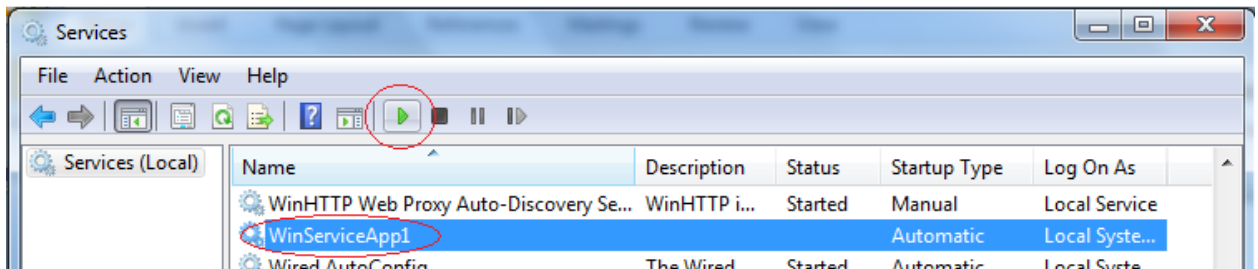


Debug Windows Service

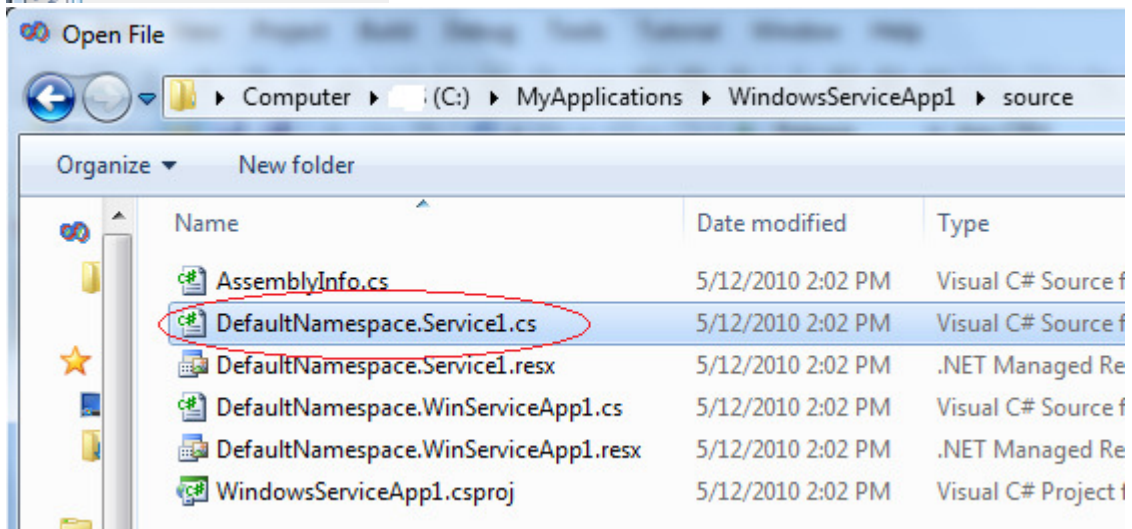
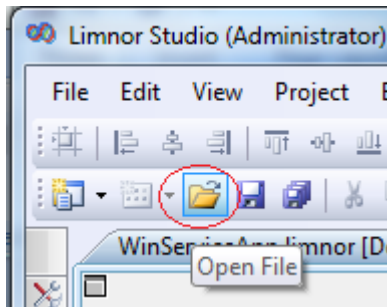
Attach Service Process to Debugger

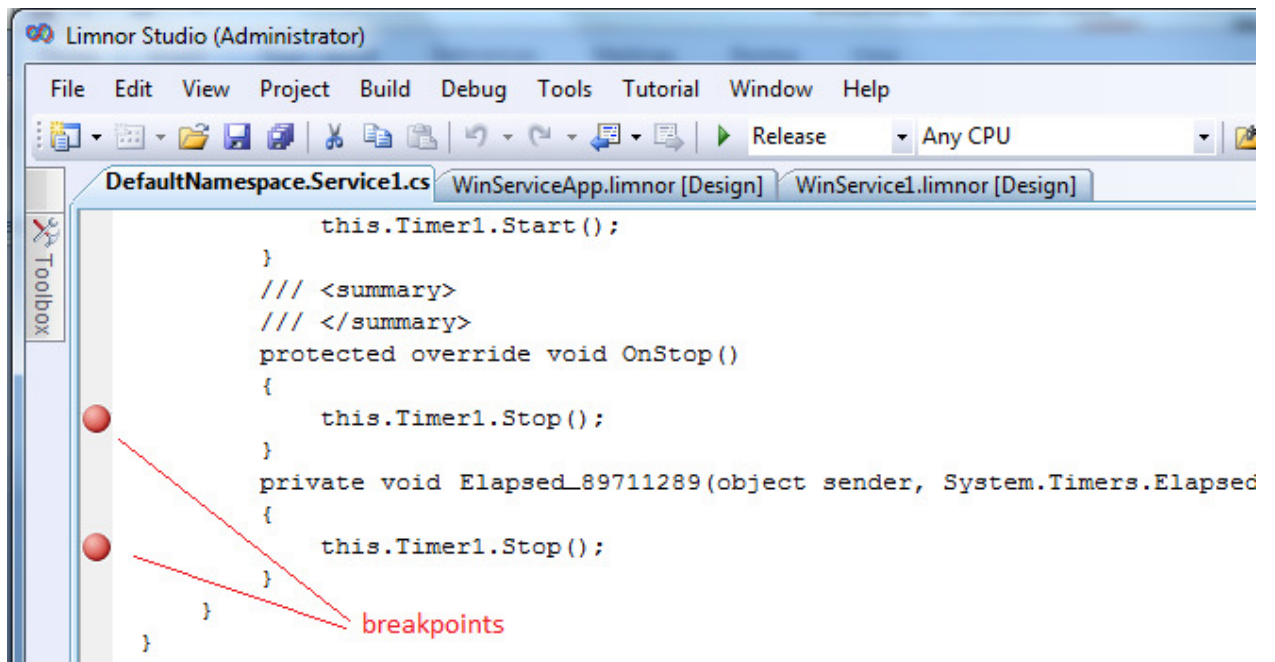
A Windows service cannot be started from within Limnor Studio. To debug a Windows service, following the steps below:

1. Install the Windows service
2. Start the service

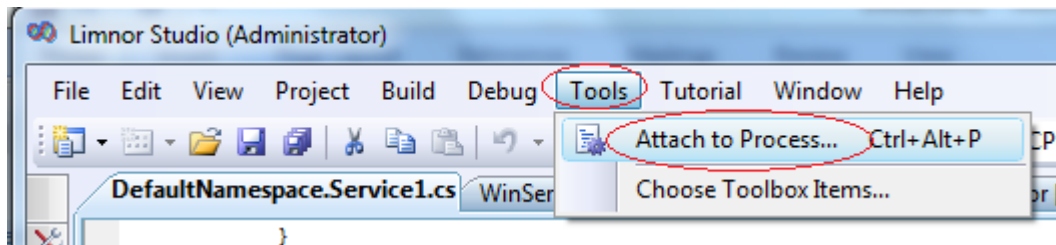


3. Open the C# source code for the service, set breakpoints in desired locations

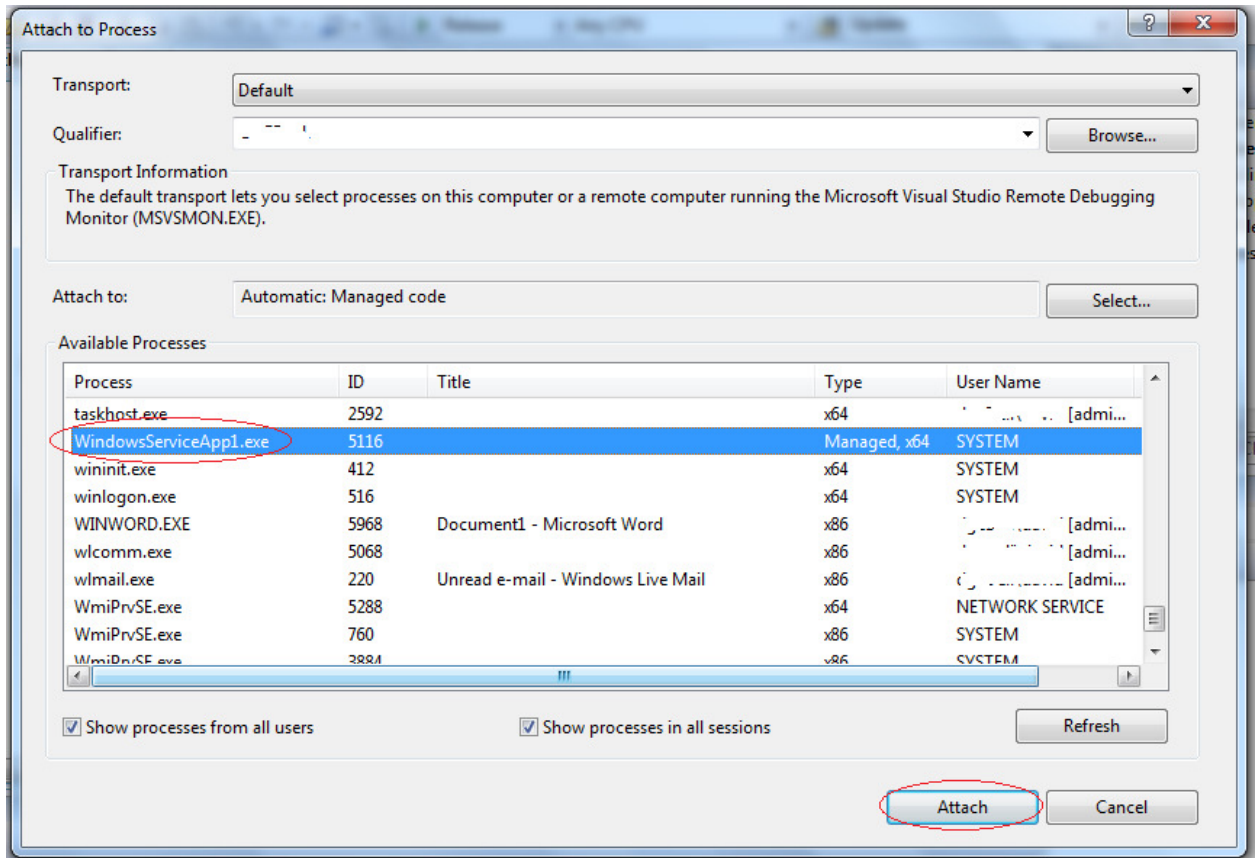




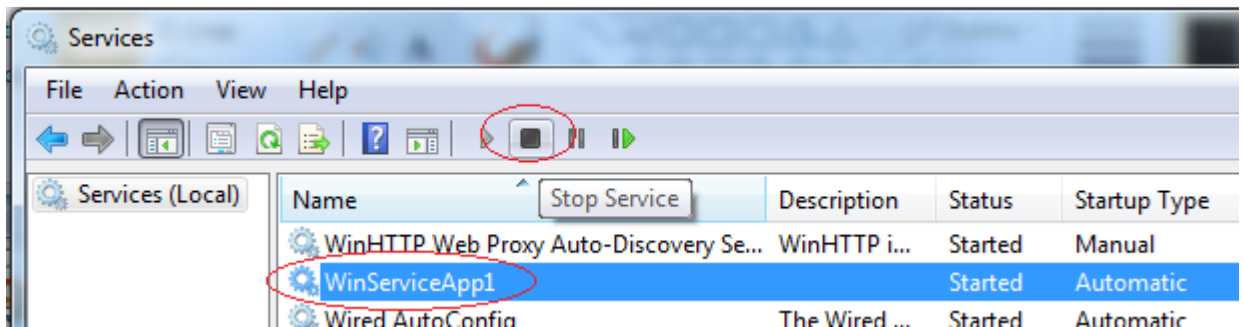
4. Attach the service to the debugger



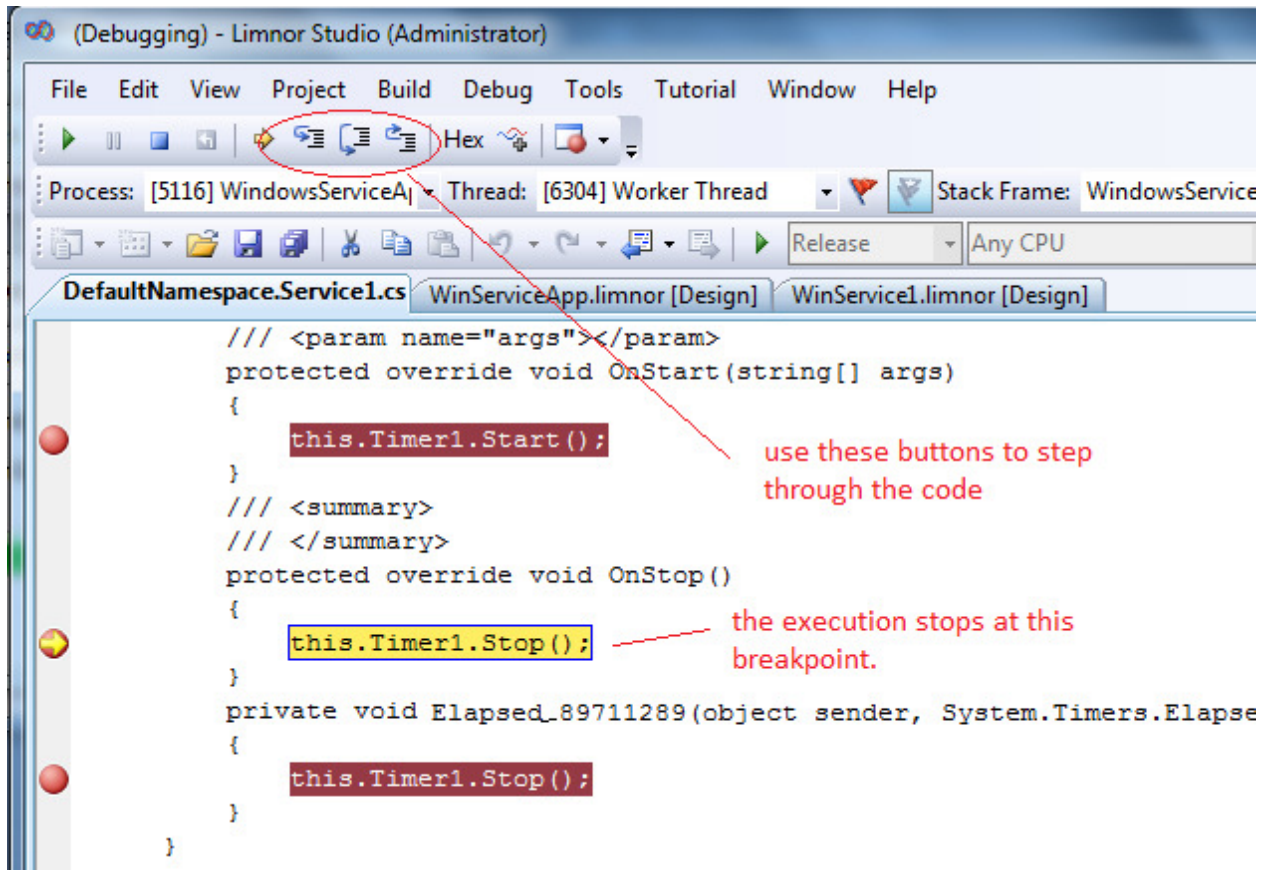
Select the service process, click Attach



5. Wait for the service to reach at a breakpoint
For example, let's stop the service:



At the time of stopping the service, OnStop method is called. The breakpoint in OnStop is hit:

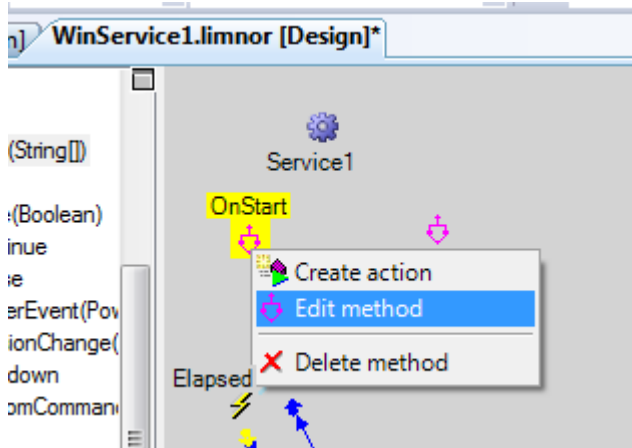


Hit Breakpoint in OnStart Method

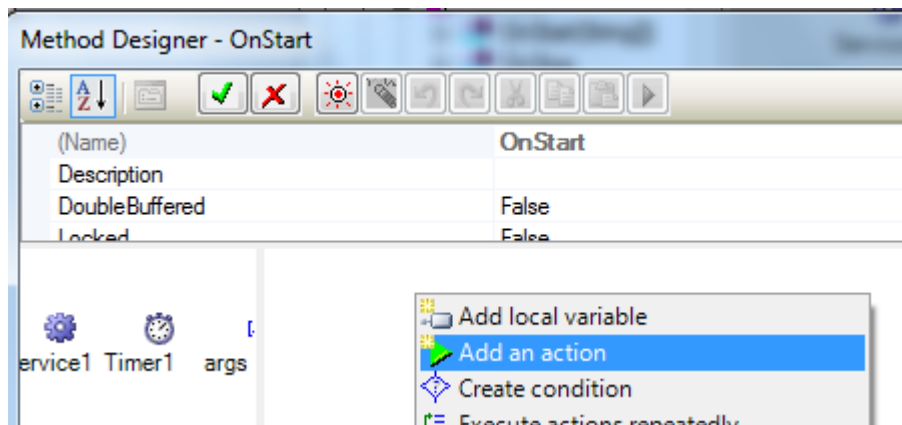
Note that the breakpoint we set in OnStart method will never get hit. When we start the service at step 2, the OnStart method is called. But at this time we have not attached the process to the debugger yet.

One way to workaroud it can be to add code to pause the execution. For example, let the execution pause for 20 seconds. Within these 20 seconds, we finish step 3 and step 4.

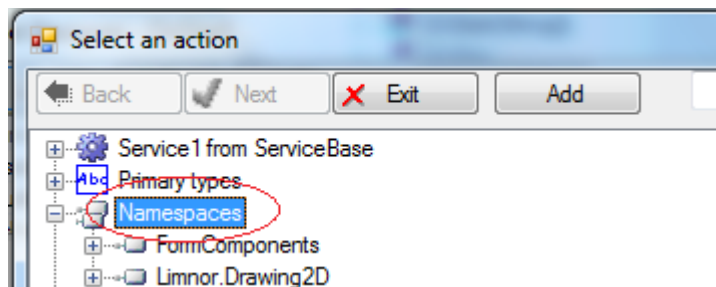
Right-click OnStart icon, choose "Edit method"



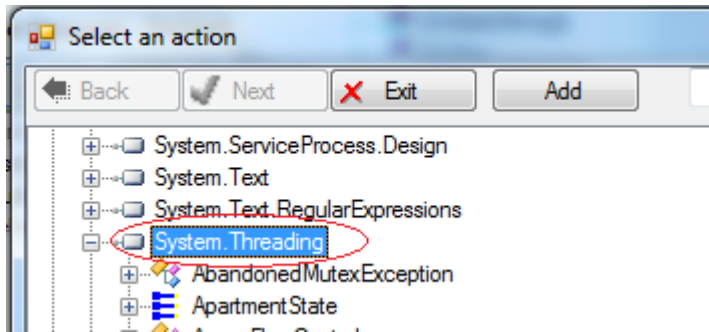
Right-click in the middle pane, choose “Add an action”:



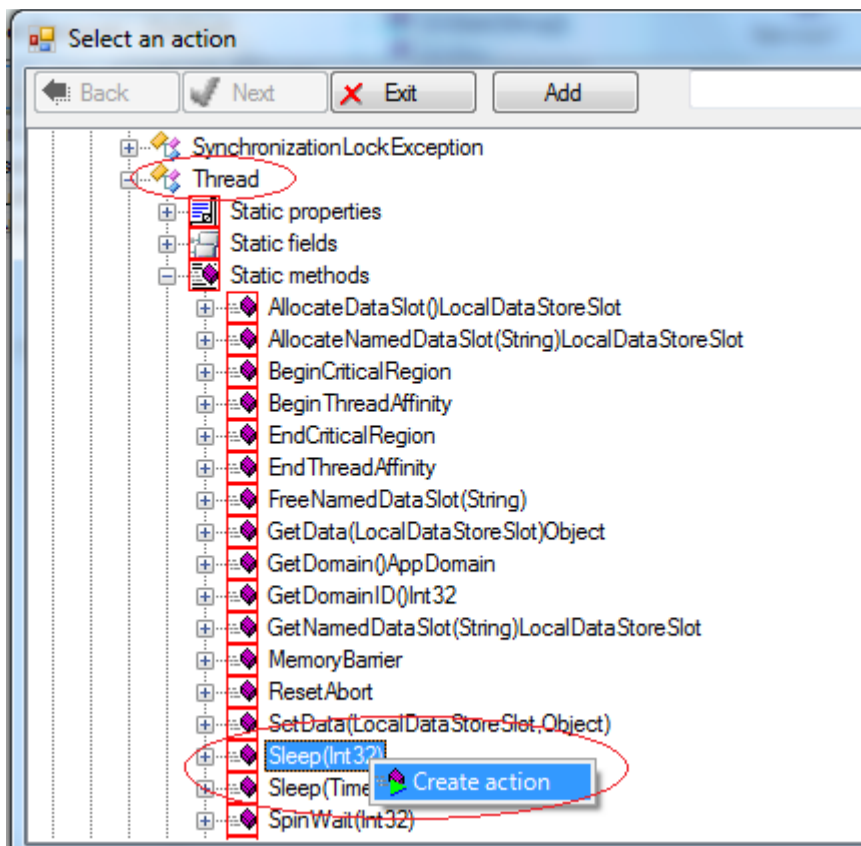
Expand Namespaces:



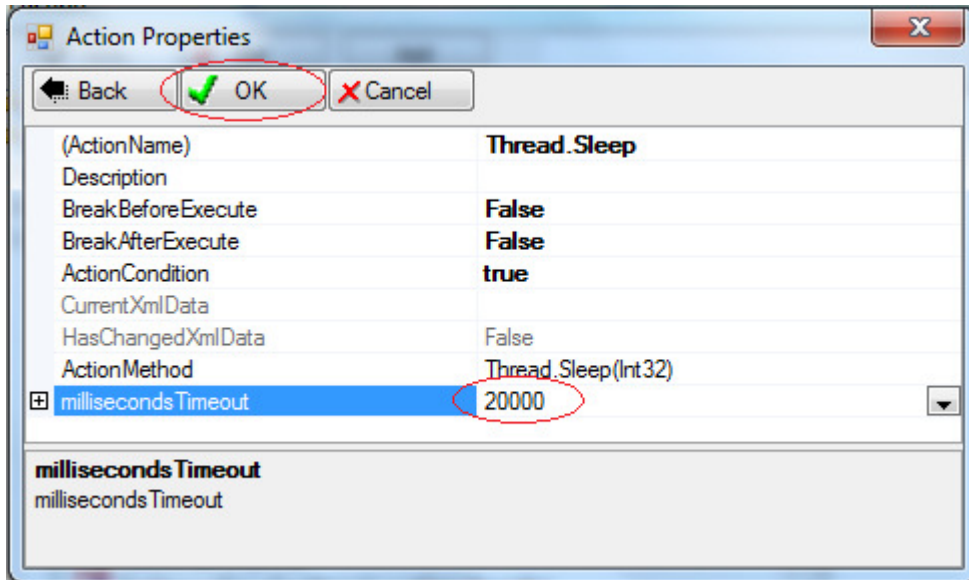
Expand System.Threading node:



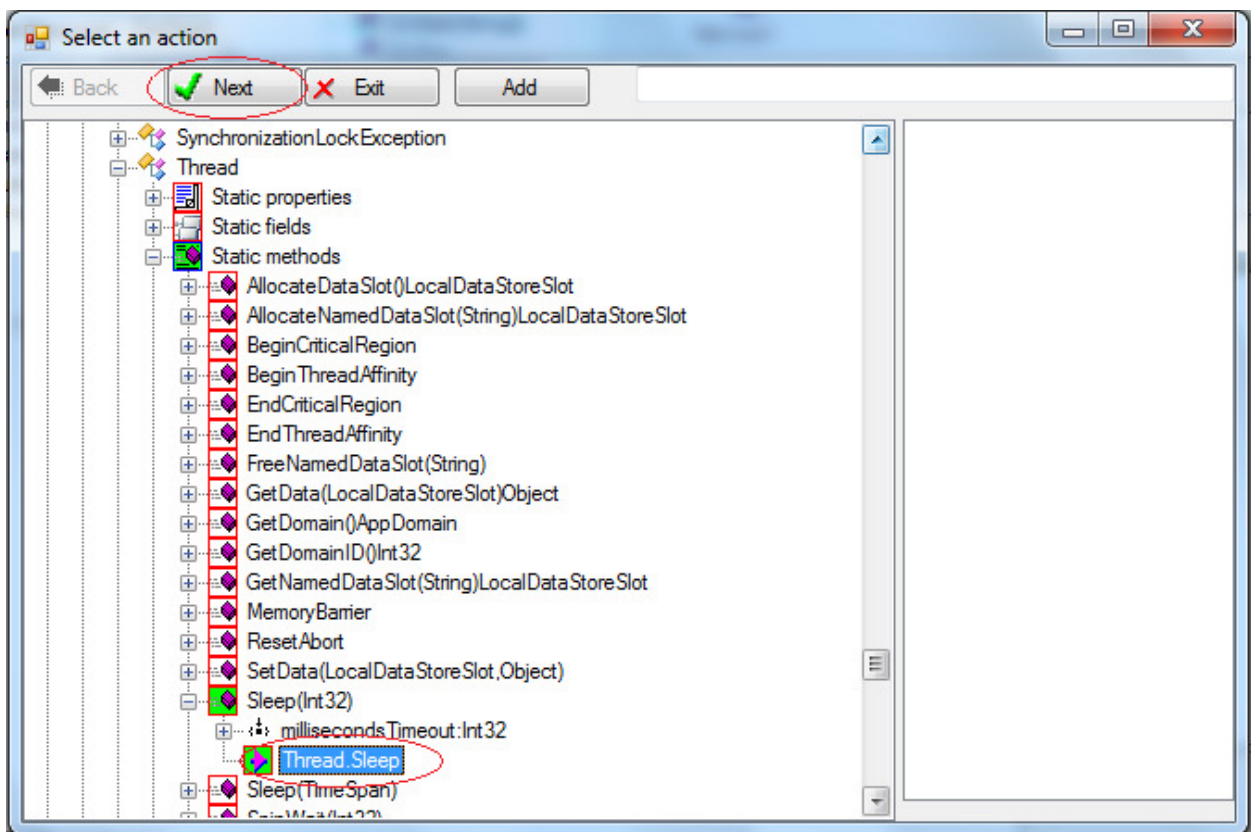
Find Sleep method under Thread class, right-click it and choose "Create action":



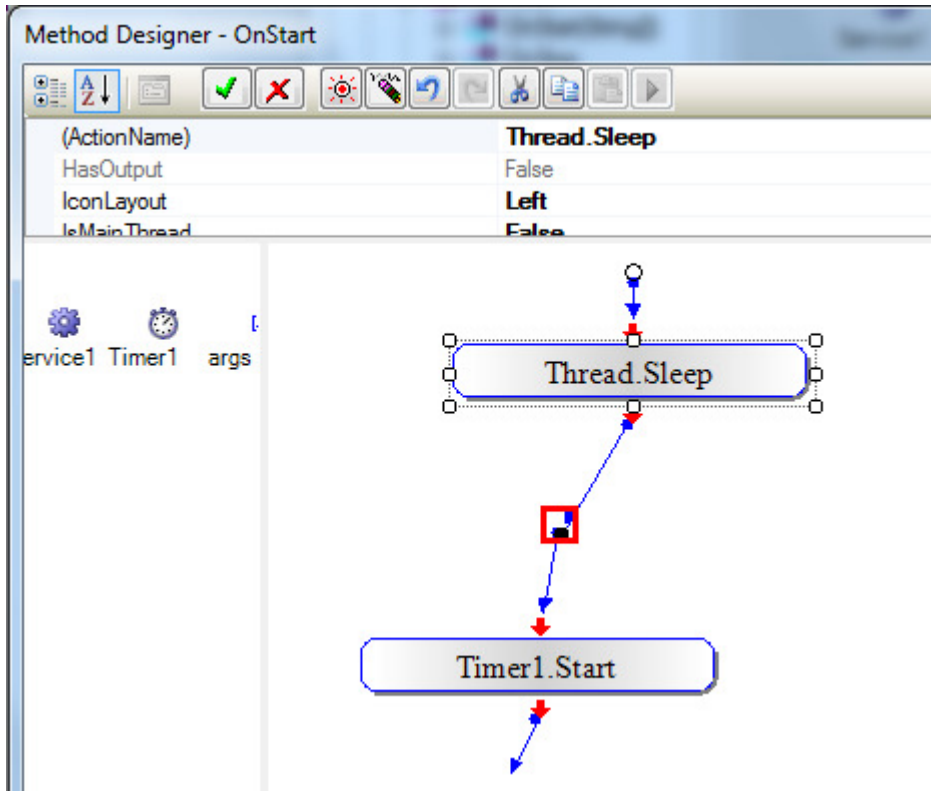
Set sleep time to 20000 milliseconds (20 seconds):



A new action, Thread.Sleep, is created. Select it, click Next:

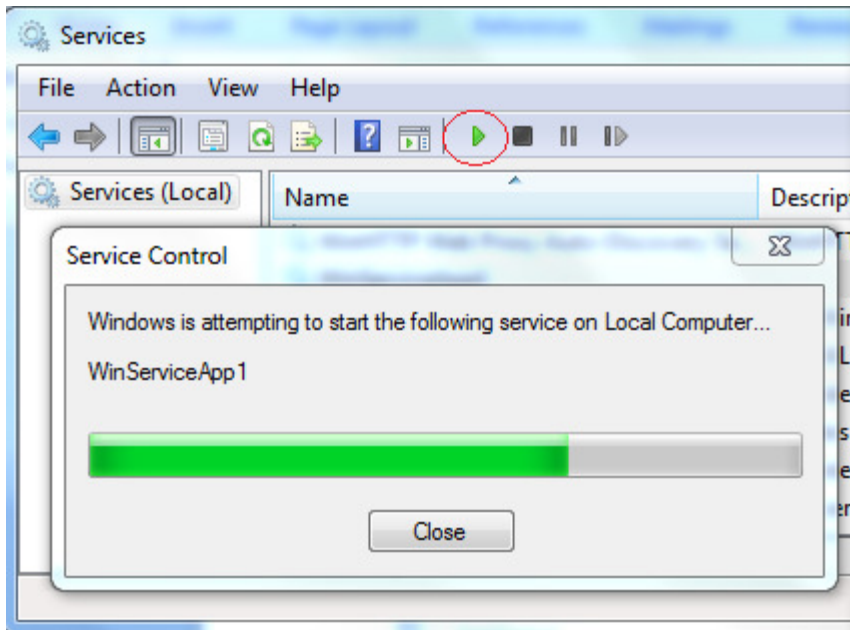


The new action appears in the method editor. Link the action with an existing action to form a sequential execution order. The Sleep action should be executed first to make other action(s) wait.

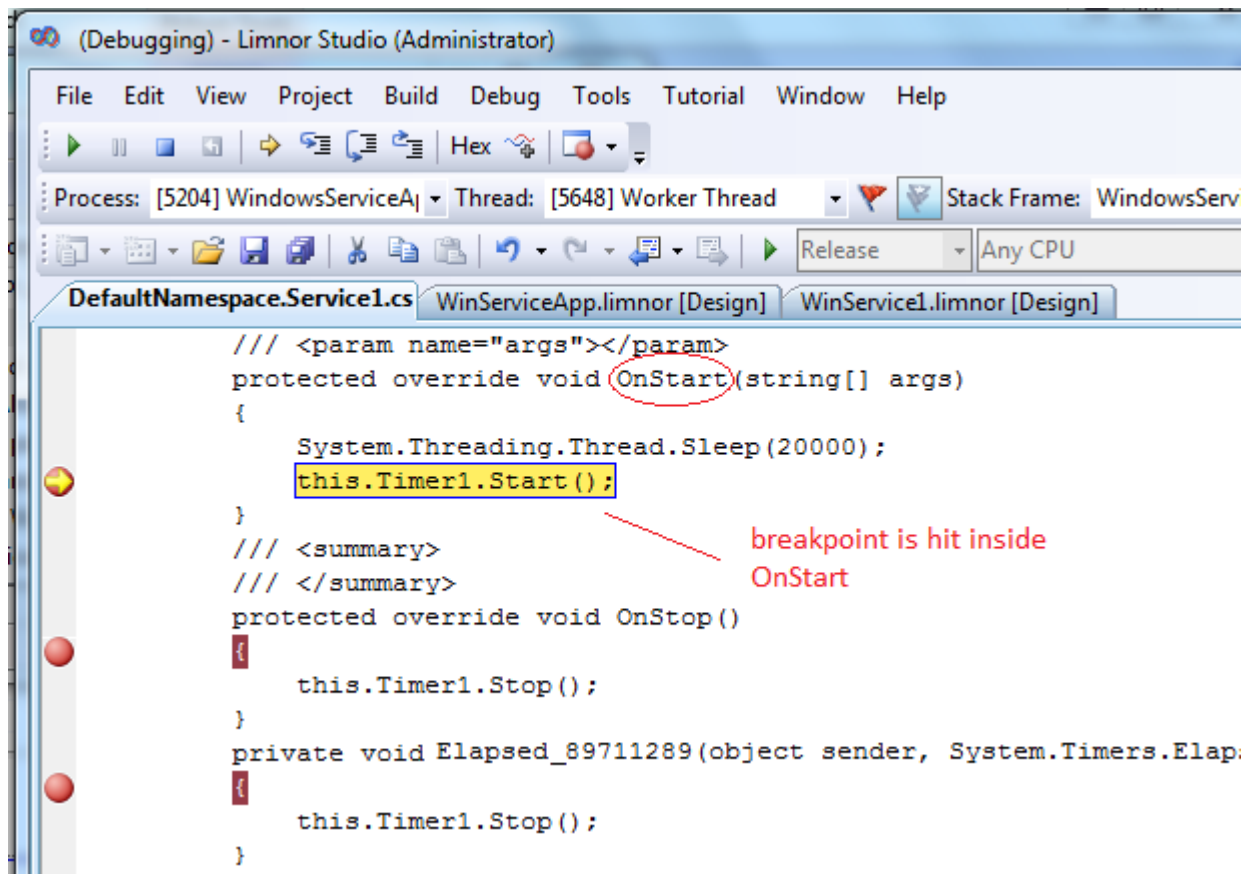


Stop the service. Compile the project to re-create the service program.

Start the service again. You will see the starting progress dialogue box:

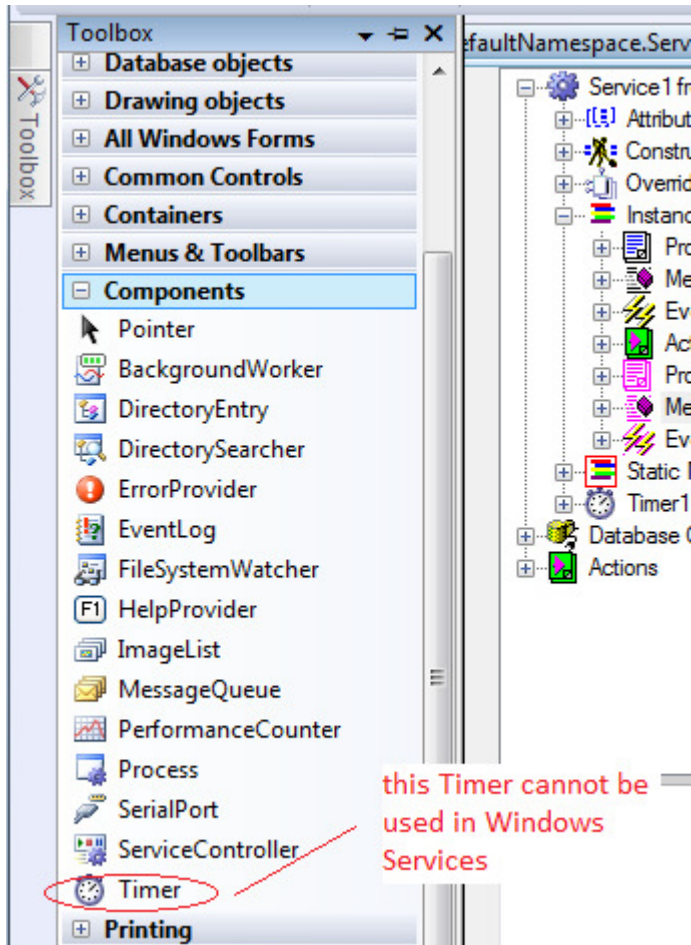


Do not wait for this dialogue box to close because it is executing the Sleep action, waiting for us to attach the process. We should quickly attach the service process to the debugger and set breakpoints in OnStart method. In 20 seconds, the Sleep action finishes and the breakpoint in the OnStart method is hit:

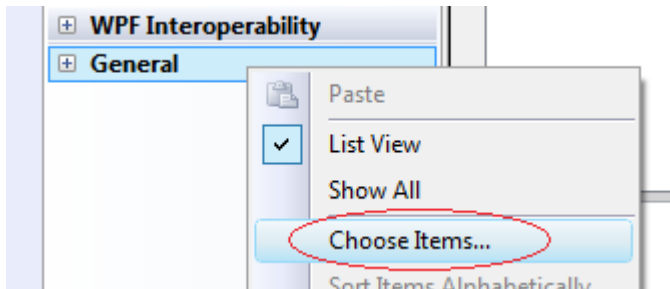


Do not forget to remove the Sleep action when you release your program to your customers. You do not want your customers to wait for 20 seconds for the service to start.

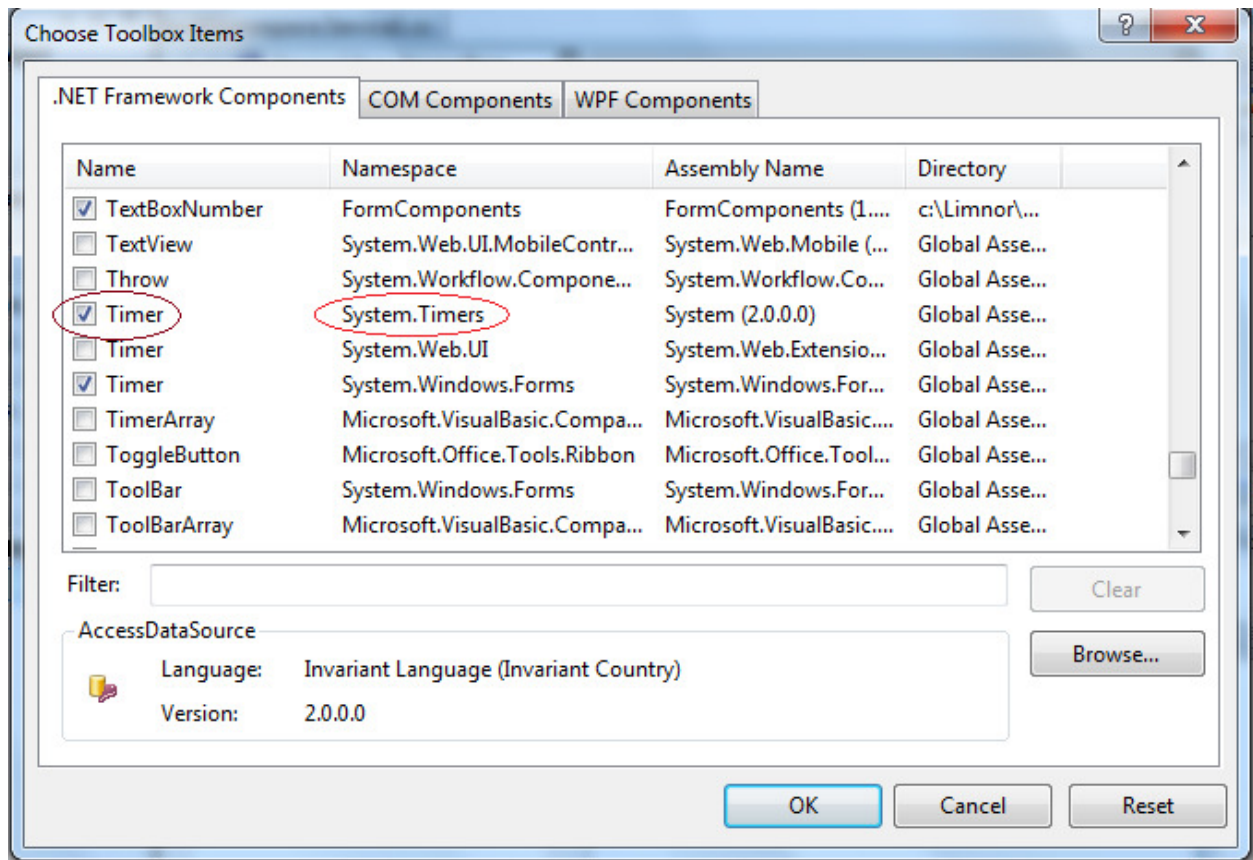
In this sample we used a Timer in the service. Note that this Timer is not the one in the Toolbox:



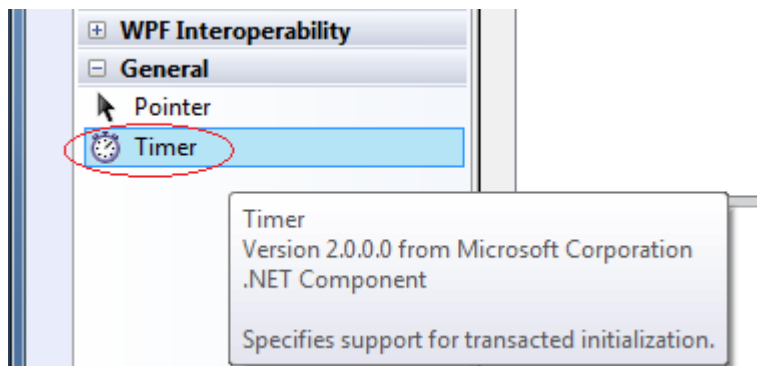
To use a Timer in service, use the Timer from System.Timer namespace. Right-click a tab in the Toolbox, choose "Choose Items...":



Check the Timer from System.Timer:



This is the Timer that can be used in a service:



This Timer has an Elapsed event. You may assign actions to this event to be executed when time is up.