

# MFC ドキュメント

## MFC ドキュメントとビューの分離

### SDI アプリケーション

#### MFC アプリケーションの開始

以下の順序で MFC アプリケーションは開始される

1. グローバルオブジェクト theApp(CWinApp の派生クラス AppWizard が作成) を生成
2. MFC ライブラリの一部であるグローバル関数 WinMain( エントリポイント) を呼び出す
3. WinMain は、CWinApp から派生されているクラスの唯一のインスタンスを探す
4. WinMain は、theApp の InitInstance メンバ関数を呼び出す
5. 派生クラスでオーバーライドされた InitInstance 関数は、ドキュメントのロードと、メインフレーム ウィンドウ および ビュー ウィンドウ の表示処理を開始
6. WinMain は theApp の Run メンバ関数を呼び出す
7. Run メンバ関数は、ウィンドウメッセージとコマンドメッセージのディスパッチ処理を開始

### ドキュメントテンプレート

- ・ InitInstance 関数に、以下のようなコードが含まれている

```
BOOL CSdiSampleApp::InitInstance()
{
    // アプリケーション用のドキュメント テンプレートを登録します。ドキュメント テンプレート
    // はドキュメント、フレーム ウィンドウとビューを結合するために機能します。
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CSdiSampleDoc),
        RUNTIME_CLASS(CMainFrame), // メイン SDI フレーム ウィンドウ
        RUNTIME_CLASS(CSdiSampleView));
    if (!pDocTemplate)
        return FALSE;
    AddDocTemplate(pDocTemplate);
}
```

- ・ 次の 3 つのクラス間の関係を定義

クラス	内容
ドキュメント クラス	CDocument クラスから派生
ビュー クラス	ドキュメント クラスのデータを表示します。このクラスは、CView、CScrollView、CFormView、または CEditView から派生
フレーム ウィンドウ クラス	ビューを保持。SDI のドキュメント テンプレートは、CFrameWnd から派生。メイン フレーム ウィンドウ の動作をカスタマイズする必要がない場合、派生クラスを作らずに直接 CFrameWnd を使用できます

空のドキュメントを生成 CWinApp::OnFileNew 関数

InitInstance 関数は、AddDocTemplate メンバ関数を呼び出した後、OnFileNew を呼び出し (CWinApp::ProcessShellCommand を介して) 次のことを行う。

処理内容	備考
1. ドキュメントオブジェクトを生成	ディスクからのデータ読み出しは行わない
2. CMainFrame メインフレームオブジェクトを生成	メインフレームウィンドウも生成するが、表示は行わない。メインフレームウィンドウは IDR_MAINFRAME メニュー、ツールバー、ステータスバーを含む
3. ビューオブジェクトを生成	ビューウィンドウも生成するが表示は行わない
4. ドキュメント、メインフレーム、ビューの各オブジェクト間の接続	AddDocTemplate 関数によるクラスの接続と混同しない
5. ドキュメントオブジェクトの CDocument::OnNewDocument 仮想メンバ関数を呼び出す	DeleteContents 仮想関数が呼び出される
6. ビューオブジェクトの CView::OnInitialUpdate 仮想メンバ関数を呼び出す	
7. フレームオブジェクトの CFrameWnd::ActivateFrame 仮想関数を呼び出す	メニュー、ビューウィンドウ、コントロールバーを含むメインフレームウィンドウを表示する

- ・ CDocument::OnNewDocument や CView::OnInitialUpdate などの関数は初期化の作業を行うのに最適

#### ファイルを開く OnFileOpen 関数

AppWizard は、アプリケーション生成時に、「ファイル - 開く」メニューを CWinApp::OnFileOpen メンバ関数にマップする。この関数が呼び出されると、以下の処理を行う。

処理内容	備考
ユーザにファイルの選択を要求	CWinApp の DoPromptFileName メンバ関数は、標準の FileOpen ダイアログを呼び出し、現在のドキュメントテンプレートから得たファイル拡張子を表示
既に存在しているドキュメントオブジェクトの CDocument::OnOpenDocument 仮想関数を呼び出す	この関数はファイルを開き、CDocument::DeleteContents を呼び出し、CArchive オブジェクトを生成しロード状態に設定。次に Serialize 関数を呼び出し、データをロード
ビューの OnInitialUpdate 関数を呼び出す	

- ・ 現在のドキュメントの内容を消去するのに適した方法は、CDocument::DeleteContents 仮想関数をオーバーライドする。
- ・ CDocument::DeleteContents 関数は、OnNewDocument 関数や、OnOpenDocument 関数、ドキュメントを閉じる場合も呼び出される。
- ・ デストラクタは、オブジェクトの終了時まで存続するアイテムの後始末に利用する

## ファイルの上書き保存、名前をつけて保存 CDocument::OnFileSave

- ・ AppWizard は、「ファイル - 上書き保存」メニューを、CDocument::OnFileSave メンバ関数にマップする

処理内容	備考
CDocument::OnFileSave メンバ関数の呼び出し。上書き保存の場合、OnFileSaveAs 関数	
CDocument::OnSaveDocument 関数の呼び出し	OnFileSave により呼び出される
CDocument::Siliarize 関数を呼び出す	

- ・ CDocument の m\_bModified データメンバはドキュメントが変更されている場合 TRUE となる
- ・ ドキュメントが変更された場合 SetModifiedFlag 関数を使ってフラグを TRUE に設定する

## MDI アプリケーション

### MFC アプリケーションの開始

- ・ 実行開始のシーケンスは、SDI アプリケーションとほとんど同じ。
- ・ MDI の InitInstance 関数は、AddDocTemplate 関数の呼び出し以降が SDI と若干ことなる。

### MDI ドキュメントテンプレートクラス

```

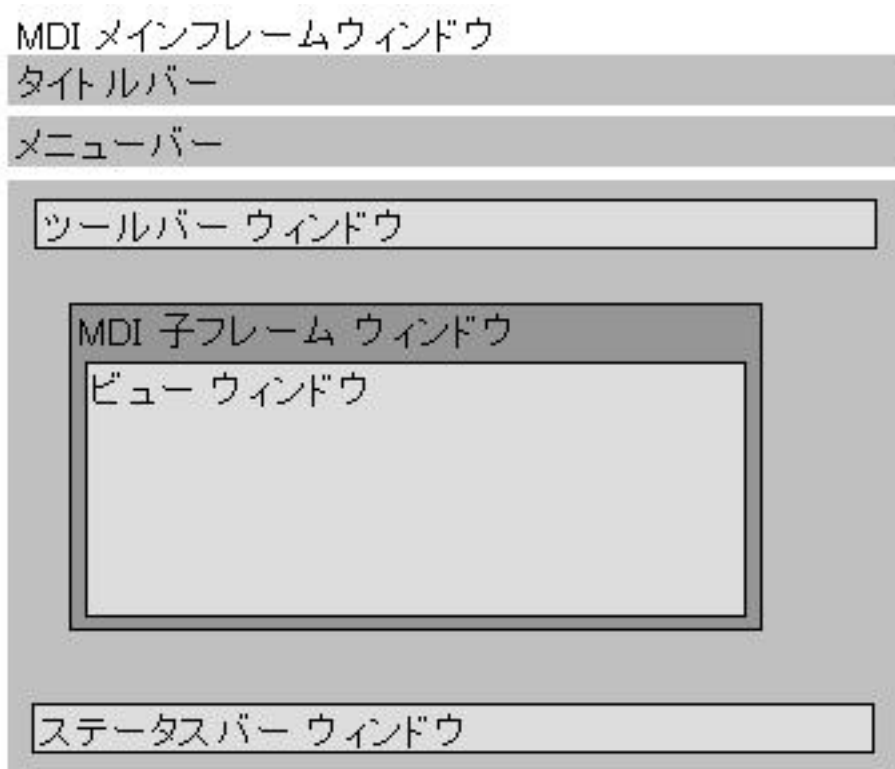
BOOL CMdiSampleApp::InitInstance()
{
    // アプリケーション用のドキュメント テンプレートを登録します。ドキュメント テンプレート
    // はドキュメント、フレーム ウィンドウとビューを結合するために機能します。
    CMultiDocTemplate* pDocTemplate;
    pDocTemplate = new CMultiDocTemplate(IDR_MdiSampleTYPE,
        RUNTIME_CLASS(CMdiSampleDoc),
        RUNTIME_CLASS(CChildFrame), // カスタム MDI 子フレーム
        RUNTIME_CLASS(CMdiSampleView));
    if (!pDocTemplate)
        return FALSE;
    AddDocTemplate(pDocTemplate);
}

```

- ・ MDI アプリケーションでは、複数のドキュメントタイプを使うことができ、同時に複数のドキュメントオブジェクトを存在させることができる

基本クラス	AppWizard 作成	オブジェクト数	メニューとコントロール	ビュー	生成
CMDIFrameWnd	CMainFrame	1 つ	あり	なし	アプリケーションクラスの InitInstance
CMDIChildWnd	CChildFrame	それぞれの子ウィンドウに 1 つずつ	なし	あり	新しい子ウィンドウが開かれたときに、アプリケーションフレームワークが生成

- ・ MDI のフレームウィンドウとビューウィンドウの関係



## 空のドキュメントの作成 CWinApp::OnFileNew 関数

- ・SDI の InitInstance 関数と同様に、MDI の InitInstance 関数は、ProcessShellCommand 関数を通じて、OnFileNew を呼び出す。
- ・このときメインフレームウィンドウは既に生成されている。

OnFileNew は、CMultiDocTemplate の OpenDocumentFile 関数の呼び出しを行ってから、次の手順を実行する

処理	備考
1. ドキュメントオブジェクトを生成	ディスクからのデータ読み込みは行わない
2. 子フレームウィンドウオブジェクト (CChildFrame) を生成	子フレームウィンドウを生成するが、表示しない。IDR_MAINFRAME メニューを IDR_[アプリ名]TYPE メニューで置き換える
3. ビューオブジェクトを生成	ビューウィンドウも生成するが表示はしない
4. ドキュメント、メインフレームウィンドウ、ビューの各オブジェクト間の接続を確立	AddDocTemplate によって確立されるクラス間の関連と混同しないこと
ドキュメントオブジェクト用の OnNewDocument 仮想メンバ関数を呼び出す	
ビューオブジェクトの OnInitialUpdate 仮想メンバ関数を呼び出す	
子フレームオブジェクト用の ActivateFrame 仮想メンバ関数を呼び出す	フレームウィンドウ、ビューウィンドウを表示

## 既存のドキュメントに追加のビューを作成する

メニューから、新規ウィンドウを選択すると、現在選択中のドキュメントにリンクされた新しい子ウィンドウを開く。

処理内容	備考
1. 子フレームオブジェクト (CChildFrame) を生成	子フレームウィンドウを生成するが表示しない
2. ビューオブジェクトを生成	ビューウィンドウも生成するが表示しない
3. 新しいビューオブジェクトと既存のドキュメントとメインフレームオブジェクト間の接続	
4. ビューオブジェクトの OnInitialUpdate 仮想メンバ関数を呼び出す	
5. 子フレームオブジェクトの ActivateFrame 仮想メンバ関数を呼び出し、フレームウィンドウとビューウィンドウを表示	

### ドキュメントのロードと保存

- ・ SDI と同様ドキュメントのロードと保存を行うが、2つの重要な相違点がある
  1. ドキュメントファイルをディスクからロードするたびに新しいドキュメントオブジェクトが生成される
  2. 子ウィンドウが閉じられるとドキュメントオブジェクトも破棄される
- ・ ロードを行う前に、ドキュメント内容の消去の心配はしなくてよいが、SDI との互換性を維持するために、CDocument::DeleteContents 関数はオーバーライドすべき

### 複数のドキュメントテンプレート

- ・ MDI アプリケーションでは、AddDocTemplate 関数を複数回呼び出すことで、複数のドキュメントをサポートすることができる。
- ・ それぞれのテンプレートは異なるドキュメント、ビュー、MDI 子フレームクラスの組み合わせを指定することができる