

Managing Windows networks using Script - Part 12: Properties of the WMI class

Back in the third part of this series, we developed a simple script called `displayTimeZone.vbs` to display the current time zone setting on your computer. This article will show you how to display all the properties of the WMI class.

Mitch Tulloch

Back in the third part of this series, we developed a simple script called `displayTimeZone.vbs` to display the current time zone setting on your computer:

```
Option Explicit
On Error Resume Next
Dim strComputer
Dim strWMINamespace
Dim strWMIQuery
Dim objWMIService
Dim colItems
Dim objItem

strComputer = "."
strWMINamespace = "root\CIMV2"
strWMIQuery = "SELECT * FROM Win32_TimeZone"

Set objWMIService = GetObject ("winmgmts:" & strComputer & strWMINamespace)
Set colItems = objWMIService.ExecQuery (strWMIQuery)

For Each objItem In colItems
WScript.Echo objItem.Caption
next
```

When I run this script, it gives the following result:

```
C: scripts>DisplayTimeZone.vbs
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

```
(GMT-06: 00) Central Time (US & Canada)
```

How do I know that the main attribute of the Win32_TimeZone class includes the information I want to display? By reading the instructions about it here, I can know it. In fact, this MSDN page tells us that the Description property basically returns *objItem.Description* and similar results.

What else does this MSDN page tell us about the Win32_TimeZone class? What if I want to find out which month does it affect the computer? If you read through this page, you will get information related to this attribute (DaylightMonth).

DaylightMonth

Data type: uint32

Access type: Read-only

Value Meaning

first

0x1 January

2

0x2 February

3

0x3 March

4

0x4 April

5

0x5 Sewing

6

0x6 June

7

0x7 July

8

0x8 August

9

0x9 September

ten

0xA October

11

0xB November

twelfth

0xC December

To use this information, I simply change the *WScript.Echo objItem.Caption* line to *WScript.Echo objItem.DaylightMonth* and this is what I get when running this script:

```
C: scripts>DisplayTimeZone.vbs
```

```
Microsoft (R) Windows Script Host Version 5.6
```

```
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

3

List the properties of a class

Now we can continue the process by changing the properties in *WScript.Echo objItem*. And so slowly work through displaying each one of the properties of the `Win32_TimeZone` class, but is this an easier way? Can we display the properties of this class in a script without naming them in the script? This is done but before doing so try and list the number of attributes of this class. This is how we do it:

```
Option Explicit
On Error Resume Next
Dim strComputer
Dim strWMINameSpace
Dim strWMIQuery
Dim objWMIService
Dim colItems
Dim objItem

strComputer = "."
strWMINameSpace = "rootCIMV2"
strWMIQuery = ": Win32_TimeZone"

Set objWMIService = GetObject ("winmgmts:" & strComputer & strWMINameSpace & strWMIQuery)
WScript.Echo "Number of properties of" & strWMIQuery & "class is" &
objWMIService.Properties_.count
```

Here is the result of running this new script, the script we will call it *DisplayClassProperties.vbs* :

```
C: scripts>DisplayClassProperties.vbs
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

```
Number of properties of: Win32_TimeZone class is 24
```

Viewing an inventory of properties at `Win32_TimeZone` MSDN will show us that the number 24 is completely correct, which means that the `Win32_TimeZone` class has 24 properties in total.

How does this new scenario work? First, note that instead of connecting to the default namespace ("`rootCIMV2`") on the local computer (".") We connect directly to the `Win32_TimeZone` class on the computer. In other words, the line:

```
Set objWMIService = GetObject ("winmgmts:" & strComputer & strWMINameSpace & strWMIQuery)
```

can be replaced by

```
Set objWMIService = GetObject ("winmgmts: .rootCIMV2: Win32_TimeZone")
```

and it will perform the work in a similar way. Another thing you will notice is one in the last line of the script:

```
objWMIService.Properties_.count
```

We know about attributes (like)

Indeed `SWbemObject.Properties_` is a collection (a type of object) and when the object has properties, you can use these properties. An attribute that a collection has is `.count`, it can return the number of properties of the object, ie `.count` returns the number of elements in the collection. This means that `SWbemObject.Properties_.count` is a `.count` property of the `SWbemObject.Properties_` object, so there are two cycles instead of a regular cycle in the syntax.

Show properties of the class

Now we can list the properties of the `Win32_TimeZone` class, how can we return the names of the properties? By appending the following lines to the end of the `DisplayClassProperties.vbs` script:

```
For Each objItem in objWMIService.Properties_  
Wscript.Echo "Property:" & objItem.name  
next
```

This is what we have when running the script:

```
C: scripts> DisplayClassProperties.vbs  
Microsoft (R) Windows Script Host Version 5.6  
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

```
Number of properties of: Win32_TimeZone class is 24  
Property: Bias  
Property: Caption  
Property: DaylightBias  
Property: DaylightDay  
Property: DaylightDayOfWeek  
Property: DaylightHour  
Property: DaylightMillisecond  
Property: DaylightMinute  
Property: DaylightMonth  
Property: DaylightName  
Property: DaylightSecond  
Property: DaylightYear  
Property: Description  
Property: SettingID  
Property: StandardBias  
Property: StandardDay  
Property: StandardDayOfWeek  
Property: StandardHour  
Property: StandardMillisecond  
Property: StandardMinute  
Property: StandardMonth  
Property: StandardName  
Property: StandardSecond  
Property: StandardYear
```

Checking out the previous MSDN page, the page described about the Win32_TimeZone class, we can see above that the name of the class properties.

Conclude

The power of this method is that it allows us to list the properties of the WMI class, so we can learn more about them (remember that in Part 3 of this series we saw how to list the WMI layer of a namespace and together with that list we can discover and see what we can manage through using WMI).

For example, if we want to list the properties of the Win32_BootConfiguration WMI class using the *DisplayClassProperties.vbs* script, all we need to do is change the line:

```
strWMIQuery = ": Win32_TimeZone"
```

into the following line:

```
strWMIQuery = ": Win32_BootConfiguration"
```

When we make this change and re-run the script, we get the result:

```
C: scripts> DisplayClassProperties.vbs  
Microsoft (R) Windows Script Host Version 5.6  
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

```
Number of properties of: Win32_BootConfiguration class is 9  
Property: BootDirectory  
Property: Caption  
Property: ConfigurationPath  
Property: Description  
Property: LastDrive  
Property: Name Value:  
Property: ScratchDirectory  
Property: SettingID  
Property: TempDirectory
```

We recommend that you use this article section along with the third part to exploit the WMI class and their other attributes, plus we can see more details about the power of using WMI to manage machines. Calculate Windows in later articles of this series, invite you to read.

You finished reading the article "**Managing Windows networks using Script - Part 12: Properties of the WMI class**" 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.