

## Пример создания библиотеки DLL и её использования в консольном приложении на Visual C++ 2005 Express Edition.

По материалам обсуждения на форуме [dxdy.ru](http://dxdy.ru) и справочным материалам Microsoft.

Руслан Шарипов.

### Создание библиотеки DLL.

ГАА в [сообщении](#) #1075716 написал:

И в качестве примера создания DLL на C по сути то же: [Создание и использование библиотеки DLL \(C++\)](#).

Спасибо. Ссылка очень ценная. Я решил следовать ей, но натолкнулся на препятствие. Пришлось искать другую ссылку: [Walkthrough: Creating and Using a Dynamic Link Library](#). По сути, это то же самое, но здесь сказано, как обойти препятствие.

Итак, что у меня есть на начало действий. Это скачанный и установленный Visual C++ 2005 Express Edition на платформе Win XP 2002 SP 3 с 32-х разрядным процессором Intel Pentium 4. Первое действие — в меню **File** выбрать **New** и далее **Project**. К счастью у меня установлен англоязычный вариант программы, так что нет мороки с поиском соответствия между исходными и переведёнными на русский язык пунктами в меню, в закладках и прочих местах. Успешно выполняю первое действие.

Открывается диалоговое окно с двумя панелями — левой и правой. На левой панели деревоподобная структура. Из корня, обозначенного **Visual C++**, растут вниз три ветви **CLR**, **Win32**, и **General**. Выбираю **Win32**. Это второе действие.

Такому выбору на правой панели соответствуют две опции: **Win32 Console Application** и **Search Online Templates**. Выбираю консольное приложение. Это третье действие.

Внизу под двумя панелями (левой и правой) три поля. Они называются **Name**, **Location** и **Solution Name**. В поле **Name** пишу **MathFuncsDll**, в поле **Solution Name** пишу **DynamicLibrary**. Названия можно выбирать свои, но я строго слеую наставлениям от Microsoft. В поле **Location** уже сделан выбор папки:

```
C:\Documents and Settings\User\My Documents\Visual Studio 2005\Projects
```

Не препятствую такому выбору, но открываю эту папку, чтобы посмотреть её содержимое. В ней пусто. Кстати, слово **User**, означает имя пользователя, с которым я вошёл в Windows. У вас оно будет своим. Это было четвёртое действие.

После нажатия кнопки **OK** взамен первого открывается новое диалоговое окно с названием **Win32 Application Wizard**. На нём нажимаю кнопку **Next**. Это было пятое действие.

Диалоговое окно сменяется новым. В нем выбор различных опций. В частности имеются опции под заголовком **Application type**. Среди них опция **DLL**. Но она у меня не активна. Высвечена серым и не может быть выбрана. В этом и была проблема, о которой я упоминал в самом начале. Но Microsoft говорит, что в некоторых версиях **Visual Studio** такое может иметь место. Взамен **DLL** рекомендуется выбрать опцию **Console application**. Это было шестое действие.

Под заголовком **Additional options** в том же диалоговом окне выбираем опцию **Empty project**. Это было седьмое действие. Восьмое действие состоит в нажатии кнопки **Finish**. После закрытия диалогового окна, в основном окне **Visual C++** видим три панели левая, правая и нижняя. Левая называется Solution Explorer. На правой панели видим вкладку Start Page. Её содержание больше напоминает рекламу.

На левой панели отобразилась деревоподобная структура. В её корне видим Solution 'DynamicLibrary' (1 project). От корня вниз растёт ветка с названием **MathFuncsDll**. Это ровно те названия, которые мы ввели при создании проекта. От ветки **MathFuncsDll** вниз растут три ветки, имеющие вид папок. Их названия **Header Files**, **Resource Files** и **Source Files**. Замечаем, что в реальной папке

```
C:\Documents and Settings\User\My Documents\Visual Studio 2005\Projects
```

появилась папка с названием `DynamicLibrary`. Внутри нее — служебные файлы. Но других папок в ней нет.

Следующие действия направлены на создание библиотеки DLL. Они опять нумеруются, начиная с первого. На левой панели, которая называется `Solution Explorer`, кликаем правой кнопкой мыши по изображению папки **Header Files** и выбираем **Add**, затем **New Item**. Открывается диалоговое окно с двумя панелями — левой и правой. Левая панель называется **Categories**. На ней из корня **Visual C++** растут четыре выбора: **UI**, **Code**, **Data**, **PropertySheets**. Выбираем **Code**. Правая панель называется **Templates**. На ней выбираем опцию **Header File (.h)**. Внизу в диалоговом окне вводим имя файла **MathFuncsDll.h** и нажимаем кнопку **Add**. Это было первое действие.

После выполнения этого действия на левой панели, которая называется `Solution Explorer`, отображается имя файла **MathFuncsDll.h** в форме ветки, растущей из папки **Header Files**. На правой панели вкладка `Start Page` закрывается и рядом появляется вкладка **MathFuncsDll.h**, внутри которой отображается текстовый редактор с пустым содержимым созданного файла **MathFuncsDll.h**. При этом в реальной папке

```
C:\Documents and Settings\User\My Documents\Visual Studio 2005\Projects\DynamicLibrary
```

появилась вложенная папка `\MathFuncsDll`, а внутри нее служебные файлы и пустой файл с именем **MathFuncsDll.h**.

В окно текстового редактора копируем и вставляем следующий текст:

```
// MathFuncsDll.h
namespace MathFuncs
{
    class MyMathFuncs
    {
    public:
        // Returns a + b
        static __declspec(dllexport) double Add(double a, double b);

        // Returns a - b
        static __declspec(dllexport) double Subtract(double a, double b);

        // Returns a * b
        static __declspec(dllexport) double Multiply(double a, double b);

        // Returns a / b
        // Throws DivideByZeroException if b is 0
        static __declspec(dllexport) double Divide(double a, double b);
    };
}
```

После этого нажимаем иконку с дискетой для того, чтобы сохранить файл. Это будет второе действие.

В тексте сохранённого файла замечаем директивы `__declspec(dllexport)`. Они означают, что создаваемая библиотека будет экспортировать четыре метода из создаваемого класса `MyMathFuncs`. Это замечание и есть третье действие.

На левой панели, которая называется `Solution Explorer`, кликаем правой кнопкой мыши по изображению папки **Source Files** и выбираем **Add**, затем **New Item**. Вновь открывается диалоговое окно с двумя панелями — левой и правой. Левая панель называется **Categories**. На ней из корня **Visual C++** растут четыре выбора: **UI**, **Code**, **Data**, **PropertySheets**. Выбираем **Code**. Правая панель называется **Templates**. На ней выбираем опцию **C++ File (.cpp)**. Внизу в диалоговом окне вводим имя файла **MathFuncsDll.cpp** и нажимаем кнопку **Add**. Это было четвертое действие.

После выполнения этого действия на левой панели, которая называется `Solution Explorer`, отображается имя файла **MathFuncsDll.cpp** в форме ветки, растущей из папки **Source Files**. На правой

панели добавляется вкладка **MathFuncsDll.h**, внутри которой отображается текстовый редактор с пустым содержимым созданного файла **MathFuncsDll.cpp**. При этом в реальной папке

C:\Documents and Settings\User\My Documents  
  \Visual Studio 2005\Projects\DynamicLibrary\MathFuncsDll

добавился пустой файл с именем **MathFuncsDll.cpp**.

Пятое действие состоит в заполнении файла **MathFuncsDll.cpp** следующим текстом:

```
// MathFuncsDll.cpp
// compile with: /EHsc /LD

#include "MathFuncsDll.h"

#include <stdexcept>

using namespace std;

namespace MathFuncs
{
    double MyMathFuncs::Add(double a, double b)
    {
        return a + b;
    }

    double MyMathFuncs::Subtract(double a, double b)
    {
        return a - b;
    }

    double MyMathFuncs::Multiply(double a, double b)
    {
        return a * b;
    }

    double MyMathFuncs::Divide(double a, double b)
    {
        if (b == 0)
        {
            throw new invalid_argument("b cannot be zero!");
        }

        return a / b;
    }
}
```

После этого нажимаем иконку с дискетой для того, чтобы сохранить файл.

Настало время обозначить то, что мы хотим получить **DLL**, а не консольное приложение. Для этого на левой панели, которая называется Solution Explorer, кликаем правой кнопкой мышки по узлу дерева с названием проекта **MathFuncsDll**. В открывшемся меню выбираем пункт Properties. В результате этого открывается диалоговое окно с двумя панелями — левой и правой. На левой панели раскрываем узел **Configuration Properties** в деревоподобной структуре и переходим к ветке **General** в дереве. После этого на правой панели в подразделе **Project Default** выбираем поле **Configuration Type** и меняем его значение на **Dynamic Library (.dll)**. После нажатия кнопки **OK** это будет завершение шестого действия.

Создание библиотеки DLL завершается после выбора функции **Build Solution** через меню **Build** или по правой кнопке мышки в узле Solution 'DynamicLibrary' (1 project) на левой панели, которая называется Solution Explorer. Это будет седьмым действием. Сам файл библиотеки **MathFuncsDll.dll** появляется внутри папки \debug, которая создается внутри папки

C:\Documents and Settings\User\My Documents\Visual Studio 2005\Projects\DynamicLibrary

Наряду с этим файлом в папке `\debug` появляется целый ряд других файлов с тем же именем, но с разными расширениями.

### Создание проекта приложения, которое использует библиотеку DLL.

В меню **File** выбираем **New** и далее **Project**. Это первое действие. В результате этого действия открывается уже знакомое нам диалоговое окно с двумя панелями.

На левой панели, которая называется **Project types**, выбираем **Win32**. Это будет второе действие.

На правой панели, которая называется **Templates**, выбираем **Win32 Console Application**. Это будет третье действие.

В том же диалоговом окне имеется поле для имени проекта. В это поле пишем **MyExecRefsDll**. Кроме того, имеется поле выбора с двумя вариантами **Create New Solution** и **Add to Solution**. Выбираем второй вариант **Add to Solution**. Это будет четвертое действие.

Нажимаем кнопку **OK**. В открывшемся диалоговом окне с именем **Win32 Application Wizard** нажимаем кнопку **Next**. Это пятое действие.

В следующем диалоговом окне выбираем опцию **Console application**. Это будет шестое действие.

Седьмое действие состоит в том, что мы убираем отметку с выбора **Precompiled header**.

И наконец, нажимаем кнопку **Finish**. Это восьмое действие. В результате проделанных действий образуется связка из двух проектов **MathFuncsDll** и **MyExecRefsDll** в рамках одного решения с именем **DynamicLibrary**. На левой панели в пределах нового проекта **MyExecRefsDll** отображаются новые файлы **stdafx.h**, **stdafx.cpp** и **MyExecRefsDll.cpp**. Появляется также текстовый файл **ReadMe.txt**. Все эти файлы расположены в папке

```
C:\Documents and Settings\User\My Documents  
\Visual Studio 2005\Projects\DynamicLibrary\MyExecRefsDll
```

### Создание самого приложения, которое использует библиотеку DLL.

Файл **MyExecRefsDll.cpp**, созданный при создании проекта, почти пустой. Для использования библиотеки DLL надо создать ссылки на неё в коде основной программы. На левой панели, которая называется **Solution Explorer**, кликаем правой кнопкой мышки наш проект **MyExecRefsDll**. В открывшемся меню выбираем пункт **References**. Открывшееся диалоговое окно почти пустое. На левой панели разворачиваем узел **Common Properties**, выбираем ветку **References** и нажимаем кнопку **Add New Reference**. Это были первое и второе действия.

Открывается диалоговое окно со вкладкой **Projects**. В ней выбираем проект нашей динамической библиотеки **MathFuncsDll** и нажимаем кнопку **OK**. Это третье действие.

На левой панели диалогового окна разворачиваем узел **Configuration Properties** и в нём узел **C/C++** и выбираем ветку **General**. На правой панели в ячейке **Additional Include Directories** печатаем полный путь к файлу **MathFuncsDll.h**.

```
C:\Documents and Settings\User\My Documents\Visual Studio  
2005\Projects\DynamicLibrary\MathFuncsDll
```

Это будет четвертое действие.

Чтобы указать местоположение самой динамической библиотеки в уже развёрнутом узле **Configuration Properties** на левой панели выбираем ветку **Debugging**. В ячейке **Environment** печатаем следующую строку:

```
PATH=C:\Documents and Settings\User\My Documents\Visual Studio  
2005\Projects\DynamicLibrary\debug
```

После нажатия кнопки **OK** это будет пятое действие.

## Примечание

Если есть желание запускать программу **MyExecRefsDll.exe** из командной строки, а не изнутри Visual Studio, Вы должны изменить вручную переменную **PATH** в переменной среды, напечатав в командной строке: `set PATH=%PATH%;C:\Documents and Settings\User\My Documents\Visual Studio 2005\Projects\DynamicLibrary\debug.`

Теперь можно использовать класс **MyMathFuncs**, экспортируемый библиотекой **MathFuncsDll.cpp**. Файл **MathFuncsDll.cpp** содержал текст

```
// MyExecRefsDll.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    return 0;
}
```

Заменяем содержимое файла следующим текстом:

```
// MyExecRefsDll.cpp
// compile with: /EHsc /link MathFuncsDll.lib

#include <iostream>

#include "MathFuncsDll.h"

using namespace std;

int main()
{
    double a = 7.4;
    int b = 99;

    cout << "a + b = " <<
        MathFuncs::MyMathFuncs::Add(a, b) << endl;
    cout << "a - b = " <<
        MathFuncs::MyMathFuncs::Subtract(a, b) << endl;
    cout << "a * b = " <<
        MathFuncs::MyMathFuncs::Multiply(a, b) << endl;
    cout << "a / b = " <<
        MathFuncs::MyMathFuncs::Divide(a, b) << endl;

    return 0;
}
```

Это будет шестое действие.

Седьмое действие состоит в том, что мы выполняем сборку программы, запуская **Build Solution** из меню **Build**.

## Проверка работы программы.

Остаётся выполнить программу **MyExecRefsDll.exe** изнутри Visual Studio. На левой панели, которая называется Solution Explorer, кликаем правой кнопкой мыши второй из наших проектов **MyExecRefsDll** и в открывшемся меню выбираем **Set As StartUp Project**. Это первое действие.

Второе и финальное действие состоит в том, что в меню **Debug** запускаем **Start Without Debugging**. На открывшейся консоли мы должны увидеть следующие четыре строки:

$$\begin{aligned} a + b &= 106.4 \\ a - b &= -91.6 \\ a * b &= 732.6 \\ \mathbf{a / b} &= \mathbf{0.0747475} \end{aligned}$$