

Fine Kernel ToolKit システム (Visual Studio 2010/2012/2013 版) セットアップマニュアル

FineKernel Project
(2014 年 5/30 版)

1 Visual Studio のバージョンについて

Fine Kernel ToolKit システム (以下 FK) は Visual Studio(以下 VS) 2010 と 2012 と 2013 上において、ほとんど同様の操作で開発を行うことができる。本マニュアルではこれらのバージョンについてまとめて解説を行っているが、メニューの項目名などが微妙に異なる場合がある。その場合は適宜読み替えること。

また VS には、通常の製品版の他に無償で使用できる Express Edition が存在するが、こちらのバージョンも問題なく使用できる。Express Edition は次の URL からダウンロードとインストールが可能である。

<http://www.microsoft.com/visualstudio/jpn/downloads>

2010 の場合は、Visual C++ 2010 Express を、2013 の場合は、Visual Studio 2013 Express for Windows Desktop を選択すること。2012 の Express Edition は現在上記の URL からは入手不能であるが、次の URL から入手できる。

<http://www.microsoft.com/ja-jp/download/details.aspx?id=34673>

2 FK システムのセットアップ

VS のインストールが完了したら、FK のセットアップを以下の手順で行う。

1. 以下の URL から、FK ToolKit VisualStudio 用インストーラの項目にあるファイルをダウンロードする。

<http://sourceforge.jp/projects/fktoolkit/releases/>

2010 版は「FK_VC10」、2012 版は「FK_VC12」、2013 版は「FK_VC13」から始まるファイル名になっているので、必要なものを選択すること。

2. インストーラを起動し、指示に従ってインストールする。標準の設定では、2010 版は“C:\FK_VC10”、2012 版は“C:\FK_VC12”、2013 版は「FK_VC13」という名称のフォルダにインストールされる。

セットアップ完了後に、VS への設定情報追加が行われる。環境によってはこの処理がうまく完了しない場合があるため、その場合は手動での設定が必要となる。詳細は本マニュアルの補足の項を参照すること。

3 プロジェクトの作成方法

VS では、アプリケーションを作成する際には“プロジェクト”を作成する必要がある。開発には単なるプログラムソースだけでなく、様々な開発用の設定ファイルが必要となるが、“プロジェクト”はそれらを全て引くくめたものだと考えれば良い。

通常、プロジェクトには様々な設定を行う必要があるが、FK のセットアップが正常に完了した場合は、「プロジェクトウィザード」を利用することで簡単に設定を済ませることができる。以下にその手順を述べる。

1. VS を起動し、メニューから「ファイル」「新規作成」「プロジェクト」を選択する。

2. 表示されたダイアログのうち、左側の「プロジェクトの種類」というツリーメニューで「Visual C++」というツリーを選択し、さらにその中の「FK」というツリーを選択する。
3. 右側の「テンプレート」から「FK ToolKit Project」を選択する。
4. 「プロジェクト名」というテキストボックス中に適当なプロジェクト名を入力する。
5. 「場所」に適当な場所を設定しておく。この場所にプロジェクトのフォルダを配置し、プログラムソースを保管することになるので、きちんと場所を把握しておく必要がある。
6. 上記の全ての入力を終わったら「OK」を押す。すると「FK ToolKit プロジェクトウィザードによるこそ」というダイアログが開くので、初期状態で生成したいソースコードの種類を選択して、「完了」ボタンを押す。
7. 作成したプロジェクトが開いたら、ツールバー内の「Debug」という項目が選択されているプルダウンメニューを「Release」に変更する。

この手順で、“プロジェクト”の作成と設定が完了する。実際にはプロジェクト自体に加えて、その上位に位置する“ソリューション”と呼ばれるファイルも同時に作成される。VS で作成したプロジェクトを開く場合は、ソリューションを通じて行う。

4 コンパイル(ビルド)

ソースファイルが作成できたらコンパイルを行い、実行ファイル(exe)を生成する。VS においては、コンパイルを初めとした一連の作業のことを“ビルド”と呼ぶ。ビルドを行うには、メニュー中の「ビルド」「ソリューションのビルド」を選択する。

エラーが発生した場合はメッセージを確認し、ソースファイルにおけるエラーの場合はそれを修正する必要がある。ヘッダファイルのインクルードや、リンクの時点でエラーが発生している場合は、これまでの設定に何らかの不備がある可能性が高いので、もう一度これまでの手順を確認すること。

5 実行

ビルドが正常に終了した場合、プロジェクトファイル(.vcxproj)と同じフォルダ内にある.exe フォルダ中に実行ファイルが生成される。実行するには、メニューの「デバッグ」「デバッグなしで開始」を選択する。

サウンド機能を利用したプログラムの場合、起動時に DLL が見つからないというメッセージが出て起動できないことがある。その場合はライブラリインストール先の「redist\oainst.exe」を実行し、bin フォルダ内の libogg.dll, ligvorbis.dll, libvorbisfile.dll を実行ファイルと同じフォルダにコピーすることで起動できる。

6 プロジェクトの保存とロード

プロジェクト全体の状態を保存するには、メニューの「ファイル」「すべてを保存」を実行する。これにより、全てのソースファイルとプロジェクトの情報が保存される。

保存したプロジェクトをロードするには、「.sln」を拡張子に持つファイルを開くか、VS を起動後メニュー中の「ファイル」「ソリューションを開く」を選択し、「.sln」を拡張子に持つファイルを選択すればよい。

7 補足

以下はプロジェクトの設定に関する補足である。前節までで一通りの解説は済んでいるので、本節以降は必要に応じて参照すること。

7.1 ファイル読み込み処理を行うプログラムを実行する場合

プログラム上から画像や形状データなどを読み込むプログラムを作成した場合、カレントフォルダはプロジェクトファイルの存在しているフォルダになるため、パスの設定に注意が必要である。fk_System::setcwd() 関数をプログラム中で使用すると、カレントフォルダを exe ファイルと同じフォルダに変更することができるので、main() 関数の冒頭でコールしておくとう便利である。詳細はクラスリファレンスを参照すること。

7.2 デバッグビルドとリリースビルドについて

FK を使用したプログラムは、通常リリースビルドで実行することを推奨する。プロジェクト作成後の初期状態ではデバッグビルドになっているが、この状態ではプログラムの処理速度が大幅に低下する。開発を行っていくうち、複雑なバグなどが発生した場合にはデバッグビルドで問題を検証するのが有効であるので、状況に応じて切り替えるとよい。

7.3 ライブラリのパス設定

本節の内容は、基本的には手動で設定を行う必要はない手順である。しかし何らかの原因でセットアップ時の設定変更がうまく行かなかったり、これらの設定が初期化されてしまった場合には、以下の手順で再設定すること。なお、この手順ではインストール先が C:\FK_VC10 であるとして記述してあるが、自分で設定したインストール先に応じて読み替えて入力すること。

1. ビルドしたいソリューションを開き、ソリューションエクスプローラ中のソリューション直下にあるプロジェクトを右クリックし、「プロパティ」を開く。
2. ダイアログの左側にあるツリーメニューの中の「構成プロパティ」というツリーを展開し、出てくるリストのうち「VC++ ディレクトリ」を選択する。
3. 「インクルードファイル」をクリックして、同じ行の右側にある ボタンを押し、「編集...」ボックスをクリックする。
4. ダイアログ右上にある 4 個のボタンのうち、一番左側のボタン (フォルダのアイコンが表示されているボタン) を押す。
5. カーソルが下に現れるので、「C:\FK_VC10\include」と入力して Enter キーを押し、「OK」ボタンを押してダイアログを閉じる。
6. 次に、「ライブラリディレクトリ」をクリックして、同じ行の右側にある ボタンを押し、「編集...」ボックスをクリックする。
7. 先ほどと同様の手順で編集ボックスを開いて、カーソルが下に現れたら、「C:\FK_VC10\lib」と入力して Enter キーを押す。
8. 「OK」ボタンを押してダイアログを閉じる。

2012,2013 版のインストーラーでは、FK のインストール先を環境変数 “FK_VC12_DIR” または “FK_VC13_DIR” に記録している。上記の手順でパスを設定する際、2012 ならば C:\FK_VC12 に相当する部分を、\$(FK_VC12_DIR) と表記することで、他の PC やユーザーの環境においても、適切にライブラリを参照できる。編集ボックスの「マクロ」ボタンを押すと “FK_VC12_DIR” にどのパスが記録されているか確認できる。環境変数が登録されていないか、正常なパスが登録されていない場合は、直接パスを設定するか、手動で環境変数を設定する必要がある。

7.4 プロジェクトの設定項目について

通常、プロジェクトはウィザードを利用することで簡易に作成できるが、他のシステムと共存させる場合や、ウィザードが利用できない場合は、以下の設定項目を参考にしてプロジェクトの設定を調整すること。手動で新規にプロジェクトを作成する場合は「Win32 アプリケーション」を選択し、ウィザード内の「空のプロジェクト」にチェックを付けて作成したものを設定変更するとよい。

- 構成プロパティ

- 全般

- * 文字セット : 「マルチバイト文字セットを使用する」
- * 共通言語ランタイム サポート : 「共通言語ランタイム サポートを使用しない」

- C/C++

- コード生成

- * ランタイムライブラリ : [Release] 「マルチスレッド (/MT)」, [Debug] 「マルチスレッド デバッグ (/MTd)」

- リンカ

- 入力

- * 追加の依存ファイル : 項目をクリックすると右側にボタンが現れるのでそれを押し、表示されるダイアログのテキストエリア中に次の文字列を入力する。

```
FK2_audio.lib FK2_util.lib FK2_base.lib
FK2_ftkWin.lib FK2_ftkErr.lib
ftkgl.lib ftkjpeg.lib ftkpng.lib
ftkimages.lib ftkforms.lib ftk.lib ftkz.lib
libvorbisfile.lib libvorbis.lib libogg.lib OpenAL32.lib
imdckv.lib freetype2MT.lib opengl32.lib glu32.lib imm32.lib winmm.lib wsock32.lib comctl32.lib
```

デバッグビルドの場合は、FK2 で始まるライブラリ名の末尾に `.d` を付ける。FK2_base.lib ならば FK2_base.d.lib となる。同一内容が最初にインストール先のフォルダ下の「doc\Lib_Release.txt」(デバッグモードの場合は「doc\Lib_Debug.txt」) というテキストファイルに記述してある。

- * 特定のライブラリの無視 : [Release] 空欄, [Debug] 「LIBCMT」

- システム

- * サブシステム : コンソールウィンドウを表示させたい場合は「コンソール」を、表示させたくない場合は「Windows」を選択する。