要查看英语原文，请勾选“英语”复选框。也可将鼠标指针移到文本上，在弹出窗口中显示英语原文。

演练：创建和使用动态链接库 (C++)

Visual Studio 2013

此分步演练演示如何创建用于 C++ 应用的动态链接库 (DLL)。 使用库是重复使用代码的一种绝佳方式。 与其在创建的每个程序中重新实现相同的例程，不如一次性编写它们，然后从需要该功能的应用中引用它们。 通过将代码置入 DLL，可以节省引用它的每个应用中的空间，也可以更新该 DLL 而无需重新编译所有应用。 有关 DLL 的详细信息，请参阅 Visual C++ 中的 DLL。

本演练涵盖以下任务：

- 创建 DLL 项目。
- 向 DLL 中添加类。
- 创建使用加载时动态链接来引用 DLL 的控制台应用。
- 在该应用中使用类中的功能。
- 运行该应用。

本演练将创建仅可从使用 C++ 调用约定的应用中调用的 DLL。 有关如何创建适用于其他语言的 DLL 的信息，请参阅从 Visual Basic 应用程序调用 DLL 函数。

系统必备

本主题假定你了解 C++ 语言的基础知识。

创建动态链接库 (DLL) 项目

1. 在菜单栏上，依次选择“**文件**”、“**新建**”、“**项目**”。
2. 在“**新建项目**”对话框的左窗格中，依次展开“**已安装**”、“**模板**”、“**Visual C++**”，然后选择“**Win32**”。
3. 在中间窗格中，选择“**Win32 控制台应用程序**”。
4. 在“**名称**”框中为项目指定名称，例如 MathFuncsDll。 在“**解决方案名称**”框中为解决方案指定名称，例如 DynamicLibrary。 选择“**确定**”按钮。
5. 在“**Win32 应用程序向导**”对话框的“**概述**”页上，选择“**下一步**”按钮。
6. 在“**应用程序设置**”页面的“**应用程序类型**”下，选择“**DLL**”。
7. 选择“**完成**”按钮创建项目。

向动态链接库添加类

1. 若要为新类创建头文件，请在菜单栏上，依次选择**“项目”**、**“添加新项”**。 在**“添加新项”**对话框的左窗格中，在**“Visual C++”**下选择**“代码”**。 在中间窗格中，选择**“头文件(.h)”**。 为头文件指定名称（例如 MathFuncsLib.h），然后选择**“添加”**按钮。 将显示一个空白头文件。

2. 将以下代码添加到头文件的开头：

C++

```
// MathFuncsDll.h

#ifdef MATHFUNCSDLL_EXPORTS
#define MATHFUNCSDLL_API __declspec(dllexport)
#else
#define MATHFUNCSDLL_API __declspec(dllimport)
#endif
```

3. 添加一个名为 MyMathFuncs 的基类，以执行常见的算术运算（例如加、减、乘和除）。 代码应类似如下:

C++

```
namespace MathFuncs
{
    // This class is exported from the MathFuncsDll.dll
    class MyMathFuncs
    {
    public:
        // Returns a + b
        static MATHFUNCSDLL_API double Add(double a, double b);

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

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

        // Returns a / b
        // Throws const std::invalid_argument& if b is 0
        static MATHFUNCSDLL_API double Divide(double a, double b);
    };
}
```

在此代码中，当 MATHFUNCSDLL_EXPORTS 符号已经被定义时，成员函数声明部分的 MATHFUNCSDLL_API 符号将被设置为 __declspec(dllexport) 修饰符。 此修饰符使函数能作为 DLL 导出，以供其他应用程序调用。 如果未定义 MATHFUNCSDLL_EXPORTS（例如，应用程序包含了头文件），则 MATHFUNCSDLL_API 将定义成员函数声明中的 **__declspec(dllimport)** 修饰符。 此修饰符将优化在应用程序中导入该函数的操作。 默认情况下，DLL 的“新建项目”模板会将 *PROJECTNAME*_EXPORTS 添加到 DLL 项目的已定义符号中。 在本例中，生成 MathFuncsDll 项目后将定义 MATHFUNCSDLL_EXPORTS。 有关详细信息，请参阅dllexport、dllimport。

说明

如果你要在命令行上生成 DLL 项目，请使用 **/D** 编译器选项来定义 MATHFUNCSDLL_EXPORTS 符号。

4. 在**“解决方案资源管理器”**的**“MathFuncsDll”**项目中，在**“源文件”**文件夹中打开 MathFuncsDll.cpp。

5. 实现源文件中 MyMathFuncs 的功能。 代码应类似如下:

C++

```
// MathFuncsDll.cpp : Defines the exported functions for the DLL application.
//

#include "stdafx.h"
#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 invalid_argument("b cannot be zero!");
        }

        return a / b;
    }
}
```

6. 通过选择菜单栏中的 生成>生成解决方案 编译动态链接库

说明

如果使用的是不显示“**生成**”菜单的 Express 版，请在菜单栏上，依次选择“**工具**”、“**设置**”、“**专家设置**”来启用它，然后依次选择“**生成**”、“**生成解决方案**”。

说明

如果在命令行中生成项目，请使用 **/LD** 编译器选项指定输出文件为 DLL。 有关详细信息，请参阅/MD、/MT、/LD（使用运行库）。 使用 **/EHsc** 编译器选项启用 C++ 异常处理。 有关详细信息，请参阅/EH（异常处理模型）。

创建引用 DLL 的应用

1. 为了创建一个项目引用你刚刚创建好的DLL，在菜单栏中选择 文件>新建>项目。

2. 在左窗格中的“**Visual C++**”下，选择“**Win32**”。

3. 在中间窗格中，选择“**Win32 控制台应用程序**”。

4. 在“**名称**”框中为项目指定名称，例如 MyExecRefsDll。 在“**解决方案**”旁边，从下拉列表中选择“**添加到解决方案**”。 这会将新项目添加到包含此 DLL 的同一解决方案中。 选择“**确定**”按钮。

5. 在“**Win32 应用程序向导**”对话框的“**概述**”页上，选择“**下一步**”按钮。

6. 在“**应用程序设置**”页的“**应用程序类型**”下，选择“**控制台应用程序**”。

7. 在“**应用程序设置**”页的“**附加选项**”下，清除“**预编译头**”复选框。

8. 选择“**完成**”按钮创建项目。

在该应用中使用类库中的功能

1. 在创建一个控制台应用程序后，一个空的程序已经为你创建好了。 源文件的名称与你之前选择的名称相同。 在此示例中，其名称为 MyExecRefsDll.cpp。

2. 若要在应用中使用在 DLL 中创建的数学例程，必须对它进行引用。 为此，请在 解决方案资源管理器 中选择 MyExecRefsDll 项目，然后在菜单栏上，选择 项目，引用。 在“**属性页**”对话框中，展开“**通用属性**”节点、选择“**框架和引用**”，然后选择“**添加新引用**”按钮。 有关“**引用**”对话框的更多信息，请参见“<Projectname> 属性页”对话框 ->“通用属性”->“框架和引用”。

3. “**添加引用**”对话框列出了可以引用的库。 “**项目**”选项卡列出了当前解决方案中的所有项目以及它们包含的所有库。 在“**项目**”选项卡上，选中“MathFuncsDll”旁边的复选框，然后选择“**确定**”按钮。

4. 若要引用 DLL 的头文件，必须修改包含的目录路径。 若要执行此操作，请在“**属性页**”对话框中，依次展开“**配置属性**”节点和“**C/C++**”节点，然后选择“**常规**”。 在“**附加包含目录**”旁边，指定 MathFuncsDll.h 头文件的位置路径。 你可以使用相对路径（例如 ..\MathFuncsDll\），然后选择“**确定**”按钮。

5. 现在即可在此应用程序中使用 MyMathFuncs 类。 将 MyExecRefsDll.cpp 的内容替换为以下代码：

C++

```
// 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;

    try
```

```
        {
            cout << "a / 0 = " <<
                MathFuncs::MyMathFuncs::Divide(a, 0) << endl;
        }
        catch (const invalid_argument &e)
        {
            cout << "Caught exception: " << e.what() << endl;
        }

        return 0;
    }
```

6. 通过在菜单栏上依次选择“**生成**”、“**生成解决方案**”来生成可执行文件。

运行应用程序

1. 请确保已将 MyExecRefsDll 选为默认项目。 在“**解决方案资源管理器**”中，选择 MyExecRefsDll，然后在菜单栏上，依次选择“**项目**”、“**设为启动项目**”。

2. 若要运行项目，请在菜单栏上依次选择“**调试**”、“**开始执行(不调试)**”。 输出应该与下面的内容类似：

```
a + b = 106.4 a - b = -91.6 a * b = 732.6 a / b = 0.0747475 捕获异常: b 不能为零！
```

请参见

任务
Visual C++ 指导教程
演练：部署程序 (C++)
概念
Visual C++ 中的 DLL
从 Visual Basic 应用程序调用 DLL 函数
其他资源
部署的桌面应用程序(Visual C++)

社区附加资源