Kernrate Usage Guide

Purpose

Kernrate is a sample profiling tool meant to help identify primarily where CPU time is being spent. Both Kernel and user mode processes can be profiled separately or simultaneously. With proper support, Kernrate can also be used to profile CPU events (sources) other than time, depending on the CPU type.

Supported OS Platforms

The version of Kernrate documented here will run under Windows 2000, Windows XP and Windows Server 2003.

Supported Hardware

The version of Kernrate documented here supports Intel x86 processors (Pentium and above), AMD equivalent processors as well as Intel 64 bit and AMD 64 bit platforms. The "Time" source is supported on all platforms. The degree of support for sources other than "Time" varies, depending on the amount of HAL support for the counters available with each processor type.

Method of Operation

Kernrate opens the process selected to be monitored and loads the process modules. After starting the profile, each module accumulates "hits" based on the number of CPU event occurrences that happened in the module address space. For pre-selected modules ("zoomed" modules), the module address space is divided into "buckets". The default bucket size is currently 16 bytes (the minimum is 4 bytes). After initializing the profile, the kernel adds hit counts to the appropriate buckets, based on the addresses where CPU events were spent in. The frequency of the profile can be controlled by setting the number of event occurrences per hit. Therefore a setting of 2,000 event occurrences per hit will generate 10 times the sample rate as compared to a setting of 20,000 event occurrences per hit. After the profiling has ended, Kernrate translates the bucket addresses into symbols and performs all the necessary statistic calculations.

When To Use Kernrate

1. Use Kernrate for preliminary identification of CPU usage patterns and CPU hogs down to API level (and even down to code sections within API's to a limited extent).
2. Use Kernrate for identifying specific CPU issues with profile sources other than the default (Time).
3. Use Kernrate to measure the effect of code changes and performance improvements on CPU usage.
4. There is little point in using Kernrate in cases where the bottleneck is not CPU related (low CPU usage), although the system-wide and process-specific summaries as well as lock information provided by Kernrate could help in initial identification of the culprits.

Limitations and Overhead Considerations

1. Kernrate is a sampling profiler. It may miss very short lived events. Increasing the sampling rate may help, but this will cost in terms of increased interrupt-rate (in some cases the machine may become unresponsive for a long period). In general, short lived events have little influence on the average CPU usage (but they may be of interest for other reasons).
2. The module address space is divided into buckets of no less than 4 bytes each (default is 16 bytes). A bucket address range may be inhabited by more than one API. Learning to use the –d, -r

or –v 2 options will help to identify who really occupies every bucket. This issue is also important in case of code optimizations that restructure the module.

3. The more processes being monitored, the more memory overhead. Besides the need to allocate memory for structures/arrays based on the number of processes, modules and sources, Kernrate also calls Imagehlp to load basic symbol information for the modules in the import list of every process being monitored at the beginning of the run. After profiling is done, Kernrate loads the appropriate deferred symbols for every zoomed module, but it releases each symbol file as soon as it is done with processing the symbols for any particular zoom module. The peak memory usage occurs during the processing phase, after profiling is done. Kernrate will also allocate more memory depending on the number of processors in the system and the bucket size. In some cases it may be useful to run the kernel profile separately from the user mode processes and divide the sampling of multiple user processes into several separate runs.
4. Kernrate may miss some of very short lived processes or some of these may go away before profiling/processing is done. Kernrate is not able to profile new processes created after its initialization.
5. Some of the optional data impose more overhead or slight delay, such as the summary of %CPU usage for all running processes on the system at the end of the profile (the –t command line option) or collecting lock contention information (the –x family of options and in particular the system lock information). Most of that overhead is incurred during the data processing phase.
6. Module and Function names are currently limited to 132 characters.
7. User command-line defined symbol path length should be less than 512 characters and the total symbol path length (environment variable + user command line defined) should not exceed 1024 characters. Exceeding these limits will cause truncation and a warning will be printed.

Early Exits and Error Messages

Kernrate will stop the run in four cases:

1. Command line error (This will result in printing an explanation of the error and in some cases a brief usage guide).
2. Memory allocation errors (a proper message will be printed).
3. Failure in calls to some API's critical for the success of the run (a proper message will be printed).
4. Trying to run on an operating system other than NT-based or older than Windows 2000.

Kernrate will not produce hits at all (not even for the default source Time) if:

1. There were no hits.
2. The particular Hal does not support profile counters or it has a bug.
3. The process being monitored exits prematurely.

Kernrate may produce no hits on some of the CPU sources specified by the user if:

1. There were no hits for a particular CPU source.
2. The specified CPU sources are incompatible to run simultaneously (on i386 and IA64 platforms, but not on AMD64 platform). This case requires switching to cyclic profiling mode.

Command line Parameters and Options

A brief usage guide can be invoked online by typing "**kernrate -?**"or "**kernrate -h**". The following is a more detailed description of the various command line options (letter case indifferent). Kernrate will accept both '-' and '/' as command line option indicators. The current version of Kernrate features many new or revised options (marked as **NEW** or **REVISED** at the end of each option description below).

Option Parameters Description

-?, -h Display a brief usage guide

-a Do a combined Kernel and User mode profile (not necessary if doing only a kernel mode profile or just a user mode process profile), **NEW**.

-av Do a combined Kernel and User mode profile and get task list and system threads info. **NEW.**

-b BucketSize Specify profiling bucket size (default = 16 bytes, minimum 4, must be a power of 2)

-c Rate in msec Change source every N milliseconds (default 1000ms), profiling one source at a time. Both '-c' and the rate are Optional. If '–c' is not specified, the default behavior is to profile all sources simultaneously. The '–c' option will cause elapsed time to be divided equally between the processes and the sources. For example, if 60 seconds are specified as the profile time with 2 processes being monitored and 3 active sources, each instance will get profiled for 60/(2*3) = 10 seconds. **REVISED.**

-d Generate output rounding buckets up and down. This will generate two output lists displaying the symbols and corresponding hits that are produced when rounding the bucket addresses up or down. See also the –r and –v 2 options. Does not apply to Managed Code modules.

-e Exclude system-wide and process specific general information (context switches, memory usage, etc.), to reduce processing overhead (default is to include that information). **NEW.**

-f Force processing the collected data at high priority (useful on busy systems if overhead is not an issue). **NEW.**

-g Rate Get interesting processor-counters statistics (Rate optional in events/hit, will apply to all sources contributing to the statistics). Not guarantied (Hal/driver support dependant). **NEW.**

-i SrcShortName Rate (in events/hit) Specify interrupt interval rate for the source specified by its ShortName. The '-i' option can be followed by only a source name (system default interrupt interval rate will then be assumed). '-i' option followed by a rate amount (no profile source short name) will change the interval rate for the default source (Time). Source 'Time' is enabled by default but profiling it can be disabled by setting the profile interval to zero (see notes). **REVISED.**

-j "SymbolPath" Prepend "SymbolPath" to the default imagehlp search path. Enclose the path in quotation marks.

-k MinHitCount Limit the output to modules that have at least MinHitCount hits (default 1). **NEW.**

-l List the default interval rates for supported sources.

-lx List the default interval rates for supported sources and then exit.

-m 0xN Generate per-CPU profiles on multi-processor machines. CPU affinity mask in Hex is optional, allowing to profile only the processors specified by the mask.

NEW.

-n ProcessName Monitor process by its name (default limited to first 8 by the same name), multiple usage allowed (see notes below). **NEW.**

-nv# N ProcessName Monitor up to N processes by the same name, 'v' will print thread info and a list of all running processes at the beginning of the run (optional). **NEW.**

-o ProcessName {CmdLine}
Create and monitor ProcessName (path OK, may be enclosed in quotes), Command Line parameters optional and must be enclosed in curly brackets. Redirection is supported within the curly brackets provided that the redirection characters ('<' or '>') are each escaped with a '^' character. Piping ('|') is not supported within this context (see notes). See also the '–wp' option. **NEW.**

-ov# N ProcessName {CmdLine}
Create N instances of ProcessName, v will print thread info and a list of all running processes (optional), {command line} optional, must be enclosed in curly brackets. See also the '-wp' option. **NEW.**

-pv ProcessId Monitor a process by its ProcessId, multiple usage allowed (see notes below). Multi-Processes are allowed. Each process ID needs to be preceded by -p except for the system process (kernel profile). 'v' will print thread info and a list of all running processes at the beginning of the run (optional). **REVISED.**

-r Raw data from zoomed modules. Print symbols and hits for every bucket, try to get bucket sharing information as well as hits in buckets with no symbol (possibly managed code sections). Try to get source-code line information for every bucket. See also the –d and –v 2 options.

-rd Raw data from zoomed modules with disassembly (currently providing only address info).

-s Seconds Collect data for N seconds (see also the '-c' and '-w' options).

-t MaxTasks Print a summary of kernel and user mode %CPU usage for all processes running during the profile. Change the maximum number of processes allowed in Kernrate's generated task list to MaxTasks (optional, default: 256, see overhead discussion). **NEW.**

-u Present symbols in undecorated form.

-w Wait for the user to press ENTER before starting to collect profile data. **NEW.**

-w Seconds Wait for N seconds before starting to collect profile data. **NEW.**

-wp Wait for the user to press enter to indicate that created processes (see -0 option) are settled (idle). **NEW.**

-wp Seconds Wait for N seconds to allow created processes settle (go idle), default is 2 seconds, (see the -o option). **NEW.**

-x Get both system and user-mode process locks contention information. Output filtered to default minimum of 1000 contention counts **NEW.**

-xk Get only system locks contention information (default for output-filter same as above). **NEW.**

-xu Get only user-mode process locks contention information (for the processes being

profiled, default for output-filter same as above). **NEW.**

-x# N — Get both system and user-mode process locks information, filter output to locks that have more than N contention counts (optional, default is 1000), the options –xk# N and –xu# N are also valid. **NEW.**

-z ModuleName — Name of module to zoom on (no extension needed by default, i.e. – z ntdll), multiple usage allowed, see notes below. The -z option requires to add the extension (.dll etc.) only if two or more binaries carry the same name and differ only by the extension. **REVISED.**

-v VerboseLevel — Verbose Printout. When specified with no level (see other verbose options below) the default printout is Imagehlp symbol path and symbol load information.

Verbose levels (can be or'ed together by the user, or will be if '–v' is specified multiple times):

1	Display ImageHlp symbol path and symbol load details.
2	Display profiling operations and per bucket information including symbol verification and bucket sharing information as well as source-code line information for every bucket (see also the –d and –r options). Note that symbol-verification information is printed to the console unless the standard-error output is redirected elsewhere. Additional sharing information totals will appear in the output summaries **REVISED.**
4	Display some Kernrate internals operations.
8	Display module related operations.

Notes:

1. A typical multi-process profiling command line (including kernel) should look like: kernrate -a -z ntoskrnl -z ntdll -z kernel32 -p 1234 -z w3svc -z iisrtl -p 4321 -z comdlg32 –z msvcrt ... (other options).
   The first group of -z denotes either kernel modules and/or modules common across processes. The other -z groups are process specific and should always follow the appropriate -p xxx. There is no need to specify the module extension with the –z option unless there is a possibility of ambiguity, such as a .exe and a .dll carrying the same name. Long module names that include one or more periods are allowed.
2. With the '-n' option, use the common modules -z option if you expect more than one process bearing the same name, i.e. kernrate – z ntdll –z iisrtl –z w3core –n w3wp …..(other options). It is assumed that the process has the extension .exe (you can bypass this assumption using the –p option).
3. Precedence rules: Only the –z option has precedence rules regarding its location on the command line.
4. No symbol files are needed when doing preliminary profiling (i.e. no zoom modules are specified using the –z option). However, correct symbols and symbol path are essential for zooming successfully.
5. There is no guaranty that all listed sources (-L or –Lx options) are supported or that they will encounter any hits. Kernrate will not fail in that case but will just print a message that no hits were encountered for the specific source.
6. The HAL will permit only certain ranges of sampling rates for any given source. Kernrate will print the actual sampling rate used for every source and an appropriate note if it failed to set the user requested rate. A margin of 25% deviation is currently allowed from the user specified rate before attempting to set the rate back to the default one. (Some machines were found to exhibit that much of a deviation between the requested rate and the actual set one).
7. The default ImageHlp/dbghelp symbol path is:

. ;%_NT_SYMBOL_PATH%;%_NT_ALTERNATE_SYMBOL_PATH%;
Symbol paths can also be specified using the –J option. In that case the specified path will be pre-pended to the previous one. ImageHlp may not be able to find the executable file if kernrate is being run from some user directory. Kernrate will append %WINDIR%\SYSTEM32\DRIVERS; %WINDIR%\SYSTEM32; %WINDIR% to the end of the path string. However be aware that the executable found may be the wrong one in case Imagehlp looked into some subdirectory (such as a service pack backup one) before looking in the right place. Do not assume that if the user specified symbol path is %windir%\symbols then ImageHlp will automatically look for sub-folders such as \dll, \exe, or \sys. Most of the reported Kernrate zooming problems are caused by bad symbol paths. Use the '–v' option to make sure you got the correct executables and associated symbol files.

8. Specifying a low minimum lock contention other than the default (1000) for filtering the output could cause the system to become unresponsive, and extend the processing time (for the system locks in particular). Not all lock addresses resolve to symbols, but use the –v option to make sure that Kernrate is looking for these symbols in the right place (see previous note).
9. Use the –k option to reduce the amount of output only after you get familiar with the typical number of hits that you get for the sampling rate being used. The –k option does not apply to verbose data.
10. The '-g' option will attempt to turn on multiple sources (currently 12 in total). One source at a time profiling mode will automatically be forced (see remarks regarding the '-c' option), so an adequate run time is needed. Output is not guarantied (Hal/driver support dependant). The run will not abort because of total lack of, or partial support and hit counts will be printed for supported sources.
11. There is no guaranty that any combination of sources can be turned on simultaneously (on x86 and IA64 platforms, AMD64 platforms do guarantee that). In addition there is a platform dependent limitation on how many sources can be turned on simultaneously (unlimited on AMD platforms, 4 on IA64 platforms, 1 on x86 platforms). If the user specified several sources in addition to TIME but did not use the '-c' command line option, kernrate will attempt to turn on these sources simultaneously but will print a warning message. Kernrate will automatically switch to cyclic profiling of one source, one process at a time if the number of sources requested exceeds the maximum number allowed to be turned on simultaneously for that platform, using the default switch rate (1000 ms).
12. On IA64 platforms, the user may actually turn off (disable) the default source (ProfileTime) by setting '-I ProfileTime 0' or even '-I 0' on the command line. This may be necessary when multiple sources are specified, because there is no guaranty any of them will work simultaneously with ProfileTime. Setting '-I ProfileTime 0' or '-I 0' on any platform will always prevent profiling of the default source (Time or ProfileTime) even if not actually disabling the source. This can be used to reduce the amount of output if CPU Time profile is not required.
13. Use of the '–c' option (profiling one source, one process at a time) consumes less system resources than the default usage (profiling all sources and processes simultaneously).
14. The '–w' option causes kernrate to initialize and wait before starting the actual profile. The '–wp' option causes kernrate to wait <u>before</u> initialization and only if there is actually any process created by kernrate.
15. Allowed redirection formats within the curly brackets of the '-o' command line option are:
    (a) Any of: ^<InputSream 1^>OutputStream 2^>ErrorStream (output and error streams allow appending as well, e.g. 1^>^>OutputStream).
    (b) Any of ^<InputSream ^>OutputStream (output appending allowed as well).
    Also:
    - Piping ('|') is not supported as an option within the curly brackets.
    - When redirection is used, the newly created process window may appear empty (blank).
    - Redirection of any output stream within the curly brackets is not allowed if the user specified more than one process to be created, by using the '**-o** Number ProcessName {parameters}' command line option.
16. Stopping the Kernrate run can be done automatically by specifying the '-s Seconds' command line

option, or it can be done manually by pressing 'ctrl-c'. In some situations one may be unable to use 'ctrl-c' because the control handler is taken by another process. In such cases one has to use the '-s Seconds' option.

17. To construct a processor affinity mask for the '-m' option select the indices of the processors to profile, do 1 << ProcessorIndex for each processor, convert to hex and add together (the index of the first processor is 0). There is no point in using the mask if the processes are not pre-bounded to processors. By default, all processors are profiled when '-m' is specified without a mask. Bogus processor entries in the mask will be ignored.

Kernrate Handling of Input and User Friendliness

Having multiple command line options increases input complexity and the possibility of user error. Kernrate parses the command line and tries to be user friendly as long as there is no ambiguity found. Here is a description of how Kernrate will handle some typical situations (reject means exit on error). Examples are given in parentheses.

Option Description	Extra Entries	Extra Letter	Associated Number	Associated String
Single Letter (-d)	none	reject any	reject	reject
Extra letters (-wp)	none	reject bogus	reject	reject
Letter combination (-xu 200)	number	reject bogus	reject if missing unless optional (-c)	reject
Letter combination (-xu 0x1a)	number	reject bogus	reject if invalid even if optional	reject
Letter combination that includes '#' (-c# 500)	number	reject bogus	reject if missing even if optional	reject
Letter combination (-s 20)	number	reject bogus	reject if missing unless optional (-c)	reject
Letter combination (-z ntdll)	string	reject bogus	N/A	reject if missing, unless optional
Letter combination (-I TIME 5000)	number & string	reject bogus	reject if missing unless optional	reject if missing unless optional
Letter combination (-I TIME 0x300)	Number & String	reject bogus	reject if invalid	reject if missing unless optional

Notes:

- Extra option-letters order is unimportant (both '–nv#' and '–n#v' will be accepted).
- Wrong number-string order will be accepted as valid as long as it is unambiguous.
- If string is mandatory but only a number is specified, the number will be treated as a string.
- If two numbers are specified, one of them will be treated as a string based on the nominal User-guide definition of order.
- A missing number will always cause rejection if '#' is specified with the option.

- Multiple entries of the command line options ‘-i’, ‘-n’, ‘-o’, ‘-p’, ‘-v’ and ‘-z’ are allowed.
- Multiple entries of the command line options ‘-d’, ‘-e’, ‘-f’, ‘-h’, ‘-r’ and ‘-u’ will be accepted (because they cannot cause any conflict) but a warning message will be printed.
- Multiple entries of the command line options ‘-a’, ‘-b’, ‘-c’, ‘-g’, ‘-j’, ‘-k’, ‘-l’, ‘-m’, ‘-s’, ‘-t’ and ‘-x’ are not allowed.
- Each of the ‘-w’ and ‘-wp’ options may be specified only once.

Managed Code Support

Kernrate can profile managed code JIT (Just In Time compilation) modules as well as pre-compiled (NGEN) modules. The support requires installing a helper library (IP2MD.DLL, not a part of Kernrate). No special command line switches are required to activate that support. Kernrate will assign unique names to each JIT module (because otherwise there may be more than one module bearing the same name). To zoom into any JIT module, the user needs to specify its unique name after the ‘–z’ option (multiple ‘-z’ usage is allowed). Pre-compiled (NGEN) modules should be treated as any regular module. At the end of the output for each process Kernrate will print a summary of the JIT modules status (Exists, New or Gone) based on snapshots taken before and after the profile. If a module is marked “GONE” do not fully trust the data related to it. Kernrate may miss some short-lived JIT modules altogether if they were created after the profile started and were destroyed before the profile ended.

On Windows 2000 one needs to install updated debugger tools before running Kernrate to profile managed code. In particular, dbgeng.dll needs to be replaced with the XP version (disable system file protection (SFP) or install it with a tool that handles SFP).

Output Notes

The printed output has been revised and enhanced. In addition to supporting some new options such as per-processor data on multi-processor machines and adaptation to the case of multi-user processes, the following system-wide and process-specific details have been added to the output:

System-Wide information:
- Time spent in User, Kernel and Idle and percentage of the elapsed profile time (this information has always existed in Kernrate)
- DPC time
- Interrupt time, count and average rate
- Total elapsed profile time in ms
- Context Switch count and average rate
- System Calls count and average rate
- Page Fault count and average rate
- I/O Read operations and average rate
- I/O Write operations and average rate
- I/O Other operations and average rate
- I/O Read bytes and average bytes per I/O
- I/O Write bytes and average bytes per I/O
- I/O Other bytes and average bytes per I/O

Process-Specific information:
- Kernel Time and %of elapsed profile time
- User Mode Time and % of elapsed profile time
- Page Fault count and average rate
- I/O Read operations and average rate

- I/O Write operations and average rate
- I/O Other operations and average rate
- I/O Read bytes and average bytes per I/O
- I/O Write bytes and average bytes per I/O
- I/O Other bytes and average bytes per I/O
- Process Threads (start – stop count, difference)
- Process Handles (start – stop counts, difference)
- Process Working Set bytes (start – stop counts, difference)
- Process Virtual Size bytes (start – stop counts, difference)
- Process Paged Pool bytes (start – stop counts, difference)
- Process Non-Paged Pool bytes (start – stop counts, difference)
- Process Page-File bytes (start – stop counts, difference)
- Process Private-Pages bytes (start – stop counts, difference)

Use the –e command line option to limit the output above. Note that if you specify both '-e' and '-t' on the command line you will get the task list summary. If you specify '-e' with any of '-av' '–nv' '-ov' or '-pv' you will get a task list for the beginning of the run but you will not get general process or thread information.

Use the –k command line option to limit the output only to modules or function calls that scored at least a given number of hits.

Note that the total number of hits reported for a module would not necessarily match the total number of accumulated hits for its zoomed function list (the latter is typically higher). This happens because the accumulated hits for the zoomed function list may contain hits counted twice or more for zoomed function calls that share the same bucket. The correct total number of hits is the one reported for the whole module. For the same reason, the hit percentage data for the zoomed list is only approximate.

The verbose level 2 information (-v 2 command line option) will print details of the actual inhabitants of zoomed module buckets (including address information, number of hits, number of shared or doubtful hits and source-code line information for each bucket) and a list of processes running on the system at the end of the profile. Source–code line information query will fail if the .pdb file does not include that information, does not match the binary file or if there are no valid code instructions in that address. Source-code line information includes the line number, the source file name and the address of the first instruction encountered in this line. The –v 2 option will also print a more detailed summary for the the zoomed module, indicating the number of shared hits and the percentage of certain (non shared) hits. The –d and –r command line option is an alternate way of presentation for the some of the output details that can be obtained with the –v 2 option.

The –t option will display a summary of all processes running on the system during the profile time and their corresponding %CPU usage (kernel and user-mode time).

The –x option(s) will display lock contention information for the system and user-mode processes being profiled, during the profile period. Lock information may be unavailable if the process being monitored is low on virtual memory (kernrate will print appropriate messages in that case).

The –g option will attempt to turn on several appropriate sources and compute the following:
- Cycles per Instruction
- Load Instructions Percentage
- Store Instructions Percentage
- Branch Instructions Percentage
- Floating Point Instructions Percentage

- Integer Instructions Percentage
- ICache Hit Percentage
- DCache Hit Percentage
- Branch Predict Hit Percentage

Output for supported sources (hit counts) will be printed but the statistics above will be calculated only if the counts exist and make some minimal sense.

Verbose Thread information (using the '-av', '-nv', '-ov', or '-pv' options):

- Process ID, Thread ID, thread start-address, module name if available.
- Thread State
- Wait Reason
- Wait Time [units of .1 uSec]
- Base Priority
- Priority
- Context Switches
- Context Switches
- User Time
- Kernel Time

Thread information obtained using the '-av', '-nv', '-ov', or '-pv' options is taken as a snapshot before and after profiling. The wait reason, thread state and priority may have some statistical meaning only if the occurrence of these particular values happens at a high frequency or for a prolonged time. Use other tools to analyze thread behavior in detail.

Brief Usage Examples

1. Get basic kernel profile (module level only, no zoom, no symbols required):

Command line:
Kernrate

Notes:

a. The basic profile is needed to identify which modules use the CPU most (candidates for zooming in).
b. Will profile 'Time' and use default settings for sample rate.
c. Will wait for user to type ctrl-c to finish the profile.

2. Get multi-processor kernel profile for 30 seconds, set sampling rate to 2000 events per hit and zoom on some modules (symbols required, symbol path assumed set with _NT_SYMBOL_PATH), get verbose Imagehlp symbol information to make sure the right symbols were picked up:

Command line:

Kernrate –m –s 30 –I 2000 –z ntoskrnl –z hal –z tcpip -v

3. Get basic profile for user processes Pid=1234 and Pid=5679 (module level only, no zoom, no symbols required), wait for user to type 'Return' before beginning to profile, then profile for 60 seconds:

Command line:

Kernrate –w –s 60 –p 1234 –p 5679

Notes:

a. Will profile 'Time' and use default settings for sample rate.

4. Get multi-processor profile for user processes 1234 and 5679 as well as for a process called "myproc", zoom in on some common modules as well as on some specific modules. Wait 10 seconds before starting the profile, then profile for 60 seconds. Set a special symbol path to "myproc" symbols and get verbose Imagehlp symbol information:

Command line:

Kernrate –m –v –w 10 –z ntdll –z kernel32 –p 1234 –z ole32 –z oleaut32 –p 5679 –z msvcrt –n myproc –z mydll_1 –z mydll_2 –j "c:\myproc symbols"

Notes:
a. Zoom modules ntdll, kernel32 will be considered "common" to all processes being profiled. Each of the common modules will be loaded for each process being profiled (unless it is not part of that process import list).
b. Modules ole32 and oleaut32 will be zoomed only for process 1234. Module msvcrt will be zoomed only for process 5679 and modules mydll_1, mydll_2 will be zoomed only for process "myproc.exe".
c. If there is more than one instance of the process "myproc.exe" running, you must put any zoomed module for that process in the common zoom list (and not as specific to that process).
d. The symbol path specified using the –j option will be placed before any other predefined symbol path.

5. Get multi-processor combined kernel and user mode profile, profile a process called "myproc.exe" as well as up to 4 worker processes by the name "myworker.exe". Zoom in on some kernel and common modules as well as some process specific modules. Get verbose profiling information regarding possibly shared buckets as well as verbose Imagehlp symbol information. Wait 20 seconds before starting the profile then profile for 60 seconds.

Command line:

Kernrate –a –v 3 –m –w 20 –s 60 –z ntoskrnl –z hal –z tcpip –z ntdll –z kernel32 –z workerdll_1 –nv myproc –z mydll_1 -n# 4 myworker

Notes:
a. Since there is more than one worker process by the name "myworker", zoom module "workerdll_1" must be placed on the common zoom list.
b. The –nv option will cause a list of running processes to be printed at the beginning of the run.
c. The combination –nv# 4 is also allowed.
d. By specifying –n# 4 the user overrides the default maximum of 8 processes bearing the same name.
e. The –v 3 option is a combination of –v 1 (Imagehlp verbose symbol information) or'ed with –v 2 (verbose profile information). Note that –v and –v 1 have the same meaning.
f. The –v 2 option will also print the full list of modules loaded for each process and a final list of processes running at the end of the profile.

6. Get the list of sources supported on a specific machine, then exit.

Command line:

Kernrate –lx

Notes:
a. The short names from this list are used for specifying the profile sources in the following case.

7. Get a multi-processor kernel profile with several sources (on a 64 bit Itanium machine), specify the sampling rate for some, sample one source at a time over a period of 120 seconds. Start sampling after user types ctrl-c. Zoom in on some modules and get raw data bucket information. Force 4 byte bucket size and high priority processing of the data. Get Imagehlp verbose symbol information. Get summary of %CPU usage for all processes running on the system during the profile, get lock contention information for the system.

Command line:

Kernrate –v –f -t -xk –b 4 –w –s 120 –m –c –r –z ntoskrnl –z hal –z tcpip –z ndis –I ProfileTime 2000 –I MercedInstRetired 100 –I MercedBranchInstructions 1000 –I MercedBranchStallCycles

Notes:
a. See also the –d and –v 2 options as alternative presentation to the –r option.
b. The –f option will alow Kernrate to continue and run at high priority during data processing. This can be useful on busy systems when momentary overhead is not an issue.
c. "ProfileTime" is the default profile (on some other systems it is called "Time"). The sampling rate for this source can also be specified without specifying the source name (i.e. –I 2000).
d. The last source (MercedBranchStallCycles) has no rate specified with it, meaning that the user chose the default rate.
e. A rate of 0 (zero) will disable the source (except for the default Time source that is always enabled).
f. The selection of the –c option in this case will cause the 120 seconds of profile time to be equally divided between the four sources, so each will be sampled for an overall of 30 seconds (the sampling will rotate continuously between the sources, changing sources every 1000 ms unless the user selected a different rotation period with the –c option).

Detailed Usage Examples and Output Description

1. Basic run, Kernel profile only (no symbols required), 2P machine, profile stopped by typing ctrl-c:
(This type of run is used to get a general idea where time is spent and which modules are most active).

Command line:
Kernrate

Output:

```
Kernel Source Profile (PID = 0): Source= Time, Using Kernrate Default Rate of 25000 events/hit
***> Press ctrl-c to finish collecting profile data
===> Finished Collecting Data, Starting to Process Results

------------Overall Summary:--------------

P0     K 0:00:07.890 ( 26%)  U 0:00:01.328 (  4%)  I 0:00:20.609 ( 69%)  DPC 0:00:00.046 (  0%)
Interrupt 0:00:00.062 (  0%)
       Interrupts= 20824, Interrupt Rate= 718/sec.

P1     K 0:00:05.500 ( 18%)  U 0:00:00.906 (  3%)  I 0:00:23.421 ( 78%)  DPC 0:00:00.140 (  0%)
Interrupt 0:00:00.109 (  0%)
       Interrupts= 20272, Interrupt Rate= 699/sec.

TOTAL  K 0:00:13.390 ( 22%)  U 0:00:02.234 (  3%)  I 0:00:44.031 ( 73%)  DPC 0:00:00.187 (  0%)
Interrupt 0:00:00.171 (  0%)
       Total Interrupts= 41096, Total Interrupt Rate= 1417/sec.
```

```
Total Profile Time = 29828 msec

                                Total      Avg. Rate
    Context Switches     ,      396270,         13664/sec.
    System Calls         ,      584312,         20149/sec.
    Page Faults          ,        1446,         50/sec.
    I/O Read Operations  ,        1200,         41/sec.
    I/O Write Operations ,         671,         23/sec.
    I/O Other Operations ,        3343,         115/sec.
    I/O Read Bytes       ,       53078,         44/ I/O
    I/O Write Bytes      ,       43118,         64/ I/O
    I/O Other Bytes      ,      425902,         127/ I/O

-----------------------------
Results for Kernel Mode:
-----------------------------

OutputResults: KernelModuleCount = 117

Time   22675 hits, 25000 events per hit --------
 Module                                  Hits       msec  %Total  Events/Sec
p3                                      14734      29840    64 %    12344168
win32k                                   5472      29840    24 %     4584450
ntoskrnl                                 1484      29840     6 %     1243297
hal                                       719      29840     3 %      602379
nv4                                       104      29840     0 %       87131
```

… (output truncated)

```
================================ END OF RUN ================================
```

Note: P3 is the (ACPI) idle loop.

2. Basic run + zooming-in on kernel module (win32k.sys, ntoskrnl), 30 seconds profile, verifying symbol path (zoom modules selected based on the previous run):

Command line:

Kernrate –v –s 30 –z win32k –z ntoskrnl

Output:

```
KERNRATE: current IMAGEHLP SymOptions: UNDNAME DEBUG
Kernel Source Profile (PID = 0): Source= Time, Using Kernrate Default Rate of 25000 events/hit
KERNRATE: IMAGEHLP symbol search path: .;C:\WIN2KSRV\symbols
CallBack: Loading symbols for ntoskrnl.exe...
DBGHELP: FindExecutableImageEx-> Looking for E:\nt\base\tools\kernrate\obj\i386\ntoskrnl.exe...
no file
DBGHELP: FindExecutableImageEx-> Searching . for ntoskrnl.exe... no file
DBGHELP: FindExecutableImageEx-> Searching C:\WIN2KSRV\symbols for ntoskrnl.exe... no file
DBGHELP: FindDebugInfoFileEx-> Looking for .\symbols\exe\ntoskrnl.dbg... path not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\symbols\exe\ntoskrnl.dbg... path
not found
DBGHELP: FindDebugInfoFileEx-> Looking for .\exe\ntoskrnl.dbg... path not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\exe\ntoskrnl.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for .\ntoskrnl.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\ntoskrnl.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for ntoskrnl.dbg... file not found
DBGHELP: FindExecutableImageEx-> Looking for E:\nt\base\tools\kernrate\obj\i386\ntoskrnl.exe...
no file
DBGHELP: FindExecutableImageEx-> Searching . for ntoskrnl.exe... no file
DBGHELP: FindExecutableImageEx-> Searching C:\WIN2KSRV\symbols for ntoskrnl.exe... no file
DBGHELP: diaLocatePDB-> Looking for .\symbols\exe\ntoskrnl.pdb... file not found
```

```
DBGHELP: diaLocatePDB-> Looking for .\exe\ntoskrnl.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for .\ntoskrnl.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for C:\WIN2KSRV\symbols\symbols\exe\ntoskrnl.pdb... file not
found
DBGHELP: diaLocatePDB-> Looking for C:\WIN2KSRV\symbols\exe\ntoskrnl.pdb... unknown pdb sig OK
Loaded symbols for \WIN2KSRV\system32\ntoskrnl.exe
CallBack: Loading symbols for win32k.sys...
DBGHELP: FindExecutableImageEx-> Looking for E:\nt\base\tools\kernrate\obj\i386\win32k.sys... no
file
DBGHELP: FindExecutableImageEx-> Searching . for win32k.sys... no file
DBGHELP: FindExecutableImageEx-> Searching C:\WIN2KSRV\symbols for win32k.sys... no file
DBGHELP: FindDebugInfoFileEx-> Looking for .\symbols\sys\win32k.dbg... path not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\symbols\sys\win32k.dbg... path not
found
DBGHELP: FindDebugInfoFileEx-> Looking for .\sys\win32k.dbg... path not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\sys\win32k.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for .\win32k.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for C:\WIN2KSRV\symbols\win32k.dbg... file not found
DBGHELP: FindDebugInfoFileEx-> Looking for win32k.dbg... file not found
DBGHELP: FindExecutableImageEx-> Looking for E:\nt\base\tools\kernrate\obj\i386\win32k.sys... no
file
DBGHELP: FindExecutableImageEx-> Searching . for win32k.sys... no file
DBGHELP: FindExecutableImageEx-> Searching C:\WIN2KSRV\symbols for win32k.sys... no file
DBGHELP: diaLocatePDB-> Looking for .\symbols\sys\win32k.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for .\sys\win32k.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for .\win32k.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for C:\WIN2KSRV\symbols\symbols\sys\win32k.pdb... file not found
DBGHELP: diaLocatePDB-> Looking for C:\WIN2KSRV\symbols\sys\win32k.pdb... unknown pdb sig OK
Loaded symbols for \??\C:\WIN2KSRV\system32\win32k.sys
Will collect profile data for 30 seconds
===> Finished Collecting Data, Starting to Process Results

------------Overall Summary:--------------

P0     K 0:00:09.609 ( 32%)  U 0:00:02.000 (  6%)  I 0:00:18.390 ( 61%)  DPC 0:00:00.156 (  0%)
Interrupt 0:00:00.046 (  0%)
       Interrupts= 17198, Interrupt Rate= 573/sec.

P1     K 0:00:08.359 ( 27%)  U 0:00:01.453 (  4%)  I 0:00:20.187 ( 67%)  DPC 0:00:00.093 (  0%)
Interrupt 0:00:00.125 (  0%)
       Interrupts= 17333, Interrupt Rate= 577/sec.

TOTAL  K 0:00:17.968 ( 29%)  U 0:00:03.453 (  5%)  I 0:00:38.578 ( 64%)  DPC 0:00:00.250 (  0%)
Interrupt 0:00:00.171 (  0%)
       Total Interrupts= 34531, Total Interrupt Rate= 1151/sec.


Total Profile Time = 30000 msec

                                Total      Avg. Rate
    Context Switches     ,       595242,         19841/sec.
    System Calls         ,       849150,         28305/sec.
    Page Faults          ,         1838,         61/sec.
    I/O Read Operations  ,         1163,         39/sec.
    I/O Write Operations ,          660,         22/sec.
    I/O Other Operations ,         4158,         139/sec.
    I/O Read Bytes       ,        70488,         61/ I/O
    I/O Write Bytes      ,        42240,         64/ I/O
    I/O Other Bytes      ,      1580946,         380/ I/O

-----------------------------
Results for Kernel Mode:
-----------------------------

OutputResults: KernelModuleCount = 117

Time   22319 hits, 25000 events per hit --------
 Module                                  Hits       msec  %Total  Events/Sec
p3                                      13104      29990    58 %    10923641
win32k                                   5756      29990    25 %     4798266
ntoskrnl                                 1869      29990     8 %     1558019
hal                                       915      29990     4 %      762754
nv4                                       483      29990     2 %      402634
```

…… (list truncated)

```
----- Zoomed module win32k (Bucket size = 16 bytes, Rounding Down) --------

Time   5731 hits, 25000 events per hit --------
 Module                                Hits       msec  %Total  Events/Sec
DoTimer                                3796      29990    66 %     3164388
TimersProc                              472      29990     8 %      393464
FindTimer                               388      29990     6 %      323441
xxxRealInternalGetMessage                93      29990     1 %       77525
xxxCallHook2                             48      29990     0 %       40013
fnHkINLPMSG                              30      29990     0 %       25008
xxxReadPostMessage                       29      29990     0 %       24174
RFONTOBJ::bGetGlyphMetricsPlus           26      29990     0 %       21673
NtUserPeekMessage                        25      29990     0 %       20840
```

……(list truncated)

```
----- Zoomed module ntoskrnl (Bucket size = 16 bytes, Rounding Down) --------

Time   1648 hits, 25000 events per hit --------
 Module                                Hits       msec  %Total  Events/Sec
RtlpStatusTable                         184      29990    11 %      153384
MiGatherMappedPages                     159      29990     9 %      132544
CcFastCopyWrite                         152      29990     9 %      126708
KeWaitForMultipleObjects                120      29990     7 %      100033
NtQueryInformationFile                   86      29990     5 %       71690
MiFlushDirtyBitsToPfn                    86      29990     5 %       71690
MiGetSystemCacheSubsection               71      29990     4 %       59186
IopParseDevice                           51      29990     3 %       42514
CcCopyWrite                              44      29990     2 %       36678
```

……(list truncated)

```
================================= END OF RUN ==================================
```

Note: In the example above _NT_SYMBOL_PATH was set to `C:\WIN2KSRV\symbols`

3. Combined Kernel+User mode run + zooming-in on selected modules, waiting 10 seconds before starting the profile, 30 seconds profile, per-processor output, process-list summary, locks information (both kernel and user modes) with non-default minimum contention count, changing bucket size to 4 bytes, asking for verbose profile detail (information about bucket sharing):

Command line:

Kernrate –a –t –x# 100 –m –w 10 –s 30 –b 4 –z hal –z win32k –z ntdll –nv iexplore –v 2

Output:

```
Profiling both Kernel and User Modes
Minimum lock contention for processing set to= 100
Profile Bucket Size Set to 4 bytes
         Pid                 Process
       -------               -----------
           0             System Idle Process
           4                       System
         324                     smss.exe
         380                    csrss.exe
         404                  winlogon.exe
         448                  services.exe
         484                     lsass.exe
```

```
         636                      svchost.exe
         684                      svchost.exe
         796                      svchost.exe
         828                      svchost.exe
         852                      svchost.exe
        1080                      spoolsv.exe
        1112                        msdtc.exe
        1348                    inetinfo.exe
        1368                      InoRpc.exe
        1388                       InoRT.exe
        1408                     InoTask.exe
        1476                      llssrv.exe
        1528                     svchost.exe
        1564                    owstimer.exe
        1604                     svchost.exe
        1636                     lserver.exe
        1812                      dfssvc.exe
        1920                     svchost.exe
        2588                         alg.exe
        3144                    explorer.exe
        3388                     Realmon.exe
        3464                    NEWSALRT.EXE
        3472                      ctfmon.exe
         432                     QSHLFED.EXE
         760                         cmd.exe
        1100                    IEXPLORE.EXE

===> Found process: IEXPLORE.EXE, Pid: 1100

        2448                    MSOFFICE.EXE
        3064                    davcdata.exe
        3552                     dllhost.exe
        3900                       EXCEL.EXE
        4084                     WINWORD.EXE
        3352                        list.exe
        3052                        calc.exe
        2540                    rundll32.exe
        3620                     notepad.exe
        1208                     notepad.exe
        3988                        srch.exe
        3276                        srch.exe
         160                         cmd.exe
         792                         cmd.exe
        2836                         cmd.exe
        3812                         cmd.exe
        2376                     notepad.exe
        3884                     notepad.exe
        2492                        srch.exe
         488                        srch.exe
         272                    IEXPLORE.EXE

===> Found process: IEXPLORE.EXE, Pid: 272

        2008                          ps.exe
        3604                         cmd.exe
        2716                        srch.exe
        3932                    IEXPLORE.EXE

===> Found process: IEXPLORE.EXE, Pid: 3932

        4004                         cmd.exe
        1416                         mmc.exe
        3260                    dmremote.exe
         236                     dmadmin.exe
        3496                         cmd.exe
        2120                    insight3.exe
        1212                      raid45.exe
        1228                    IEXPLORE.EXE

===> Found process: IEXPLORE.EXE, Pid: 1228

        3256                        list.exe
         568                        CW32.EXE
```

```
        1568                        cmd.exe
         672                    OUTLOOK.EXE
        1908                    Visio32.EXE
        3140                   wmiprvse.exe
        3640                   kernrate.exe

Requested Rate= 25000 Events/Hit, Actual Rate= 25000 Events/Hit
PID = 1228: Source= Time, Using Kernrate Default Rate of 25000 events/hit
Requested Rate= 25000 Events/Hit, Actual Rate= 25000 Events/Hit
PID = 3932: Source= Time, Using Kernrate Default Rate of 25000 events/hit
Requested Rate= 25000 Events/Hit, Actual Rate= 25000 Events/Hit
PID = 272: Source= Time, Using Kernrate Default Rate of 25000 events/hit
Requested Rate= 25000 Events/Hit, Actual Rate= 25000 Events/Hit
PID = 1100: Source= Time, Using Kernrate Default Rate of 25000 events/hit
Requested Rate= 25000 Events/Hit, Actual Rate= 25000 Events/Hit
Kernel Source Profile (PID = 0): Source= Time, Using Kernrate Default Rate of 25000 events/hit

------------Overall Summary:--------------

P0     K 0:00:05.140 (12.8%)  U 0:00:05.640 (14.1%)  I 0:00:29.234 (73.1%)  DPC 0:00:00.312
( 0.8%)  Interrupt 0:00:00.000 ( 0.0%)
       Interrupts= 61620, Interrupt Rate= 1540/sec.

P1     K 0:00:04.000 (10.0%)  U 0:00:03.359 ( 8.4%)  I 0:00:32.656 (81.6%)  DPC 0:00:00.250
( 0.6%)  Interrupt 0:00:00.078 ( 0.2%)
       Interrupts= 23411, Interrupt Rate= 585/sec.

TOTAL  K 0:00:09.140 (11.4%)  U 0:00:09.000 (11.2%)  I 0:01:01.890 (77.3%)  DPC 0:00:00.562
( 0.7%)  Interrupt 0:00:00.078 ( 0.1%)
       Total Interrupts= 85031, Total Interrupt Rate= 2125/sec.


Total Profile Time = 40015 msec

                                Total      Avg. Rate
    Context Switches     ,       182278,          4555/sec.
    System Calls         ,       343212,          8577/sec.
    Page Faults          ,         6929,          173/sec.
    I/O Read Operations  ,         3500,          87/sec.
    I/O Write Operations ,          356,          9/sec.
    I/O Other Operations ,         4587,          115/sec.
    I/O Read Bytes       ,      1336870,          382/ I/O
    I/O Write Bytes      ,       448777,          1261/ I/O
    I/O Other Bytes      ,       353583,          77/ I/O



Locks Contention Info:

Address, Contention-Diff., Rate(per sec.), Thread, Type, Recursion, Waiting-Shared, Waiting-
Exclusive, Symbol-Information
81D7A618,      55892,       1397,      0x0, RESOURCE,  N/A,             0,          0
81D96C40,        382,         10,      0x0, RESOURCE,  N/A,             0,          0

--- Process List and Summary At The End of Data Collection ---

Found 73 processes at the start point, 73 processes at the stop point

    ProcessID, Process Name,                    Kernel Time,   User-Mode Time,   Idle Time

          0,             System Idle Process,       0.00%,       0.00%,      77.3%
          4,                          System,       0.82%,       0.00%
        324,                       smss.exe,       0.02%,       0.00%
        380,                      csrss.exe,       4.86%,       0.04%
        404,                    winlogon.exe,       0.00%,       0.00%
        448,                    services.exe,       0.00%,       0.00%
        484,                       lsass.exe,       0.00%,       0.00%
        636,                     svchost.exe,       0.00%,       0.00%
        684,                     svchost.exe,       0.00%,       0.00%
        796,                     svchost.exe,       0.12%,       0.02%
        828,                     svchost.exe,       0.00%,       0.00%
        852,                     svchost.exe,       0.00%,       0.00%
       1080,                     spoolsv.exe,       0.00%,       0.02%
```

```
        1112,                       msdtc.exe,       0.00%,       0.00%
        1348,                     inetinfo.exe,      0.00%,       0.00%
        1368,                       InoRpc.exe,      0.00%,       0.00%
        1388,                        InoRT.exe,      0.16%,       0.29%
        1408,                      InoTask.exe,      0.02%,       0.00%
        1476,                       llssrv.exe,      0.00%,       0.00%
        1528,                      svchost.exe,      0.00%,       0.00%
        1564,                     owstimer.exe,      0.00%,       0.00%
        1604,                      svchost.exe,      0.00%,       0.00%
        1636,                      lserver.exe,      0.00%,       0.00%
        1812,                       dfssvc.exe,      0.00%,       0.00%
        1920,                      svchost.exe,      0.00%,       0.00%
        2588,                          alg.exe,      0.00%,       0.00%
        3144,                     explorer.exe,      0.06%,       0.04%
        3388,                      Realmon.exe,      0.00%,       0.00%
        3472,                       ctfmon.exe,      0.00%,       0.00%
         432,                      QSHLFED.EXE,      0.00%,       0.00%
        1100,                     IEXPLORE.EXE,      0.00%,       0.00%
        2448,                     MSOFFICE.EXE,      0.02%,       0.00%
        3064,                     davcdata.exe,      0.00%,       0.00%
        3552,                      dllhost.exe,      0.00%,       0.00%
        3900,                        EXCEL.EXE,      0.31%,       0.06%
        4084,                      WINWORD.EXE,      1.64%,       0.35%
        2540,                     rundll32.exe,      0.00%,       0.00%
        3884,                      notepad.exe,      0.04%,       0.00%
         272,                     IEXPLORE.EXE,      0.00%,       0.02%
        2008,                           ps.exe,      0.00%,       0.00%
        3604,                          cmd.exe,      0.00%,       0.00%
        3932,                     IEXPLORE.EXE,      0.00%,       0.00%
        3260,                     dmremote.exe,      0.00%,       0.00%
         236,                      dmadmin.exe,      0.00%,       0.00%
        1228,                     IEXPLORE.EXE,      6.85%,      10.84%
         672,                      OUTLOOK.EXE,      0.12%,       0.06%
        1908,                      Visio32.EXE,      0.02%,       0.00%
        3640,                     kernrate.exe,      0.55%,       0.23%

-----------------------------

Results for Kernel Mode:
-----------------------------

OutputResults: KernelModuleCount = 119

Time   28018 hits, 25000 events per hit --------
 Module                                Hits       msec  %Total  Events/Sec
p3                                    23269      40021    83 %    14535493
     0   10713  40021    38 %     6692111
     1   12556  40021    44 %     7843382
ntoskrnl                               1899      40021     6 %     1186252
     0    1142  40021     4 %      713375
     1     757  40021     2 %      472876
win32k                                 1606      40021     5 %     1003223
     0     830  40021     2 %      518477
     1     776  40021     2 %      484745
hal                                     745      40021     2 %      465380
     0     464  40021     1 %      289847
     1     281  40021     1 %      175532
.........(List truncated)


 -------------- VERBOSE PROFILE DATA FOR ZOOMED MODULE win32k ----------------
Module Name, Parent Base Address, Module Base address, Current Bucket Index, Current Bucket
Address, Total Current Bucket Hits, Per CPU Hits, Remarks

fsg_QueryTwilightElement, 0xbf800000, 0xbf85c019, 94224, 0xbf85c040, 1, 1, 0 (line 987 in d:\nt\
windows\core\ntgdi\fondrv\tt\scaler\fsglue.c, 0xbf85c03b)
fsg_QueryTwilightElement, 0xbf800000, 0xbf85c019, 94228, 0xbf85c050, 2, 2, 0 (line 989 in d:\nt\
windows\core\ntgdi\fondrv\tt\scaler\fsglue.c, 0xbf85c04b)
Time, fsg_QueryTwilightElement - Module Total Count = 3, Total Doubtful or Shared Counts = 0

vSetGrayState__FONTCONTEXT, 0xbf800000, 0xbf85ab8c, 92956, 0xbf85ac70, 1, 1, 0 (line 1641 in d:\
nt\windows\core\ntgdi\fondrv\tt\scaler\fsglue.c, 0xbf85ac51) , Actual Hits Should be Attributed
```

```
to or Shared with ===> fsg_GridFit+0x38 (line 1656 in d:\nt\windows\core\ntgdi\fondrv\tt\scaler\
fsglue.c, 0xbf85ac99)
Time, vSetGrayState__FONTCONTEXT - Module Total Count = 1, Total Doubtful or Shared Counts = 1

itrp_MPPEM, 0xbf800000, 0xbf85df93, 96232, 0xbf85dfa0, 1, 1, 0 (line 5159 in d:\nt\windows\core\
ntgdi\fondrv\tt\scaler\interp.c, 0xbf85df9d)
Time, itrp_MPPEM - Module Total Count = 1, Total Doubtful or Shared Counts = 0

GreSetDCOrg, 0xbf800000, 0xbf869c3c, 108324, 0xbf869c90, 1, 1, 0 (line 636 in d:\nt\windows\core\
ntgdi\gre\dcgdi.cxx, 0xbf869c8b)
Time, GreSetDCOrg - Module Total Count = 1, Total Doubtful or Shared Counts = 0

_DrawIconEx, 0xbf800000, 0xbf86814e, 106639, 0xbf86823c, 1, 1, 0 (SymGetLineFromAddr64 failed,
Error Code= 1e7 - ERROR_INVALID_ADDRESS) , Actual Hits Should be Attributed to or Shared with
===> BltIcon+0x2c (line 222 in d:\nt\windows\core\ntgdi\gre\bltlnkfn.cxx, 0xbf868268)
_DrawIconEx, 0xbf800000, 0xbf86814e, 106649, 0xbf868264, 1, 1, 0 (line 218 in d:\nt\windows\core\
ntgdi\gre\bltlnkfn.cxx, 0xbf868254) , Actual Hits Should be Attributed to or Shared with ===>
vRop2Function8+0x26 (SymGetLineFromAddr64 failed, Error Code= 1e7 - ERROR_INVALID_ADDRESS)
Time, _DrawIconEx - Module Total Count = 2, Total Doubtful or Shared Counts = 2

vSolidFillRect1, 0xbf800000, 0xbf867d43, 106408, 0xbf867ea0, 1, 1, 0 (line 660 in d:\nt\windows\
core\ntuser\kernel\mnsys.c, 0xbf867e9e) , Actual Hits Should be Attributed to or Shared with ===>
_SetMenuDefaultItem+0x8b (SymGetLineFromAddr64 failed, Error Code= 1e7 - ERROR_INVALID_ADDRESS)
Time, vSolidFillRect1 - Module Total Count = 1, Total Doubtful or Shared Counts = 1

PDEVOBJ__DestroyFont, 0xbf800000, 0xbf83b6b9, 60859, 0xbf83b6ec, 1, 1, 0 (line 222 in d:\nt\
windows\core\ntgdi\gre\usersrv.cxx, 0xbf83b6eb)
Time, PDEVOBJ__DestroyFont - Module Total Count = 1, Total Doubtful or Shared Counts = 0

CreateHalftoneBrushPat, 0xbf800000, 0xbf8ddfcd, 227960, 0xbf8de9e0, 1, 1, 0 (line 196 in d:\nt\
windows\core\ntuser\kernel\newmouse.c, 0xbf8de9dd) , Actual Hits Should be Attributed to or
Shared with ===> DoNewMouseAccel+0x1b9 (line 1960 in d:\nt\windows\core\ntuser\kernel\ntinput.c,
0xbf8deb95)
Time, CreateHalftoneBrushPat - Module Total Count = 1, Total Doubtful or Shared Counts = 1

AllocateW32Process, 0xbf800000, 0xbf873fac, 118807, 0xbf87405c, 1, 0, 1 (line 158 in D:\NT\
windows\core\ntgdi\gre\i386\locka.asm, 0xbf87405c) , Actual Hits Should be Attributed to or
Shared with ===> HmgLock+0x1a (line 173 in D:\NT\windows\core\ntgdi\gre\i386\locka.asm,
0xbf874075)
Time, AllocateW32Process - Module Total Count = 1, Total Doubtful or Shared Counts = 1

.........(Output truncated)



----- Zoomed module win32k (Bucket size = 4 bytes, Rounding Down) --------

Time   2550 hits, 25000 events per hit --------
 Module                                Hits       msec  %Total  Events/Sec
zzzUpdateCursorImage                    205      40021     8 %      128057
     0      70  40021     2 %       43727
     1     135  40021     5 %       84330
RawInputThread                          185      40021     7 %      115564
     0      52  40021     2 %       32482
     1     133  40021     5 %       83081
TimersProc                              170      40021     6 %      106194
     0      45  40021     1 %       28110
     1     125  40021     4 %       78084
DoTimer                                 112      40021     4 %       69963
     0      72  40021     2 %       44976
     1      40  40021     1 %       24986
vSpUpdateSpriteVisRgn                    51      40021     2 %       31858
     0      22  40021     0 %       13742
     1      29  40021     1 %       18115
FindTimer                                46      40021     1 %       28734
     0      28  40021     1 %       17490
     1      18  40021     0 %       11244
InternalSetTimer                         46      40021     1 %       28734
     0      28  40021     1 %       17490
     1      18  40021     0 %       11244
ENUMUNDERLAYS__bEnum                     44      40021     1 %       27485
     0      31  40021     1 %       19364
     1      13  40021     0 %        8120
```

```
pSpFindInZ                                39      40021     1 %       24362
     0      28  40021     1 %       17490
     1      11  40021     0 %        6871
.........(Output truncated)

---------------------------------------------------------------

Results for User Mode Process IEXPLORE.EXE (PID = 1228)

    User Time                  = 10.84% of the Elapsed Time
    Kernel Time                = 6.85% of the Elapsed Time

                                 Total      Avg. Rate
    Page Faults          ,         4601,         115/sec.
    I/O Read Operations  ,         1732,          43/sec.
    I/O Write Operations ,          224,           6/sec.
    I/O Other Operations ,         2712,          68/sec.
    I/O Read Bytes       ,         1732,          0/ I/O
    I/O Write Bytes      ,          224,          1/ I/O
    I/O Other Bytes      ,         2712,          1/ I/O

                            Start-Count     Stop-Count       Diff.
    Threads              ,           25,           31,           6
    Handles              ,          662,          746,          84
    Working Set Bytes    ,     13848576,     24981504,     11132928
    Virtual Size Bytes   ,    213188608,    219713536,      6524928
    Paged Pool Bytes     ,        97124,       102604,         5480
    Non Paged Pool Bytes ,        25288,        32944,         7656
    Pagefile Bytes       ,     23212032,     24522752,      1310720
    Private Pages Bytes  ,     23212032,     24522752,      1310720

Locks Contention Info:

Address, Contention-Diff., Rate(per sec.), Thread, Type, Recursion, Waiting-Shared, Waiting-
Exclusive, Symbol-Information
74A81620,        626,         16,      0x0, CRITICAL_SECTION,           0,     N/A,
N/A ,base= 74810000 - mshtml!g_csHeap
---------------------------------------------------------------------

OutputResults: ProcessModuleCount (Including Managed-Code JITs) = 105

Time   3598 hits, 25000 events per hit --------
 Module                                   Hits       msec  %Total  Events/Sec
swflash                                   1526      40021    42 %      953249
     0    1115  40021    30 %      696509
     1     411  40021    11 %      256740
mshtml                                    1270      40005    35 %      793650
     0     797  40005    22 %      498062
     1     473  40005    13 %      295588
ntdll                                      236      40005     6 %      147481
     0     151  40005     4 %       94363
     1      85  40005     2 %       53118
MSLS31                                      90      40021     2 %       56220
     0      59  40021     1 %       36855
     1      31  40021     0 %       19364
USP10                                       65      40005     1 %       40619
     0      41  40005     1 %       25621
     1      24  40005     0 %       14998
kernel32                                    60      40005     1 %       37495
     0      35  40005     0 %       21872
     1      25  40005     0 %       15623
SHLWAPI                                     50      40005     1 %       31246
     0      37  40005     1 %       23122
     1      13  40005     0 %        8123
.........(Output truncated)


 -------------- VERBOSE PROFILE DATA FOR ZOOMED MODULE ntdll ----------------
Module Name, Parent Base Address, Module Base address, Current Bucket Index, Current Bucket
Address, Total Current Bucket Hits, Per CPU Hits, Remarks

LdrpGetProcedureAddress, 0x77f50000, 0x77f5f2e2, 15625, 0x77f5f424, 1, 0, 1 (line 1404 in d:\nt\
base\ntdll\ldrapi.c, 0x77f5f423)
```

```
Time, LdrpGetProcedureAddress - Module Total Count = 1, Total Doubtful or Shared Counts = 0

RtlCheckHeldCriticalSections, 0x77f50000, 0x77f69c4f, 26388, 0x77f69c50, 1, 1, 0 (line 1905 in
d:\nt\base\ntdll\resource.c, 0x77f69c4f)
Time, RtlCheckHeldCriticalSections - Module Total Count = 1, Total Doubtful or Shared Counts = 0

RtlpInterlockedPushEntrySList, 0x77f50000, 0x77fb59a4, 104041, 0x77fb59a4, 1, 1, 0 (line 296 in
D:\NT\base\ntos\rtl\i386\slist.asm, 0x77fb59a4)
RtlpInterlockedPushEntrySList, 0x77f50000, 0x77fb59a4, 104047, 0x77fb59bc, 1, 0, 1 (line 319 in
D:\NT\base\ntos\rtl\i386\slist.asm, 0x77fb59b8)
RtlpInterlockedPushEntrySList, 0x77f50000, 0x77fb59a4, 104048, 0x77fb59c0, 3, 1, 2 (line 337 in
D:\NT\base\ntos\rtl\i386\slist.asm, 0x77fb59c0)
Time, RtlpInterlockedPushEntrySList - Module Total Count = 5, Total Doubtful or Shared Counts = 0

__SEH_epilog, 0x77f50000, 0x77fa4a3f, 86674, 0x77fa4a48, 1, 1, 0 (line 56 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a42)
Time, __SEH_epilog - Module Total Count = 1, Total Doubtful or Shared Counts = 0

RtlSizeHeap, 0x77f50000, 0x77f82496, 51549, 0x77f82574, 1, 1, 0 (line 4389 in d:\nt\base\ntos\
rtl\heap.c, 0x77f82571)
Time, RtlSizeHeap - Module Total Count = 1, Total Doubtful or Shared Counts = 0

_NLG_Notify, 0x77f50000, 0x77fa4b26, 86730, 0x77fa4b28, 1, 1, 0 (line 286 in D:\NT\base\crts\
crtw32\misc\i386\exsup.asm, 0x77fa4b28)
Time, _NLG_Notify - Module Total Count = 1, Total Doubtful or Shared Counts = 0

__SEH_prolog, 0x77f50000, 0x77fa4a04, 86658, 0x77fa4a08, 1, 0, 1 (line 32 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a04)
__SEH_prolog, 0x77f50000, 0x77fa4a04, 86661, 0x77fa4a14, 1, 0, 1 (line 36 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a14)
__SEH_prolog, 0x77f50000, 0x77fa4a04, 86664, 0x77fa4a20, 2, 2, 0 (line 41 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a20)
__SEH_prolog, 0x77f50000, 0x77fa4a04, 86666, 0x77fa4a28, 1, 0, 1 (line 45 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a28)
__SEH_prolog, 0x77f50000, 0x77fa4a04, 86671, 0x77fa4a3c, 2, 1, 1 (line 49 in d:\nt\base\crts\
crtw32\misc\i386\sehprolg.asm, 0x77fa4a38)
Time, __SEH_prolog - Module Total Count = 7, Total Doubtful or Shared Counts = 0

RtlReAllocateHeap, 0x77f50000, 0x77f8c86e, 62105, 0x77f8ca64, 1, 1, 0 (line 766 in d:\nt\base\
ntos\rtl\heapdll.c, 0x77f8ca5d)
RtlReAllocateHeap, 0x77f50000, 0x77f8c86e, 62724, 0x77f8d410, 2, 2, 0 (line 1257 in d:\nt\base\
ntos\rtl\heapdll.c, 0x77f8d3fc)
Time, RtlReAllocateHeap - Module Total Count = 3, Total Doubtful or Shared Counts = 0

RtlpCoalesceHeap, 0x77f50000, 0x77f57826, 7734, 0x77f578d8, 2, 1, 1 (line 4728 in d:\nt\base\
ntos\rtl\heapdll.c, 0x77f578d6)
Time, RtlpCoalesceHeap - Module Total Count = 2, Total Doubtful or Shared Counts = 0

RtlDosPathNameToNtPathName_U, 0x77f50000, 0x77f5980e, 9747, 0x77f5984c, 1, 1, 0 (line 2537 in d:\
nt\base\ntdll\curdir.c, 0x77f59842)
Time, RtlDosPathNameToNtPathName_U - Module Total Count = 1, Total Doubtful or Shared Counts = 0

_KiUserCallbackDispatcher, 0x77f50000, 0x77f75030, 37900, 0x77f75030, 2, 2, 0 (line 139 in D:\NT\
base\ntos\rtl\i386\userdisp.asm, 0x77f75030) , Actual Hits Should be Attributed to or Shared with
===> KiUserCallbackDispatcher+0x0 (line 139 in D:\NT\base\ntos\rtl\i386\userdisp.asm, 0x77f75030)
Time, _KiUserCallbackDispatcher - Module Total Count = 2, Total Doubtful or Shared Counts = 2

RtlTimeFieldsToTime, 0x77f50000, 0x77f8fa0e, 65168, 0x77f8fa40, 1, 1, 0 (line 892 in d:\nt\base\
ntos\rtl\time.c, 0x77f8fa3e)
RtlTimeFieldsToTime, 0x77f50000, 0x77f8fa0e, 65209, 0x77f8fae4, 1, 1, 0 (line 924 in d:\nt\base\
ntos\rtl\time.c, 0x77f8fa48)
Time, RtlTimeFieldsToTime - Module Total Count = 2, Total Doubtful or Shared Counts = 0
.........(Output truncated)

----- Zoomed module ntdll (Bucket size = 4 bytes, Rounding Down) --------

Time   251 hits, 25000 events per hit --------
 Module                                Hits       msec  %Total  Events/Sec
_RtlEnterCriticalSection                 57      40005    22 %       35620
     0      37  40005    14 %       23122
     1      20  40005     7 %       12498
RtlAllocateHeap                          32      40005    12 %       19997
     0      21  40005     8 %       13123
```

```
     1      11  40005     4 %        6874
_RtlLeaveCriticalSection                  32      40005    12 %       19997
     0      21  40005     8 %       13123
     1      11  40005     4 %        6874
RtlFreeHeap                               15      40005     5 %        9373
     0      11  40005     4 %        6874
     1       4  40005     1 %        2499
ExpInterlockedPopEntrySListResume          8      40005     3 %        4999
     0       0  40005     0 %           0
     1       8  40005     3 %        4999
RtlpInterlockedPopEntrySList               7      40005     2 %        4374
     0       1  40005     0 %         624
     1       6  40005     2 %        3749
__SEH_prolog                               7      40005     2 %        4374
     0       3  40005     1 %        1874
     1       4  40005     1 %        2499
_RtlInitUnicodeString                      5      40005     1 %        3124
     0       4  40005     1 %        2499
     1       1  40005     0 %         624
.........(Output truncated)

.........(Output for other instances of IEXPLORE.EXE truncated)

================================ END OF RUN ================================
```

Notes:

In the case of Kernel locks (ERESOURCE type) kernrate found two instances with a contention count higher than 100, but no symbol info is available (only the address). In the case of the user mode process IEXPLORE.EXE one lock (CRITICAL_SECTION) was found as well as symbol information.

The verbose level 2 output (as well as the raw data option) gives details about shared buckets. It will also attempt to give source code file and line information if available.