

Materia: Programación orientada a objetos

Nombre: Miramontes Luis Uriel

Numero de control: 191140009

29-1-2020

Archivos generados
(aplicación de consola
y Windows Form)

Maestra: Cinthia Anahí Mata Bravo



Índice

Aplicación de consola	3
Designer.cs.....	3
Archivo .cs.....	4
Creación de archivos de recursos para aplicaciones .NET.....	4
Recursos en archivos de texto	5
Recursos en archivos .resx.....	6
Recursos en archivos .resources.....	6
App.config en Visual Studio.....	7
Windows Forms	7
Conclusión.....	8

Aplicación de consola

Una aplicación de consola es un programa de computadora diseñado para usarse a través de una interfaz de computadora de solo texto, como un terminal de texto , la interfaz de línea de comando de algunos sistemas operativos (Unix , DOS , etc.) o la interfaz basada en texto incluida con la mayoría Sistemas operativos de interfaz gráfica de usuario (GUI), como la consola Win32 en Microsoft Windows , la Terminal en macOS y xterm en Unix. Un usuario generalmente interactúa con una aplicación de consola usando solo un teclado y pantalla de visualización , a diferencia de las aplicaciones GUI, que normalmente requieren el uso de un mouse u otro dispositivo señalador . Muchas aplicaciones de consola, como los intérpretes de línea de comandos, son herramientas de línea de comandos, pero también existen numerosos programas de interfaz de usuario basada en texto (TUI).

A medida que la velocidad y la facilidad de uso de las aplicaciones GUI han mejorado con el tiempo, el uso de aplicaciones de consola ha disminuido considerablemente, pero no ha desaparecido. Algunos usuarios simplemente prefieren las aplicaciones basadas en consola, mientras que algunas organizaciones aún confían en las aplicaciones de consola existentes para manejar tareas clave de procesamiento de datos.

La capacidad de crear aplicaciones de consola se mantiene como una característica de los entornos de programación modernos como Visual Studio y .NET Framework en Microsoft Windows porque simplifica enormemente el proceso de aprendizaje de un nuevo lenguaje de programación al eliminar la complejidad de una interfaz gráfica de usuario (ver un ejemplo en el artículo de C #).

Para las tareas de procesamiento de datos y la administración de la computadora, estos entornos de programación representan el siguiente nivel del sistema operativo o el control del procesamiento de datos después de las secuencias de comandos . Si el programador original y / o algunos colegas solo van a ejecutar una aplicación, es posible que no sea necesaria una interfaz gráfica de usuario bonita, lo que hace que la aplicación sea más ágil, más rápida y más fácil de mantener.

Designer.cs

Designer.cs contiene la declaración y la inicialización de los controles de la interfaz de usuario y el diseño del formulario. El formulario se representa en base a la información provista en designer.cs. Este archivo se genera automáticamente cuando se crea un formulario en modo de diseño.

Archivo .cs

Los archivos con la extensión CS contienen código fuente escrito en el lenguaje de programación orientado a objetos C#. Se utilizan habitualmente con Microsoft .NET Framework y están destinados al desarrollo de aplicaciones. Estas aplicaciones pueden variar desde simples aplicaciones de escritorio hasta programas complejos, todos desarrollados utilizando C# como lenguaje de programación principal.

Los archivos CS están principalmente relacionados con el programa Microsoft Visual C#, que forma parte de Microsoft Visual Studio. Otros programas externos que soportan el resaltado sintáctico también pueden visualizar los archivos CS. Para ejecutar correctamente un archivo CS es necesario compilarlo primero.

Creación de archivos de recursos para aplicaciones .NET

Puede incluir recursos, como cadenas, imágenes o datos de objeto, en los archivos de recursos para hacer que estén disponibles fácilmente en su aplicación. .NET Framework ofrece cinco mecanismos para crear archivos de recursos:

Crea un archivo de texto que contenga recursos de cadena. Puede usar el generador de archivos de recursos (Resgen.exe) para convertir el archivo de texto en un archivo de recursos binario (.resources). Después, puede insertar el archivo de recursos binario en el ejecutable de una aplicación o en una biblioteca de aplicaciones mediante un compilador del lenguaje, o bien puede insertarlo en un ensamblado satélite mediante Assembly Linker (Al.exe). Para obtener más información, vea la sección recursos en archivos de texto.

Crea un archivo de recursos XML (.resx) que contenga una cadena, una imagen o datos de objeto. Puede usar el generador de archivos de recursos (Resgen.exe) para convertir el archivo .resx en un archivo de recursos binario (.resources). Después, puede insertar el archivo de recursos binario en el ejecutable de una aplicación o en una biblioteca de aplicaciones mediante un compilador del lenguaje, o bien puede insertarlo en un ensamblado satélite mediante Assembly Linker (Al.exe). Para obtener más información, vea la sección Recursos en archivos .resx.

Crea un archivo de recursos XML (.resx) mediante programación utilizando los tipos del espacio de nombres System.Resources. Puede crear un archivo .resx, enumerar sus recursos y recuperar recursos específicos por nombre. Para obtener más información, vea el tema Working with .resx Files Programmatically (Trabajar con archivos .resx mediante programación).

Crea un archivo de recursos binario (.resources) mediante programación. Después, puede insertar el archivo en el ejecutable de una aplicación o en una biblioteca de aplicaciones mediante un compilador del lenguaje, o bien puede insertarlo en un ensamblado satélite mediante Assembly Linker (Al.exe). Para obtener más información, vea la sección Recursos en archivos .resources.

Use Visual Studio para crear un archivo de recursos e incluirlo en el proyecto. Visual Studio proporciona un editor de recursos que le permite agregar, eliminar y modificar recursos. En tiempo de compilación, el archivo de recursos se convierte automáticamente en un archivo .resources binario y se incrusta en un ensamblado de la aplicación o un ensamblado satélite.

Recursos en archivos de texto

Puede usar archivos de texto (.txt o .restext) únicamente para almacenar recursos de cadena. Para los recursos que no son de cadena, use archivos .resx o créelos mediante programación. El formato de archivo de recursos de los archivos .txt y .restext es idéntico. La extensión de archivo .restext solo sirve para permitir que los archivos de texto se identifiquen de forma inmediata como archivos de recursos basados en texto.

Los recursos de cadena aparecen como pares name/value, donde name es una cadena que identifica el recurso y value es la cadena de recurso que se devuelve al pasar name a un método de recuperación de recursos como ResourceManager.GetString. name y value deben separarse mediante el signo igual (=).

Las cadenas vacías (es decir, un recurso cuyo valor es String.Empty) se permiten en los archivos de texto. A partir de .NET Framework 4.5 y en todas las versiones de .NET Core, los archivos de texto admiten compilación condicional con las construcciones #ifdefsymbol... #endif y #if !symbol... #endif. Se puede usar el modificador /define con el Generador de archivos de recursos (Resgen.exe) para definir símbolos.

Cada recurso requiere su propia construcción #ifdefsymbol... #endif o #if !symbol... #endif. Si se usa una instrucción #ifdef y se ha definido symbol, el recurso asociado se incluye en el archivo .resources. En caso contrario, no se incluye. Si se usa una instrucción #if ! y no se ha definido symbol, el recurso asociado se incluye en el archivo .resources. En caso contrario, no se incluye.

En los archivos de texto, los comentarios son opcionales y se incluyen al principio de la línea precedidos por un símbolo de punto y coma (;) o un signo de número (#). Las líneas que contienen los comentarios se pueden colocar en cualquier parte del archivo. Los comentarios no se incluirán en un archivo .resources compilado que se crea mediante el Generador de archivos de recursos (Resgen.exe).

Las líneas en blanco de los archivos de texto se consideran espacios en blanco y no se tienen en cuenta.

Si el archivo de texto contiene instancias duplicadas de name, el Generador de archivos de recursos (Resgen.exe) muestra una advertencia y omite el segundo nombre.

El valor no puede contener caracteres de nueva línea, pero puede usar caracteres de escape de estilo de lenguaje C como `\n` para representar una línea nueva y `\t` para representar una tabulación. También puede incluir un carácter de barra diagonal inversa si se trata de un carácter de escape (por ejemplo, `"\"`). Además, se admite el uso de una cadena vacía.

Debe guardar los recursos en formato de archivo de texto utilizando la codificación UTF-8 o UTF-16 con el orden de bytes little endian o big endian. Pero el generador de archivos de recursos (Resgen.exe), que convierte un archivo .txt en un archivo .resources, trata los archivos como UTF-8 de forma predeterminada. Si desea que Resgen.exe reconozca un archivo codificado con UTF-16, es preciso incluir una marca de orden de bytes Unicode (U+FEFF) al principio del archivo.

Para insertar un archivo de recursos en formato de texto en un ensamblado .NET, debe convertirlo en un archivo de recursos binario (.resources) mediante el Generador de archivos de recursos (Resgen.exe). Después, puede insertar el archivo .resources en un ensamblado .NET mediante un compilador del lenguaje o insertarlo en un ensamblado satélite mediante Assembly Linker (Al.exe).

Recursos en archivos .resx

A diferencia de los archivos de texto, que solo pueden almacenar recursos de cadena, los archivos de recursos XML (.resx) pueden almacenar cadenas, datos binarios (como imágenes, iconos y clips de sonido) y objetos de programación. Los archivos .resx contienen un encabezado estándar, en el que se describe el formato de las entradas de los recursos y se especifica la información sobre el control de versiones correspondiente al código XML que se usa para analizar los datos. Los datos del archivo de recursos siguen lo establecido en el encabezado XML. Cada elemento de datos se compone de un par nombre/valor incluido en una etiqueta data. Su atributo name define el nombre del recurso y la etiqueta value anidada contiene el valor del recurso. En el caso de los datos de cadena, la etiqueta value contiene la cadena.

Recursos en archivos .resources

Puede usar la clase `System.Resources.ResourceWriter` para crear mediante programación un archivo de recursos binario (.resources) directamente desde el código. También puede usar el generador de archivos de recursos (Resgen.exe) para crear un archivo .resources a partir de un archivo de texto o un archivo .resx. El archivo .resources puede contener datos binarios (matrices de bytes) y datos de objeto además de datos de cadena. Al crear un archivo .resources mediante programación, se requieren los siguientes pasos:

- Cree un objeto `ResourceWriter` con un nombre de archivo único. Puede hacerlo especificando el nombre de archivo o una secuencia de archivos en un constructor de clase `ResourceWriter`.

- Llame a una de las sobrecargas del método `ResourceWriter.AddResource` para que cada recurso con nombre se agregue al archivo. El recurso puede ser una cadena, un objeto o una colección de datos binarios (una matriz de bytes).
- Llame al método `ResourceWriter.Close` para que escriba los recursos en el archivo y cierre el objeto `ResourceWriter`.

App.config en Visual Studio

Al utilizar una `app.config` para configurar la cadena de conexión a la base de datos la principal ventaja que tiene es que evitaras tener que recompilar tu aplicación si es que cambiar algún dato de la cadena de conexión, además también esta la facilidad de modificarlos. Si en cada formulario usas la cadena de conexión, y un día debes cambiar algo, tendrías que entrar a cada Form y cambiar el valor, mientras que en `My.Settings` solo lo cambias una vez y está listo. El archivo de configuración podrías editarlo con el notepad y como internamente es representado como xml, podrás editar la configuración y tu aplicación en el cliente seguirá funcionando sin tener que hacer un nueva compilación y actualización en cada pc del usuario. Ahora para guardar la cadena de conexión en una `app.config` ya sea en Visual Basic o C# se siguen estos pasos, creamos un nuevo proyecto sea en VB o C# y luego desde el Solution Explorer damos click derecho sobre la aplicación y seleccionamos propiedades

Windows Forms

Windows Forms (o **formularios Windows**) es el nombre dado a la interfaz de programación de aplicación gráfica (API) que se incluye como parte de Microsoft .NET Framework, que proporciona acceso a los elementos de la interfaz de Microsoft Windows nativas envolviendo la API de Windows existente en código administrado.

Una aplicación de Windows Forms es aquella que se ejecuta en la computadora de escritorio. Una aplicación de formularios de Windows normalmente tendrá una colección de controles como etiquetas, cuadros de texto, cuadros de lista, etc.

- Una aplicación de formulario llamada `Forms1.cs`. Este archivo contendrá todo el código para la aplicación Windows Form.
- El programa principal llamado `Program.cs` es un archivo de código predeterminado que se crea cuando se crea una nueva aplicación en Visual Studio. Este código contendrá el código de inicio de la aplicación en su conjunto.

Conclusión

En conclusión, para poder empezar a usar Visual Studio se necesita obtener conocimientos acerca de los tipos de archivos que generan estos proyectos para poder empezar a realizar programas de manera correcta, es por eso que es necesario aprender varios conceptos acerca de lo que sería la aplicación de consola y Windows Form, la información recabada aquí tiene este objetivo, poder facilitar el aprendizaje y el conocimiento acerca de la programación orientada a objetos.