クラス : wxApp

wxWidgets アプリケーションは、メイン関数を持たない。それと等価なものは、wxApp から継承したクラスで定義される OnInit メンバである。OnInit は通常、最低限、トップウィンドウを作成する。

初期の wxWidgets と異なり、OnInit はフレームを返さない。その代わりに、プロセスを続けるべき (true) か、終了すべき (false) かを示す bool 値を返す。wxWidgets にトップウィンドウを通知するために、wxApp:SetTopWindow を使用する。

特筆すべきは、argc と argv で現されるプログラムのコマンドライン引数は、wxApp メンバ関数により利用可能である。

アプリケーションは、全てのウィンドウを破棄することにより終了する。アプリケーションが終了するためには全てのフレームが破棄されなければならないため、新しいフレームを作成する場合には、できるだけ親フレームを使用することが望ましい。そうすれば、トップレベルフレームを削除するときに自動的に子フレームが削除される。別の方法は、トップレベルフレームの wxCloseEvent ハンドラ内で確実に子フレームを削除することである。

緊急時には、アプリケーションをキルすることができる wxExit 関数がアプリケーションを自動的に終了させる。以下を参照：

アプリケーションの定義例：

```
class DerivedApp : public wxApp
{
public:
    virtual bool OnInit();
};

IMPLEMENT_APP(DerivedApp)

bool DerivedApp::OnInit()
{
    wxFrame *the_frame = new wxFrame(NULL, ID_MYFRAME, argv[0]);
    ...
    the_frame->Show(true);
    SetTopWindow(the_frame);

    return true;
}
```

特筆すべきは、IMPLEMENT_APP(appClass) の使用であり、wxWidgets に対して、wxWidgets 初期化の適切な場所でアプリケーションオブジェクトのインスタンスを自動的に作成することを許可する。以前のバージョンの wxWidgets は、グローバルなアプリケーションオブジェクトの作成を当てにしていたが、その方法はもう推奨されない。なぜなら、要求されたグローバルな初期化がアプリケーションオブジェクトの構築時に実行されない可能性があるからである。

ヘッダファイル内で DECLARE_APP(appClass) を使用すると、アプリケーションオブジェクトへの参照を返す wxGetApp を宣言することができる。他の方法では、wxApp* 型のグローバルなポインタ変数 wxTheApp を使用するしかない。

Application の終了

アプリケーションは通常、最後のトップレベルウィンドウが閉じられた時に終了する。これは通常予期されることであり、プログラムが単一のトップレベルウィンドウを持っている場合には、"終了"メニューコマンドに対応した Close() を呼ぶだけで充分であることを意味する。もしこの動作が望ましくなければ、動作を変更するために wxApp::SetExitOnFrameDelete を呼べばよい。[wxWidgets 2.3.3](#) からは、このようなロジックはプログラムがメインループに入る前に表示されるウィンドウには適用されないことに注意が必要である。言い換えれば、wxApp::OnInit からダイアログを安全に表示することができ、このダイアログ(しばらくの間は、最後のトップレベルウィンドウである)が閉じられるときにアプリケーションが終了することに関して気にする必要がない。

アプリケーションが終了するときの別の局面は、アプリケーションは終了するが [wxWidgets](#) が内部データをクリアする前に OnExit が呼ばれる場合である。OnExit が終了する前に作成した [wxWidgets](#) オブジェクトを全て削除しなければならない。特に、アプリケーションクラスのデストラクタで破棄してはならない。

例えば、以下のコードはクラッシュを引き起こす。：

```
class MyApp : public wxApp
{
public:
    wxCHMHelpController m_helpCtrl;
    ...
};
```

その理由は、m_helpCtrl はメンバオブジェクトであり、MyApp デストラクタで破棄される。しかし、MyApp オブジェクトは、wxCHMHelpController が依存を持つ [wxWidgets](#) データが破棄された後に削除される。この解決方法は、OnExit で HelpCtrl を破棄することである。：

```
class MyApp : public wxApp
{
public:
    wxCHMHelpController *m_helpCtrl;
    ...
};

bool MyApp::OnInit()
{
    ...
    m_helpCtrl = new wxCHMHelpController;
    ...
}

int MyApp::OnExit()
{
    delete m_helpCtrl;
    return 0;
}
```