GET_PROC

```
using System;

using System.Diagnostics;

class GetProc

{

 public static void Main()

 {

  Process thisProc = Process.GetCurrentProcess();

  string procName = thisProc.ProcessName;

  DateTime started = thisProc.StartTime;

  int procID = thisProc.Id;

  int memory = thisProc.VirtualMemorySize;

  int priMemory = thisProc.PrivateMemorySize;

  int physMemory = thisProc.WorkingSet;

  int priority = thisProc.BasePriority;

  ProcessPriorityClass priClass = thisProc.PriorityClass;

  TimeSpan cpuTime = thisProc.TotalProcessorTime;

  Console.WriteLine("Process: {0}, ID: {1}", procName, procID);

  Console.WriteLine("  started: {0}", started.ToString());

  Console.WriteLine("  CPU time: {0}", cpuTime.ToString());

  Console.WriteLine(

   "  priority class: {0} priority: {1}", priClass, priority);

  Console.WriteLine("  virtual memory: {0}", memory);
```

```
   Console.WriteLine("  private memory: {0}", priMemory);
   Console.WriteLine("  physical memory: {0}", physMemory);
   Console.WriteLine("\n  trying to change priority...");
   thisProc.PriorityClass = ProcessPriorityClass.High;
   priClass = thisProc.PriorityClass;
   Console.WriteLine("  new priority class: {0}", priClass);
  }
}
```

LIST_PROC

```
using System;
using System.Diagnostics;
class ListProcs
{
  public static void Main()
  {
   int totMemory = 0;
   Console.WriteLine("Info for all processes:");
   Process[] allProcs = Process.GetProcesses();
   foreach(Process thisProc in allProcs)
   {
     string procName = thisProc.ProcessName;
     DateTime started = thisProc.StartTime;
```

```
    int procID = thisProc.Id;

    int memory = thisProc.VirtualMemorySize;

    int priMemory = thisProc.PrivateMemorySize;

    int physMemory = thisProc.WorkingSet;

    totMemory += physMemory;

    int priority = thisProc.BasePriority;

    TimeSpan cpuTime = thisProc.TotalProcessorTime;

    Console.WriteLine("Process: {0}, ID: {1}", procName, procID);

    Console.WriteLine("  started: {0}", started.ToString());

    Console.WriteLine("  CPU time: {0}", cpuTime.ToString());

    Console.WriteLine("  virtual memory: {0}", memory);

    Console.WriteLine("  private memory: {0}", priMemory);

    Console.WriteLine("  physical memory: {0}", physMemory);

   }

  Console.WriteLine("\nTotal physical memory used: {0}", totMemory);

 }

}
```

GET_THREAD

```
using System;

using System.Diagnostics;

class GetThreads
```

```
{
  public static void Main()
  {
   Process thisProc = Process.GetCurrentProcess();
   ProcessThreadCollection myThreads = thisProc.Threads;
   foreach(ProcessThread pt in myThreads)
   {
     DateTime startTime = pt.StartTime;
     TimeSpan cpuTime = pt.TotalProcessorTime;
     int priority = pt.BasePriority;
     ThreadState ts = pt.ThreadState;
     Console.WriteLine("thread: {0}", pt.Id);
     Console.WriteLine("  started: {0}", startTime.ToString());
     Console.WriteLine("  CPU time: {0}", cpuTime);
     Console.WriteLine("  priority: {0}", priority);
     Console.WriteLine("  thread state: {0}", ts.ToString());
   }
  }
}
```

LIST_THREAD

```
using System;
using System.Diagnostics;
class ListThreads
```

```
{
  public static void Main()
  {
   Process[] allProcs = Process.GetProcesses();
   foreach(Process proc in allProcs)
   {
     ProcessThreadCollection myThreads = proc.Threads;
     Console.WriteLine("process: {0}, id: {1}", proc.ProcessName, proc.Id);
     foreach(ProcessThread pt in myThreads)
     {
      DateTime startTime = pt.StartTime;
      TimeSpan cpuTime = pt.TotalProcessorTime;
      int priority = pt.BasePriority;
      ThreadState ts = pt.ThreadState;
      Console.WriteLine(" thread: {0}", pt.Id);
      Console.WriteLine("  started: {0}", startTime.ToString());
      Console.WriteLine("  CPU time: {0}", cpuTime);
      Console.WriteLine("  priority: {0}", priority);
      Console.WriteLine("  thread state: {0}", ts.ToString());
     }
   }
  }
}
```