

Top 5 leading WMI tools

Windows Management Instrumentation, or WMI, is a technology that makes it easier to manage Microsoft Windows-based servers and workstations.

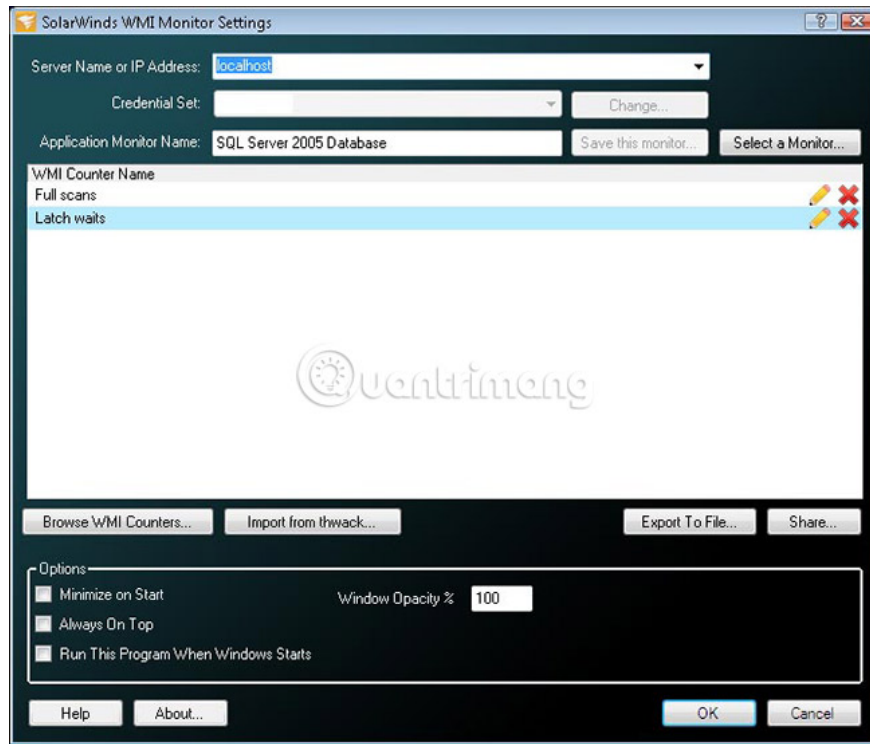
Windows Management Instrumentation, or WMI, is a technology that makes it easier to manage Microsoft Windows-based servers and workstations. It collects and reports configuration information to the computer using the standard interface. In addition to viewing configuration information, WMI can also work with Windows computers by configuring the computer's configuration. WMI is not a configuration tool, but a framework builder for remote viewing and configuration of individual Windows computers. Before WMI was born in the mid-1990s, managing remote Windows computers was a very difficult task. The first WMI deployment was quite limited, but this has changed to include support for a variety of hardware and software configurations.

WMI is very similar to a network management protocol that most network administrators are familiar with: SNMP. Like SNMP, WMI is between management applications, scripts and configuration settings on the server. However, unlike SNMP, WMI is designed from scratch to support a multitude of configuration items. This means that WMI can report and control complex Windows configuration items such as mount points. The most popular WMI tools are divided into two categories: Browser (Browser) and manipulator. As the name implies, the browser uses to browse the WMI and manipulator objects used to make changes. There are countless WMI tools, but the following 5 tools are the best options.

Top 5 leading WMI tools

1. Solarwinds Free WMI Monitor (RECOMMENDATIONS)
2. WMI Explorer (CodePlex)
3. WMI Explorer (Sapien)
4. WMI Explorer (Marc van Orsouw)
5. Powershell

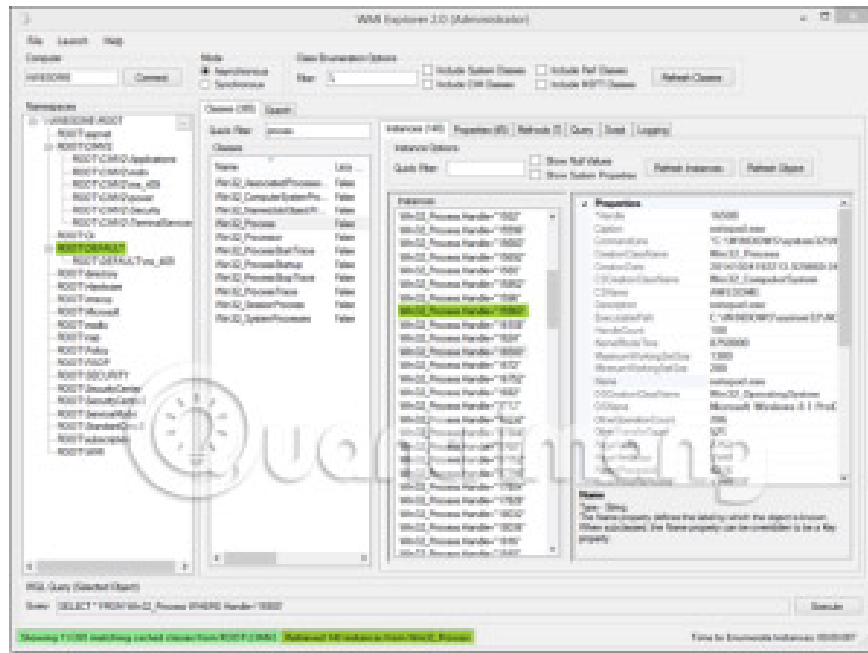
1. Solarwinds Free WMI Monitor (RECOMMENDATIONS)



Solarwinds Free WMI Monitor monitors every Windows application or server, giving you insight into real-time performance. You can use custom templates, developed by the community, built-in or customized to start tracking right after downloading. Fast and easy WMI Monitor gives you an overview of the performance of servers and applications with a new desktop dashboard. You can modify or design your own application templates with the integrated WMI browser - no more writing WMI scripts.

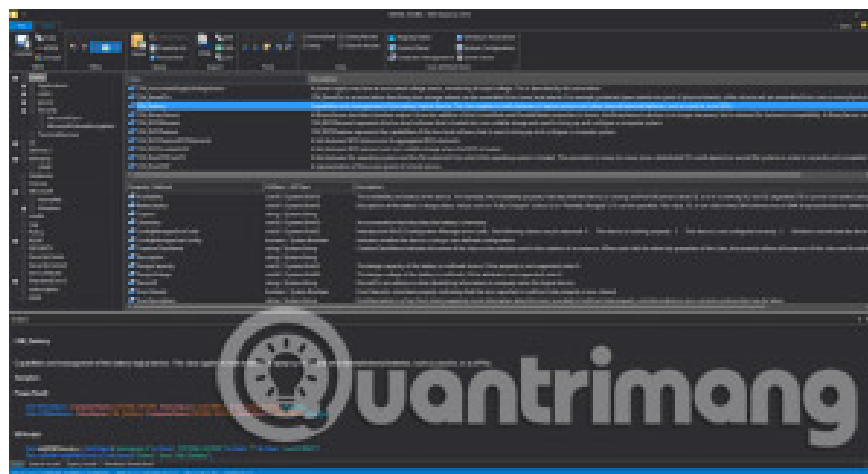
Solarwinds also has a great tutorial on how to configure and use WMI Monitor:

2. WMI Explorer (CodePlex)



WMI Explorer distributed by CodePlex is a popular tool. It provides a modern, intuitive interface. This tool displays the WQL query to receive the selected objects or attributes, as well as creating scripts to provide Powershell / VBScript users with the corresponding to get them. This can be particularly useful when working with parts of the Windows system that Powershell does not include with Cmdlets. One of the most popular features about this WMI Explorer is that it is free, so you can install it at will without worrying about buying a license.

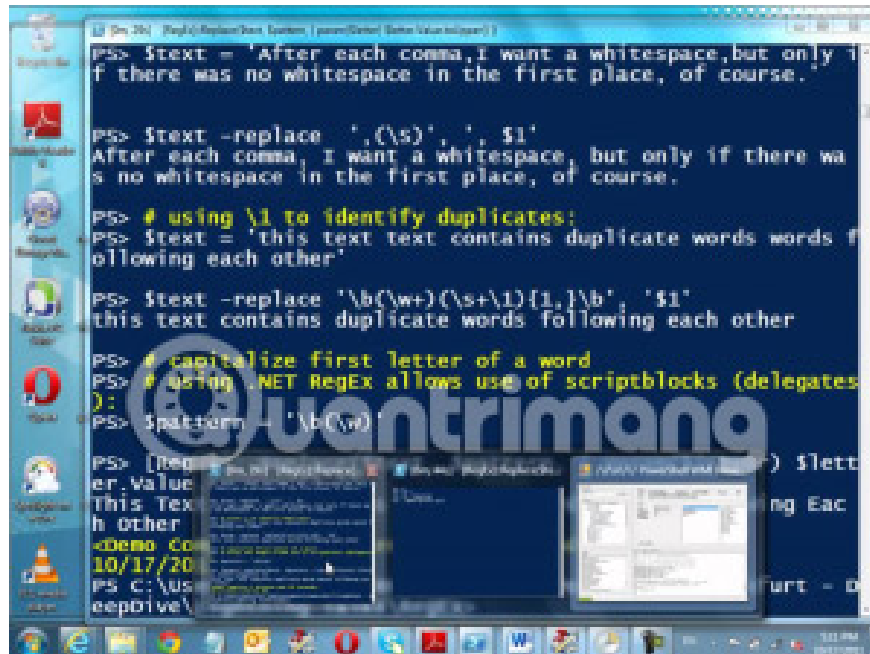
3. WMI Explorer (Sapien)



Although not free, Sapien's WMI Explorer shows some very nice features that CodePlex WMI Explorer doesn't have. A great example is the integrated Technet button, which will search your selected WMI object via Technet. The user interface is also a bit more intuitive, with the tabs below showing Powershell and VBScript code to retrieve the selected item, along with the Powershell dashboard embedded for testing with the browser. The user interface also seems to be styled according to the new Ribbon-based Microsoft Office menus, making the

interface more familiar.

4. WMI Explorer (Marc van Orsouw)



```
PS> $text = 'After each comma,I want a whitespace,but only if there was no whitespace in the first place, of course.'
PS> $text -replace ',(\s)', ', $1'
After each comma, I want a whitespace, but only if there was no whitespace in the first place, of course.
PS> # using \1 to identify duplicates:
PS> $text = 'this text text contains duplicate words words following each other'
PS> $text -replace '\b(\w+)(\s+\1){1,}\b'; '$1'
this text contains duplicate words following each other
PS> # capitalize first letter of a word
PS> # using .NET RegEx allows use of scriptblocks (delegates)
PS> $pattern = '\b(\w)'
PS> [regex]::Replace($text, $pattern, { $1.ToUpper() })
This Text Contains Duplicate Words Following Each Other
PS C:\Users\eeprive\...
```

The big advantage of Marc van Orsouw WMI Explorer is that it is completely based on Powershell. That means it will run on Windows Server 2008 or newer operating system without installation. This can be a great advantage if you want to place WMI Explorer on all your systems or if you just want to use it quickly without having to go through a complicated installation process. It is quite typical compared to some other options, but it is useful if you just want to quickly find a WMI object for the Powershell script.

5. Powershell

1. About PowerShell

```
PS C:\> Get-Childitem 'MediaCenter\Media' -rec |
?? where { -not $_.FileContainer -and $_.Extension -match 'wmimp?' } |
?? Measure-Object -property length -sum -min -max -ave

Count           : 1392
Average         : 5491276.89563887
Sum             : 7177997857
Maximum        : 22785267
Minimum        : 3235
Property        : Length

PS C:\> Get-WmiObject CIM_MIOElement | select bios#, man#, oem# | Format-List

BIO#Version : 1000000000 - 1000000000, Ver 1.00P000000
Manufacturer : 1000000000
SerialNumber : 0000000000

PS C:\> (Get-WmiObject -class 'MIOElement') |
?? SELECT * FROM CIM_MIOElement
?? WHERE Priority > 1
?? "Bios.get()" | Format-Custom
??

- Last ManagementObjectFromWmi: class 'MIOElement'
{
  Document = Managed Manifests - Public
  JobId = 4
  JobStatus =
  Owner = User
  Priority = 42
  Size = 1024000
  Name = Epson Stylus COLOR 740 ESC/P 2, 4
}

PS C:\> $url = "http://blogs.mdn.com/powershell/res.aspx"
PS C:\> $blog = Get-WmiObject System.Net.WebClient).DownloadString($url)
PS C:\> $blog.res.channels.item | select title -first 3

title
---
MFE: What's Coming in PowerShell UE
PowerShell Presence at MFE
MFE Talk: System Center Foundation Technologies

PS C:\> $host.version.ToString().Insert(0, "Windows PowerShell: ")
Windows PowerShell: 1.0.0.0
PS C:\>
```

The tools discussed focused on finding WMI objects, but Powershell became special because it helped control the found objects. This is evidenced by the tight integration between Powershell and other WMI Explorer. When combined with information about attributes from one of the WMI Explorer, Powershell is the easiest way to access basic data from local or remote WMI store.

Administrators do not lack the tools at hand to work with WMI objects. In the Windows world, WMI often replaces SNMP, becoming the protocol of choice for tracking configuration information. The ability to view information about remote systems easily and incorporate centralized management software makes WMI an important part of the administrator's toolkit.

See more:

1. What is WMI Provider Host (WmiPrvSE.exe) and why is it using too much CPU?
2. Managing Windows networks using Script - Part 12: Properties of the WMI class
3. Top sites that support computer troubleshooting and technology

You finished reading the article "**Top 5 leading WMI tools**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.