Arquitectura de programación de aplicaciones de servicio

Las aplicaciones de servicio de Windows se basan en una clase que hereda de la clase System.ServiceProcess.ServiceBase. Reemplace los métodos procedentes de esta clase y defina su funcionalidad para determinar el comportamiento del servicio.

Las principales clases implicadas en la creación de servicios son las siguientes:

- **System.ServiceProcess.ServiceBase**: reemplace los métodos de la clase **ServiceBase** cuando cree un servicio y defina el código para determinar cómo funciona el servicio en esta clase heredada.
- System.ServiceProcess.ServiceProcessInstaller y System.ServiceProcess.ServiceInstaller: utilice estas clases para instalar y desinstalar su servicio.

Además, una clase denominada ServiceController se puede utilizar para manipular el servicio. Esta clase no está implicada en la creación de servicios, pero puede utilizarse para iniciar y detener el servicio, pasarle comandos y devolver una serie de enumeraciones.

Definir el comportamiento del servicio

En la clase de servicio, reemplace las funciones de clase base que determinan lo que ocurre cuando se cambia el estado del servicio en el Administrador de control de servicios. La clase **ServiceBase** expone los siguientes métodos, que puede reemplazar para agregar comportamientos personalizados.

Método	Reemplácelo para
OnStart	Indicar qué acciones deben ejecutarse cuando empieza a funcionar el servicio. Para que el servicio ejecute un trabajo útil, deberá escribir código en este procedimiento.
OnPause	Indicar qué debe ocurrir cuando se pausa el servicio.
OnStop	Indicar qué debe ocurrir cuando se detenga la ejecución del servicio.
OnContinue	Indicar qué debe ocurrir cuando el servicio reanuda su funcionamiento normal tras una pausa.
OnShutdown	Indicar qué debe ocurrir justo antes de que el sistema se cierre, en caso de que se esté ejecutando el servicio en ese momento.

OnCustomCommand	Indicar qué debe ocurrir cuando el servicio reciba un comando personalizado. Para obtener más información acerca de los comandos personalizados, consulte MSDN Online.
OnPowerEvent	Indicar cómo debe responder el servicio cuando se reciba un evento de administración de energía, por ejemplo, una batería agotada o una operación suspendida.

Nota

Estos métodos representan los estados a través de los cuales se mueve el servicio a lo largo de su ciclo de vida, es decir, las transiciones del servicio de un estado al siguiente. Por ejemplo, no podrá hacer que el servicio responda a un comando **OnContinue** antes de llamar a **OnStart**.

Hay otras propiedades y métodos de interés. Éstos incluyen:

- El método Run en la clase **ServiceBase**. Este es el punto de entrada principal al servicio. Al crear un servicio mediante la plantilla de servicios de Windows, se inserta código en el método `Main` de la aplicación para ejecutar el servicio. Este código tiene el siguiente aspecto:

C# | **VB** — Copiar

```
System.ServiceProcess.ServiceBase[]
ServicesToRun;
ServicesToRun = new
System.ServiceProcess.ServiceBase[]
  { new Service1() };
System.ServiceProcess.ServiceBase.Run(Serv
```

J# — Copiar

```
System.ServiceProcess.ServiceBase
ServicesToRun[];
ServicesToRun = new
System.ServiceProcess.ServiceBase[]
  { new Service1() };
System.ServiceProcess.ServiceBase.Run(Serv
```

Nota

Estos ejemplos utilizan una matriz de tipo **ServiceBase**, a la que puede agregarse cada uno de los servicios que contiene la aplicación para que todos los servicios se puedan ejecutar en conjunto. No obstante, si sólo va a crear un único servicio, puede elegir no utilizar la matriz y declarar simplemente un nuevo objeto que se herede de **ServiceBase** y, a continuación, ejecutarlo. Para obtener un ejemplo, vea Cómo: Crear servicios mediante programación.

- Una serie de propiedades de la clase **ServiceBase**. Estas propiedades determinan a qué métodos se puede llamar en el servicio. Por ejemplo, cuando la propiedad CanStop se establece en **true**, se puede llamar al método **OnStop** en su servicio. Cuando la propiedad CanPauseAndContinue se establece en **true**, se puede llamar a los métodos **OnPause** y **OnContinue**. Cuando establezca una de estas propiedades en **true**, es recomendable reemplazar y definir los procesos para los métodos asociados.

Nota

Su servicio debe reemplazar al menos **OnStart** y **OnStop** para ser útil.

También puede utilizar un componente denominado **ServiceController** para comunicarse y controlar el comportamiento de un servicio existente. Para obtener información sobre cómo utilizar **ServiceController**, vea Supervisar servicios de Windows.

Vea también

Tareas
Cómo: Crear servicios de Windows
Conceptos
Introducción a las aplicaciones de servicios de Windows
Otros recursos
Supervisar servicios de Windows