

連載 ソフトウェア技術

MFC 入門～簡単に GUI アプリケーションを書いてみよう～

2TK 庄子

ここでは、Windows で簡単に GUI アプリケーションを作る「MFC」の”触り方”について、簡単に解説していききたいと思います。

あまり細かいことに触れず、とりあえずアプリケーションを作るために必要な流れを解説していきます。想定している開発環境は「VisualStudio2005」です。

まず、VisualStudio2005 を起動し、「ファイル」→「新規作成」→「プロジェクト」の順にクリックします。



ここからプロジェクトの必要な設定を行います。

- ・「プロジェクトの種類」は「Visual C++」の中の「MFC」にする。ここまで来たら、画面右の「VisualStudio にインストールされたテンプレート」を「MFC アプリケーション」にする。
- ・「プロジェクト名」に、適当なプロジェクト名を入力(ここでは MFC_Sample とした)
- ・「場所」に、このプロジェクトを保存する場所を指定する(ここでは”C:\MFC”とした)。
- ・「ソリューション名」はそのままにする。基本的にはプロジェクト名と同じ名前になっているはず。

ここで OK を押し、これから作るアプリケーションの基本的な設定を行います。

(1) アプリケーションの種類

- ・「アプリケーションの種類」を「シングルドキュメント」にする。
- ・「ユニコードライブラリを使用する」のチェックを外す。

(2) 生成されたクラス

- ・「基本クラス」を「CFormView」にする。



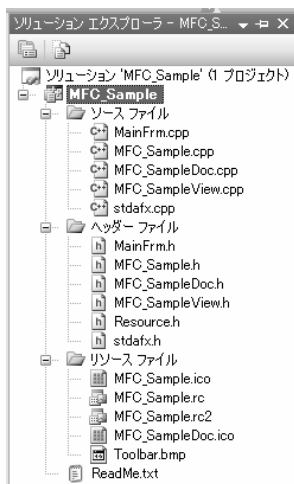
(1) の設定画面



(2) の設定画面

これで基本的な設定を終えたので完了を押します。

右のようなダイアログが出て、とりあえず無視して OK を押しておきましょう。



さて、プロジェクトの基本設定が終わったら、「表示」→「ソリューションエクスプローラ」をクリックし、このアプリケーションで用いるソースファイル等の一覧を表示しておきましょう。

左図に示すように、MFC ではアプリケーションの設定を行うだけでこれだけのファイルが自動生成されます。あとは、画面上にボタンなどの部品を張り付けたり、命令を記述することでアプリケーションを完成させていきます。

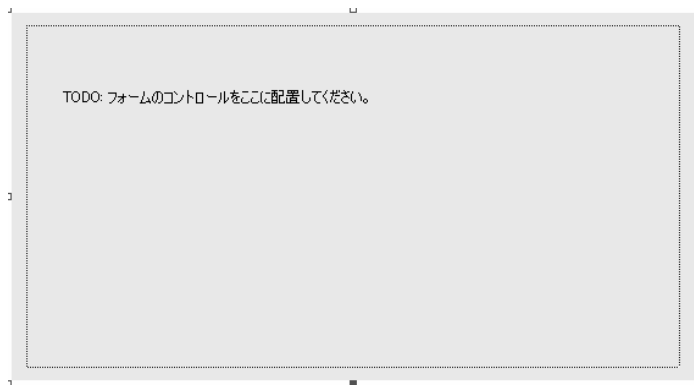
ここでは、とりあえず画面にボタンを一つ置いて、それを押

すと「こんにちは！」というダイアログを出す簡単なアプリケーションを作ってみましょう。

まず、画面にボタンを配置するために、土台となる画面を表示しておきましょう。

「表示」→「その他のウィンドウ」→「リソースビュー」をクリックし、出てきたウィンドウの中から”IDD_(プロジェクト名)_FORM”をダブルクリックします。

この”IDD_(プロジェクト名)_FORM”が、アプリケーションの画面の元(土台)となります。



上のような「画面の土台」が表示されたら、ここにボタンを張り付けましょう。

「表示」→「ツールボックス」をクリックし、画面に貼り付けることのできる部品の集まりを表示します。

“Button”, “Check Box”, “Edit Control”など、さまざまな部品があります。今回はボタンを張り付けるので、“Button”をクリックし、土台の上でドラッグしてやります。



ボタンを張り付けた状態

さて、このままではただボタンがあるだけで何もしないアプリケーションになってしまいます。そこで、「ボタンを押したら、ダイアログを出す」という処理を書いてやります。

まず、画面に貼り付けたボタンをダブルクリックします。すると、何やらC言語っぽいコードがたくさん書いてある画面に移ったと思います。

これは、画面の土台に貼り付けたボタンをダブルクリックしたことで、「ボタンをクリックしたときに行う処理を書く『場所』」ができたことを意味しています。

『場所』ができたので、今度はそこに「ボタンをクリックしたときに行う『処理』」を書いてやります。つまり、「”こんにちは!”というダイアログを出す」処理を書いてやればよいこととなります。

ちなみに、その『場所』ですが…

・ (プロジェクト名)View.cpp

というファイルになります。ここに、「このボタンが押されたらこういう処理をせよ」とか、「このメニューが押されたらこういう処理をせよ」というようにして、アプリケーションの処理を記述していきます。

さて、ボタンをダブルクリックしたときに表示されたコードをみると、

```
「void C(プロジェクト名)View::OnBnClickedButton1 ()」
```

という関数が見つかると思います。この関数の中に、「ボタンが押された時の処理」を書いてやればよいのです。

今回は「ダイアログを出す」という処理を行います。ここでは「AfxMessageBox ()」関数を使って、ダイアログを出現させる処理を実現しましょう。

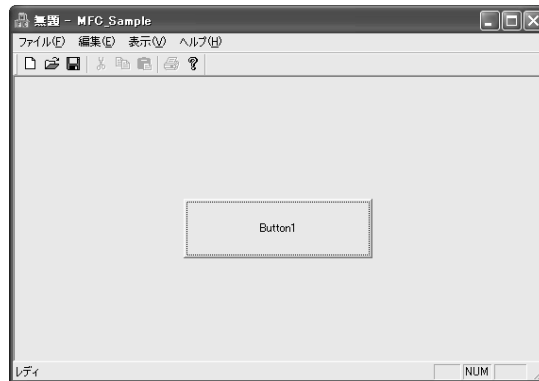
```
void CMFC_SampleView::OnBnClickedButton1 ()
{
    // TODO: ここにコントロール通知ハンドラコードを追加します。
    AfxMessageBox("こんにちは!"); //左の1行を追加する。
}
```

さて、これでアプリケーションは出来た…と言いたいところですが、土台となる画面を見ると”TODO: フォームのコントロールをここに配置してください。”という余計なものがかっついていて、邪魔なので消しちゃいましょう。

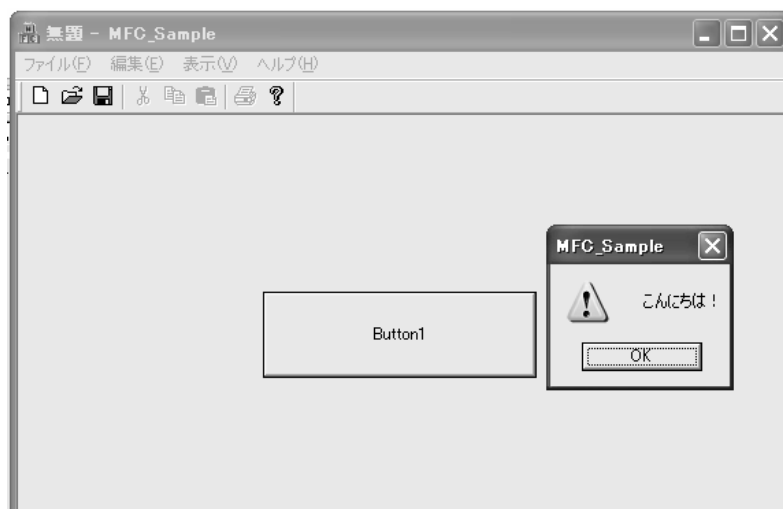
土台の画面にある”TODO: フォームのコントロールを〜”の上でクリックし、Delete キーで削除できます。

これで本当に完成です。では、出来上がったアプリケーションを実際に動かしてみましょう。

「デバッグ」→「デバッグ開始」をクリックすると、このアプリケーションのソースコードや土台の画面等々を組み合わせでコンパイルが行われます。ここでエラーがなければ、実際にアプリケーションが実行され、画面が表示されます。



できあがったアプリケーションの画面

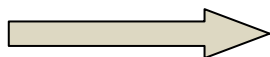


Button1 と書かれたボタンを押すと、
ちゃんと” こんにちは！” とダイアログボックスが表示される。

さて、とても大急ぎでMFCの触り方の基礎を解説しましたが、なんとなくイメージはつかんでもらえたでしょうか。

最初の方でプロジェクトの設定やら何やらやりましたが、それより重要なのは

ボタンなどの部品を、画面に貼り付ける



それに対応した動作を書いてやる

という”作り方”です。これだけは覚えておいてもらいたいと思います。