

Tutorial: Creación y uso de una biblioteca estática

13/04/2020 • Tiempo de lectura: 5 minutos •  

En este artículo

[Requisitos previos](#)

[Creación de un proyecto de biblioteca estática](#)

[Adición de una clase a la biblioteca estática](#)

[Creación de una aplicación de consola de C++ que haga referencia a la biblioteca estática](#)

[Uso de la funcionalidad de la biblioteca estática en la aplicación](#)

[Ejecución de la aplicación](#)

[Vea también](#)

En este tutorial paso a paso se muestra cómo crear una biblioteca estática (archivo .lib) para su uso con aplicaciones de C++. Utilizar una biblioteca estática es una excelente manera de reutilizar el código. En lugar de volver a implementar las mismas rutinas en todas las aplicaciones que requieran la funcionalidad, escribálas una vez en una biblioteca estática y, a continuación, haga referencia a ellas desde las aplicaciones. El código vinculado desde una biblioteca estática pasa a formar parte de la aplicación (no es necesario instalar otro archivo para usar el código).

En este tutorial se tratan las siguientes tareas:

- [Creación de un proyecto de biblioteca estática](#)
- [Adición de una clase a la biblioteca estática](#)

• [Creación de una aplicación de consola de C++ que haga referencia a la biblioteca](#)

- [Creación de una aplicación de consola de C++ que haga referencia a la biblioteca estática](#)
- [Uso de la funcionalidad de la biblioteca estática en la aplicación](#)
- [Ejecución de la aplicación](#)

Requisitos previos

Descripción de los fundamentos del lenguaje C++.

Creación de un proyecto de biblioteca estática

Las instrucciones para crear el proyecto varían en función de la versión de Visual Studio. Para ver la documentación de su versión preferida de Visual Studio, use el control de selector **Versión**. Se encuentra en la parte superior de la tabla de contenido de esta página.


Para crear un proyecto de biblioteca estática en Visual Studio 2019

1. En la barra de menús, seleccione **Archivo > Nuevo > Proyecto** para abrir el cuadro de diálogo **Crear un nuevo proyecto**.
2. En la parte superior del cuadro de diálogo, establezca **Lenguaje** en **C++**, establezca **Plataforma** en **Windows** y establezca **Tipo de proyecto** en **Biblioteca**.
3. En la lista filtrada de tipos de proyecto, seleccione **Asistente para escritorio de Windows** y, después, **Siguiente**.
4. En la página **Configure su nuevo proyecto**, escriba *MathLibrary* en el cuadro **Nombre del proyecto** para especificar un nombre para el proyecto. Escriba *StaticMath* en el cuadro **Nombre de la solución**. Haga clic en el botón **Crear** para abrir el cuadro de diálogo **Proyecto de escritorio de Windows**.
5. En el cuadro de diálogo **Proyecto de escritorio de Windows**, en **Tipo de aplicación**, seleccione **Biblioteca estática (.lib)**.
6. En **Opciones adicionales**, desactive la casilla **Encabezado precompilado** si está activada. Active la casilla **Proyecto vacío**.
7. Seleccione **Aceptar** para crear el proyecto.

Adición de una clase a la biblioteca estática

Para agregar una clase a la biblioteca estática

1. Si quiere crear un archivo de encabezado para una nueva clase, haga clic con el botón derecho para abrir el menú contextual del proyecto **MathLibrary** en el **Explorador de soluciones** y, después, elija **Agregar > Nuevo elemento**.
2. En el cuadro de diálogo **Agregar nuevo elemento**, seleccione **Visual C++ > Código**. En el panel central, seleccione **Archivo de encabezado (.h)**. Especifique un nombre para el encabezado de archivo, por ejemplo, *MathLibrary.h*, y después elija el botón **Agregar**. Se muestra un archivo de encabezado casi en blanco.
3. Agregue una declaración para una clase denominada *Arithmetic* para hacer operaciones matemáticas comunes, como sumas, restas, multiplicaciones y divisiones. El código debería ser similar al siguiente:

C++	 Copiar
<pre>// MathLibrary.h #pragma once namespace MathLibrary { class Arithmetic { public: // Returns a + b static double Add(double a, double b); // Returns a - b static double Subtract(double a, double b); // Returns a * b static double Multiply(double a, double b); // Returns a / b static double Divide(double a, double b); }; }</pre>	

4. Para crear un archivo de código fuente para la nueva clase, abra el menú contextual para el proyecto **MathLibrary** en el **Explorador de soluciones** y, después, elija **Agregar > Nuevo elemento**.

5. En el cuadro de diálogo **Agregar nuevo elemento**, en el panel central, seleccione **Archivo C++ (.cpp)**. Especifique un nombre para el archivo de código fuente, por ejemplo, *MathLibrary.cpp*, y después elija el botón **Agregar**. Se muestra un archivo de código fuente en blanco.
6. Use este archivo de código fuente para implementar la funcionalidad de la clase *Arithmetic*. El código debería ser similar al siguiente:

C++	 Copiar
<pre>// MathLibrary.cpp // compile with: cl /c /EHsc MathLibrary.cpp // post-build command: lib MathLibrary.obj #include "MathLibrary.h" namespace MathLibrary { double Arithmetic::Add(double a, double b) { return a + b; } double Arithmetic::Subtract(double a, double b) { return a - b; } double Arithmetic::Multiply(double a, double b) { return a * b; } double Arithmetic::Divide(double a, double b) { return a / b; } }</pre>	

7. Para compilar la biblioteca estática, seleccione **Compilar > Compilar solución** en la barra de menús. La compilación crea una biblioteca estática, *MathLibrary.lib*, que pueden usar otros programas.

ⓘ Nota

Cuando compile desde la línea de comandos de Visual Studio, debe compilar el programa en dos pasos. Primero, ejecute `cl /c /EHsc MathLibrary.cpp` para

compilar el código y crear un archivo de objeto denominado *MathLibrary.obj*. (El comando `cl` invoca el compilador, `Cl.exe` y la opción `/c` especifica compilar sin vinculación. Para obtener más información, consulte el artículo [/c \(Compilar sin vincular\)](#). En segundo lugar, ejecute `lib MathLibrary.obj` para enlazar el código y crear la biblioteca estática *MathLibrary.lib*. (El comando `lib` invoca el Administrador de bibliotecas, `Lib.exe`. Para obtener más información, vea [LIB Reference](#)).

Creación de una aplicación de consola de C++ que haga referencia a la biblioteca estática

Para crear una aplicación de consola de C++ que haga referencia a la biblioteca estática en Visual Studio 2019

1. En el **Explorador de soluciones**, haga clic con el botón derecho en el nodo superior, **Solución "StaticMath"**, para abrir el menú contextual. Elija **Agregar > Nuevo proyecto** para abrir el cuadro de diálogo **Agregar un nuevo proyecto**.
2. En la parte superior del cuadro de diálogo, establezca el filtro **Tipo de proyecto** en **Consola**.
3. En la lista de plantillas de proyecto, seleccione **Aplicación de consola y Siguiente**. En la página siguiente, escriba *MathClient* en el cuadro **Nombre** para especificar un nombre para el proyecto.
4. Haga clic en el botón **Crear** para crear el proyecto de cliente.
5. Después de crear una aplicación de consola, se crea un programa vacío. El nombre del archivo de código fuente será el mismo que el elegido anteriormente. En este ejemplo, se denomina `MathClient.cpp`.

Uso de la funcionalidad de la biblioteca estática en la aplicación

Para utilizar la funcionalidad de la biblioteca estática en la aplicación


1. Para poder usar las rutinas de coincidencia de la biblioteca estática, debe hacer referencia a ella. Abra el menú contextual del proyecto **MathClient** en el **Explorador de soluciones** y, luego, elija **Agregar > Referencia**.
2. El cuadro de diálogo **Agregar referencia** muestra las bibliotecas a las que puede hacer referencia. La pestaña **Proyectos** enumera los proyectos de la solución actual y las bibliotecas a las que hacen referencia. En la pestaña **Proyectos**, active la casilla **MathLibrary** y, después, haga clic en el botón **Aceptar**.
3. Para hacer referencia al archivo de encabezado `MathLibrary.h`, debe modificar la ruta de acceso de los directorios de inclusión. En el **Explorador de soluciones**, haga clic con el botón derecho **MathClient** para abrir el menú contextual. Elija **Propiedades** para abrir el cuadro de diálogo **Páginas de propiedades de MathClient**.
4. En el cuadro de diálogo **Páginas de propiedades de MathClient**, establezca el valor de la lista desplegable **Configuración** en **Todas las configuraciones**. Establezca el valor de la lista desplegable **Plataforma** en **Todas las plataformas**.
5. Seleccione la página de propiedades **Propiedades de configuración > C/C++ > General**. En la propiedad **Directorios de inclusión adicionales**, especifique la ruta de acceso del directorio **MathLibrary** o búsquela.

Para buscar la ruta de acceso del directorio:

- a. Abra la lista desplegable del valor de propiedad **Directorios de inclusión adicionales** y, a continuación, elija **Editar**.
- b. En el cuadro de diálogo **Directorios de inclusión adicionales** haga doble clic en la parte superior del cuadro de texto. A continuación, presione el botón de puntos suspensivos (...) al final de la línea.
- c. En el cuadro de diálogo **Seleccionar directorio**, desplácese a un nivel y, a continuación, seleccione el directorio **MathLibrary**. Después, haga clic en el botón **Seleccionar carpeta** para guardar su selección.
- d. En el cuadro de diálogo **Directorios de inclusión adicionales**, haga clic en el botón **Aceptar**.
- e. En el cuadro de diálogo **Páginas de propiedades**, elija el botón **Aceptar** para guardar los cambios en el proyecto.

6. Ahora puede usar la clase `Arithmetic` en esta aplicación incluyendo el encabezado

#include "MathLibrary.h" en el código. Reemplace el contenido de MathClient.cpp por el código siguiente:

C++	
<pre>// MathClient.cpp // compile with: cl /EHsc MathClient.cpp /link MathLibrary.lib #include <iostream> #include "MathLibrary.h" int main() { double a = 7.4; int b = 99; std::cout << "a + b = " << MathLibrary::Arithmetic::Add(a, b) << std::endl; std::cout << "a - b = " << MathLibrary::Arithmetic::Subtract(a, b) << std::endl; std::cout << "a * b = " << MathLibrary::Arithmetic::Multiply(a, b) << std::endl; std::cout << "a / b = " << MathLibrary::Arithmetic::Divide(a, b) << std::endl; return 0; }</pre>	

7. Para compilar el archivo ejecutable, elija **Compilación > Compilar solución** en la barra de menús.

Ejecución de la aplicación

Para ejecutar la aplicación

1. Compruebe que **MathClient** está seleccionado como proyecto predeterminado. Para seleccionarlo, haga clic con el botón derecho para abrir el menú contextual de **MathClient** en el **Explorador de soluciones** y, después, elija **Establecer como proyecto de inicio**.
2. Para ejecutar el proyecto, en la barra de menús, seleccione **Depurar > Iniciar sin depurar**. El resultado debería ser similar al siguiente:

Output	
--------	---

```
a + b = 106.4  
a - b = -91.6  
a * b = 732.6  
a / b = 0.0747475
```

Vea también

[Tutorial: Creación y uso de una biblioteca de vínculos dinámicos \(C++\)](#)

[Aplicaciones de escritorio \(Visual C++\)](#)

¿Le ha resultado útil esta página?

 Yes  No
