

Windows Performance Analysis Tools

The Windows Performance Tools (WPT) Kit contains performance analysis tools that are new to the Windows SDK for Windows Server 2008 and .NET Framework 3.5. The WPT Kit is useful to a broad audience, including system builders, hardware manufacturers, driver developers, and general application developers. These tools are designed for measuring and analyzing system and application performance on Windows Vista, Windows Server 2008, and later.

Toolkit

Windows Performance Tools are designed for analysis of a wide range of performance problems including application start times, boot issues, deferred procedure calls and interrupt activity (DPCs and ISRs), system responsiveness issues, application resource usage, and interrupt storms.

These tools ship with the Windows SDK (starting with Feb'08 Windows Server 2008 SDK). Latest QFEs of these tools are also available for download on this developer center. The MSIs containing these tools are available in the SDK bin directory (one per architecture).

What Tools Do I Get?

Windows Performance Toolkit is released as an MSI installer, one per architecture, and presently contains the Performance Analyzer tool suite consisting of the following three tools:

Tool Name	File Name	Function
Trace Capture, Processing, and Command-Line Analysis tool	<i>Xperf.exe</i>	Captures traces, post-processes them for use on any machine, and supports command-line (action-based) trace analysis.
Visual Trace Analysis tool	<i>Xperfview.exe</i>	Presents trace content in the form of interactive graphs and summary tables.
On/Off Transition Trace Capture tool	<i>Xbootmgr.exe</i>	Automates on/off state transitions and captures traces during these transitions.

How Do These Tools Fit Together?

The tools use a trace capture and analysis model that follows this general flow:

1. ETW tracing is enabled using `xperf` (or `xbootmgr`).
2. User scenario of interest is performed.
3. ETW tracing is stopped using `xperf` (or automatically by `xbootmgr`) and the data is saved to an ETL trace file.
4. Trace files can then further be processed with `xperf` and/or viewed with `xperfview`.

Traces can be processed on the machine on which they were taken, or copied to another machine for analysis (including cross-architecture). Everything needed for analysis is stored in the trace file.

What Makes These Tools Possible?

Performance Analyzer is built on top of the Event Tracing for Windows (ETW) infrastructure. ETW enables Windows and applications to efficiently generate events, which can be enabled and disabled at any time without requiring system or process restarts. ETW collects requested kernel events and saves them to one or more files referred to as "trace files" or "traces." These kernel events provide extensive details about the operation of the system. Some of the most important and useful kernel events available for capture and analysis are context switches, interrupts, deferred procedure calls, process and thread creation and destruction, disk I/Os, hard faults, processor P-State transitions, and registry operations, though there are many others.

One of the great features of ETW, supported in WPT, is the support of symbol decoding, sample profiling, and capture of call stacks on kernel events. These features provide very rich and detailed views into the system operation. WPT also supports automated perf testing. Specifically, `xperf` is designed for scripting from the command line and can be employed in automated performance gating infrastructures (it is the core of Windows PerfGates). `xperf` can also dump the trace data to an ANSI text file, which allows you to write your own trace processing tools that can look for performance problems and regressions from previous tests.