FREE eBook

LEARNING

wmi

Free unaffiliated eBook created from
Stack Overflow contributors.

#wmi

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: wmi

It is an unofficial and free wmi ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official wmi.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with wmi

Remarks

This section provides an overview of what wmi is, and why a developer might want to use it.

It should also mention any large subjects within wmi, and link out to the related topics. Since the Documentation for wmi is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

The WMI Framework

Windows Management Instrumentation (WMI) is included in all modern versions of Windows and Windows Server. Windows Management Instrumentation (WMI) CORE 1.5 was once available from Microsoft to install WMI on computers running Windows 95/98 or Windows NT 4.0.

Even though all Windows computers have WMI, not all WMI Providers, Classes, or even Methods and Properties are included in every version of Windows. For example, the Win32_WinSAT class is not available in versions of Windows greater than 8.1 and the Win32_Service class is not supported on versions of Windows prior to Vista.

WMI Development

Any language that can handle ActiveX objects (.NET, VBScript, PowerShell, Python, JScript, etc) can interact with WMI.

Read Getting started with wmi online: https://riptutorial.com/wmi/topic/7610/getting-started-with-wmi

Chapter 2: Using WMI in VBScript

Examples

Getting a WMI Object and listing some of its properties

This example will list the preferred resolution for all the connected monitors.

The Code:

```
On Error Resume Next
strComputer = "."
strQuery = "SELECT PreferredMonitorSourceModeIndex, MonitorSourceModes " & _
           "FROM WmiMonitorListedSupportedSourceModes"

Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\ROOT\WMI")
Set colItems = objWMIService.ExecQuery(strQuery, , 48)

For Each objItem In colItems
  i = objItem.PreferredMonitorSourceModeIndex
  wscript.stdout.writeline "InstanceName: " & objItem.instancename
  wscript.stdout.writeline "Horizontal: " &
objItem.MonitorSourceModes(i).HorizontalActivePixels
  wscript.stdout.writeline "Vertical: " & objItem.MonitorSourceModes(i).VerticalActivePixels
Next
```

We first get the WMI Service. It is not creatable.

```
Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\ROOT\WMI")
```

Next, set up our query using WQL. WQL is very similar to SQL.

```
strQuery = "SELECT PreferredMonitorSourceModeIndex, MonitorSourceModes " & _
           "FROM WmiMonitorListedSupportedSourceModes"
```

The WMI class WmiMonitorListedSupportedSourceModes has 5 properties: InstanceName, Active, MonitorSourceModes, NumOfMonitorSourceModes, and PreferredMonitorSourceModeIndex. MonitorSourceModes is an array, and we must query PreferredMonitorSourceModeIndex to determine which element of the array contains the information we seek.

Now let's execute our query

```
Set colItems = objWMIService.ExecQuery(strQuery, , 48)
```

and loop through the results:

```
For Each objItem In colItems
  i = objItem.PreferredMonitorSourceModeIndex
```

```
  wscript.stdout.writeline "InstanceName: " & objItem.instancename
  wscript.stdout.writeline "Horizontal: " &
objItem.MonitorSourceModes(i).HorizontalActivePixels
  wscript.stdout.writeline "Vertical: " & objItem.MonitorSourceModes(i).VerticalActivePixels
Next
```

Executing a WMI Method

Some WMI Classes expose Methods that allow you to *do* something with that object. For example, the Win32_Printer class has 11 methods for interacting with a printer, one of which is the PrintTestPage method. The following code demonstrates how to select a specific printer and print a test page.

```
'Specify the name of the target computer
strComputer = "."
'Note: Backslash is a special character that must be escaped with a backslash
'This means the UNC \\Network\Path\PrinterName must be written like the following
strQuery = "SELECT * FROM Win32_Printer WHERE DeviceID='\\\\Network\\Path\\PrinterName'"

Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\ROOT\cimv2")
Set colItems = objWMIService.ExecQuery(strQuery)

'ExecQuery returns a collection object, even when there's only 1 item in the collection
For Each objItem In colItems
    'The PrintTestPage method takes no parameters and returns a UINT32
    intTestPageReturnCode = objItem.PrintTestPage
Next
'PrintTestPage will return 0 (Successs) or 5 (Failure)
Select Case intTestPageReturnCode
    Case 0
        WScript.StdOut.WriteLine "Test page successfully printed"
    Case 5
        WScript.StdOut.WriteLine "Test page printing failed"
    Case Else
        WScript.StdOut.WriteLine "An unknown error occurred while printing a test page"
End Select
```

Read Using WMI in VBScript online: https://riptutorial.com/wmi/topic/7611/using-wmi-in-vbscript

Credits

S. No	Chapters	Contributors
1	Getting started with wmi	Community, Tim
2	Using WMI in VBScript	Tim