

# **Получение данных об программном и аппаратном обеспечении**

# WMI

WMI (Windows Management Instrumentation) в C# позволяет получать информацию о системе и оборудовании, используя пространство имен `System.Management`

- Данные об оборудовании
- Управление процессами и службами
- Отслеживать системные события в реальном времени

# Архитектура и язык запросов

Для извлечения данных используется WQL (WMI Query Language) — похож на язык запросов SQL.

**Синтаксис:**

```
SELECT [свойства] FROM [WMI_Класс] WHERE [Условие]
```

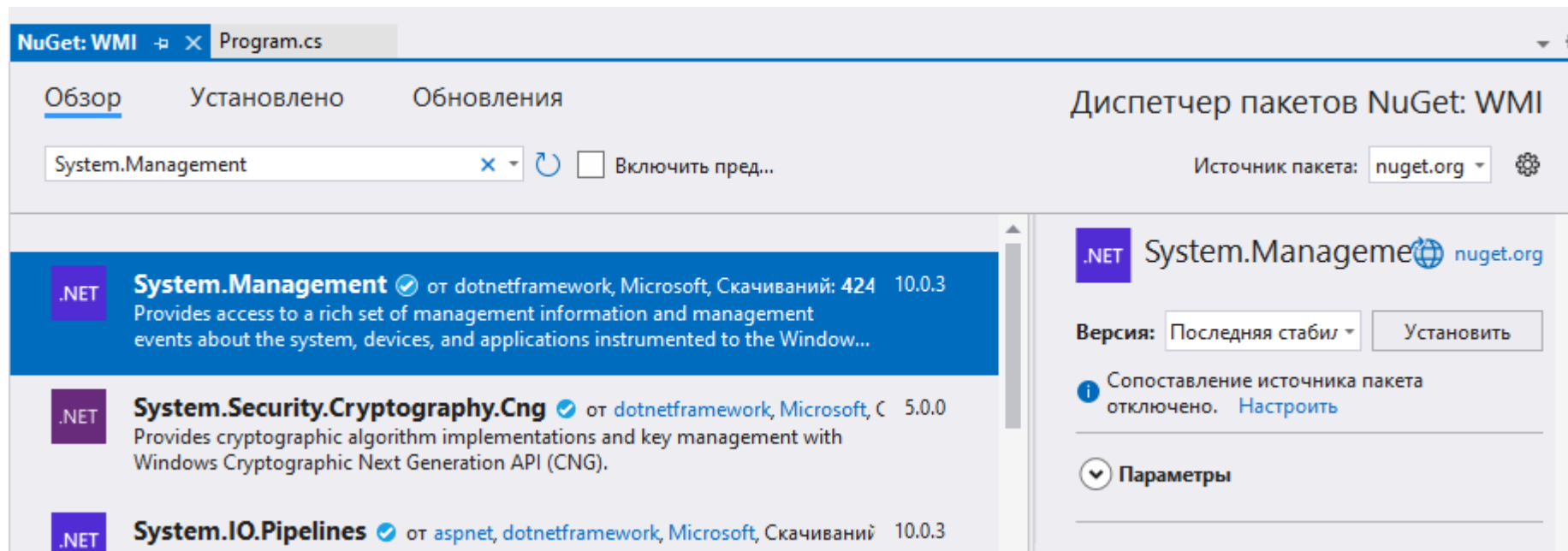
**Пример:**

```
SELECT Name, FreeSpace FROM Win32_LogicalDisk  
WHERE DriveType = 3
```

# WMI Классы

Класс	Описание
Win32_Processor	Информация о процессоре
Win32_OperatingSystem	Данные об ОС
Win32_LogicalDisk	Сведения о логических дисках и разделах
Win32_Process	Список запущенных процессов
Win32_Service	Информация об установленных системных службах

# Установка



# Пример

```
using System.Management;

void GetProcessorInfo(){
    // WQL-запрос для получения процессоров

    ManagementObjectSearcher searcher = new ManagementObjectSearcher("SELECT * FROM
Win32_Processor");

    // Перебираем найденные объекты (экземпляры класса Win32_Processor)
    foreach (ManagementObject obj in searcher.Get()){
        Console.WriteLine("Процессор: " + obj["Name"]);
        Console.WriteLine("Количество ядер: " + obj["NumberOfCores"]);
    }
}
```

GetProcessorInfo();  
Консоль отладки Microsoft Visual Studio

Процессор: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx  
Количество ядер: 4



# Пример

```
void GetDiskSpace(){  
    var searcher = new ManagementObjectSearcher("SELECT Name, Size, FreeSpace FROM  
Win32_LogicalDisk WHERE DriveType = 3");  
    foreach (ManagementObject disk in searcher.Get()){  
        Console.WriteLine($"Диск: {disk["Name"]}");  
        Console.WriteLine($"Всего: {disk["Size"]} байт");  
        Console.WriteLine($"Свободно: {disk["FreeSpace"]} байт");  
        Console.WriteLine("-----");  
    }  
}
```

Диск: C:

Всего: 480085274624 байт

Свободно: 141662171136 байт

-----

Диск: D:

Всего: 998864056320 байт

Свободно: 763075350528 байт

-----

# Управление устройствами

1. Управление питанием и состоянием — Класс Win32\_PnPEntity через методы Disable и Enable можно программно «выткнуть» и «воткнуть» устройство
2. Управление сетевыми адаптерами — Класс Win32\_NetworkAdapterConfiguration, через EnableStatic установить статический IP удаленно, SetGateways прописать шлюз

# Управление устройствами

3. Управление дисками и разделами — позволяет работать с дисками и томами на низком уровне. Класс `Win32_Volume`. Действия: `Format` — отформатировать диск. `Defragment` — запустить дефрагментацию. `Chkdsk` — начать проверку диска на ошибки.

# Управление устройствами

4. Мониторы и яркость. Для управления яркостью экрана ноутбука используется специальное пространство имен `root\WMI`. Метод: `WmiSetBrightness`, например, можно написать программу которая плавно снижает яркость экрана вечером, если в системе нет датчика освещенности.

# ВАЖНО

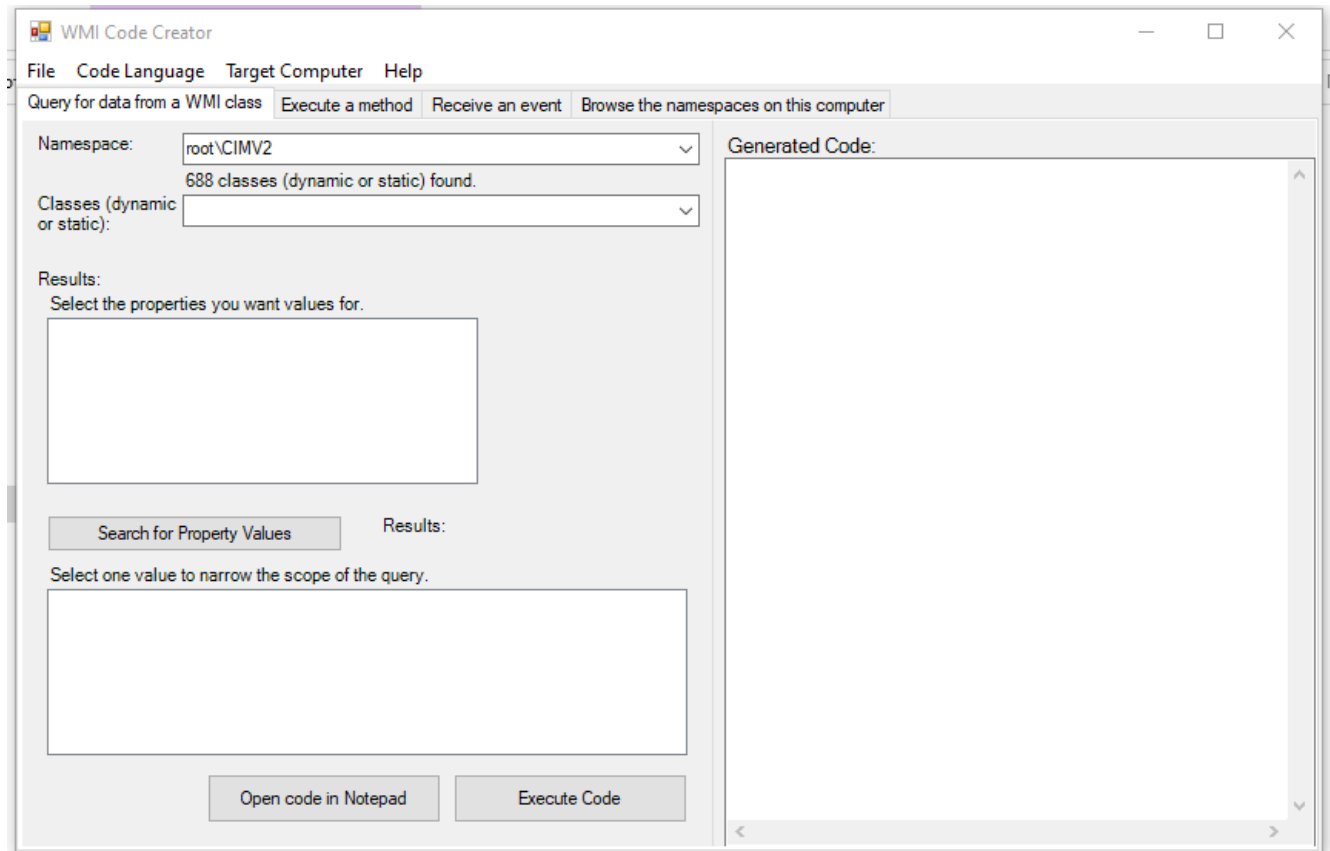
- Права доступа
- Read-Only свойства
- Альтернатива CIM
- Разрешения в Брандмауэре
- Права администратора
- включенная RPC (для удаленного управления)

# Пример: Изменение яркости ноутбука

[Ссылка](#)

# WMI Code Creator

Ссылка





# Как пользоваться утилитой

**Query for data:** Генерация кода для получения информации (например, модель процессора, свободное место на диске).

**Execute a method:** Создание кода для действий (запуск процесса, перезагрузка, изменение яркости).

**Receive an event:** Генерация кода для отслеживания событий (например, когда вставляется флешка или запускается программа).

# Как пользоваться утилитой

**Namespace:** Обычно используется `root\cimv2` для общих данных или `root\WMI` для специфических настроек железа.

**Classes:** Выберите интересующий класс. Например, `Win32_Process` для процессов или `Win32_LogicalDisk` для дисков

**Properties/Methods:** Отметьте галочками нужные свойства

Namespace: root\cimv2

688 classes (dynamic or static) found.

Classes (dynamic or static): Win32\_Keyboard

[Get documentation for this class from the online MSDN Library.](#)

Results: 23 properties found.

Select the properties you want values for.

Availability  
Caption  
ConfigManagerErrorCode  
ConfigManagerUserConfig  
CreationClassName  
Description  
DeviceID  
ErrorCleared

Search for Property Values

Results:

Select one value to narrow the scope of the query.

Open code in Notepad

Execute Code

Generated Code:

```
using System;
using System.Management;
using System.Windows.Forms;

namespace WMISample
{
    public class MyWMIQuery
    {
        public static void Main()
        {
            try
            {
                ManagementObjectSearcher searcher =
                    new ManagementObjectSearcher("root\\cimv2",
                        "SELECT * FROM Win32_Keyboard");

                foreach (ManagementObject queryObj in searcher.Get())
                {
                    Console.WriteLine("-----");
                    Console.WriteLine("Win32_Keyboard instance");
                    Console.WriteLine("-----");
                    Console.WriteLine("Caption: {0}", queryObj["Caption"]);
                }
            }
            catch (ManagementException e)
            {
                MessageBox.Show("An error occurred while querying for WMI data");
            }
        }
    }
}
```

Namespace:

root

Classes (dynamic  
or static):

Win32\_Keyboard

[Get documentation for this class from the online MSDN Library.](#)

Results: 23 properties

Select the properties

Availability

Caption

ConfigManagerErrorCo

ConfigManagerUserCo

CreationClassName

Description

DeviceID

ErrorCleared

Search for Prop

Select one value to na

Win32\_Keyboard instance

Caption: Расширенная клавиатура (101 или 102 клавиши)

Win32\_Keyboard instance

Caption: Расширенная клавиатура (101 или 102 клавиши)

Win32\_Keyboard instance

Caption: Расширенная клавиатура (101 или 102 клавиши)

Win32\_Keyboard instance

Caption: Расширенная клавиатура (101 или 102 клавиши)

C:\Windows\Microsoft.NET\Framework\v2.0.50727&gt;

Администратор: C:\Windows\system32\cmd.exe

Версия компилятора Microsoft (R) Visual C# 20058.00.50727.9149

для Microsoft (R) Windows (R) 2005 Framework версии 2.0.50727

Авторские права (C) Microsoft Corporation 2001-2005. Все права защищены.

using System.Management;  
using System.Windows.Forms;

namespace WMISample

public class MyWMIQuery

public static void Main()

ManagementObjectSearcher searcher =  
new ManagementObjectSearcher("root\\CIMV2",  
"SELECT \* FROM Win32\_Keyboard");

foreach (ManagementObject queryObj in searcher.Get())

Console.WriteLine("-----");  
Console.WriteLine("Win32\_Keyboard instance");  
Console.WriteLine("-----");  
Console.WriteLine("Caption: {0}", queryObj["Caption"]);

MessageBox.Show("An error occurred while querying for WMI data");

Namespace:

root\cimv2

688 classes (dynamic or static) found.

Classes (dynamic or static):

Win32\_Keyboard

[Get documentation for this class from the online MSDN Library](#)

Results: 23 properties found.

Select the properties you want values for.

Availability

Caption

ConfigManagerErrorCode

ConfigManagerUserConfig

CreationClassName

Description

DeviceID

ErrorCleared

Search for Property Values

Results:

Select one value to narrow the scope of the query.

Open code in Notepad

Execute Code

Администратор: C:\Windows\system32\cmd.exe

Сервер сценариев Windows (Microsoft ®) версия 5.812  
Copyright (C) Корпорация Майкрософт 1996-2006, все права

-----  
Win32\_Keyboard instance  
-----

Caption: Расширенная клавиатура (101 или 102 клавиши)

ConfigManagerErrorCode: 0

ConfigManagerUserConfig: Ложь

CreationClassName: Win32\_Keyboard

Description: Клавиатура HID  
-----Win32\_Keyboard instance  
-----

Caption: Расширенная клавиатура (101 или 102 клавиши)

ConfigManagerErrorCode: 0

ConfigManagerUserConfig: Ложь

CreationClassName: Win32\_Keyboard

Description: Стандартная клавиатура PS/2

C:\Windows\system32&gt;

Namespace: root\WMI

53 classes with methods found.

Classes with methods: WmiMonitorBrightnessMethods

4 methods found.

Methods: WmiSetBrightness

Method [in] parameters:

Select the method [in] parameters you want to assign a value to (some may be optional).

Brightness = 70  
Timeout = 1

Select the instance you want to execute the query on. The values in the list are the values of the key property for this class.

InstanceName='DISPLAY\BOE07B4\5&amp;5b45f8b&amp;1&amp;UID256'

Open code in Notepad

Execute Code

## Generated Code:

```
using System;
using System.Management;
using System.Windows.Forms;

namespace WMISample
{
    public class CallWMIMethod
    {
        public static void Main()
        {
            try
            {
                ManagementObject classInstance =
                    new ManagementObject("root\\WMI",
                        "WmiMonitorBrightnessMethods.InstanceName='DISPLAY\\BOE" +
                        null);

                // Obtain in-parameters for the method
                ManagementBaseObject inParams =
                    classInstance.GetMethodParameters("WmiSetBrightness");

                // Add the input parameters.
                inParams["Brightness"] = 70;
                inParams["Timeout"] = 1;

                // Execute the method and obtain the return values.
                ManagementBaseObject outParams =
                    classInstance.InvokeMethod("WmiSetBrightness", inParams, null);

                // No outParams
            }
            catch (ManagementException ex)
```

Query for data from a WMI class Execute a method Receive an event Browse the namespaces on this computer

Namespace: root\WMI

53 classes with methods found.

Classes with methods: WmiMonitorBrightnessMethods

4 methods found.

Methods: WmiSetBrightness

Select the method [in] parameters you want to assign a value to (some may be optional).

Method [in] parameters:

Brightness = 100  
Timeout = 1

Select the instance you want to execute the query on. The values in the list are the values of the key property for this class.

InstanceName="DISPLAY\BOE07B4\5&amp;5b45f8b&amp;1&amp;UID256"

&lt; &gt;

Open code in Notepad

Execute Code

Generated Code:

```
strComputer = "."  
Set objWMIService = GetObject("winmgmts:\\." & strComputer & "\root\WMI")  
' Obtain an instance of the the class  
' using a key property value.  
Set objShare = objWMIService.Get("WmiMonitorBrightnessMethods.Instance_0")
```

rs object specific

```
are.Methods_("WmiSetBrightness"). _  
Instance_0)
```

```
ers.  
.Item("Brightness") = 70  
.Item("Timeout") = 1
```

```
nd obtain the return status.  
ject in objOutParams
```

```
to created by the provider.  
Set objOutParams = objWMIService.ExecMethod("WmiMonitorBrightnessMe
```

' No outParams

Enter in-parameter

Assign a value to the Brightness parameter.  
The parameter is of type: UInt8.

70

OK

Cancel

# Зачем это нужно

- Автоматизация инвентаризации (сбор данных о железе и ПО с тысяч компьютеров по сети)
- Удаленное управление
- Лицензирование и защита
- Установщики
- Мониторинг вторжений
- Оптимизация



# Пример

КОД

# Другие ОС

WMI — это эксклюзивная технология Microsoft. В других ОС её нет, т. к. она глубоко завязана на реестр и архитектуру ядра Windows.

# Другие ОС

## Аналоги для других систем:

[illegible]

## 2. В Линуксе все является файлом, поэтому можно получать/менять параметры прямо через файлы

# Другие ОС

Пример с яркостью:

```
Echo 500 >
```

```
/sys/class/backlight/intel_backlight/brightness
```

Пример с железом:

```
/proc/cpuinfo
```

# Другие ОС

## 3. macOS: I/O Kit и system\_profiler

Для взаимодействия с железом используется фреймворк I/O Kit

Для получения данных из терминала используется команда `system_profiler` (аналог SELECT запросов в WMI)

# Другие ОС

4. Кроссплатформенное решение SNMP

5. Node Exporters и Promertheus

# Задание

- Достать модель процессора, объем оперативной памяти, серийный номер материнской платы, список разделов диска с указанием свободного места в ГБ.