

Fun with WMI Filters in Group Policy

By kexugit

Archived: 2026-04-06 03:17:15 UTC

Hi, [Ned](#) here again. You may remember Mike Stephens writing about [importing and exporting WMI filters](#) back in May. A common follow up question we got from that blog post was: “*Hey cool. So, uh, what are WMI filters again?*”

Group Policy WMI filters were introduced with Windows XP, and are supported in Windows Server 2003, Windows Vista, and Windows Server 2008. They are *not* supported in Windows 2000, so if you have an all-2000 environment you’re out of luck (10 years is a long time to go without upgrading :-P).

For those still with us...

You can use WMI filters to add a decision on when to apply a given group policy. This can be very useful when users or computers are located in a relatively flat structure instead of specific OU’s, for example. Filters can also help when you need to apply certain policies based on server roles, operating system version, network configuration, or other criteria. Windows evaluates these filters in the following order of overall Group Policy Processing:

1. Policies in hierarchy are located.
2. *WMI Filters are checked.*
3. Security settings are checked.
4. Finally, once everything has ‘passed’, a policy is applied.

So we find all the policies that exist in the user/computer’s Local, Site, Domain, and OU hierarchy. Then we determine if the WMI filter evaluates as *TRUE*. Then we verify that the user/computer has Read and Apply Group permissions for the GPO. This means that WMI filters are still less efficient than hierarchical linking, but can definitely use filters to make decisions in a non-hierarchical Active Directory design.

You configure WMI filters using the WMI Filters node in **GPMC.MSC**.

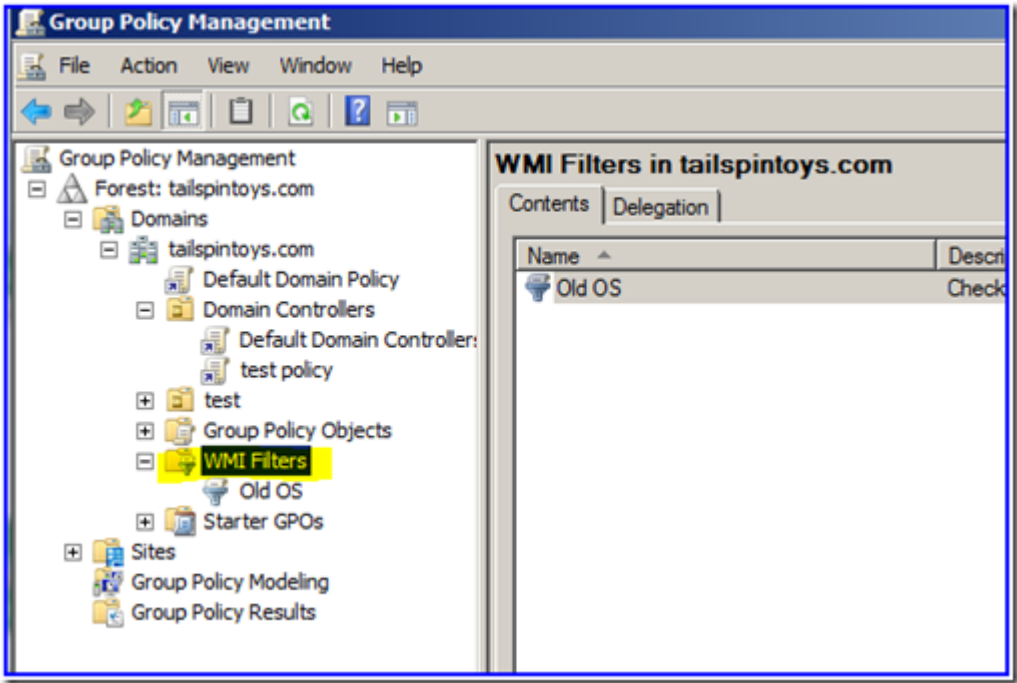


Figure 1 – GPMC WMI Filters Node

Then you can create, delete or edit a filter.

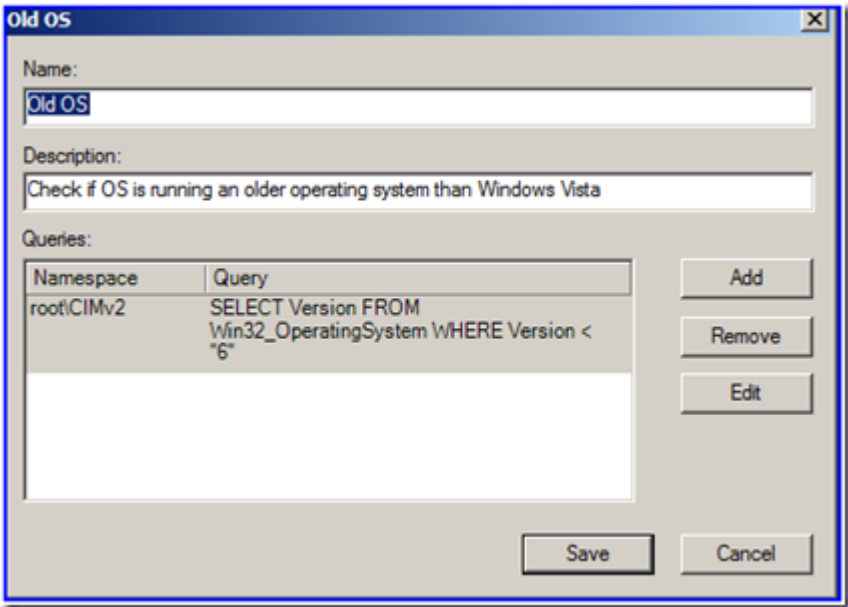


Figure 2 – WMI Filter Editor

Then you can link the WMI filter to any GPO you like (or more than one GPO), like below:

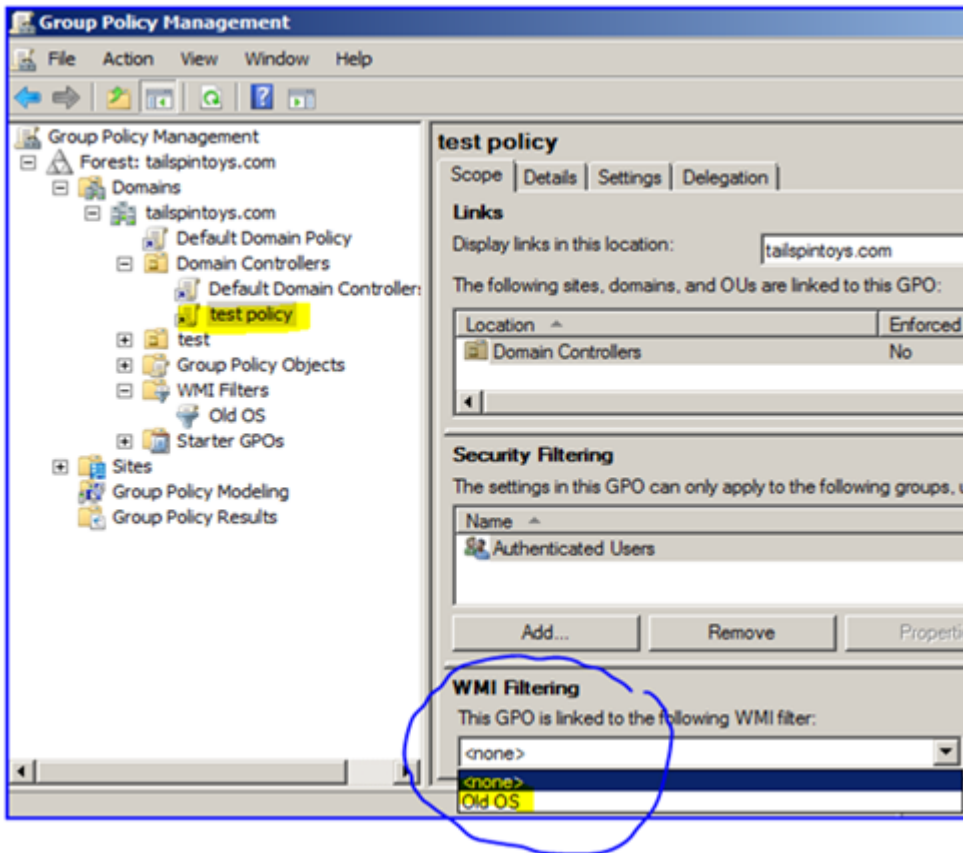
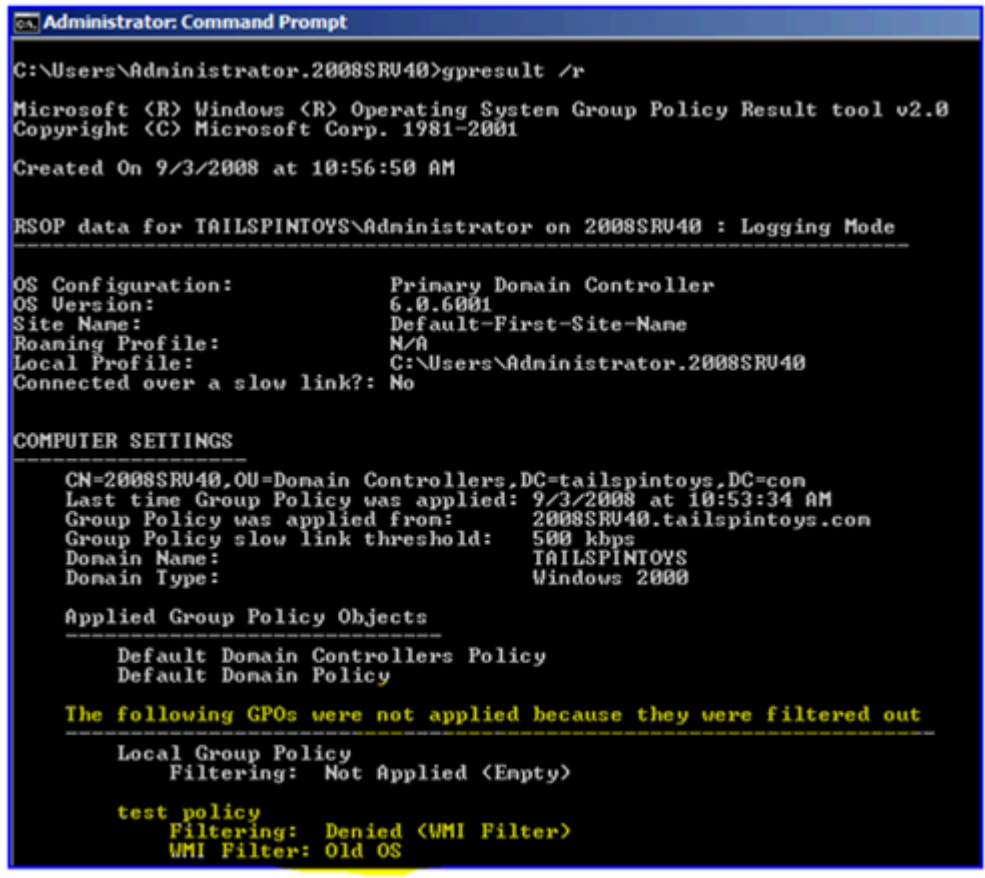


Figure 3 – GPMC Filter Dropdown

So in this case, I created a filter (you will see more on this below) that allows a GPO to apply to operating systems earlier than Windows Vista. I linked the WMI filter to a GPO that is applied to Windows Server 2008 computers –

so the GPO *shouldn't* apply. If I force Group Policy processing using **GPUPDATE /FORCE** then run



```
Administrator: Command Prompt
C:\Users\Administrator.2008SRU40>gpresult /r
Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0
Copyright (C) Microsoft Corp. 1981-2001

Created On 9/3/2008 at 10:56:50 AM

RSOP data for TAILSPINTOYS\Administrator on 2008SRU40 : Logging Mode
-----
OS Configuration:           Primary Domain Controller
OS Version:                 6.0.6001
Site Name:                  Default-First-Site-Name
Roaming Profile:            N/A
Local Profile:              C:\Users\Administrator.2008SRU40
Connected over a slow link?: No

COMPUTER SETTINGS
-----
CN=2008SRU40,OU=Domain Controllers,DC=tailspintoys,DC=com
Last time Group Policy was applied: 9/3/2008 at 10:53:34 AM
Group Policy was applied from: 2008SRU40.tailspintoys.com
Group Policy slow link threshold: 500 kbps
Domain Name:                TAILSPINTOYS
Domain Type:                Windows 2000

Applied Group Policy Objects
-----
Default Domain Controllers Policy
Default Domain Policy

The following GPOs were not applied because they were filtered out
-----
Local Group Policy
  Filtering: Not Applied (Empty)

test_policy
  Filtering: Denied (WMI Filter)
  WMI Filter: Old OS
```

GPRESULT /R, I see:

Figure 4 – GPRESULT output

Slick!

WMI filters use a language called *WQL*, which will be very familiar to anyone that has ever written a SQL query. The nice thing about learning WMI queries is that it forces you to learn more about the extremely powerful WMI system as a whole and its [massive repository of data](#) within it. WMI works within a construct of Namespaces and Classes. 99% of every WQL query will operate in the CIMV2 namespace, like all of the examples below.

So let's look at some syntax examples:

Only for certain operating systems

It is common to want Group Policy objects to apply to a computer using a specific operating system or service pack installed. Here are some examples that cover a few bases:

```
SELECT Version FROM Win32_OperatingSystem WHERE Version < "6"
```

The above WQL query returns true for any operating systems older than Vista (so Windows XP and Windows Server 2003).

```
SELECT Version FROM Win32_OperatingSystem WHERE Version LIKE "6.0%"
```

The above WQL query returns true for only Windows Vista or Windows Server 2008 operating systems.

SELECT Version FROM Win32_OperatingSystem WHERE Version = "5.1.2600"

The above WQL query returns true only if the operating system is Windows XP Service Pack 2.

SELECT * FROM Win32_OperatingSystem WHERE Version LIKE "6.0.%" AND ProductType <> "1"

The above WQL query returns true only if the computer is running Windows Server 2008 regardless of service pack. Why so complex, you ask? Remember that Windows Server2008 and Vista SP1 share the same codebase, so they actually have the same exact version. Choosing a product type not equal to 1 (which is Workstation) returns only servers or domain controllers running Windows Server 2008.

Only on Windows Server 2008 Core servers

What if you have a GPO that you want to apply only to servers running Windows Server 2008 Core installations? Here is a sample query (wrapped for readability, this should be done as a single line in the filter dialog):

SELECT OperatingSystemSKU FROM Win32_OperatingSystem WHERE OperatingSystemSKU = 12 OR OperatingSystemSKU = 39 OR OperatingSystemSKU= 14 OR OperatingSystemSKU = 41 OR OperatingSystemSKU = 13 OR OperatingSystemSKU = 40 OR OperatingSystemSKU = 29

These values map back to HEX values, which map back to:

Value	Meaning
PRODUCT_DATACENTER_SERVER_CORE0x0000000C	Server Datacenter Edition (core installation)
PRODUCT_DATACENTER_SERVER_CORE_V0x00000027	Server Datacenter Edition without Hyper-V (core installation)
PRODUCT_ENTERPRISE_SERVER_CORE0x0000000E	Server Enterprise Edition (core installation)
PRODUCT_ENTERPRISE_SERVER_CORE_V0x00000029	Server Enterprise Edition without Hyper-V (core installation)

PRODUCT_STANDARD_SERVER_CORE0x0000000D	Server Standard Edition (core installation)
PRODUCT_STANDARD_SERVER_CORE_V0x00000028	Server Standard Edition without Hyper-V (core installation)
PRODUCT_WEB_SERVER_CORE0x0000001D	Web Server Edition (core installation)

If you want GPOs to apply only to computers NOT running Windows Server 2008 Core (and you can probably think of some reasons to do that), then you would change all the equal signs (=) in the above query to signs above to angled brackets (<>).

(See <https://msdn2.microsoft.com/en-us/library/ms724358.aspx> for details and the non-CORE values.)

Only on a certain day of the week

Yes this is possible! Yes, customers have asked how to do this! No, I have no idea why! Ok, kidding about that last one, but it sure seems like an odd request at first. It turns out that some companies like to do things like set a specific *message of the day* for their legal notice. Or have a separate screensaver running every day of the week for their users. Different strokes for different folks, I suppose.

To do this, your WQL queries (*one filter per GPO* that you wanted to set, remember) would be:

Select DayOfWeek from Win32_LocalTime where DayOfWeek = 1

Select DayOfWeek from Win32_LocalTime where DayOfWeek = 2

Select DayOfWeek from Win32_LocalTime where DayOfWeek = 3

You get the idea. One is Monday, two is Tuesday, etc.

Wrapping it up

Hopefully you've found some new things to think about regarding WMI filters and Group Policy. A closing note: not all WMI filters are created equal. Not everything in WMI is as optimized as we'd like it to be, and some WMI queries are not as performant as we'd like. Avoid loose wildcard queries when possible as they will run slower (for example, *Select * from Win32_LocalTime where DayOfWeek = 5* will run slightly slower than the samples provided above). And above all, always test before deploying to production, using the *slowest hardware you can find* so that you get a good idea about baseline performance.

Got a filter question or a good sample to share? Hit the comments section below.

- Ned Pyle

- **Anonymous**

September 11, 2008

Great post, thanks. I wish you wrote it a bit earlier ;) A couple of months ago I was evaluating Windows Server Security Guide. Default GPOs from there turn on UAC for user accounts (even administrators). That's generally a good thing, but not on Server Core. It's simply unable to display the UAC prompt, so the user can never elevate. So I had to find out myself how to filter out the Cores using WMI.

- **Anonymous**

September 11, 2008

The comment has been removed

- **Anonymous**

September 12, 2008

Group Policy MVP Darren Mar-Elia had a contest on his mailing list for a tough WMI problem. The original poster wanted a filter for desktop (no laptops) computers only. A guy named Joel won the contest with this nice filter "Select * from Win32_PhysicalMemory where FormFactor != 12 Returns True on computers that do not have SoDimm form factor memory and False on computers with SoDimm form factor memory. The assumption is that all laptops will have this style memory and desktops will not." Very good filter that does the job for almost all hardware configs.

- **Anonymous**

September 12, 2008

The comment has been removed

- **Anonymous**

October 01, 2008

Auch das Active Directory Team betreibt ein Blog , dass ich interessant zu lesen finde. Gestolpert darüber

- **Anonymous**

October 01, 2008

This is great! If I need to filter just for non-server OS, will the following work? SELECT ProductType FROM Win32_OperatingSystem WHERE ProductType <> "1" For WMI filters in general, do they add more significant load on domain controllers because the filters are processed at GPO refresh interval? Thank you.

- **Anonymous**

October 01, 2008

The comment has been removed

- **Anonymous**

October 07, 2008

EXCELLENT! This is real-world useful stuff! Especially the text to use for certain OS type filters - KUDOS!

- **Anonymous**

October 24, 2008

Hey!! So, I have a WMI filter question for you related to above, I have half of it already, thanks. But, I need to filter out a Windows 2008 64 bit server, that is not a DC from everything else. I have this so far, but somehow my Vista clients still get it, which I really dont want to. Here is my filter. Select OSArchitecture,Version,ProductType from Win32_OperatingSystem where (OSArchitecture="64-bit") AND (Version like "6.0%") AND (ProductType="3") Thanks Daniel

- **Anonymous**

October 24, 2008

WBEMTEST returns true for that query on my 2008 x64 member server, and returns false on my Vista x64 machine. Do you get those same results with WBEMTEST? Just do Start, Run, WBEMTEST. Then click Connect, leave it as rootcimv2, then click Query, and paste in your query as-is, then click Apply. If you get 0 objects returned that is a False, 1 object returned is a True.

- **Anonymous**

November 19, 2008

We've been at this for over a year (since August 2007), with more than 100 posts (127 to be exact), so

- **Anonymous**

March 06, 2009

For various reasons, you may want to restrict certain configuration processes to a subset of your environment

- **Anonymous**

May 15, 2009

Just do Start, Run, WBEMTEST. Then click Connect, leave it as rootcimv2, then click Query, and paste in your query as-is, then click Apply. If you get 0 objects returned that is a False, 1 object returned is a True.

- **Anonymous**

May 28, 2009

I have half of it already, thanks. But, I need to filter out a Windows 2008 64 bit server, that is not a DC from everything else.

- **Anonymous**

May 28, 2009

This works for me (will apply to any 64-bit OS non-DC Win2008 server: SELECT OSArchitecture,ProductType FROM Win32_OperatingSystem WHERE OSArchitecture = "64-bit" and ProductType = "3" Is that what you're looking for?

- **Anonymous**

August 21, 2009

It's always interesting to read your articles. I wish I could write as good as you!

- **Anonymous**

August 21, 2009

The comment has been removed

- **Anonymous**

October 31, 2009

I must add - for various reasons, you may want to restrict certain a-type configuration processes to a subset of your virtual environment. This is true in many ways and it's good to keep it in mind.

- **Anonymous**

November 30, 2009

I need a WMI Query (Filter) for IIS 6 or above, any Idea what class I should use?

- **Anonymous**

November 30, 2009

Hi Yoels, Interesting problem. Here's one way to do it, you can explore the theme here. Make sure you test on Win2003 and later, as I only tested on Win2008. Also, make sure you use the double backslashes like I did below: `SELECT Version FROM CIM_DataFile WHERE Name = "c:\windows\system32\inetsrv\iisres.dll" AND Version >= "6"` This will see if the IISRES.DLL that IIS uses is installed and if it's version is equal to or greater than 6. Let me know if that works for you. I must point out that IIS 5.0 only exists on Win2000 though, and since you cannot use WMI filters on Win2000, the version is rather moot. This would be more useful to see if it was IIS 7 or higher, for example.

- **Anonymous**

January 18, 2010

The comment has been removed

- **Anonymous**

January 18, 2010

The comment has been removed

- **Anonymous**

January 18, 2010

The comment has been removed

- **Anonymous**

May 10, 2010

This works when the systemroot is installed on `SELECT Version FROM CIM_DataFile WHERE Name = "c:\windows\system32\inetsrv\iisres.dll" AND Version >= "6"` But in our environment we also have servers with the `systemroot` installed on D:. We have 2000 IIS with the `systemroot` installed on C: or D:.

- **Anonymous**

July 03, 2010

Check out my blog post that includes Windows 7 and server 2008 R2: derek858.blogspot.com/.../wmi-gpo-filters-for-operating-system.html

- **Anonymous**
September 18, 2014
Fun with WMI Filters in Group Policy - Ask the Directory Services Team - Site Home - TechNet Blogs
- **Anonymous**
September 22, 2014
Fun with WMI Filters in Group Policy - Ask the Directory Services Team - Site Home - TechNet Blogs
- **Anonymous**
October 06, 2014
Fun with WMI Filters in Group Policy - Ask the Directory Services Team - Site Home - TechNet Blogs
- **Anonymous**
October 26, 2014
Fun with WMI Filters in Group Policy - Ask the Directory Services Team - Site Home - TechNet Blogs
- **Anonymous**
October 31, 2014
Fun with WMI Filters in Group Policy - Ask the Directory Services Team - Site Home - TechNet Blogs
- **Anonymous**
September 09, 2015
Hey, is it possible using WMI filtering to restrict the ntuser to login one at a time?

Source: <https://blogs.technet.microsoft.com/askds/2008/09/11/fun-with-wmi-filters-in-group-policy/>