GC Notifications

.NET Framework 3.5.1 added notification features which allow you to determine when a full GC is about to take place, allowing you to take action if necessary. A server application, for example, could redirect requests to other servers. Basically, with NET Framework 3.5.1, you can do the following things:

- register for full GC notifications
- determine when a full GC is approaching
- determine when a full GC has finished
- unregister for full GC notifications.

In Listing 3.6 there is some example code that continually adds 1,000 bytes to an `ArrayList` in order to create increasing memory pressure, and I've added the line...

```
GC.RegisterForFullGCNotification(10, 10);
```

Listing 3.5.

...to indicate that I want to receive full GC notifications. The parameters are threshold values for the SOH and LOH respectively. They are values between 1 and 99, and indicate when the notification should be raised, based on the number of objects promoted to Gen 2 (for the SOH) and the number of objects allocated (for the LOH). A large value will notify you a longer time before collection takes place, and vice versa. The trick is to experiment and get these settings about right for your situation, so that you have enough time to take appropriate action.

```
System.Collections.ArrayList data = new ArrayList();
bool carryOn = true;
private void b1_Click(object sender, EventArgs e)
{
   GC.RegisterForFullGCNotification(10, 10);
   Thread t = new Thread(new ThreadStart(ChecktheGC));
   t.Start();
   while (carryOn)
   {
      data.Add(new byte[1000]);
   }
   GC.CancelFullGCNotification();
}
```

Listing 3.6: Registering for full GC notification.

In my example (Listing 3.6) I am spawning a thread to monitor the GC status calling method " `ChecktheGC` " (see Listing 3.7).

In Listing 3.7 I have a `while` loop continually checking the return from `GC.WaitForFullGCApproach()` .

When it returns `GCNotificationStatus.Succeeded` I know a full GC is approaching and I can take some action, e.g. do own collection, or redirect requests, etc.

```
private void ChecktheGC()
{
 while (true) // Wait for an Approaching Full GC
 {
    GCNotificationStatus s = GC.WaitForFullGCApproach();
    if (s == GCNotificationStatus.Succeeded)
    {
       Console.WriteLine("Full GC Nears");
       break;
    }
 }
 while (true) // Wait until the Full GC has finished
 {
   GCNotificationStatus s = GC.WaitForFullGCComplete();
   if (s == GCNotificationStatus.Succeeded)
   {
      Console.WriteLine("Full GC Complete");
      break;
   }
 }
 carryOn = false;
}
```

Listing 3.7: Polling for full GC notification messages.

Once the approaching GC has been detected and handled, I then wait for the full GC to take place so that I can take additional action. In this case, I may stop redirecting requests and start processing them again.

If you have a situation where garbage collection is causing you a problem, then knowing when it's about to happen may be useful to you. When and if that's the case, you may find the above pattern useful, though rather cumbersome.