

РАЗРАБОТКА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ ДЛЯ ПОЛУЧЕНИЯ ИНФОРМАЦИИ О ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВА WMI

И.А. Бунеев, Д.Р. Брославский, Н.Ю. Юдина

ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Аннотация. В статье описывается программный продукт, который был создан и используется для сбора сведений о компьютерах в корпоративной сети ВГЛТУ. В качестве инструментария для управления данными и операциями на базе ОС Windows нами был выбран инструментарий управления Windows (WMI), который предоставляет возможность создавать скрипты и приложения. Разработанные программные модули могут быть использованы для решения задач администратора сети.

Ключевые слова: инструментарий WMI, класс, данные, устройства, приложение, NET Framework, Newtonsoft.Json, сериализация, десериализация.

DEVELOPMENT OF A CLIENT APPLICATION FOR OBTAINING INFORMATION ABOUT A PERSONAL COMPUTER USING THE WMI TOOL

I.A. Buneev, D.R. Broslavsky, N.Yu. Yudina

Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. The article describes a software product that was created and is used to collect information about computers in the VSTU corporate network. As a tool for managing data and operations based on Windows OS, we selected the Windows Management Instrumentation (WMI), which provides the ability to create scripts and applications. The developed software modules can be used to solve network administrator tasks. WMI provides a powerful infrastructure for managing data and operations in Windows operating systems.

Keywords: WMI instrumentation, class, data, devices, application, NET Framework, Newtonsoft.Json, serialization, deserialization.

Для определения производительности сети и выявления недостатков ее инфраструктуры необходимо периодически проводить мониторинг состояния сети. Сбор данных о компьютерах в сети проводится в трех направлениях – мониторинг сети, биллинг трафика, обнаружение вторжений и выявление сетевых атак.

В качестве инструментария для управления данными и операциями на базе ОС Windows нами был выбран инструментарий управления Windows (WMI), который предоставляет возможность создавать скрипты и приложения. Разработанные программные модули могут быть использованы для решения задач администратора сети. Инструментарий WMI предоставляет мощную инфраструктуру для управления данными и операциями в операционных системах Windows. Также имеется возможность для создания скриптов WMI и приложений для автоматизации задач на удаленных компьютерах.

Нами было разработано приложение, которое обеспечивает получение информации о компьютере в корпоративной сети ВГЛТУ с использованием средств WMI.

Для разработки приложения будет использоваться язык программирования C# и интегрированная среда разработки Visual Studio 2022.

Для начала разработки необходимо создать консольный проект на платформе .NET Framework.

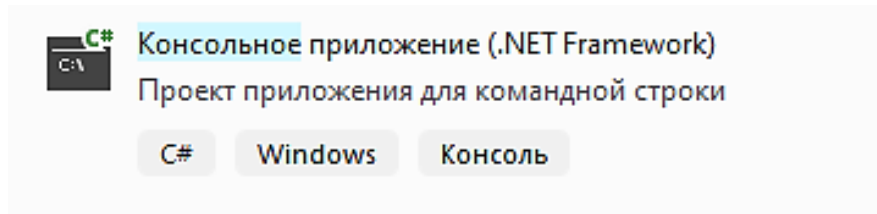


Рисунок 1 – Консольное приложение на платформе .NET Framework

Далее происходит подключение следующих библиотек Newtonsoft.Json, System.Management. Newtonsoft.Json – это библиотека, которая используется для сериализации и десериализации файла JSON. Библиотека System.Management предоставляет доступ к управляющей информации и событиям управления системой, устройствам и приложениям, подключёнными к инфраструктуре инструментария управления Windows.

Класс CompInfo описывает данные, которые будут собираться в ходе работы приложения.

```
public class CompInfo
{
    //Пользователь под которым была запущена программа
    Свойство 1
    public string CurrentUser { get; set; }

    //Название ПК
    Свойство 1
    public string Name { get; set; }

    //ОС
    Свойство 1
    public string OSVersion { get; set; }

    //Список IP адресов
    Свойство 1
    public List<string> IpAddresses { get; set; } = new List<string>();

    //Список процессоров
    Свойство 1
    public List<string> Processors { get; set; } = new List<string>();

    //Материнская плата
    Свойство 1
    public string BaseBoard { get; set; }

    //Напряжение Q3V
    Свойство 1
    public ulong Memory { get; set; }

    //Видеокарта
    Свойство 1
    public List<string> VideoControllers { get; set; } = new List<string>();

    //Вестные диски и твердотельные накопители (Название : Объем)
    Свойство 1
    public Dictionary<string, string> Disks { get; set; } = new Dictionary<string, string>();

    //Установленный софт
    Свойство 2
    public List<Soft> InstalledSoft { get; set; } = new List<Soft>();

    Свойство 2
    public class Soft
    {
        Свойство 2
        public string Name { get; set; }
        Свойство 2
        public string Vendor { get; set; }
        Свойство 2
        public string Date { get; set; }
    }
}
```

Рисунок 2 – Класс CompInfo

Свойства класса CompInfo приведены ниже.

Свойство `CurrentUser` определяет имя пользователя, который запустил программу. Свойство `Name` – это имя персонального компьютера в домене. Версия операционной системы описывается в свойстве `OSVersion`. Список IP-адресов хранится в свойстве `IpAddresses`, а список процессоров в свойстве `Processors`. Сведения о материнской плате записаны в свойстве `BaseBoard`. Объем оперативной памяти в гигабайтах записан в свойстве `Memory`. Список видеоадаптеров – свойство `VideoControllers`. Свойство `Disks` – это словарь, где ключ – название жесткого диска или твердотельного накопителя, а значение – объем в гигабайтах. В свойстве `InstlledSoft` содержится список, содержащий информацию об установленных программах на персональном компьютере (название, издатель, дата установки).

Для получения информации об имени пользователя обращаемся к окружению и соответствующему свойству `UserName`. Для получения названия ПК в домене необходимо обратиться к классу `Dns` и вызвать метод `GetHostName()`.

```
CompInfo compInfo = new CompInfo();  
//Пользователь под которым была запущена программа  
compInfo.CurrentUser = System.Environment.UserName;  
//Название ПК  
compInfo.Name = Dns.GetHostName();
```

Рисунок 3 – Получения информации о имени пользователя и названии ПК

Далее для получения информации о комплектующих персонального компьютера создаем объект класса `ManagementObjectSearcher`. В конструктор необходимо передать область «`root\CIMV2`» и SQL запрос. В качестве примера приведем получение названий процессоров, установленных в ПК.

Список литературы

1. <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
2. С# 10 и .NET 6. Современная кроссплатформенная разработка. – 2023. – Марк Прайс.
3. Ник, Р. Visual Studio 2010 для профессионалов / Р. Ник. – М.: Диалектика / Вильямс, 2012. – 703 с.
4. Албахари Дж. С# 6.0. Справочник. Полное описание языка / Дж. Албахари, Б. Албахари. – 6-е изд. – Москва: Вильямс, 2016. – 1040 с.

References

1. <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
2. C# 10 and .NET 6. Modern Cross-Platform Development. – 2023. – Mark Price.
3. Nick, Randolph Visual Studio 2010 for Professionals / Randolph Nick. – M.: Dialectics / Williams, 2012. - 703 p.
4. Albahari J. C# 6.0. Handbook. Complete Description of the Language / J. Albahari, B. Albahari. – 6th ed. – Moscow: Williams, 2016. – 1040 p.