

C/C++開発環境の構築

ABHB の開発言語は処理スピードの最速化の為に C 言語としています。下記の手順で開発環境を構築します。

1. Visual Studio 2022 を Microsoft のサイトからダウンロード、インストール
2. <https://abhb.forgevision.com/> から ABHB_download_kit_vers1.0.0 をダウンロード、インストール
3. ABHB を使うための共通部分のプロジェクトを新規作成して、正常にビルドできることを確認します。
 - ① Visual Studio 2022 を起動して、新規作成⇒プロジェクト
 - ② C/C++コンソールアプリを選択⇒次へ (図 1_2_1)
 - ③ プロジェクト名を"ABHB_basic_form"、場所は任意の場所を指定、“新しいソリューションを作成する”、ソリューション名はデフォルト、“ソリューションとプロジェクトを同じディレクトリに配置する”のチェックボックスは外す ⇒作成 (図 1_2_2)
 - ④ インストールした ABHB フォルダの"src"フォルダ内の¥Interface_2024.9月号_Cソースコードファイル¥C ソースコード 1_5.txt の全文字列をコピーして、ABHB_basic_form.cpp のエディターに上書きコピー (図 1_2_3)
 - ⑤ 上部のメニューバーの下の"Debug"を"Release"に変更、“x64”はそのまま (図 1_2_4)
 - ⑥ 上部付近にある文字列"_CRT_SECURE_NO_WARNINGS"をコピーしておく (図 1_2_4)
 - ⑦ 右のソリューションエクスプローラーの空白部分を右クリックしてプロパティーを選択 (図 1_2_4)
 - ⑧ 構成プロパティー/詳細 文字セットを“マルチバイト文字セットを使用する”に変更 (図 1_2_5)
 - ⑨ C/C++ / 全般 追加のインクルードディレクトリを編集 ABHB フォルダの"include"フォルダを指定⇒適用 (図 1_2_6)
 - ⑩ C/C++ / プリプロセッサ プリプロセッサの定義を編集 文字列末尾で改行⇒文字列"_CRT_SECURE_NO_WARNINGS"をペースト⇒適用 (図 1_2_7)
 - ⑪ リンカー / 全般 追加のライブラリディレクトリを編集 ABHB フォルダの"lib"フォルダを指定⇒適用 (図 1_2_8)
 - ⑫ リンカー / 入力 追加の依存ファイルを編集 “ABHB.lib”の文字列をインプット⇒適用⇒OK (図 1_2_9)
 - ⑬ メニューバーの ビルド / ソリューションのビルド を選択してビルド実行。③で指定した場所/x64/Release/に ABHB_basic_form.exe が生成されていることを確認
4. 下記の手順で実行してみます
 - A) Windows の標準アプリ コマンドプロンプトを立ち上げ
 - B) 上記⑬で確認した exe があるディレクトリにチェンジディレクトリ

引数 1 : unsigned char** gRed	クエリ画像の Red 2 次元配列
引数 2 : unsigned char** gGreen	クエリ画像の Green 2 次元配列
引数 3 : unsigned char** gBlue	クエリ画像の Blue 2 次元配列
引数 4 : unsigned char** gBuff_red	処理用/結果画像の Red 2 次元配列
引数 5 : unsigned char** gBuff_green	処理用/結果画像の Green 2 次元配列
引数 6 : unsigned char** gBuff_blue	処理用/結果画像の Blue 2 次元配列
引数 7 : int width_pixel	横ピクセル数
引数 8 : int height_pixel	縦ピクセル数

“main_process”は必要に応じて自由に引数を追加設定できます。

- ⑤ 結果画像を書きだします。”ABHB_bmp_read_write()”を第 1 引数 = 1 で呼び出します。

引数 1 : int mode	1 : 書き出し
引数 2 : char* file_name_dump	書き出しファイル名 (フルパス)
引数 3 : unsigned char** gBuff_red	書き出し画像の Red 2 次元配列
引数 4 : unsigned char** gBuff_green	書き出し画像の Green 2 次元配列
引数 5 : unsigned char** gBuff_blue	書き出し画像の Blue 2 次元配列
引数 6 : int* width_pixel	横ピクセル数ポインタ
引数 7 : int* height_pixel	縦ピクセル数ポインタ

(引数 6 と 7 は、書き出す画像の横縦ピクセル数を格納して呼び出します)

- ⑥ 確保したメモリを解放、実行時間を表示してリターンします