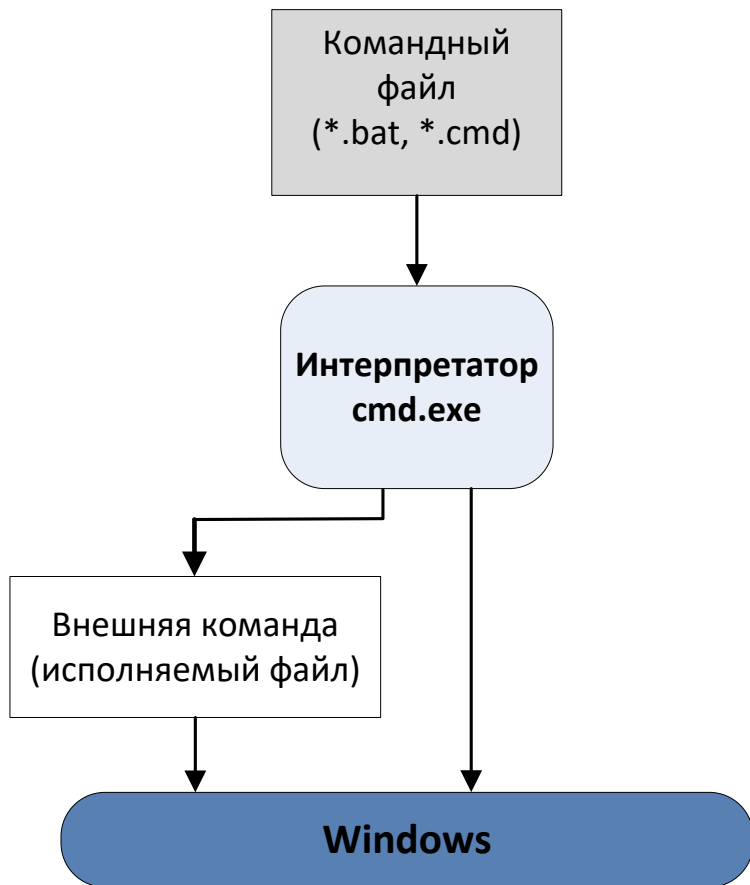


Лекция 13.

Сценарии Windows Script Host



Достоинства cmd.exe:

- Простота и надежность
- Два режима: интерактивный и пакетный

Недостатки:

- Непривычный синтаксис языка
- Нет массивов, структур, логических операций в условиях, удобных процедур и функций
- Отсутствуют мощные и удобные инструменты для обработки текста вывод команд
- Несогласованный синтаксис внешних команд
- Нельзя работать с объектами операционной системы

Три варианта улучшения сценариев Windows

- Увеличить количество утилит командной строки и существенно расширить язык интерпретатора `cmd.exe`, превратив его в полноценный алгоритмический язык с поддержкой ООП (+ совместимость с прежним синтаксисом языка командных файлов).

Три варианта улучшения сценариев Windows

- Увеличить количество утилит командной строки и существенно расширить язык интерпретатора `cmd.exe`, превратив его в полноценный алгоритмический язык с поддержкой ООП (+ совместимость с прежним синтаксисом языка командных файлов).
- Создать совсем новый язык.

Три варианта улучшения сценариев Windows

- Увеличить количество утилит командной строки и существенно расширить язык интерпретатора `cmd.exe`, превратив его в полноценный алгоритмический язык с поддержкой ООП (+ совместимость с прежним синтаксисом языка командных файлов).
- Создать совсем новый язык.
- Выбрать для сценариев Windows другой простой язык программирования (из существующих).

Язык для сценариев Windows

Недостатки cmd.exe:

- Непривычный синтаксис
- Нет массивов, структур, логических операций в условиях, удобных процедур и функций
- Для взаимодействия программ приходится анализировать текстовый вывод команд
- Несогласованный синтаксис внешних команд
- Нет графического интерфейса
- Нельзя работать с COM-объектами в операционной системе

Нужен язык для сценариев:

- Простой и привычный для пользователей Windows
- С поддержкой стандартных алгоритмических возможностей
- Позволяющий пользоваться внешними COM-объектами, имеющимися в Windows



Язык для сценариев Windows

Недостатки cmd.exe:

- Непривычный синтаксис
- Нет массивов, структур, логических операций в условиях, удобных процедур и функций
- Для взаимодействия программ приходится анализировать текстовый вывод команд
- Несогласованный синтаксис внешних команд
- Нет графического интерфейса
- Нельзя работать с COM-объектами в операционной системе

Нужен язык для сценариев:

- Простой и привычный для пользователей Windows
- С поддержкой стандартных алгоритмических возможностей
- Позволяющий пользоваться внешними COM-объектами, имеющимися в Windows





Интерпретируемый язык с доступом
к внешним COM-объектам



BASIC для скриптов*!

*VBScript = Visual Basic Script Edition

VBScript - преимущества

1. Развитость и распространенность. BASIC – стандартный язык программирования для персональных компьютеров в 1970-1990 годы

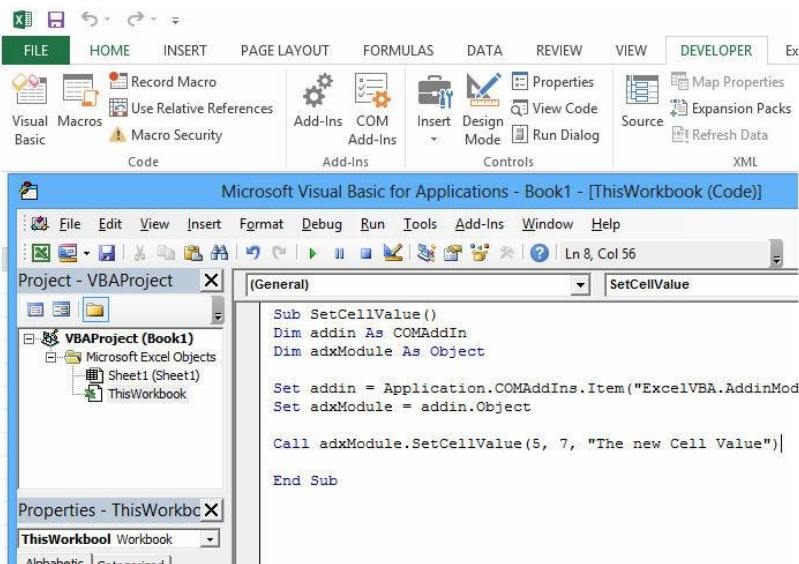
- **B**egginners **A**llpurpose **S**ymbolic **I**nstruction **C**ode (1964, Дартмутский колледж)
- Altair BASIC (1975, Microsoft)
- До конца 1980-х BASIC был на всех персоналках
- Microsoft QBasic поставлялся с Windows 9x
- Microsoft Visual Basic – стандартный ЯП для Windows



VBScript - преимущества

2. Универсальность. Один язык для сценариев в разных областях:

- Автоматизация настольных приложений (VBA)
- Динамические HTML-страницы (клиентская часть)
- Серверные ASP-страницы



```
Untitled - Notepad
File Edit Format View Help
<html>
<head>
<title>Simple VBScript Example</title>
<script language="vbscript">
  Sub ButtonClicked
    window.alert("You clicked on the button!")
  End Sub
</script>
</head>
<body>
  <button name="Button1" type="BUTTON" onclick="ButtonClicked">
    Click Me If You Can!!!
  </button>
</body>
</html>
```

VBScript - преимущества

3. Расширяемость за счет внешних объектов.

Функция `CreateObject("ProgID")`

"ProgID" = "Библиотека_Типов.Имя_класса"

1. Интерпретатор сценария через ProgID получает из реестра путь к файлам нужной библиотеки типов.
2. С помощью этой библиотеки в память загружается экземпляр запрашиваемого объекта.
3. Ссылка на созданный объект сохраняется в переменной.
4. Через эту переменную мы получаем доступ к свойствам и методам объекта.

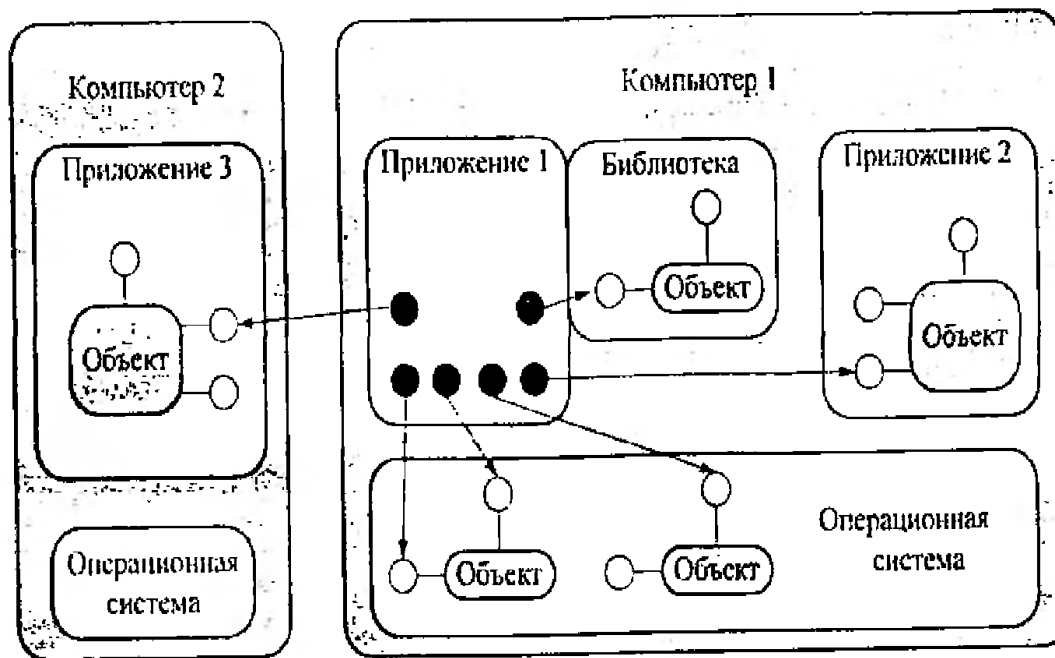
```
Set WA = CreateObject("Word.Application")
```

```
Set Shell = CreateObject("WScript.Shell")
```

```
Set FSO = CreateObject("Scripting.FileSystemObject")
```

VBScript - преимущества

3. Расширяемость за счет внешних объектов.



Подключение COM-объектов в Приложении 1

VBScript - преимущества

4. Простой графический интерфейс через html

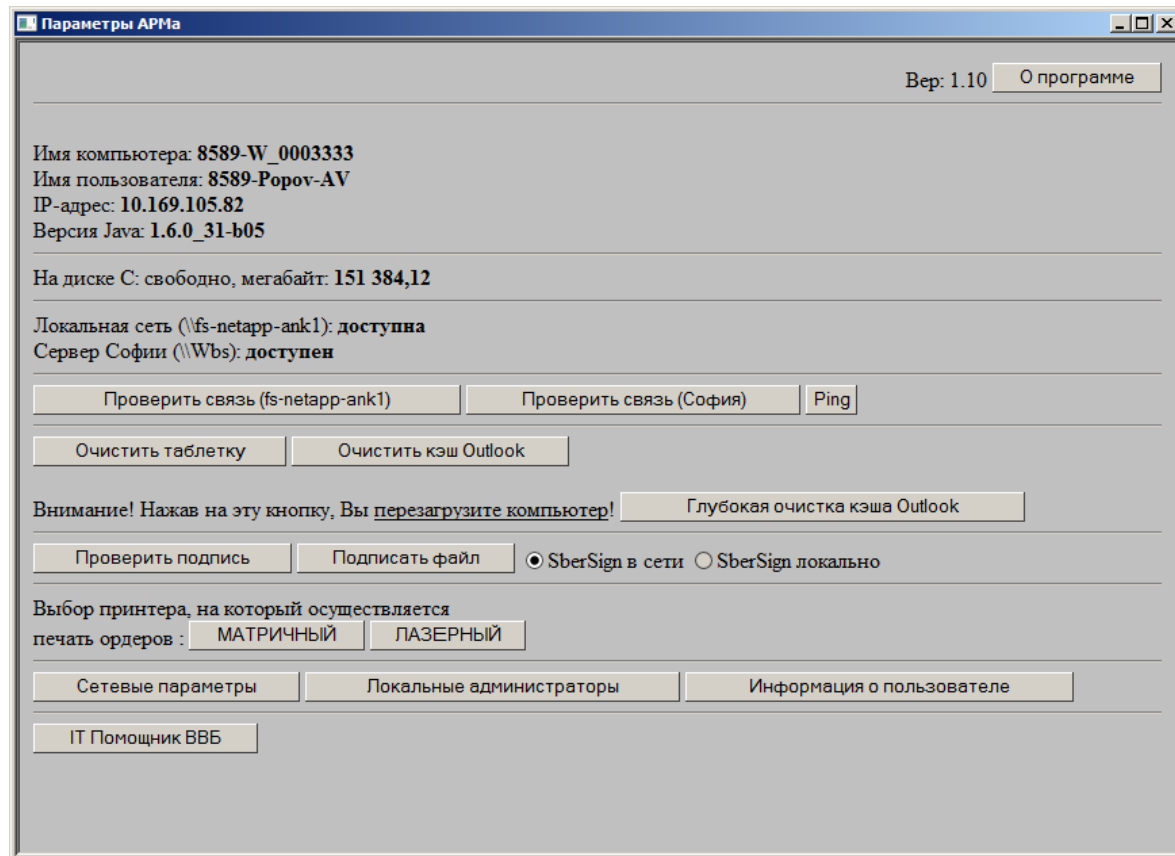
```
<html>

<head>
<title>HTA Test</title>
<HTA:APPLICATION
  APPLICATIONNAME="HTA Test"
  SCROLL="yes"
  SINGLEINSTANCE="yes"
>
</head>

<script language="VBScript">
  Sub TestSub
    MsgBox "Hello, world!!!"
  End Sub
</script>

<body>
  <input type="button" value="Run Script"
    name="run_button" onClick="TestSub">
</body>

</html>
```



JavaScript – еще один язык для сценариев Windows

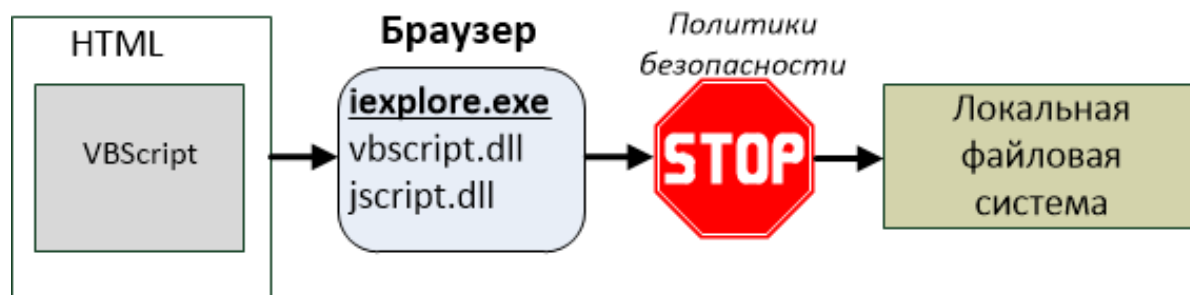
JavaScript – реализация стандарта ECMAScript, аналогичен языку JavaScript.

JavaScript (сначала LiveScript) создан в 1995 году Бренданом Айком (компания Netscape) для встраивания в HTML-страницы и выполнения на клиенте (интерпретатор – в браузере).

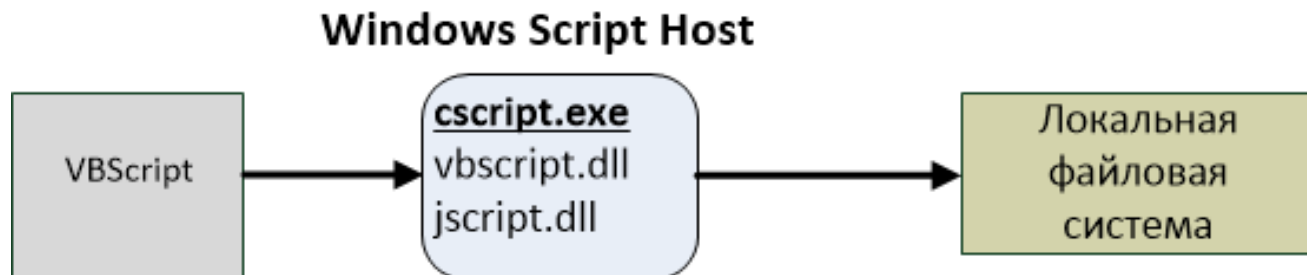
```
1 <html>
2 <head>
3   <script type="text/javascript" src="https://www.google.com/jsapi"></script>
4   <script type="text/javascript">
5     window.onload = function(){
6       google.load("visualization", "1", {packages:["corechart"]});
7       google.setOnLoadCallback(drawChart);
8       function drawChart() {
9         var data = new google.visualization.DataTable();
10        var categories = ['Western Restaurant', 'Khmer Restaurant', 'Hotels', 'Resort'];
11        var values = [100, 49, 120, 30];
12
13        for (var i = 0, len = categories.length; i < len; i++){
14          data.addColumn('number', categories[i]);
15        }
16
17        data.addRow(1);
18
19        for (var j = 0, jlen = values.length; j < jlen; j++){
20          data.setValue(0, j, values[j]);
21        }
22
23        var chart = new google.visualization.ColumnChart(document.getElementById('chart_div'));
24        chart.draw(data, {width: 600, height: 180, chartArea: {left:50,top:30,width:"50%",height:"75%"},
25        backgroundColor:{strokeWidth:0}});
26      }
27    }
28  </script>
29 </head>
30 <body>
31   <div id="chart_div"></div>
32 </body>
33 </html>
```

VBScript/JScript - недостатки

Проблема. Если запускать скрипт внутри html, то он будет работать в контексте безопасности браузера



Решение: Создадим компонент ОС (Windows Script Host), с помощью которого можно будет запускать сценарии непосредственно в операционной системе (без браузера)



VBScript /JScript - недостатки

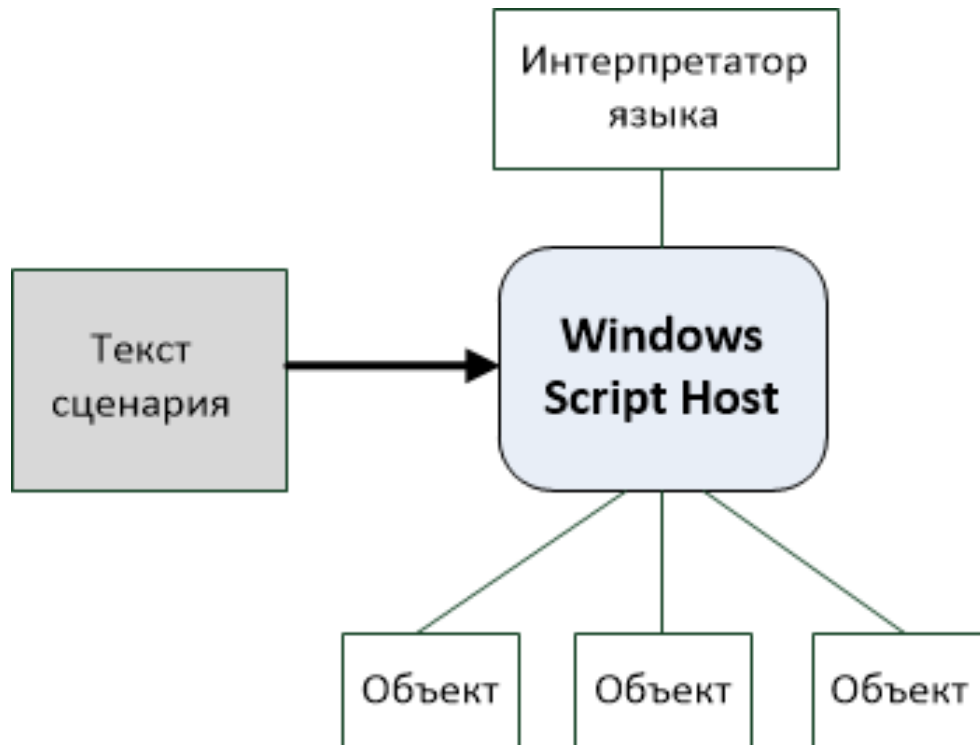
Проблема. В языках сценариев нет специфической функциональности по управлению компонентами ОС

Решение. Создадим и зарегистрируем в системе объекты с нужными методами и свойствами

FileSystemObject	Обеспечивает доступ к файловой системе
WshShell	Позволяет запускать независимые процессы, создавать ярлыки, работать с переменными среды, системным реестром и специальными папками Windows
WshSpecialFolders	Обеспечивает доступ к специальным папкам Windows
WshShortcut	Позволяет работать с ярлыками Windows
WshEnvironment	Предназначен для просмотра, изменения и удаления переменных среды
WshNetwork	Содержит сетевую информацию для локального компьютера, позволяет подключать сетевые диски и принтеры
...	

Скрипты на VBScript через WSH

Windows Script Host – посредник между интерпретатором языка сценариев и внешними объектами.



Скрипты на VBScript/JScript через WSH

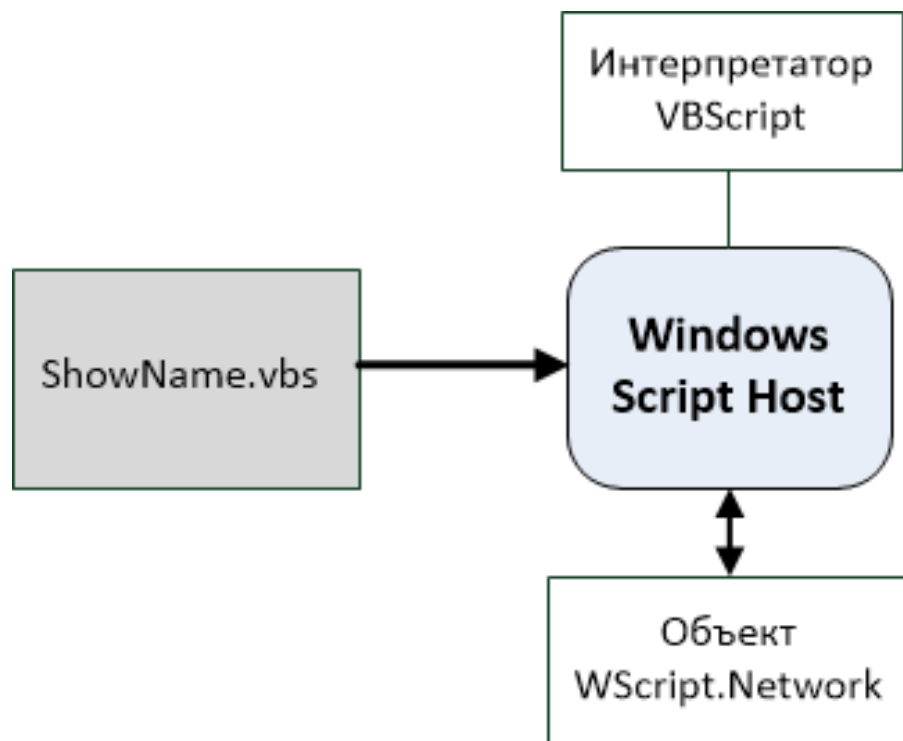
Windows Script Host (WSH):

1. Позволяет запускать в контексте безопасности пользователя файлы *.vbs (VBScript) и *.js (JScript) в консольном (cscript.exe) или графическом (wscript.exe) режиме.
2. Обеспечивает доступ из сценариев к свойствам и методам внешних объектов.
3. Регистрирует в системе объекты, обеспечивающие функциональность для работы с компонентами операционной системы Windows.

Скрипты на VBScript/JScript через WSH

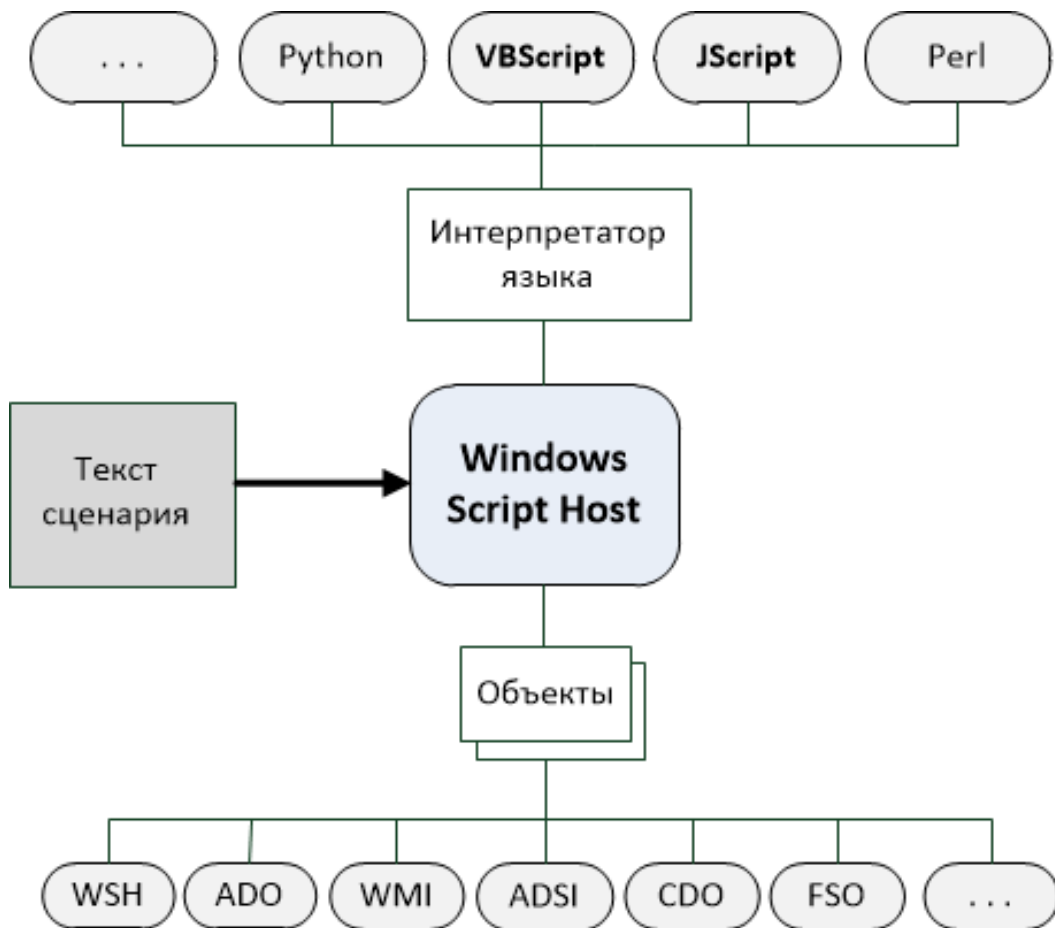
Пример. ShowName.vbs

```
set wnet = CreateObject("WScript.Network")  
uname = wnet.UserName  
MsgBox "Ваше имя пользователя – " & uname
```



Сценарии через WSH

Написание сценариев складывается из двух составляющих:



Внутренние элементы языка программирования

Свойства и методы внешних объектов
(не зависят от выбора языка программирования)

Внутренние элементы языка сценариев

Visual Basic Scripting Edition
Statements

Visual Basic Scripting Edition
Functions

[This topic is part of a beta release a

[This topic is part of a beta release and is subject to change in future releases. Blank topics are included as placeholders.]

In This Section

- [Call Statement](#)
- [Class Statement](#)
- [Const Statement](#)
- [Dim Statement](#)
- [Do...Loop Statement](#)
- [Erase Statement](#)
- [Execute Statement](#)
- [ExecuteGlobal Statement](#)
- [Exit Statement](#)
- [For Each...Next Statement](#)
- [For...Next Statement](#)
- [Function Statement](#)
- [If...Then...Else Statement](#)
- [On Error Statement](#)
- [Option Explicit Statement](#)
- [Private Statement](#)
- [Property Get Statement](#)
- [Property Let Statement](#)
- [Property Set Statement](#)
- [Public Statement](#)
- [Randomize Statement](#)
- [ReDim Statement](#)
- [Rem Statement](#)
- [Select Case Statement](#)
- [Set Statement](#)
- [Sub Statement](#)
- [While...Wend Statement](#)
- [With Statement](#)

The following table contains the VBScript functions.

Abs	Array	Asc	Atn
CBool	CByte	CCur	CDate
CDBl	Chr	CInt	CLng
Conversions	Cos	CreateObject	CSng
Date	DateAdd	DateDiff	DatePart
DateSerial	DateValue	Day	Derived Maths
Eval	Exp	Filter	FormatCurrency
FormatDateTime	FormatNumber	FormatPercent	GetLocale
GetObject	GetRef	Hex	Hour
InputBox	InStr	InStrRev	Int, Fixs
IsArray	IsDate	IsEmpty	IsNull
IsNumeric	IsObject	Join	LBound
LCase	Left	Len	LoadPicture
Log	LTrim; RTrim; and Trims	Maths	Mid
Minute	Month	MonthName	MsgBox
Now	Oct	Replace	RGB
Right	Rnd	Round	ScriptEngine
ScriptEngineBuildVersion	ScriptEngineMajorVersion	ScriptEngineMinorVersion	Second
SetLocale	Sgn	Sin	Space
Split	Sqr	StrComp	String
Tan	Time	Timer	TimeSerial
TimeValue	TypeName	UBound	UCase
VarType	Weekday	WeekdayName	Year

Внешние объекты

<p>Собственные объекты WSH</p>	<ul style="list-style-type: none">• Работа со стандартными потоками ввода/вывода• Управление дочерними процессами• Работа с локальной сетью• Просмотр и изменение переменных среды• Доступ к специальным папкам Windows• Работа с ярлыками• Работа с системным реестром
<p>Scripting.FileSystemObject</p>	<ul style="list-style-type: none">• Управление файловой системой (создание, копирование, удаление файлов/папок, ...)• Чтение/запись текстовых файлов
<p>Windows Management Instrumentation (WMI)</p>	<p>Обеспечивают программный интерфейс управления всеми компонентами операционной модели.</p> <p>На базе схемы CIM (Common Information Model) представления физической и логической структуры компьютерной системы в виде единой расширяемой объектно-ориентированной информационной модели с едиными интерфейсами для получения информации о любом компоненте модели.</p>

Внешние объекты

Active Directory Scripting Interface (ADSI)	Работа с каталогами пользователей в доменных сетях и сетях на базе рабочих групп (Active Directory, Microsoft Exchange Server, Microsoft IIS, любой LDAP-каталог)
Collaboration Data Object (CDO)	Работа с сообщениями электронной почты и почтовыми серверами
Active Database Object (ADO)	Работа с базами данных
Приложения Microsoft Office	Управление офисными приложениями и документами
Сторонние приложения	

COM-объекты в системе

HKEY_CLASSES_ROOT

HKEY_CLASSES_ROOT\CLSID

Редатор реестра

Файл Правка Вид Избранное Справка

Имя	Тип	Значение
(По умолчанию)	REG_SZ	{0D43FE01-F093-11CF-8940-00A0C9054228}

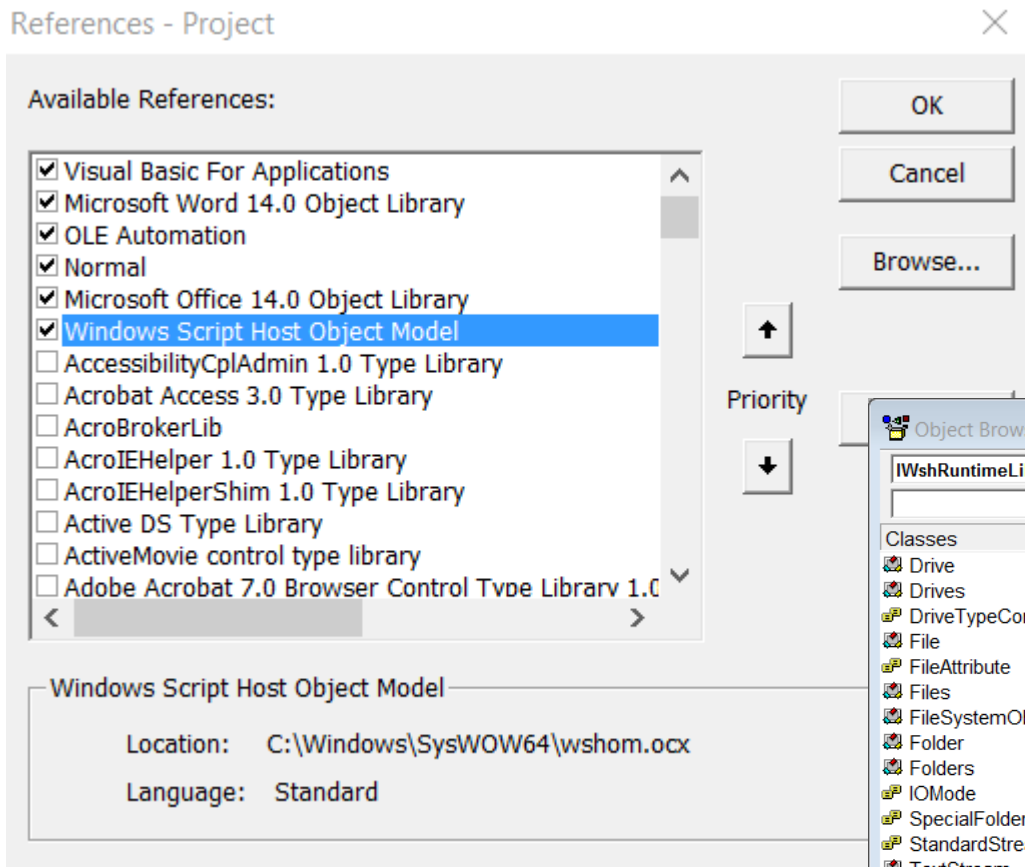
Редатор реестра

Файл Правка Вид Избранное Справка

Имя	Тип	Значение
(По умолчанию)	REG_SZ	C:\Windows\System32\scrrun.dll
ThreadingModel	REG_SZ	Both

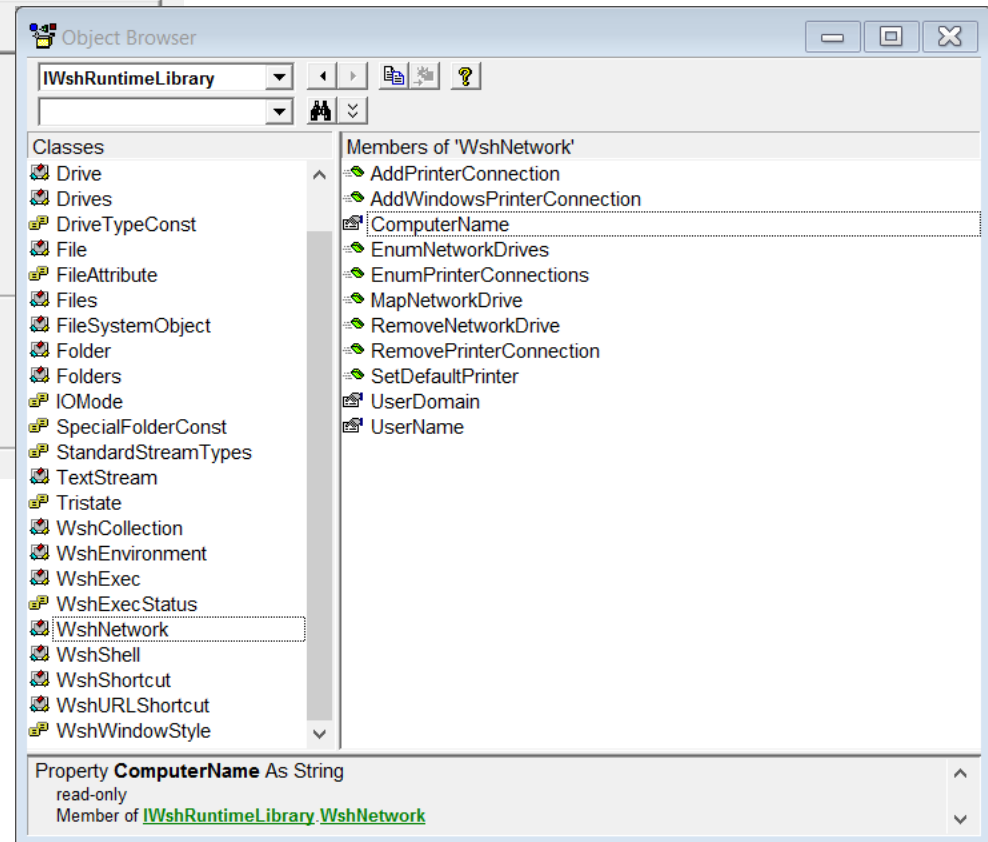
Компьютер\HKEY_CLASSES_ROOT\CLSID\{0D43FE01-F093-11CF-8940-00A0C9054228}\InprocServer32

COM-объекты в системе

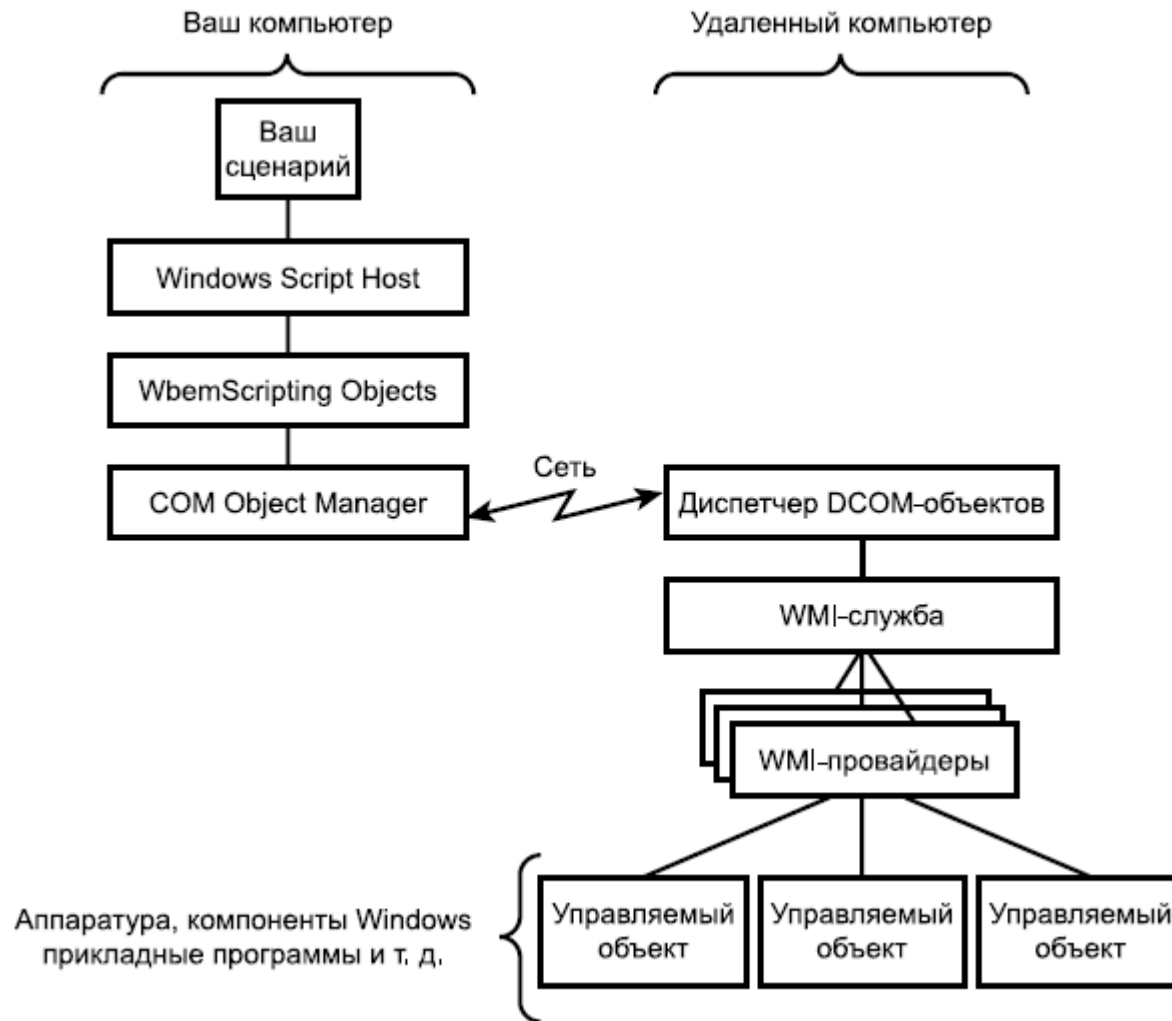


MS Office, Visual Basic

1. Tools\References
2. View\Object Browser

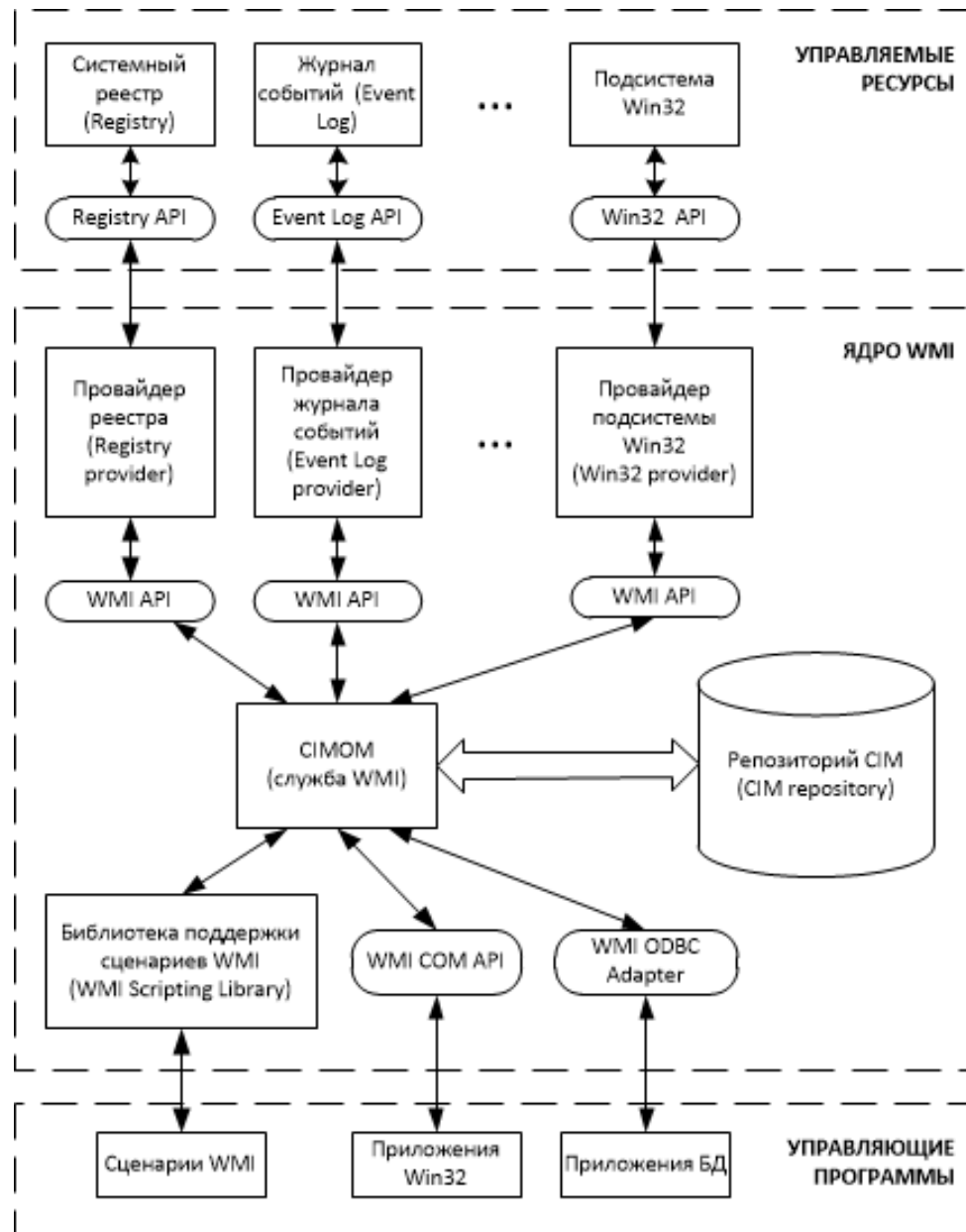


WMI

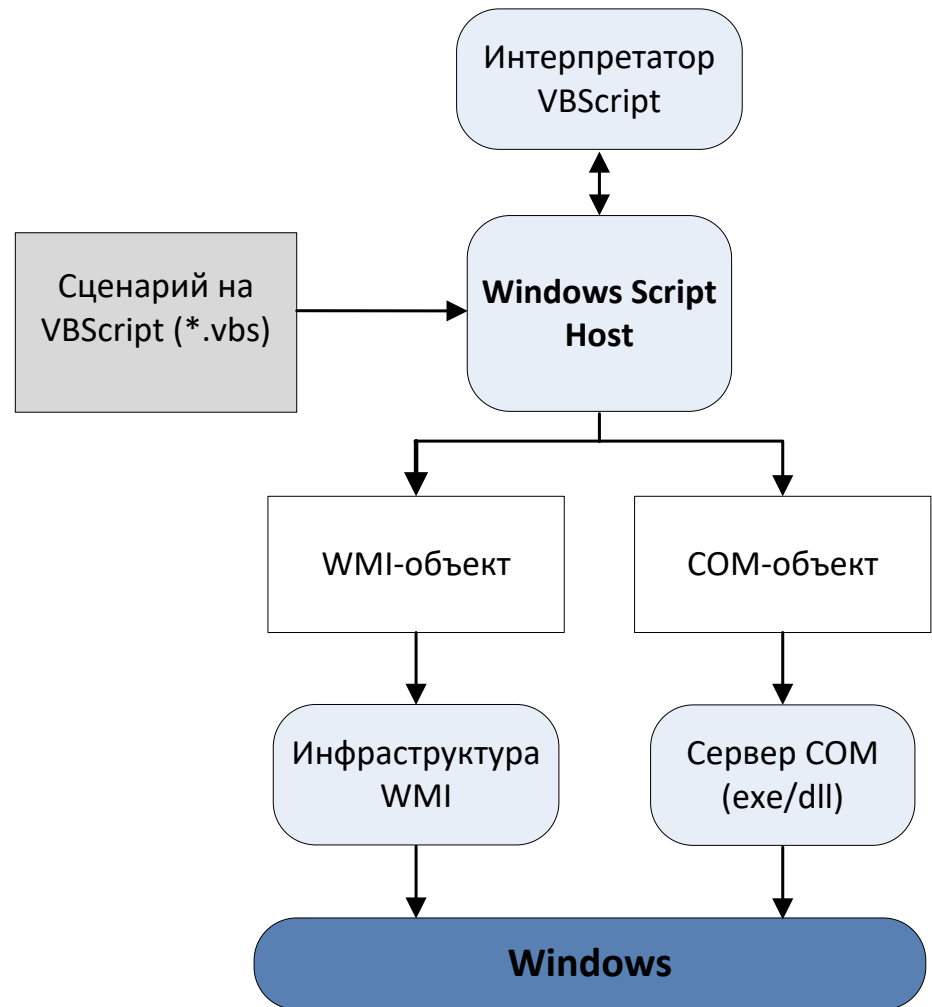
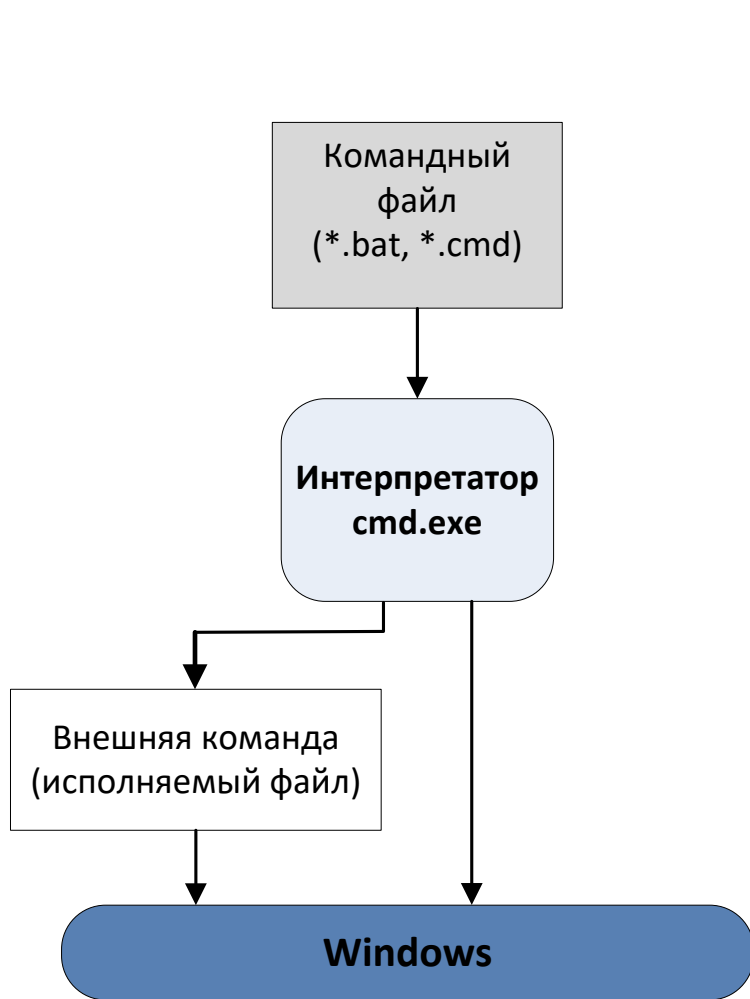


Локальный сценарий взаимодействует с WMI-службой на удаленном компьютере, которая углубляется в настройки Windows посредством служб WMI-провайдеров.

Архитектура WMI



Надежность сценариев



Выбор языка: VBScript или JScript

	VBScript	JScript
Синтаксис	+ <i>Один любит арбуз...</i>	+ <i>Другой свиной хрящик</i>
Применимость	- Только Windows	+ Во всех html-страницах на любых платформах
Встроенные функции	Много функций для работы со строками, датой и временем	Функций мало, но есть встроенные объекты Array, Date, Math, String со множеством методов
Перебор элементов коллекции	+ Напрямую в в цикле For Each ... Next	- Через вспомогательный элемент Enumerator и цикл for с условием завершения.
Вызов методов объектов WMI	+ Вызываются напрямую	- Вызов через специальный объект SWbemObject
Готовые примеры в Интернете	+ Большинство примеров на языке VBScript	

Грамотный код сценариев

- Сценарий должен быть максимально универсален (не должно быть магических переменных, фиксированных путей и т.п.)
- Соблюдение правил структурного программирования
- Комментирование
- Отключение неявного объявления переменных (VBScript)
- Проверка синтаксиса параметров, с которыми запущен сценарий
- Проверка существования файлов и папок перед обращением к ним

Инструментарий

Минимум

- Любой текстовый редактор

Дополнительно

- Специальный редактор (VBSEdit, PrimalScript, notepad++, VS Code, ...)
- Файловый менеджер (Total Commander, FAR, ...)

Для полного счастья не хватает...

- Интерактивности
- Выполнения однострочных команд
- Перенаправления ввода/вывода внутри сценариев
- Подключения в скриптах внешних файлов
- Нет простого и быстрого способа узнать, какие свойства и методы содержит объект
- Работать с WMI слишком сложно

Вывод

Сценарии WSH – это универсальный инструмент, который в любой версии операционной системы Windows позволяет решать задачи автоматизации практически любой степени сложности, но требует при этом большой работы по изучению самих языков сценариев и ряда смежных технологий управления операционной системой (WMI, ADSI и т.п.).