

Contents For APIMON Help

Introduction

[What Is APIMON?](#)

[How To Use APIMON](#)

Usage

[Command Line Options](#)

Options

[Miscellaneous Options](#)

[File Names Options](#)

[Known DLLs Options](#)

What Is APIMON?

APIMON is a Application Monitor. It monitors a running application for all Application Programming Interface (**API**) calls. APIMON counts and times all API calls. It will also, optionally, monitor page faults caused by the monitored application and report them by API call.

APIMON is used as a performance monitor to help tune your application. The data can be used to determine where the hot spots are in your application by examining the APIs with the largest times and/or counts. You can also look at which APIs are used by the application, taking a special look at calls that cause ring transitions to kernel mode or client/server transitions. Large numbers of these calls can be culprits for the cause of a slow application.

APIMON can also generate 2 types of reports: a report containing all API calls showing their counts and times, and a report showing a trace of all APIs as they occurred in time.

How To Use APIMON

APIMON is launched like any other application, either from a command shell, program manager, or the explorer. After APIMON starts you must open a program for monitoring. You can optionally specify the program name as an argument to APIMON.

After a program is opened you can start the program by *pushing* the *go traffic light* or selecting the *Start Monitor* menu item under the *Tools* menu. When the application starts two windows appear. The first window is the API counter and time and the second is the current DLL list window.

Command Line Options

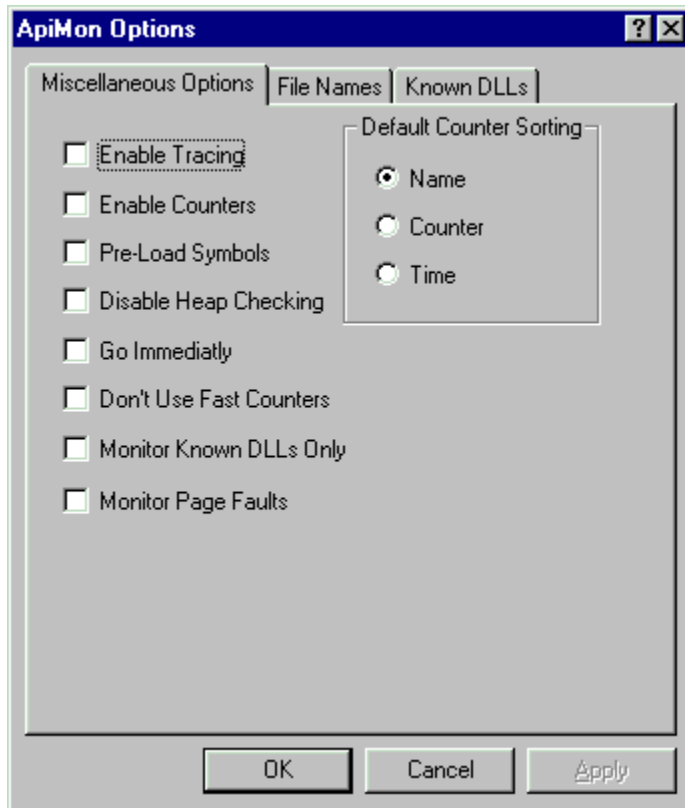
APIMON takes one argument in addition to the options, the application name that is to be monitored. The command line syntax is as follows:

APIMON [options] [appname]

<u>Option</u>	<u>Description</u>
-g	Start the Application Immediately

Miscellaneous Options

Use this property page to set the options that control the behavior of APIMON.



To get help on an option just click on the text for the option.

Enable Tracing

Turns on API tracing. This causes APIMON to create a trace file containing one line of text for each API call made by the monitored application. The trace contains the following information:

1. Return Address
2. First four arguments
3. API Name

Enable Counters

Turns on API counting and timing.

Pre-Load Symbols

Enabling this option causes APIMON to load the debug symbols for all DLLS while the monitored program is executing. This option is turned on automatically when *Monitor Page Faults* is turned on.

Disable Heap Checking

When a process is being debugged, NTDLL's heap manager turns on a set of heap checking flags. This ensures that if the process corrupts its heap that the debugger will catch the error. This behavior also slows down the process and changes the characteristics of the heap. This option disables this behavior and allows the process to run as if it were not being debugged.

Go Immediately

This option causes APIMON to start the monitor for the application as soon as the program is opened through the *File.Open* menu selection or when a program name is specified on the APIMON command line.

Don't Use Fast Counters

On the Intel Pentium and the Alpha AXP processors APIMON can use special instructions to do the API timings. The other processors must use the WIN32 API called `GetPerformanceCounter()`. This option disables the use of these instructions if APIMON is running on one of the supported platforms.

Monitor Known DLLs Only

By default APIMON monitors API calls to all DLLs that the monitored process loads, either statically or dynamically. This option provides for the monitoring of a specific list of DLLs to be monitored. To create the DLL list see the [Known DLLs Options](#) .

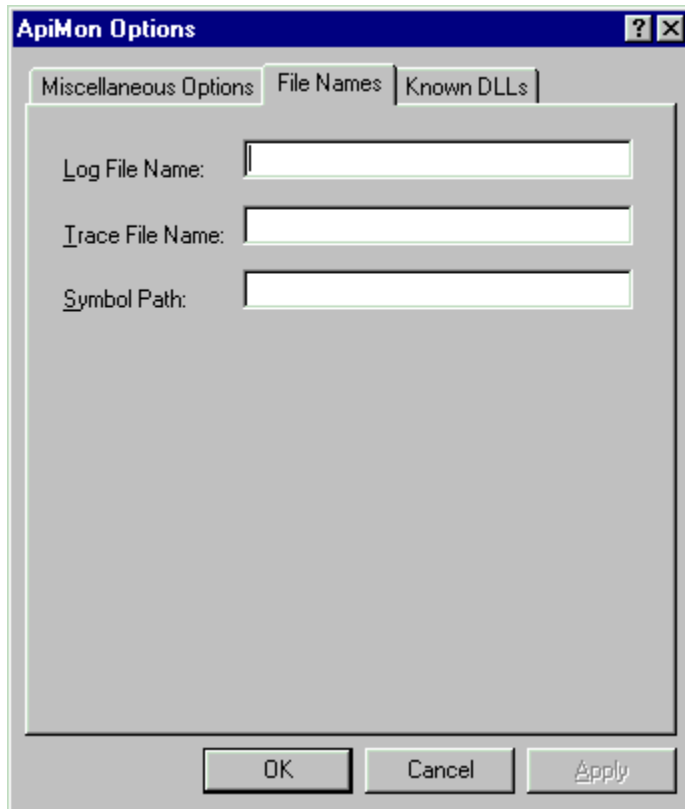
Monitor Page Faults

Enabling this option causes APIMON to create a new window that displays page fault information. The window displays one entry for each API from all enabled DLLs that has caused a page fault.

Default Counter Sorting

This option specifies the default sort order for the API counter window. The sort order can be changed while the monitor is running.

File Names Options



The image shows a screenshot of the "ApiMon Options" dialog box. The title bar is blue with the text "ApiMon Options" and a help icon (?). The dialog has three tabs: "Miscellaneous Options", "File Names", and "Known DLLs". The "File Names" tab is selected. Inside the dialog, there are three text input fields with labels: "Log File Name:", "Trace File Name:", and "Symbol Path:". Each label has a small underlined letter (L, T, S) indicating a mnemonic. At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Apply".

To get help on a file name just click on the text for the file name.

Log File Name

Specify a full path and file name for the APIMON log file. You may specify a file on your local machine or a UNC name to a network path and name. This file is created when the menu command *File.Write Data To Log File* is selected. The file is overwritten each time the log file is created.

Trace File Name

Specify a full path and file name for the Trace file. You may specify a file on your local machine or a UNC name to a network path and name. This file is created when the [Enable Tracing](#) option is turned on. The file is overwritten each time the trace file is created.

Symbol Path

This string is a path or multiple paths separated by semicolons. Each path points to a tree containing debug symbols. The debug symbols can be either *.DBG* files, *.EXE* files, or *.PDB* files. This string is necessary if the [Pre-Load Symbols](#) option is turned on.

Known DLLs Options

```
{bml DLG3.BMP}
```

This property page is used to specify a list of *Known DLLs* that are initially monitored by APIMON. This can be nice if you don't want APIMON to clutter the display and log file with statistics that you don't care about.

To add a new entry to the list just click on an empty entry and an edit box appears. Enter the DLL name and press enter. The DLL name **MUST** not contain a full path, just the DLL name with a *.DLL* extension.

To delete an entry just click on the desired entry, delete all of the text, and press enter. The entry is then removed from the list.

In order for this DLL list to take effect the [Monitor Known DLLs Only](#) option must be turned on.

