

BREW & EZアプリ ゲームプログラミング

第2回 BREWモジュール

BREWモジュール

BREWアプリケーションは、実機にダウンロードできるようになるまで時間が掛かるので、開発初期の段階ではSDKに付属のBREWシミュレータで動作確認を行います。BREWシミュレータで実行するBREWアプリケーションの実体は、拡張子がdllのファイルで、これをBREWモジュールといいます。BREWモジュールを実行するには、BREWモジュールに関する基本的な情報を記述したMIFファイルが必要です。MIFファイルはBREWモジュールと同じ名前で作成しなければなりません。たとえば、SampleGame.dllモジュールには、SampleGame.mifというMIFファイルが必要になります。2つのファイルは、必ずペアで扱います。どちらかが欠けていたり、正しい場所に保存されていない場合、そのアプリケーションは起動できません。

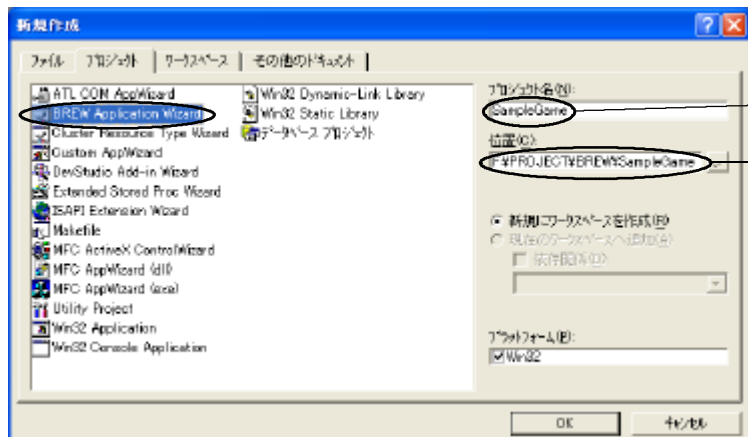
BREWモジュールには、複数のBREWアプリケーションを含ませることができます。このBREWアプリケーションを、正確にはBREWアプレットといい、携帯電話のアプリ一覧に表示されるアイコンひとつひとつに対応します。

BREWアプレットは、クラスIDを持ちます。クラスIDとは、BREWアプレットを識別するための32ビットの値で、BREWアプレットごとに一意に割り当てられます。このIDは、正式にはQualcommやKDDIに申請して発行してもらうものですが、開発の段階ではどんなIDを使用しても構いません。

課題

BREWアプリ用のプロジェクトを作成しましょう。

- (1) BREWアプリのプロジェクトを保存するフォルダを作成しましょう(日本語は使用できません)。このフォルダに、プロジェクトのフォルダとMIFファイルを保存します(MIFファイルは各プロジェクトのフォルダではなく、ここで作成したBREWプロジェクト全体のフォルダに保存します)。
- (2) BREWアプリのプロジェクトを作成します。Visual C++を起動しましょう。
- (3) メニューの「ファイル(F)」 「新規作成(N)」を選択しましょう。新規作成ダイアログが表示されます。
- (4) 「プロジェクト」タブの「BREW Application Wizard」を選択した状態で、「位置(C)」に(1)で作成したフォルダを設定します。その後、「プロジェクト名(N)」を入力しましょう。



プロジェクト名を入力(英数字と_のみ)

「BREWアプリのプロジェクトを保存するフォルダ」+「プロジェクト名」であることを確認

正しく設定できたら、OKボタンをクリックします。

(5) 「BREW Application Wizard - ステップ 1 / 2」が表示されます。



ここでは、BREWアプリケーションで使用する機能を選択します。以下のような機能があります。

項目名	機能
File	ファイル入出力
Network	ネットワーク通信
Database	データベース
TAPI/SMS	電話制御 / ショートメッセージ
Sound	サウンド

使用する機能を選択すると、機能を提供するAPIのヘッダーファイルが、生成されるソースコードにインクルードされます。なお、いつでも手動でインクルードすることができるので、ここで設定しなくてもかまいません。

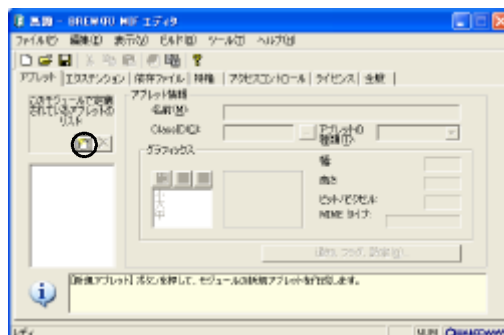
今回は、いずれの機能も選択しないで、そのまま次へボタンをクリックします。

(6) 「BREW Application Wizard - ステップ 2 / 2」が表示されます。



ここでは、MIFファイルを作成することができます。中央の「MIF Editor」ボタンをクリックしましょう。

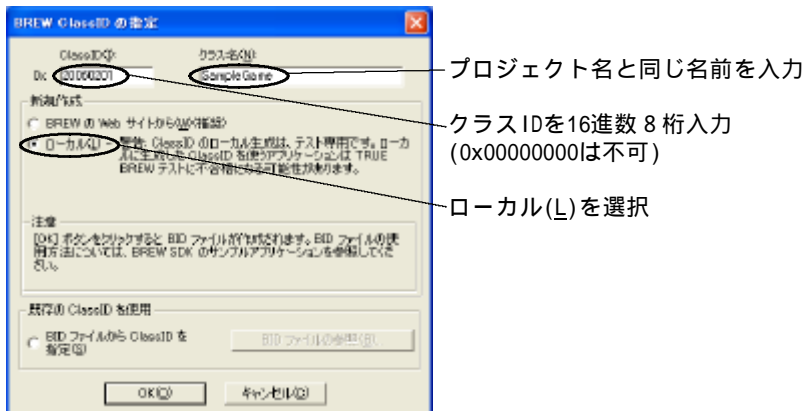
(7) MIFエディタが起動します(起動しない場合もあります。そのときは手動で起動してください)。



ここでは、MIFファイルを作成することができます。「新規アプレット」ボタンをクリックします。

(8) 「ClassIDの生成」ダイアログが表示されます。

ここでは、クラスIDを作成することができます。「名前(N)」にプロジェクト名と同じ名前を入力し、「ローカル(L)」を選択します。「ClassID(I)」に適当な値を16進数で8桁入力し、「生成(G)」ボタンをクリックします。「BREW ClassID をローカルに生成してよろしいですか?」というメッセージが表示されるので、「はい」を選択します。



(9) bidファイルの保存ダイアログが表示されるので、(4)で作成したプロジェクトのフォルダに保存しましょう。

(10) MIFエディタに戻ります。ここでは、作成するアプリケーションの名前や種類、アイコンを設定します。

ClassIDが設定されていることを確認します。設定されていない場合はBIDファイルの参照ボタンをクリックし、(9)で作成したbidファイルを選択します。

「名前(M)」にアプリケーションの名前を入力します。プロジェクト名と同じでなくてもかまいません。ここで設定した名前が携帯電話のアプリ一覧に表示されます。ただし、日本語は正しく表示されません。

「アプレットの種類(I)」を設定します。

アイコンを設定する場合は、「アイコン(I)」にアイコンとなる画像のファイルを設定します。

モジュールに複数のアプレットを含める場合は、(7)から(10)を繰り返します。

(11) MIFファイルを保存します。メニューの「ファイル(F)」 「名前を付けて保存(A)」を選択し、「プロジェクト名.mif」(BREW MIF v 2 ファイル)という名前で、(1)で作成したフォルダに保存しましょう。

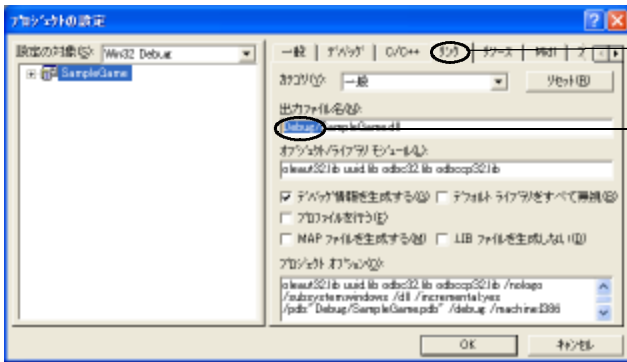
(12) MIFエディタを終了してVisual C++に戻り、「BREW Application Wizard - ステップ 2 / 2」ダイアログの終了ボタンをクリックしましょう。プロジェクトの情報が表示されるので、OKボタンをクリックすると、ソースファイルが自動生成されます。

(13) プロジェクトの設定を行います。

ビルドされるモジュールファイル(.dll)は、(4)で作成したプロジェクトのフォルダ直下になければ実行できません。Visual C++では、DebugフォルダやReleaseフォルダにファイルがビルドされるので、プロジェクトの設定を変更します。

メニューの「プロジェクト(P)」 「設定(S)」(VC++2002以降はプロジェクト(P) プロパティ(P))を選択してプロジェクトの設定ダイアログを表示し、リンクタブをクリックすると、リンク時の設定が行えます。

「出力ファイル名(N)」の「Debug/」や「Release/」を削除しましょう。



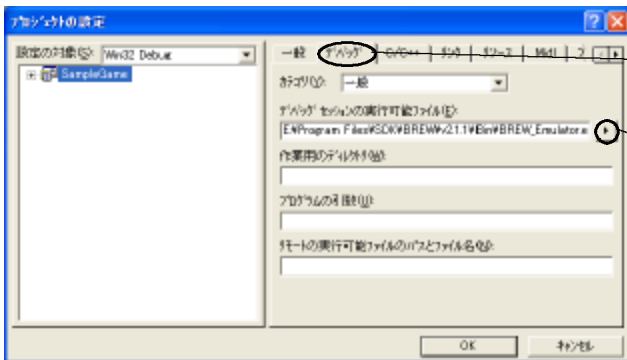
リンクをクリック

「Debug/」や「Release/」を削除

(14) 実行ファイルの設定を行います。

BREWモジュールファイルなどの拡張子がDLLのファイルは、アプリケーションから呼び出される関数を集めたもので、そのままでは実行できません。Visual C++でも、DLLファイルを直接実行できないので、代わりにBREWシミュレータを起動させ、そこからDLLファイルを呼び出すようにします。

(13)のプロジェクトの設定ダイアログのデバッグタブをクリックすると、実行ファイルの設定が行えます。「デバッグセッションの実行可能ファイル(E)」のボタン(VC++2002以降は構成 - デバッグ - コマンドラインの参照)をクリックし、シミュレータの実行ファイル「BREW_Simulator.exe」を選択します。このファイルは、BREW SDKをインストールしたフォルダの「bin」フォルダにあります。



デバッグをクリック

をクリックして「BREW_Simulator.exe」を指定

(15) プログラムを実行します。Shift+F5キーを押し、BREWエミュレータを起動しましょう。

BREWシミュレータの初回起動時は、古い機種で起動します。機種を変更するには、BREWシミュレータのメニューから「ファイル(F)」 「デバイスの読み込み(L)」を選択し、「A5501T.qsc」を開きます。

機種が変更されたら、実行したいアプリケーションを左右で選んで選択(Enter)キーをクリックすると、プログラムが起動します。

(16) デバイスの設定を変更します。

シミュレータの画面が右下にずれている場合があります。これは、機種を定義する「BREWデバイスコンフィギュレータドキュメントファイル」(拡張子.qsc)の設定が正しくないためです。一部の機種の実行可能ファイルでは、このように、画面やボタンの設定が若干ずれている場合があります。

デバイスの設定は、BREW SDK 2.1に付属の「BREWデバイスコンフィギュレータ」というツールで編集するか、テキストエディタで直接編集することにより、自由に変更することができます。

A5501T.qscをテキストエディタで開き、88行目を以下のように変更しましょう。

誤 : SCREEN 67 167 307 465 AVS_SCREEN_0 240 298 0.000000 0.000000 INCH 16 1

正 : SCREEN 66 163 306 461 AVS_SCREEN_0 240 298 0.000000 0.000000 INCH 16 1