The CLR has support for a notion called **contexts**.  A context is a logical grouping of objects.  When a method call is made on an object that is inside your context, nothing in particular happens except for the method itself.  When a method call is made on an object that is outside your context, then "something" might happen **before** the method itself is **executed**.  All .NET classes derived from ContextBoundObject will be associated with a context at runtime.

An example of that "something" that can happen before the method is called would be enforcing synchronization.  An object that derives from ContextBoundObject and specifies the [Synchronization] attribute will get automatic synchronization services from the runtime.  This means that only one thread will ever be able to execute within the object's context.

The exitContext parameter of the WaitOne method specifies whether to leave the synchronization context before issuing the wait.  This enables reentrancy.  Here's a scenario where this is necessary:

Code Snippet

[Synchronization]  
public class MyCounter : ContextBoundObject {

 private int \_expectedCounterVal;  
 private int \_currentCounterVal;  
 private ManualResetEvent \_event = new ManualResetEvent(false);

 public void WaitUntilCounterIs(int counterVal) {  
  \_expectedCounterVal = counterVal;  
  \_event.WaitOne(TimeSpan.FromDays(1), true);  
 }

 public void IncrementCounter() {  
  if (++\_currentCounterVal >= \_expectedCounterVal) {  
   \_event.Set();  
  }  
 }  
}

In this case, if the WaitUntilCounterIs method is issued without the exitContext=true parameter of the WaitOne method, then no other thread will be able to ever call the IncrementCounter method on the object, resulting in a deadlock.  However, once exitContext=true is specified, other threads can enter the IncrementCounter method and signal the event at some point, even though the WaitUntilCounterIs method has not yet returned.

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/ea5aea17-555c-4549-8d32-bbab14b35a77/waitone-exitcontext?forum=netfxbcl>