

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Лабораторная работа №1.5

дисциплина «Администрирование распределенных вычислительных сетей»  
по теме «Windows Script Host. Создание сценариев. Сценарии пользователей и  
компьютеров»

Выполнил: студент группы ВТ-31  
Проверил:

Макаров Д.С.  
Федотов Е.А.

Белгород 2020

# Лабораторная работа №1.5

## «Windows Script Host. Создание сценариев. Сценарии пользователей и компьютеров»

**Цель работы:**изучить назначение Windows Script Host и сценариев, способы их запуска; научиться создавать сценарии пользователей и компьютеров..

### Краткие теоритические сведения

VBScript — язык сценариев, созданный компанией Microsoft на основе языка Visual Basic, предназначенный для применения в приложениях, использующих технологию Active Scripting.

В Visual Basic работают следующие правила:

- длина строки не ограничена;
- регистр символов не учитывается;
- количество пробелов между параметрами не учитывается;
- строку команды можно разрывать, а на месте разрыва нужно вставлять символ «\_»;
- максимальная длина имени переменной 255 символов;
- комментарии обозначаются символом «'». несколько команд можно расположить на одной строке, разделив их символом «:»

### Задание к лабораторной работе

#### 1. Создание простейшего сценария и запуск его различными способами

- Запустите программу Блокнот, где наберите текст скрипта:

```
WScript.Echo "MS Windows Server 2003"
```

Сохраните файл с расширением \*.vbs Запустите сценарий следующими способами:

- с помощью командной строки в консольном режиме (утилита cscript.exe);
- с помощью командной строки в графическом режиме (утилита wscript.exe);
- с помощью меню Пуск > Выполнить;
- с помощью Проводника.

Запустите тот же сценарий из командной строки в консольном режиме, но с использованием параметров, рассмотренных в теоретических сведениях.Измените текст сообщения в сценарии на "Hello".

Сохраните его в файл logon.vbs. Создайте копию сценария logon.vbs. Измените текст сообщения в сценарии на "Goodbye". Сохраните его в файл logoff.vbs.

## 2. Управление сценариями пользователей и компьютеров

Поместите созданный выше сценарий `logon.vbs` Откройте консоль Групповая политика (Group Policy) для работы с нужным сайтом, доменом или ОП (оснастка Редактор объектов групповой политики).

В узле Конфигурация компьютера (Computer Configuration) дважды щелкните папку Конфигурация Windows (Windows Settings). Затем щелкните Сценарии (Scripts). Щелкните правой кнопкой элемент Автозагрузка (Startup) для работы со сценариями загрузки или элемент Завершение работы (Shutdown) для работы со сценариями завершения работы. Выберите команду Свойства (Properties).

Щелкните кнопку Показать файлы (Show Files). Если вы скопировали сценарий компьютера в нужное место в папке политик, вы должны увидеть его.

Чтобы назначить сценарий, щелкните Добавить (Add). В поле Имя сценария (Script Name) введите имя сценария, скопированного в папку соответствующей политики, В поле Параметры сценария (Script Parameters) наберите аргументы командной строки, которые следует передать в сценарий серверу сценариев.

Таким образом, добавьте сценарий `logon.vbs` для запуска во время загрузки компьютера, а сценарий `logoff.vbs` – перед завершением работы. Если необходимо, кнопками Вверх (Up) и Вниз (Down) задайте очередность выполнения сценариев при загрузке или завершении работы. Чтобы изменить имя сценария или параметры, выделите сценарий в списке и щелкните Изменить (Edit). Проверьте работоспособность обоих сценариев.

### 1. Назначение сценариев входа и выхода пользователя

Сценарии пользователей назначают одним из трех способов:

- Сценарии входа и выхода назначают как часть групповой политики. Таким образом, все пользователи — члены сайта, домена и/или ОП — автоматически исполняют сценарии при входе или выходе.
- Сценарии входа в консоли Active Directory — пользователи и компьютеры (Active Directory Users and Computers) назначают индивидуально. Так, каждому пользователю или группе можно приписать отдельный сценарий входа.
- Индивидуальные сценарии входа активизируют с помощью Мастера планирования заданий (Scheduled Task Wizard).

Создайте новый сценарий по подобию рассмотренных выше.

Для облегчения управления скопируйте нужные сценарии в папку соответствующей политики. Откройте консоль Групповая политика (Group Policy) для работы с нужным сайтом, доменом или ОП (оснастка Редактор объектов групповой политики).

В узле Конфигурация пользователя (User Configuration) дважды щелкните папку Конфигурация Windows (Windows Settings). Затем щелкните Сценарии

(Scripts). Щелкните правой кнопкой элемент Вход в систему (Logon) для работы со сценариями входа или элемент Выход из системы (Logoff) для работы со сценариями выхода из системы. Выберите команду Свойства (Properties). Щелкните кнопку Показать файлы (Show Files). Если вы скопировали сценарий пользователя в нужное место в папке политик, вы должны увидеть его.

Чтобы назначить сценарий, щелкните Добавить (Add). В поле Имя сценария (Script Name) введите имя сценария, скопированного в папку соответствующей политики или В поле Параметры сценария (Script Parameters) введите аргументы командной строки, которые нужно передать в сценарий серверу сценариев. Повторите этот пункт, чтобы добавить другие сценарии.

Кнопками Вверх (Up) и Вниз (Down) задайте очередность выполнения сценариев при входе в систему или выходе из нее. Чтобы изменить имя сценария или параметры, выделите сценарий в списке и щелкните Изменить (Edit).

Проверьте работоспособность сценария. Чтобы удалить сценарий, выделите его в списке и щелкните Удалить (Remove).

## 1. Просмотр свойств домена

Создайте файл сценария VBScript со следующим текстом:

```
strObjectDN = "dc=contoso,dc=com"
DisplayAttributes("LDAP://" & strObjectDN)
Function DisplayAttributes(strObjectADsPath)
set objObject = GetObject(strObjectADsPath)
objObject.GetInfo
set dicADsType = CreateObject("Scripting.Dictionary")
dicADsType.Add 0, "INVALID"
dicADsType.Add 1, "DN_STRING"
dicADsType.Add 2, "CASE_EXACT_STRING"
dicADsType.Add 3, "CASE_IGNORE_STRING"
dicADsType.Add 4, "PRINTABLE_STRING"
dicADsType.Add 5, "NUMERIC_STRING"
dicADsType.Add 6, "BOOLEAN"
dicADsType.Add 7, "INTEGER"
dicADsType.Add 8, "OCTET_STRING"
dicADsType.Add 9, "UTC_TIME"
dicADsType.Add 10, "LARGE_INTEGER"
dicADsType.Add 11, "PROV_SPECIFIC"
dicADsType.Add 12, "OBJECT_CLASS"
dicADsType.Add 13, "CASEIGNORE_LIST"
dicADsType.Add 14, "OCTET_LIST"
dicADsType.Add 15, "PATH"
dicADsType.Add 16, "POSTALADDRESS"
dicADsType.Add 17, "TIMESTAMP"
dicADsType.Add 18, "BACKLINK"
dicADsType.Add 19, "TYPEDNAME"
dicADsType.Add 20, "HOLD"
dicADsType.Add 21, "NETADDRESS"
dicADsType.Add 22, "REPLICAPOINTER"
dicADsType.Add 23, "FAXNUMBER"
dicADsType.Add 24, "EMAIL"
dicADsType.Add 25, "NT_SECURITY_DESCRIPTOR"
```

```

dicADsType.Add 26, "UNKNOWN"
dicADsType.Add 27, "DN_WITH_BINARY"
dicADsType.Add 28, "DN_WITH_STRING"

for intIndex = 0 to (objObject.PropertyCount - 1)
set objPropEntry = objObject.Item(intIndex)
for each objPropValue in objPropEntry.Values
value = ""
if (dicADsType(objPropValue.ADsType) = "DN_STRING") then
value = objPropValue.DNString
elseif (dicADsType(objPropValue.ADsType) = "CASE_EXACT_STRING") then
value = objPropValue.CaseExactString
elseif (dicADsType(objPropValue.ADsType) = "CASE_IGNORE_STRING") then
value = objPropValue.CaseIgnoreString
elseif (dicADsType(objPropValue.ADsType) = "PRINTABLE_STRING") then
value = objPropValue.PrintableString
elseif (dicADsType(objPropValue.ADsType) = "NUMERIC_STRING") then
value = objPropValue.NumericString
elseif (dicADsType(objPropValue.ADsType) = "BOOLEAN") then
value = objPropValue.Boolean
elseif (dicADsType(objPropValue.ADsType) = "INTEGER") then
value = objPropValue.Integer
elseif (dicADsType(objPropValue.ADsType) = "LARGE_INTEGER") then
set objLargeInt = objPropValue.LargeInteger
value = objLargeInt.HighPart * 2^32 + objLargeInt.LowPart
elseif (dicADsType(objPropValue.ADsType) = "UTC_TIME") then
value = objPropValue.UTCTime
else
value = "<" & dicADsType.Item(objPropEntry.ADsType) & ">"
end if

WScript.Echo objPropEntry.Name & " : " & value
next
next
End Function

```

Запустите его через командную строку в консольном режиме. Проанализируйте текст сценария и выводимые им сообщения.

## 5. Просмотр свойств объекта RootDSE

Первоначально объект RootDSE был определен в документе RFC 2251 как часть спецификации LDAP версии 3. Поэтому его имя отсутствует в пространстве имен Active Directory. Каждый контроллер домена самостоятельно поддерживает этот искусственный объект.

Доступ к объекту RootDSE может осуществляться анонимно, поэтому имя и пароль пользователя не упоминаются. В решениях на основе командной строки и VBScript используется бессерверное подключение к объекту RootDSE.

В таких случаях для поиска контроллера домена в том домене, где вы зарегистрированы, применяется локатор контроллеров домена. При использовании программы LDP вы достигнете того же эффекта, если просто не зададите имя сервера в диалоговом окне Connect (Подключение).

Объект RootDSE служит основой переносимых приложений для работы с Active Directory. Кроме всего прочего, он предоставляет механизм программного определения отличительных имен различных контекстов именования, благодаря чему отпадает необходимость в жестком кодировании их в сценариях и программах.

Создайте файл сценария VBScript со следующим текстом:

```
set objRootDSE = GetObject("LDAP://RootDSE")
objRootDSE.GetInfo
for i = 0 to objRootDSE.PropertyCount - 1
set strProp = objRootDSE.Item(i)
WScript.Echo strProp.Name & " "
for each strPropval in strProp.Values
WScript.Echo " " & strPropval.CaseIgnoreString
next
next
```

Запустите его через командную строку в консольном режиме. Проанализируйте текст сценария и выводимые им сообщения.

### Ход работы

1. Был созданы файлы logon.vbs, logoff.vbs и запущенные различными способами. В зависимости от способа запуска изменялась форма вывода сообщения. Для консольного запуска сообщение выводилось в консоль, для остальных случаев появлялось диалоговое окно.

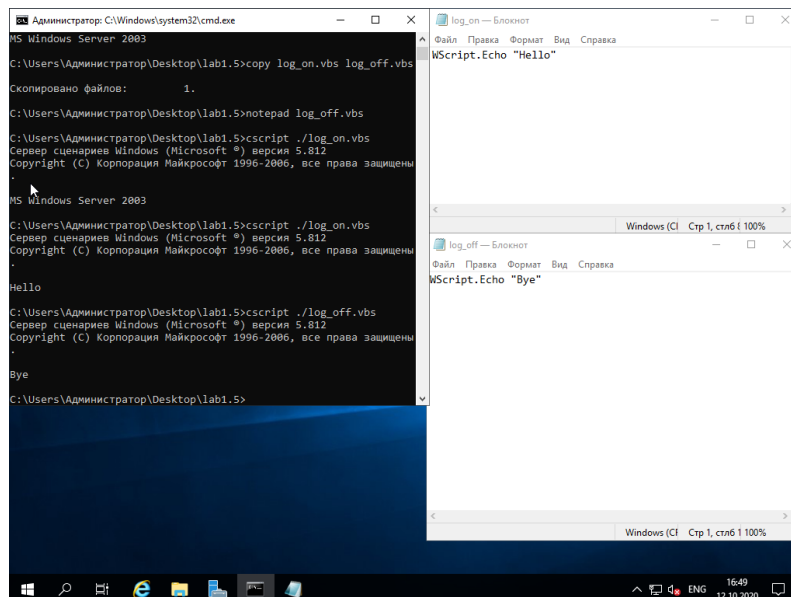


Рис. 1: Запуск скрипта при помощи cscript

2. В групповой политике домена АД в пункте Автозагрузка и Завершение работы были добавлены скрипты logon.vbs logoff.vbs соответственно. А так же проверена их работоспособность.

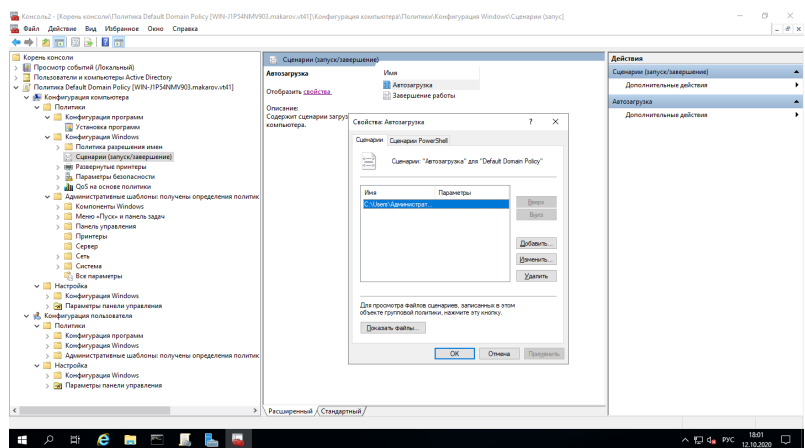


Рис. 2: Добавление скрипта в Автозагрузку

- В групповой политике домена AD в пункте Сценарии входа пользователя и Сценарии выхода были добавлены скрипты logon.vbs logoff.vbs соответственно. А так же проверена их работоспособность.

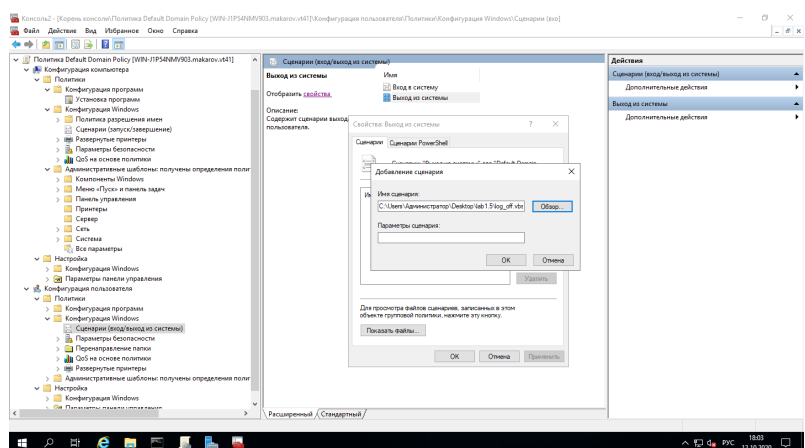


Рис. 3: Добавление скрипта в Сценарии входа пользователя

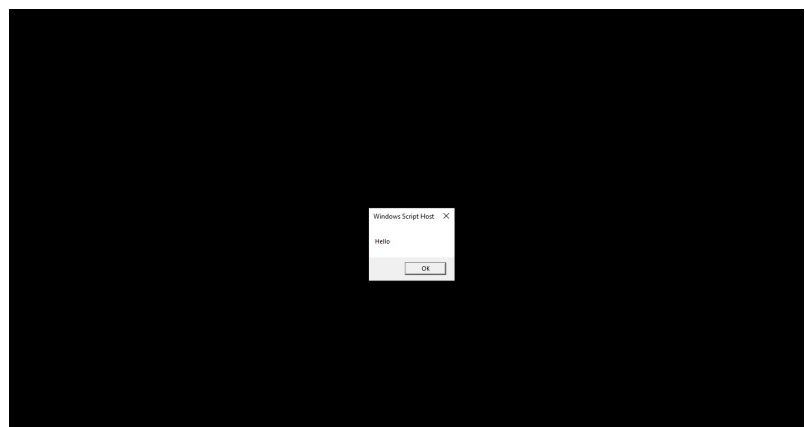


Рис. 4: Работоспособность скрипта

4. Был создан файл с указанным содержимым и запущен через консоль. Результат выполнения сценария - информация о корневом объекте Active Directory.

```
objectClass : top
objectClass : domain
objectClass : domainDNS
distinguishedName : DC=makarov,DC=vt41
instanceType : 5
whenCreated : 07.10.2020 23:38:54
whenChanged : 14.10.2020 21:56:18
subRefs : DC=forestnsZones,DC=makarov,DC=vt41
subRefs : DC=DomainDnsZones,DC=makarov,DC=vt41
subRefs : CN=Configuration,DC=makarov,DC=vt41
usnCreated : 4099
usnSignature : <OCTET_STRING>
usnChanged : 24583
ntSecurityDescriptor : <NT_SECURITY_DESCRIPTOR>
name : makarov
objectGUID : <OCTET_STRING>
creationTime : 1,3247185749186E+17
forceLogoff : -9,22337203685478E+18
lockoutDuration : -22294967296
lockoutObservationWindow : -22294967296
lockoutThreshold : 0
maxPwdAge : -362880000000000
minPwdAge : -868294967296
minPwdLength : 7
modifiedCountAtLastProm : 0
nextRid : 1000
pwdProperties : 0
pwdHistoryLength : 24
objectSid : <OCTET_STRING>
serverState : 1
usnCompat : 1
modifiedCount : 1
auditingPolicy : <OCTET_STRING>
ntMixedDomain : 0
RIDManagerReference : CN=RID Manager$,CN=System,DC=makarov,DC=vt41
FSMORoleOwner : CN=NTDS Settings,CN=WIN-J1P54NMV903,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=makarov,DC=vt41
systemFlags : -1946157056
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
wellKnownObjects : <DN_WITH_BINARY>
objectCategory : CN=Domain-DNS,CN=Schema,CN=Configuration,DC=makarov,DC=vt41
isCriticalSystemObject : 1
```

Рис. 5: Вывод сценария

5. Был создан файл с указанным содержимым и запущен через консоль.

```
domainFunctionality
7
forestFunctionality
7
domainControllerFunctionality
7
rootDomainNamingContext
DC=makarov,DC=vt41
ldapServiceName
makarov.vt41:win-j1p54nmv903$@MAKAROV.VT41
isGlobalCatalogReady
TRUE
supportedSASLMechanisms
GSSAPI
GSS-SPNEGO
EXTERNAL
DIGEST-MD5
supportedLDAPVersion
3
2
supportedLDAPPolicies
MaxPoolThreads
MaxPercentDirSyncRequests
MaxDatagramRecv
MaxReceiveBuffer
InitRecvTimeout
MaxConnections
MaxConnIdleTime
MaxPageSize
MaxBatchReturnMessages
MaxQueryDuration
MaxDirSyncDuration
MaxTempTableSize
MaxResultSetSize
MinResultSets
MaxResultSetsPerConn
MaxNotificationPerConn
MaxValRange
MaxValRangeTransitive
ThreadMemoryLimit
SystemMemoryLimitPercent
supportedControl
1.2.840.113556.1.4.319
1.2.840.113556.1.4.801
1.2.840.113556.1.4.473
1.2.840.113556.1.4.528
1.2.840.113556.1.4.417
```

Рис. 6: Результат выполнения сценария



**Вывод:** Я ознакомился с языком сценариев VBS его синтаксисом, ограничениями. Научился создавать сценарии, запускать их различными способами и работать с объектами AD при помощи инструментария VBS.

### Контрольные вопросы

1. Сколькими способами можно запустить сценарий через командную строку? Перечислите их.
2. Выдаст ли ошибку сценарий при запуске, если там будет введен следующий текст? Почему?

```
Const Message = "Hello World!"  
WScript.Echo meSSaGe
```

3. Каково по умолчанию время выполнения сценария?
4. Напишите сценарий получения объекта Пользователя Иван Иванов, находящийся в организационном подразделении Работники домена Gazprom.com.
5. Что такое LDAP?
6. Недостатки языка VBScript?

### Ответы

1. Запуск скрипта в командной строке можно осуществить при помощи 2 программ:
  - cscript - консольный режим
  - wscript - графический режим
2. Ошибка возникает в следствии неправильного обращения к переменной Message, язык VBS - регистрозависимый.
3. Время исполнения сценария по умолчанию - не ограничено.
4. PowerShell скрипт для получения объекта:

```
Get-ADObject -Identity "CN= Иван Иванов,OU= Работники,DC=Gazprom,DC=com"
```

5. LDAP — протокол прикладного уровня для доступа к службе каталогов. Является одной из составных частей AD.
6. Недостаток языка VBScript в том что Microsoft прекратила его поддержку. В качестве замены в современных системах используется язык сценариев Powershell.