Using TaskMeter Directly

(without using the rest of the Alpaca Unit Testing Framework)

1. Install Alpaca (this will put the taskometer dll into the GAC).
2. Add 3 project references:  
     
    **Microsoft.Concurrency.UnitTestingFramework.dll** **Microsoft.Concurrency.TaskoMeter.dll**  
   (Both are installed with Alpaca and should be visible in the Add Reference dialog box in VS)  
     
    **Scenario.Managed.dll**  
   (to find this one you have to browse to C:\Program Files (x86)\Microsoft Alpaca\bin)
3. Include the namespace:

using Microsoft.Concurrency.TestTools.UnitTesting;

using Microsoft.Concurrency.TestTools.TaskoMeter;

1. In your code, delimit the start/end of your measured section  
     
   Metering.Start()  
   /\* the section of code within which to perform measurements \*/  
   Metering.End()
2. To measure details in your code, construct TaskMeter objects with name and color, then start and stop them. Each TaskMeter is displayed as a separate row in the timeline. Meters can measure multiple intervals, and can be used by multiple threads concurrently.  
     
   Construct (must be called before Metering.Start())  
    TaskMeter mymeter = new TaskMeter(“sometask”, TaskMeter.Color.Orange)  
     
   Measure (must be called in between Metering.Start() and Metering.End())  
    mymeter.Start();  
    /\* code of interest \*/  
    mymeter.End();

# Example Screenshot

The blue bar shows the overall time between Metering.Start() and Metering.End()

There are two taskmeters, “tile” and “renderloop”

Tile shows the rendering time taken to render a tile (and you can see there are many threads performing these), while renderloop shows the time taken in each frame.

