

# LabVIEW で DLL を作り、Visual C++ 6.0 で読み込む方法

このページで紹介しているリソースは ZIP 形式の圧縮ファイルとして添付されています。解凍ソフトを使って解凍してお使いください。

コンテンツ:

- [LabVIEW での DLL の作成方法](#)
- [DLL を Visual C++ 6.0 で利用する方法](#)

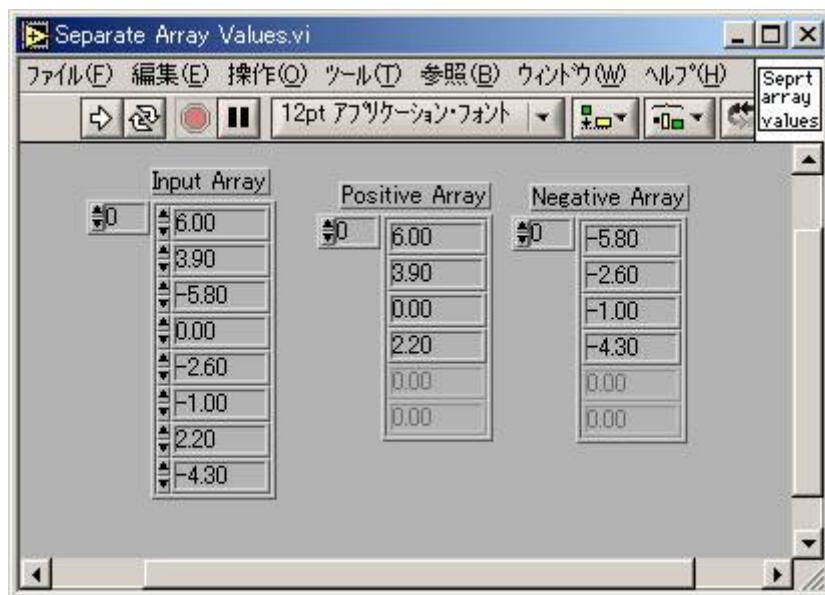
## LabVIEW での DLL の作成方法

### LabVIEW での DLL の作成方法

1. C:\arraydll フォルダを作成して、全てのファイルを置いておきます。

Separate Array Values.vi を LabVIEW で開き、実行してみます。

2. 下記の図のように、入力された配列には正負の値が含まれていますが、正のみを含む配列と負のみを含む配列が表示される結果となります。これを DLL 化していきます。

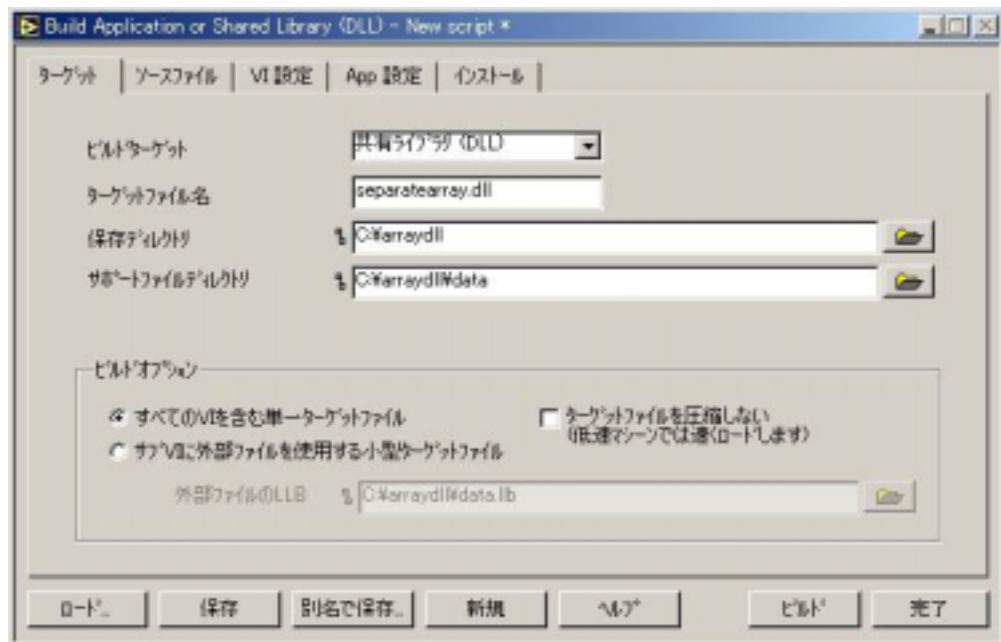


3. Separate Array Values.vi を保存して閉じて、新規 VI だけを開いた状態にします。

4. 下記の図のように「DLL 作成」を選択します。



下記の図のように設定をします。



5. ソースファイルのタブを選択し、「Add Exported VI」をクリックして、Separate Array Values.vi を選択します。このとき、下記のようなウィンドウが開きます。



ここでは、「標準呼び出し規約」を選択します。

6. Separate Array Values.vi では、入力配列 1 つと出力配列の 2 つの入出力があるので、「+」を 3 回クリックすると、下記のようなウィンドウになります。



7. 下記の図のように NegativeArray の下の\_を選択して、VI名を len にして下さい。



8. 下記の図のように、InputArray を選択して、長さ入力を len に選択して下さい。



9. 図のように、PositiveArray を選択して、長さ入力を len に、長さ出力を(なし)を選択して下さい。

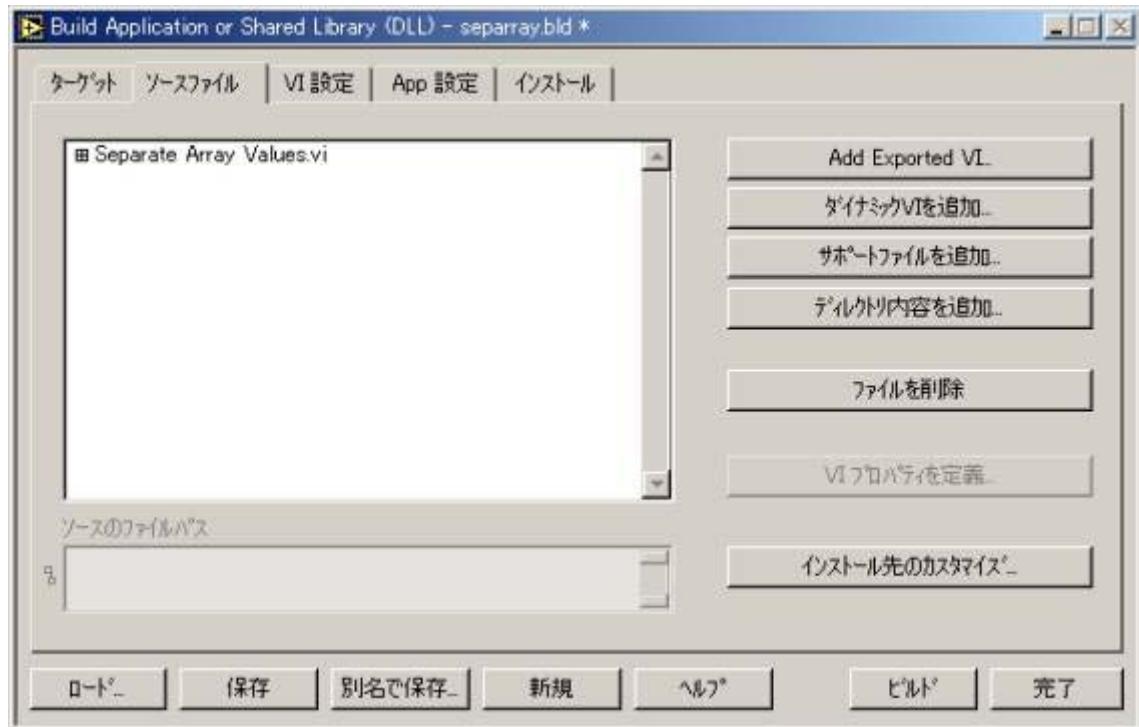


10. NegativeArray を選択して、長さ入力を len に、長さ出力を(なし)を選択して下さい。



11.OK をクリックして下さい。

12.「ビルド」をクリックします。



ビルドが終了したら、完了を押して、LabVIEW を閉じます。

(もし、ビルドに成功しないなどの不具合が生じた場合は、上記の図の「ロード」を押して、separamarray.bld を選択すれば、ここで説明されているビルドの条件が整えられますので、お試しください。)

これで、C:\arraydll\separatearray.dll と separatearray.h と separatearray.lib ファイルが作成されました。

### DLL を Visual C++ 6.0 で利用する方法

DLL を Visual C++ 6.0 で読み込みます。

1. C:\arraydll\separamarray.dsw を Visual C++ 6.0 で開きます。

下記の図のように FileView の中には、SourceFiles として separamarray.c、HeaderFiles として separatearray.h、ResourceFiles として separatearray.lib ファイルがあるはずです。

The screenshot shows the Microsoft Visual Studio interface. The title bar says "separarray - Microsoft Visual C++ - [main.c]". The menu bar includes "ファイル" (File), "編集" (Edit), "表示" (View), "挿入" (Insert), "プロジェクト" (Project), "ツール" (Tools), "ヘルプ" (Help). The toolbar has icons for New, Open, Save, Find, Replace, and others. The Solution Explorer on the left shows the project structure:

- ワーカーの「separarray」
- separarray ファイル
  - ソース ファイル:
    - separarray.c
  - ヘッダー ファイル:
    - separatearray.h
  - リソース ファイル
  - separatearray.lib
  - 外部依存関係

The code editor displays the file "main.c" with the following content:

```
#include "separatearray.h"
#include "stdio.h"

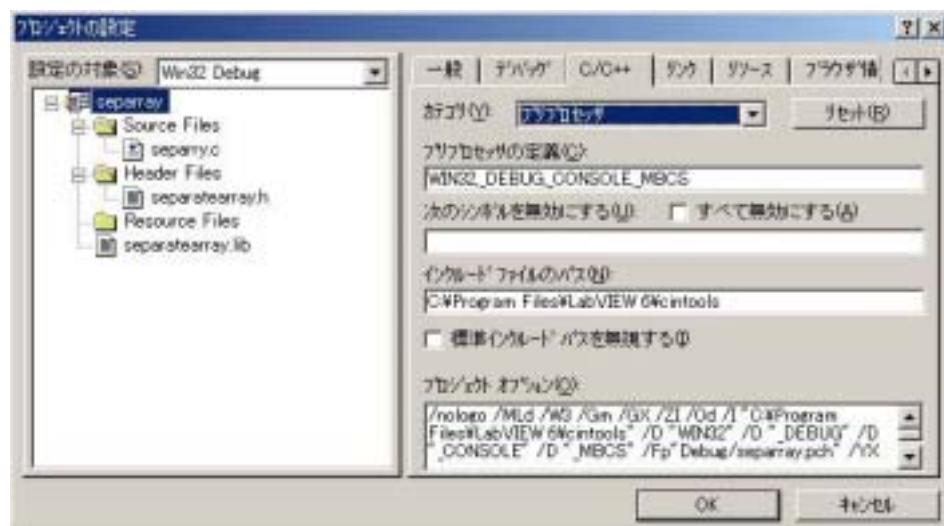
float64 inputarray[10] = [-5.46, -5.39, 3.32, 2.32, 4.3
float64 positivearray[5];
float64 negativearray[5];
int len1 = 10;
int len2 = 5;
int len3 = 5;
int i;

int main(int argc, char *argv[])
{
    SeparateArrayValues(inputarray, positivearray, negativearray);

    printf("The values you entered are:\n");
    for (i = 0; i<10; i++) {
        printf("%f ", inputarray[i]);
    }
    printf("\n");
    printf("The positive values you entered are:\n");
    for (i = 0; i<5; i++) {
        printf("%f ", positivearray[i]);
    }
}
```

## 2. FileView の separarray ファイル上で右クリックして、「設定」を選択します。

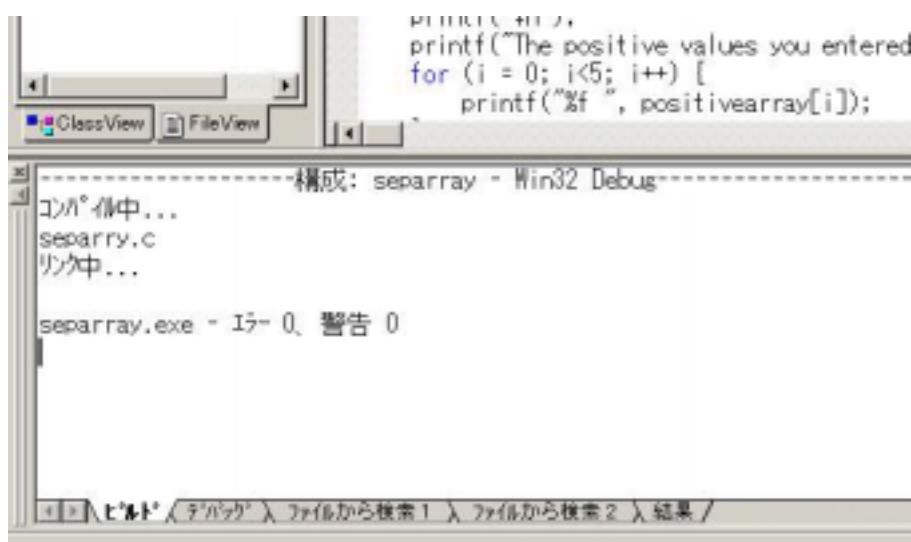
下記のようなプロジェクトの設定が現れますので、「C/C++」のタブを選択して、カテゴリを「プリプロセッサ」にします。



インクルードファイルパスの設定は、LabVIEW6\cintools\の中の extcode.h ファイルがあるディレクトリ名までのパスを入れてください。ここでは、C:\Program Files\LabVIEW 6\cintools としています。

- OK を押して、Separray.exe にビルドしてください。

下記のようにエラーなし、警告なしになるはずです。Separray.exe が作成されました。



(もし、ビルドがうまく出来ない場合は、ZIP ファイルの中にある separatearray.dll や separatearray.h や separatearray.lib を使用して、お試しください。)

- Separray.exe の動作の確認をします。



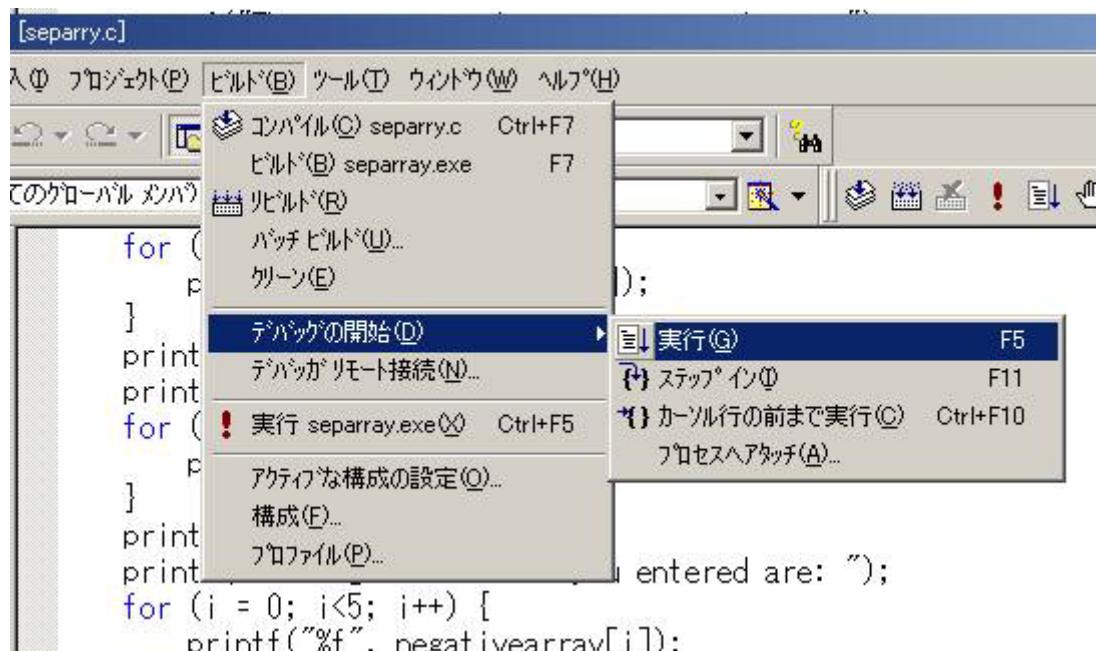
上記の図のように、c ファイルの return 0; をブレークポイントとして設定します。

5. 下記の図のように、ブレークポイントが出来ます。

```
    }
    printf("\n");
    printf("The positive values you entered are: ");
    for (i = 0; i<5; i++) {
        printf("%f ", positivearray[i]);
    }
    printf("\n");
    printf("The negative values you entered are: ");
    for (i = 0; i<5; i++) {
        printf("%f ", negativearray[i]);
    }
    printf("\n");
}

return 0;
}
```

6. デバッグモードで実行します。



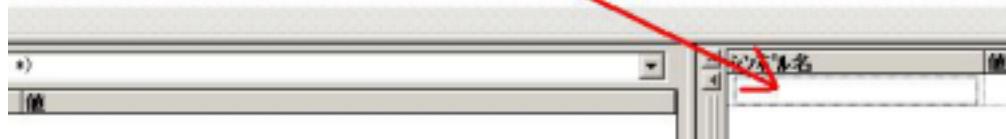
7. LabVIEW の dll に対する入力データである inputarray をシンボル名にすれば、この inputarray の内容を確認できます。

```

lues you entered are:");
for( i=0; i<10; i++) {
    cout << "Value at index " << i << ": " << inputarray[i] << endl;
}

cout << "Positive values you entered are: ";
for( i=0; i<10; i++) {
    if( inputarray[i] > 0) {
        cout << "Value at index " << i << ": " << positivearray[i] << endl;
    }
}

```



8. 同様に、positivearray、negativearray も確認できます。下記の図に、結果を示しています。

シンボル名	値
inputarray	0x00426a30 _inputarray
[0]	-5.46000000000000
[1]	-5.39000000000000
[2]	3.32000000000000
[3]	2.32000000000000
[4]	4.33000000000000
[5]	-7.32000000000000
[6]	-6.33000000000000
[7]	7.23000000000000
[8]	-9.34000000000000
[9]	4.35000000000000
positivearray	0x00427c20 _positivearray
[0]	3.32000000000000
[1]	2.32000000000000
[2]	4.33000000000000
[3]	7.23000000000000
[4]	4.35000000000000
negativearray	0x00427c60 _negativearray
[0]	-5.46000000000000
[1]	-5.39000000000000
[2]	-7.32000000000000
[3]	-6.33000000000000
[4]	-9.34000000000000

入力された配列 `inputarray` には正負の値が含まれていますが、正のみを含む配列 `positivearray` と負のみを含む配列 `negativearray` が実行されています。

**Note:**Visual C++ 6.0 は、マイクロソフト社の商標であります。弊社では Visual C++ 6.0 に対するビルドの方法、結果を表示するインターフェースの作成等を含むプログラミングの方法についてのサポートは一切行なっておりません。

これらの記述内容は、あくまでも参考でありますので、使用しているプログラミング環境によっては変更が必要な場合がありますことをご理解ください。