Delegates & events

Generalmente la comunidad presenta dificultades al diferenciar un evento y un delegado, no sirve de nada declarar eventos que están respaldados por una variable delegada del mismo nombre

**Tipos de *Delegates***

Los delegados se pueden pensar como un interfaz de un solo método. Tiene una firma de un mismo método, cuando esta es instanciada por un delegado puede llamarlo como si fuera el mismo método y firma. Los delegados permiten hacer instanciaciones con una misma firma y esa es su razón de ser. Tienen una referencia a un método y una referencia al objeto de destino al que se hará llamar el método.

Se declaran con la palabra reservada “Delegate”. Pueden ser únicos o anidados dentro de una clase. Aunque se haya declarado el “Delegate” no siempre estará declarado. Se usa la misma palabra reservada para crear instancias del “Delegate” usando métodos anónimos.

La diferencia entre Delegate y MultiDelegate es histórica, en versiones beta de .NET 1.0. Microsoft decidió fusionar ambos, aunque fue demasiado tarde para el lanzamiento realizar un cambio importante.

Los delegados que se crean son miembros heredados de sus tipos primordiales: un constructor con parámetros de objeto un IntPtr y tres métodos adicionales: Invoke, BeginInvoke y Endlvoke. Los métodos no se pueden heredar según las firmas ya que varían con la que es declarado un delegado

**Crear instancian delegadas**

Como se menciono anteriormente, clave de los datos de los delegados son el método al que se refiere el delegado y una referencia para llamar al método en el destino.

**Llamar instancias delegadas**

Se llaman como si fuesen propios métodos. El método al que se hace referencia la instancia de delegado se llama en el objeto de destino y se devuelve el resultado.

**Delegados combinados**

Los delegados se pueden combinar de modo que cuando llame al delegado, se llame a una lista completa de métodos, potencialmente con diferentes objetivos. La alternativa es una instancia de delegado que es una lista de delegado simples, todos del mismo tipo.

**Eventos**

Los eventos no son instancias delegadas. Los eventos son pares de métodos, adecuadamente decorados en IL para unirlos y hacer que los idiomas sepan que los métodos representan eventos

**Delegar instancias: otros métodos**

Anteriormente vimos cómo una llamada a someDelegate (10) es realmente una abreviatura de someDelegate.Invoke (10). Los tipos de delegados también pueden permitir un comportamiento asincrónico utilizando el par BeginInvoke / EndInvoke. Estos son opcionales en lo que respecta a la especificación CLI, pero los tipos de delegado de C # siempre los proporcionan. Siguen el mismo modelo para la ejecución asincrónica que el resto de .NET, lo que permite proporcionar un controlador de devolución de llamada, junto con un objeto para almacenar información de estado. Los delegados se ejecutan en subprocesos creados por el grupo de subprocesos del sistema.

**Conclusión**

Ahora que he leído el articulo me doy cuenta de la utilidad y beneficio de los delegados en C# ya que facilitan en gran medida a la hora de instanciar objetos y sus métodos realizando así muchos métodos pero con solo una instanciación del objeto y hacer múltiples tareas es fascinante la velocidad que otorga un delegado aunque este se ramifique de uniones de versiones anteriores de .NET en general tengo un mayor conocimiento y ahora podre usarlos mas apropiadamente.