Runtime GC Latency Control

.NET Framework 3.5.1 allows the latency of the GC to be controlled programmatically, which ultimately overrides the `gcConcurrent` setting within your `config` file.

To achieve this, the `System.Runtime.GCSettings.LatencyMode` property can be set to one of three modes.

- **`GCLatencyMode.Batch`** – designed to give maximum throughput and performance for sections of an app where UI responsiveness isn't important.
- **`GCLatencyMode.LowLatency`** – this mode reduces the impact of GC to a minimum, which is ideal for times when things like UI responsiveness are critical, e.g. animation.
- **`GCLatencyMode.Interactive`** – Workstation GC with Concurrency switched on, giving a balance between GC efficiency and app responsiveness.

An obvious use of `LatencyMode` is to change it for a short period during execution of critical code that needs maximum UI or batch processing performance, and then change it back on completion.

```
using System.Runtime;
…
// Store current latency mode
GCLatencyMode mode = GCSettings.LatencyMode;
// Set low latency mode
GCSettings.LatencyMode = GCLatencyMode.LowLatency;
try
{
   // Do some critical animation work
}
finally
{
   // Restore latency mode
   GCSettings.LatencyMode = mode;
}
```

Listing 3.4: Using GC `LatencyMode`.