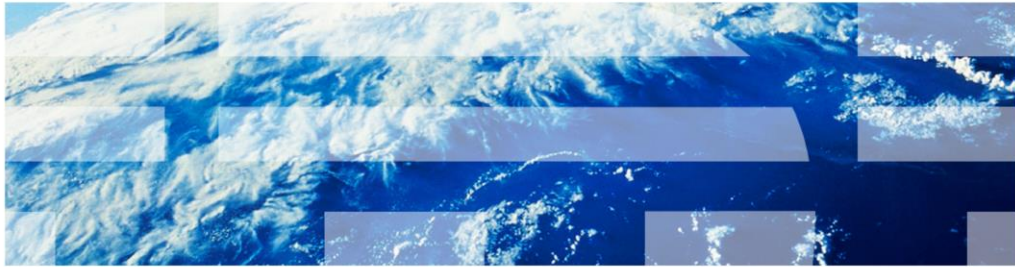


# IBM Tivoli Monitoring V6.2.3

## How to debug Windows performance objects issues - Overview and tools



© 2012 IBM Corporation

IBM Tivoli® Monitoring V6.2.3, how to debug issues with Windows® performance objects issues - overview and tools.

## Assumptions

Before you proceed, the module designer assumes that you have these skills and knowledge:

- Basic knowledge of IBM Tivoli Monitoring
- Basic knowledge of IBM Tivoli Monitoring Windows Agent
- Basic knowledge of the Windows operating system

The module developer assumes that you understand basic IBM Tivoli Monitoring concepts and have a basic knowledge of the Windows OS agent.

It is also useful if you have a basic knowledge of the Windows operating system to better understand Windows-specific concepts like the Windows performance objects libraries.

## Objectives

When you complete this module, you can perform these tasks:

- Describe performance objects and the tools that are used to investigate their problems
- Describe when a Windows OS agent failure can be related to Windows performance objects issues
- Perform actions on Windows performance objects to correct problems

When you complete this module, you can describe performance objects and the tools that are used to investigate their problems. You can also describe when a Windows OS agent failure is related to Windows performance objects issues and perform actions to correct the problem.

## Windows performance objects and counters (1 of 2)

- They are operating system components that are available by default with all the Windows Servers family
- They obtain data about performance and the availability of specific Windows components  
For example: LogicalDisk object contains counters to monitor Disks activity
- Performance objects are available for the most important hardware and software OS components
- Windows applications that deliver, install, and manage performance objects:
  - Microsoft® SQL Server
  - Microsoft Exchange Server
  - Windows Internet Name Service (WINS)
  - Others

The Windows Server family of operating systems gathers performance data about specific services, or about the operating system itself, through the implementation and activation of performance objects.

The performance object is typically named for the component that is generating the data.

For example, the *Logical Disk* object is a collection of performance data about disk volumes that are attached to your system.

Performance objects are built and run in the operating system, typically corresponding to the major hardware and software components such as processors, disks, and network.

Other services might install their own performance objects.

For example, services such as Windows Internet Name Service (WINS) or server programs such as Microsoft SQL Server provide their own performance objects.

## Windows performance objects and counters (2 of 2)

- Each performance object contains many counters that represent specific performance or availability data for system or services
- Performance objects and counters are implemented through specific Performance DLL libraries  
For example, **perfdisk.dll**, **perfproc.dll**

Each performance object provides performance counters that represent data on specific aspects of the operating system or service.

For example, the *Logical Disk* object provides the **Pages % Disk Write Time** counter that tracks the percentage of time that is spent for the write operation in the last collection period.

Performance Library DLL are the software modules that are installed on the machine responsible for gathering specific performance data.

There is one specific Performance Library DLL responsible for gathering data for each performance object and its counters.

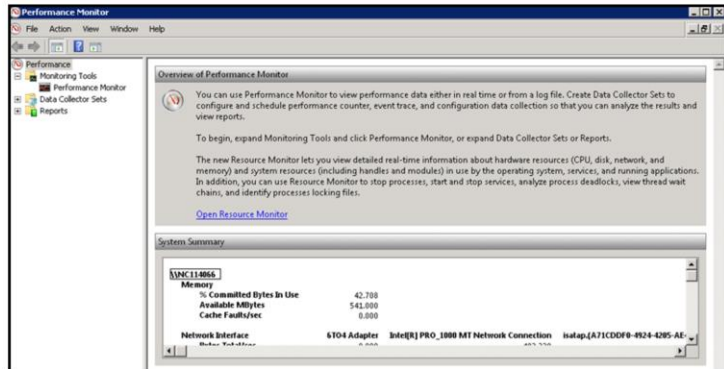
Many Performance Library DLL gather data for more than one object.

For example, the performance library **PerfDisk.dll** contains definition for disk performance objects and counters.

## Troubleshooting tools: Microsoft performance monitor tool

- The performance monitor tool is useful to:
  - Verify available performance objects and counters
  - Create reports and data collection
  - Verify that the Windows OS agent is collecting the right data
- The performance monitor tool is started from the command line or from the Run panel with the command:

***perfmon***



6

How to debug Windows performance objects issues - Overview and tools

© 2012 IBM Corporation

Some tools are available to troubleshoot issues with the performance objects activity.

The first one is the Microsoft performance monitor tool, also known as **perfmon**.

The performance monitor tool is available by default for all the different version of Windows operating systems, including both *Servers* or the *Workstation Edition*.

It is a good and quick method to verify the available performance objects and related counters.

With the performance monitor tool, you can perform other additional tasks like creating data collection for specific objects and counters or create reports.

In the Tivoli Monitoring context, it is used for a troubleshooting purpose. For example, to verify whether a specific counter in the performance monitor tool and the Windows OS agent provides the same value.

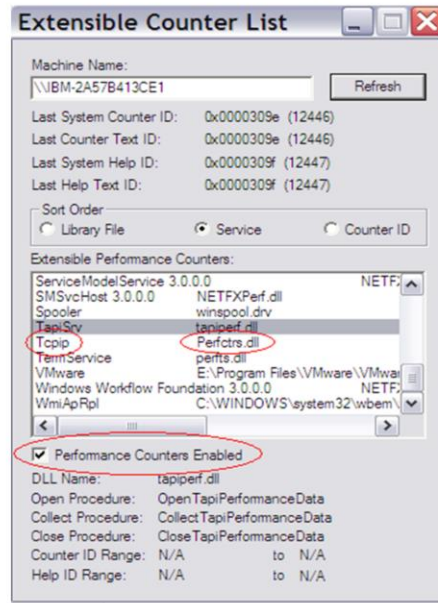
The performance monitor tool can be run from the **Run** panel, or directly from a command line, typing the command: **perfmon**.

## Troubleshooting tools: Extensible performance counter list (1 of 2)

- Available with the Windows Resource Kit or downloadable
- Started with the binary **exctrlist.exe**
- Returns useful information about the DLL file that contains specific objects and counters and the status of the service (enabled / disabled)
- Allows you to enable or disable performance counters that are related to a performance library

Windows Resource Kit download:

<http://www.microsoft.com/downloads/details.aspx?familyid=7ff99683-b7ec-4da6-92ab-793193604ba4&displaylang=en>



7

How to debug Windows performance objects issues - Overview and tools

© 2012 IBM Corporation

Another useful tool is the **Extensible Performance Counter List** that is also known as **exctrlist**: the name of the executable file that is used to invoke the tool.

This tool is part of the Windows Resource Kit, available with all the known releases of Windows Servers, including Windows 2008.

You can also download it as a single program from the Microsoft link shown in this slide.

The tool is useful to verify the status of specific performance object libraries, for example, to check whether the related service is enabled or not.

The Windows OS agent might not return data in the Network workspace views for TCPIP statistics.

The first step is to check whether the related performance service is disabled by running the **exctrlist** command.

You must select the pertinent row in the list of the Extensible Performance Counters and select the **Performance Counters Enabled** check box.

In this slide, you can see that this check box is enabled.

## Troubleshooting tools: Extensible performance counter list (2 of 2)

- Typical problem scenario with the Windows OS Agent:
  - Missing performance metrics: Check if the related performance service is disabled
  - Receiving errors with the performance counters activity: Disable the related performance service
  - Memory leak: Performance counters libraries are corrupted

The operating system can automatically disable performance counters in case the invocation of the related libraries results in repeated errors.

If you experience missing performance metrics for the Windows Agent, check whether the related performance service is enabled or not.

If the performance counter is not enabled, investigate why it was deactivated. It is possible that the performance objects or the DLL file are corrupted.

If you are facing problems with the Windows OS agent. For example, if you are getting errors for the Performance Counters activity, you can implement a quick workaround by clearing the check box.

In this way, the performance service is disabled and all the users of the service, like the Windows OS agent, do not use it.

A typical problem scenario is the memory leak of the Windows OS agent process. This problem can happen when the performance counters libraries are corrupted and this situation can cause the memory leak.

When you identify the failing performance object, you can use the **exctrlst** tool to disable it temporary, allowing the Windows OS agent to work until the performance library is recovered.



## Services, libraries, and performance objects

- Services are the providers of the performance data. The performance objects clients use the services to gather the performance metrics
- Libraries are the physical files that contain the definition of performance objects and counters
- Performance objects identify the performance area, for example, Logical Disk or Processor, and can be seen as the container of the performance counters

Service	Library	Objects
PerfOS	PerfOs.dll	Cache, memory, Object, paging File, Processor, System
PerfDisk	PerfDisk.dll	LogicalDisk, PhysicalDisk
PerfNet	PerfNet.dll	Browser, Redirector, Server
PerfProc	PerfProc.dll	Job Object, Job Object Details, Process, Thread, Thread Details
Spooler	Winspool.dvr	Print Queue
TapiSrv	TapiPerf.dll	Telephony
Tcpip	Perfctrs.dll	IP, ICMP, NBTC Connections, Network Interface, TCP, UDP

9

How to debug Windows performance objects issues - Overview and tools

© 2012 IBM Corporation

You already know what a *performance object* is. You can see it as the container of performance counters for a specific performance area, for example, Logical Disk with its counters or Processor with its counters.

The definitions of the performance objects and counters are specified within the library files. These files have DLL as their extension. Each DLL can contain a definition for one or more performance objects.

The services are the providers of the performance data. Any programs that want to gather performance metrics, see the service that delivers the needed performance object and counters.

For example, if you want to gather information about Thread Details, you need to have the **PerfProc** service that is enabled and available.

It can happen that even if the service is shown as enabled, the metric is not collected, or shows unexpected values. Typically this means that the performance object is corrupted and needs to be reloaded.

## Useful Microsoft links

- How to manually rebuild Performance Counter Library values
  - <http://support.microsoft.com/kb/300956>
    - Provides detailed steps about:
      - How to reload extensible counters from the ini file
      - How to manually add counters in the registry in case the keys are missing or corrupted
      - How to rebuild all the performance counters
- Performance Object Is Not Displayed in Performance Monitor
  - <http://support.microsoft.com/default.aspx?scid=kb;en-us;248993>
  - Provides the following information:
    - Possible reason about why Perfmon does not show performance objects
    - How to re-enable the Performance Counters by using **exctrlst** tools
    - How to re-enable the Performance Counters manually from the registry

Put these two Microsoft technotes in your bookmark list. They help with most of the scenarios where your Tivoli Monitoring Windows OS agent is not working because of the performance counter misbehavior.

The first link provides details about how to reload counters from an .ini file. This operation is useful when the DLL file is corrupted and rebuilding the counters from the installed DLL does not help.

This technote also describes the steps that are used to manually add registry keys of the performance objects, in case you recognize them as missing or corrupted.

This Technote also provides instruction about how to rebuild the whole set of performance counters with a single command.

The second technote provides information about the possible reasons why the Performance Monitor tool does not show the expected performance objects and related counters.

If the Performance Monitor tool does not see the performance objects, then the Windows OS agent is not able to gather the related performance metrics.

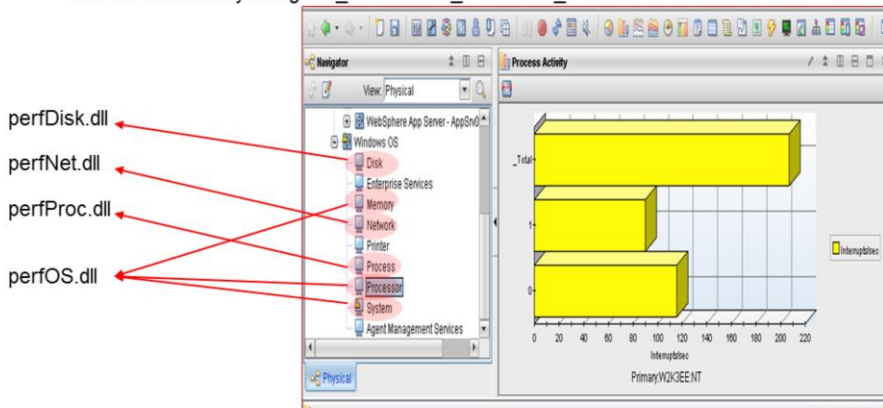
For this reason, this technote is useful to troubleshoot problems with the Windows OS agent. In fact, the Windows OS agent is a perflib client like the **perfmon** tool.

The technote also provides instructions on enabling performance counters. You can use the **exctrlst** tool that is described in the previous slides or you can do it manually by altering specific registry keys.

It is always better to make the Windows System Administrator aware of your intentions before changing the registry, either by making manual changes or by using the **exctrlst** tool.

## Windows OS agent and performance objects

- The Windows OS agent collects most of the performance metrics from the Windows performance object libraries
- At initialization time, the agent:
  - Loads the **perflib.dll**
  - Obtains the list of the available performance objects and counters
  - Verify whether one or more of the counters can lead to memory leak and in case exclude it.  
Can be disabled by using `NT_PERFMON_MEMORY_CHECK=0` in the **KNTENV** file



11

How to debug Windows performance objects issues - Overview and tools

© 2012 IBM Corporation

This slide introduces concepts about the Windows performance counters because the Tivoli Monitoring Windows OS agent uses these components to gather performance metrics.

The Windows OS agent collects most of the data from the Windows performance objects. When the agent is started, it loads the main performance library that is named **perflib.dll**, and obtains from it the list of available objects and counters.

In Windows, a specific index is associated to each counter, and you can see from the agent log the list of retrieved and active counters by their name and index.

After the agent retrieves the list of the counters, it checks the behavior of the counters in order to verify whether they can lead to unexpected memory leaks.

If there is a memory leak, the specific counter is excluded.

This feature was introduced after corruptions or misbehavior of performance libraries caused unexpected problems to the agent itself.

The `NT_PERFMON_MEMORY_CHECK` environment variable can be set in the **KNTENV** file to disable the memory check on the performance counters when the agent starts.

As you can see from this slide, there is a close link between the Microsoft DLL performance libraries and the Windows OS agent metrics.

You can see that **Disk** metric relies on a library that is called **perfDisk.dll**. Most of the Network statistics are gathered from the library **perfNet.dll**. Process information is collected from **perfProc.dll**, and by using the Windows API. Memory, Processor, and System metrics are collected from the **PerfOS.dll** that is one of the most important libraries because it contains the most commonly used metrics.

## Summary

Now that you have completed this module, you can perform these tasks:

- Describe performance objects and the tools that are used to investigate their problems
- Describe when a Windows OS agent failure is related to Windows performance objects issues
- Perform actions on Windows performance objects to correct problems

Now that you have completed this module, you can describe performance objects and the tools that are used to investigate their problems. You can also describe when a Windows OS agent failure can be related to Windows performance objects issues and perform actions to correct the problems.

## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.