

Generic Delegate

A delegate can define its own type parameters. Code that references the generic delegate can specify the type argument to create a closed constructed type, just like when instantiating a generic class or calling a generic method, as shown in the following example:

```
. . .  
public delegate void Del<T>(T item);  
. . .  
public static void Notify(int i) { }  
. . .  
Del<int> m1 = new Del<int>(Notify);  
. . .
```

Generic delegates are especially useful in defining events based on the typical design pattern because the sender argument can be strongly typed and no longer has to be cast to and from Object.

Example:

```
delegate void StackEventHandler<T, U>(T sender, U eventArgs);  
  
class Stack<T>  
{  
    public class StackEventArgs : System.EventArgs { }  
    public event StackEventHandler<Stack<T>, StackEventArgs> stackEvent;  
  
    protected virtual void OnStackChanged(StackEventArgs a)  
    {  
        stackEvent(this, a);  
    }  
}  
  
class SampleClass  
{  
    public void HandleStackChange<T>(Stack<T> stack, Stack<T>.StackEventArgs  
args) { }  
}  
  
public static void Test()  
{  
    Stack<double> s = new Stack<double>();  
    SampleClass o = new SampleClass();  
    s.stackEvent += o.HandleStackChange;  
}
```