

各ペインに表示された構成要素をダブルクリックすることにより、ドキュメントウィンドウに呼び出して編集することができます。

アウトプットウィンドウ

アウトプットウィンドウは、ビルド、デバッグなどの処理で出力されるメッセージを表示します。
"ビルド", "デバッグ", "ファイルからの検索"といったペインから構成されます。



アウトプットウィンドウ

ワークスペースとプロジェクト

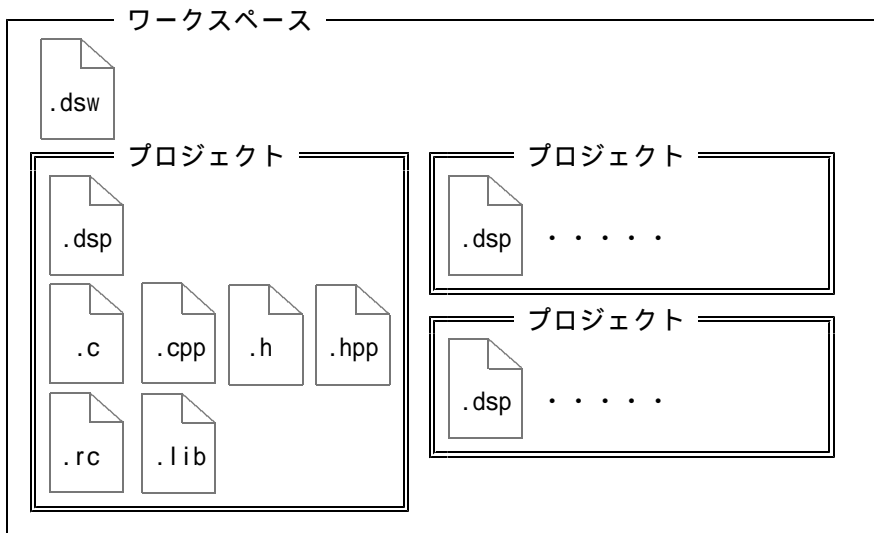
VisualC++で作成するアプリケーションは、非常に多くのファイルから構成されます。たとえば、プログラムが書かれたソースファイルや、コンパイルのための情報が書かれたファイル、アイコンやカーソルなどのリソースファイルです。

VisualC++では、これらをひとまとめにして「プロジェクト」と呼び、アプリケーションの実行に必要なプロジェクトをひとまとめにして「ワークスペース」と呼びます。たとえば、1つの実行ファイル(.exe)と2つのDLLファイルから構成するアプリケーションの場合は、ワークスペースに3つのプロジェクトが存在することになります。

プログラマはプロジェクトを「ビルド」することで目的のプログラムを作成することができます。プロジェクトには一般に以下のような情報が含まれています。

プロジェクトの名前 作成するファイルの名前 ソースファイル一覧とコンパイルの方法 ソースファイルにインクルードされているヘッダファイルの一覧と依存関係 コンパイルおよびリンクコマンドを呼び出す際のオプション

これらの情報はプロジェクトワークスペースファイル(.dsw)とプロジェクトファイル(.dsp)の2種類のファイルに書き込まれています。



VisualC++では、ワークスペースおよびプロジェクトは、上で挙げた2つのファイル

- .dsw (ワークスペースファイル)
- .dsp (プロジェクトファイル)

があれば復元することができます。したがって、プロジェクトを持ち運ぶときには、ソースファイルやリソースファイルのほかにこれらのファイルがあればプロジェクトを復元することができます。たいていの場合、以下のファイル

- .opt (プロジェクトオプションファイル)
- .ncb (クラス情報ファイル)
- .plg (プロジェクトログファイル)

が作成されており、またその他にも以下のような数種類のバイナリファイルが、DebugフォルダやReleaseフォルダなどに作成されます。これらは必要に応じて作成されるので保存しておく必要はありませんが、保存しておくことで最初のビルド時間が大幅に短縮されます。

- .obj (オブジェクトファイル。ソースファイルが翻訳されたファイル)
- .res (リソースファイル。リソーススクリプトが翻訳されたファイル)
- .aps (リソースID定義ファイル。リソースIDと数値の対応を定義)
- .pch (プリコンパイル済みファイル。次回からのコンパイル時間を短縮するために作成されます)
- .ilk (インクリメンタルリンク情報。次回からのリンク時間を短縮するために作成されます)
- .idb (デバッグ情報が格納されているファイル)
- .pdb (デバッグ情報が格納されているファイル。リンク時に存在しないとデバッグできなくなります)

アプリケーションのビルド

アプリケーションを開発する際には、さまざまな種類のファイルが必要とします。VisualC++で開発する際にたとえば以下のようなファイルを使用することになります。

- C ソースファイル(.c)
- C++ソースファイル(.cppまたは.cxx)
- ヘッダファイル(.hまたは.hpp)
- オブジェクトファイル(.obj)
- スタティックライブラリ(.lib)
- DLL定義ファイル(.def)
- リソーススクリプトファイル(.rc)
- リソースファイル(.res)
- アイコンファイル(.ico)

VisualC++はこれらのファイルを組み合わせて1つアプリケーションを構築します。この作業を「ビルド」と呼びます。通常のビルドでは、次のような処理が行われます。

リソースコンパイル

リソーススクリプト(.rc)をシステムが読み込める形式(.res)に変換します。

コンパイル

CやC++で書かれたソースコード(.c, .cpp, .cxxなど)をシステムの理解できるオブジェクトファイル形式(.obj)に変換します。

リンク

コンパイルやリソースコンパイルなどの結果(.res, .objなど)をまとめ、システムが読み込んで実行できる形式(.exe, .dll, .ocxなど)を作成します。

ブラウザ情報作成

デバッグ時などにソースコードを参照するための情報を作成します。

アクティブな構成の設定

VisualC++によるアプリケーションのビルドには、"Win32 Debug"と"Win32 Release"の2種類があります。デフォルトでは"Win32 Debug"が指定されており、デバッグ情報が実装された状態でアプリケーションがビルドされます。この状態で作成されたファイルは、デバッグ情報によりサイズが肥大化しています。デバッグが完了した製品出荷時には"Win32 Release"を設定し、ビルドの再実行を行います。"Win32 Release"では、プログラムの無駄を省いたり実行速度を高めるための最適化が行われてビルドされます。

これらの2つのモードを変更するには、メニューから"ビルド (B) アクティブな構成の設定(0)"を選択します。