LOGMAN.exe

Manage Performance Monitor & performance logs from the command line.

```
Syntax
      Logman [create | query | start | stop | delete | update | import | export] [options]

Verbs:
  create           Create a new data collector.
  query            Query data collector properties.
                   If no name is given all data collectors are listed.
  start            Start an existing data collector and set the begin time to manual.
  stop             Stop an existing data collector and set the end time to manual.
  delete           Delete an existing data collector.
  update           Update an existing data collector's properties.
  import           Import a data collector set from an XML file.
  export           Export a data collector set to an XML file.

Adverbs:
  counter          Create a counter data collector.
  trace            Create a trace data collector.
  alert            Create an alert data collector.
  cfg              Create a configuration data collector.
  providers        Show registered providers.

Options (counter):
   -c path [path [...]]
                   The performance counters to collect.
                   To collect remotely, prefix with the \\machine name.
   -cf filename    File listing performance counters to collect,one per line.
   -f {bin|bincirc|csv|tsv|sql}
                   The log format for the data collector. For SQL database format,
                   you must use the -o option in the command line with the DNS!log option.
                   The defaults is binary.
   -sc value       Maximum number of samples to collect with a performance counter data collector.
   -si [[hh:]mm:]ss  Sample interval for performance counter data collectors.

Options (trace):
   -f {bin|bincirc|csv|tsv|sql}
                   The log format for the data collector. For SQL database format,
                   you must use the -o option in the command line with the DNS!log option.
                   The defaults is binary.
   -mode trace_mode  Event Trace Session logger mode.
   -ct {perf|system|cycle}
                   The clock resolution to use when logging the time stamp for
                   each event: query performance counter, system time, or CPU cycle.
   -ln logger_name   Logger name for Event Trace Sessions.
   -ft [[hh:]mm:]ss  Event Trace Session flush timer.
   -[-]p provider [flags [level]]
                   A single Event Trace provider to enable.
                   The terms 'Flags' and 'Keywords' are synonymous in this context.
```

```
   -pf filename      File listing multiple Event Trace providers to enable.
   -[-]rt            Run the Event Trace Session in real-time mode.
   -[-]ul            Run the Event Trace Session in user mode.
   -bs value         Event Trace Session buffer size in kb.
   -nb min max       Number of Event Trace Session buffers.

Options (alert):
   -[-]el             Enable/Disable event log reporting.
   -th threshold [threshold [...]]
                      Specify counters and a threshold values for an alert.
   -[-]rdcs name      Data collector set to start when alert fires.
   -[-]tn task        Scheduled Task to run when alert fires.
   -[-]targ argument  Scheduled Task arguments.
   -si [[hh:]mm:]ss   Sample interval for performance counter data collectors.

Options (cfg):
   -[-]ni                    Enable/Disable network interface query.
   -reg path [path [...]]    Registry values to collect.
   -mgt query [query [...]]  WMI objects to collect.
   -ftc path [path [...]]    Full path to the files to collect.

Options:
   -?                        Display context sensitive help.
   -s computer               Perform the command on specified remote system.
   -config filename          Settings file containing command options.
   [-n] name                 Name of the target object.
   -pid pid                  Process identifier.
   -xml filename             Name of the XML file to import or export.
   -as                       Perform the requested operation asynchronously.
   -[-]u user [password]     User to Run As. Entering a * for the password produces a prompt
                             The interactive password is not displayed on screen.

   -m [start] [stop]         Change to manual start or stop instead of a scheduled begin or end time.
   -rf [[hh:]mm:]ss          Run the data collector for the specified period of time.
   -b dd/MM/yyyy HH:mm:ss[AM|PM] Begin the data collector at specified time.
   -e dd/MM/yyyy HH:mm:ss[AM|PM] End the data collector at specified time.
   -[-]r                     Repeat the data collector daily at the specified begin and end times.

   -o {path|dsn!log}         Path of the output log file or the DSN and log set name in a
                             SQL database. The default path is '%systemdrive%\PerfLogs\Admin'.
   -[-]a                     Append to an existing log file.
   -[-]ow                    Overwrite an existing log file.
   -[-]v {nnnnnn | mmddhhmm} Attach file versioning information to the end of the log name.
   -[-]rc task               Run the command specified each time the log is closed.
   -[-]max value             Maximum log file size in MB or number of records for SQL logs.
   -[-]cnf [[hh:]mm:]ss      Create a new file when the specified time has elapsed or when the max size is exceeded.

   -y                        Answer yes to all questions without prompting.
   -fd                       Flush all the active buffers of an existing Event Trace Session to disk.
   -ets                      Send commands to Event Trace Sessions directly without saving or scheduling.
```

Note: Where [-] is listed, an extra - negates the option. For example --u turns off the -u option.

Examples

Create a counter to Log the % Processor Time on the local machine:

```
C:\> set _mycounters="\Processor(_Total)\%% Processor Time"
C:\> set _mylogfile=C:\docs\ss64.blg
C:\> Logman.exe create counter ss64_CPU -f bincirc -v mmddhhmm -max 250 -c %_mycounters% -o %_mylogfile%
```

Start the counter running (and then run whatever other process you wish to monitor):

```
C:\> Logman.exe start ss64_CPU
```

Stop the counter:

```
C:\> Logman.exe stop ss64_CPU
The above creates a file like C:\docs\ss64_09031235.blg
```

To convert this to text/CSV format:

```
C:\> relog C:\docs\ss64_09031235.blg -f csv -o proc_time.csv -t 2
```

We can now delete the counter (unless intending to reuse it again):

```
C:\> Logman.exe delete ss64_CPU
```

Run a scheduled task if % Processor Time > 5, this assumes that "demo_task" already exists:

```
C:\> Logman.exe create alert ss64_alert -th "\Processor(_Total)\%% Processor Time>5" -tn "demo_task"
```

More examples:

```
logman start perf_log
logman update perf_log -si 10 -f csv -v mmddhhmm

logman create counter perf_log -c "\Processor(_Total)\% Processor Time"
logman create counter perf_log -c "\Processor(_Total)\% Processor Time" -max 10 -rf 01:00

logman create trace trace_log -nb 16 256 -bs 64 -o c:\logfile

logman create alert new_alert -th "\Processor(_Total)\% Processor Time>50"

logman create cfg cfg_log -reg "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\\"
logman create cfg cfg_log -mgt "root\cimv2:SELECT * FROM Win32_OperatingSystem"

logman query providers
logman query providers Microsoft-Windows-Diagnostics-Networking

logman start process_trace -p Microsoft-Windows-Kernel-Process 0x10 win:Informational -ets
logman start process_trace -p Microsoft-Windows-Kernel-Process -mode newfile -max 1 -o output%d.etl -ets

logman start usermode_trace -p "Service Control Manager Trace" -ul -ets
logman query usermode_trace -p "Service Control Manager Trace" -ul -ets
logman stop usermode_trace -p "Service Control Manager Trace" -ul -ets

logman start "NT Kernel Logger" -o log.etl -ets
logman start "NT Kernel Logger" -p "Windows Kernel Trace" (process,thread) -ets
```

“Painting is just another way of keeping a diary” ~ Pablo Picasso

Related commands

Microsoft Help page
LODCTR - Load PerfMon performance counters.
SYSMON - Monitor and log system activity to the Windows event log.
TypePerf - Write performance data to a log file.
EVENTCREATE - Add a message to the Windows event log.
Equivalent PowerShell: New-Object System.Diagnostics.PerformanceCounter.
