

ИВАН КОРОБКО

# АДМИНИСТРИРОВАНИЕ СЕТЕЙ WINDOWS С ПОМОЩЬЮ СЦЕНАРИЕВ

**Windows Script Host, Visual Basic  
Script Edition, ASP, ASP.NET**

**Основы программирования  
Active Directory: WinNT и LDAP**

**Microsoft Windows Management  
Instrument**

**Сценарии регистрации  
пользователей в сети**

**Автоматическая установка  
программного обеспечения**

Иван Коробко

**АДМИНИСТРИРОВАНИЕ  
СЕТЕЙ  
WINDOWS  
С ПОМОЩЬЮ СЦЕНАРИЕВ**

Санкт-Петербург

«БХВ-Петербург»

2007

УДК 681.3.06  
ББК 32.973.26-018.2  
К68

**Коробко И. В.**

К68 Администрирование сетей Windows с помощью сценариев. — СПб.: БХВ-Петербург, 2007. — 368 с.: ил. — (Системный администратор)

ISBN 978-5-9775-0140-8

Книга представляет собой подробное руководство по автоматизации различных процессов в сети с помощью сценариев, а также содержит большое количество справочной информации. Приведенные примеры наглядно иллюстрируют возможные способы решения задач, возникающих перед пользователями и системными администраторами. Описываются достоинства и недостатки каждого способа. Рассмотрены основы разработки сценариев с использованием Windows Script Host, Visual Basic Script Edition, ASP и ASP.NET. Показано, как программно управлять реестром и файловой системой. Рассмотрены инструменты WMI. Уделено большое внимание созданию сценариев регистрации пользователей в сети на базе языка программирования KIXTart. Описано программное управление Active Directory с помощью LDAP и WinNT. Рассмотрены вопросы автоматизации процесса установки ОС и ПО, клонирования жестких дисков и др.

*Для системных администраторов, сотрудников службы технической поддержки*

УДК 681.3.06  
ББК 32.973.26-018.2

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.08.07.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 29,67.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0140-8

© Коробко И. В., 2007  
© Оформление, издательство "БХВ-Петербург", 2007

# Оглавление

<b>Благодарности .....</b>	<b>1</b>
<b>Предисловие .....</b>	<b>3</b>
<b>Введение .....</b>	<b>5</b>
Ответственность .....	8
<b>Глава 1. Основы программирования сценариев .....</b>	<b>9</b>
Программирование с использованием объектов.....	9
Понятие объекта в программировании.....	9
Наборы.....	11
Методы .....	11
Обзор систем исчислений, используемых в программировании.....	12
Десятичная система исчисления .....	12
Двоичная система исчисления .....	12
Шестнадцатеричная система исчисления .....	13
Обзор языков программирования .....	15
Сценарии на базе командной строки .....	16
WinBatch .....	17
JavaScript .....	17
Visual Basic Script Edition .....	18
Visual Basic for Application .....	18
Windows Scripting Host.....	18
Hiddensof AutoIt .....	19
KIXtart.....	20
Анализ предлагаемых решений .....	20
ActiveX-компоненты .....	23
Инструмент разработки сценариев. Primal Script 4.0 .....	24
Интегрирование собственных шаблонов .....	24
Создание новой объектной модели.....	28

Добавление новых функций в существующие языки программирования.....	30
Шифрование скриптов.....	31
Шифрование файлов из командной строки .....	31
Использование Microsoft Script Encoder .....	32
Шифрование KIX-сценариев .....	32
<b>Глава 2. Visual Basic Script Edition.....</b>	<b>33</b>
Основы синтаксиса .....	33
Переменные.....	33
Соглашение об именах.....	33
Комментарии.....	34
Непрерывные строки.....	34
Типы данных .....	34
Оператор <i>Option Explicit</i> .....	36
Константы.....	37
Объявление переменных.....	37
Присвоение значений переменным.....	37
Объявление массивов .....	38
Переопределение размеров массива .....	39
Определение границ массива .....	39
Операторы .....	39
Операторы конкатенации.....	40
Преобразование типов.....	41
Инструкции .....	42
Инструкции <i>If...Then</i> .....	42
Инструкция <i>Select...Case</i> .....	43
Инструкция безусловного перехода .....	43
Управляемые циклы.....	44
Циклы с параметрами: оператор <i>For...Next</i> .....	44
Циклы с условием.....	45
Цикл <i>Do...While</i> .....	45
Цикл <i>While...Wend</i> .....	46
Оператор <i>For...Each</i> .....	46
Подпрограммы и классы .....	47
Функция.....	47
Процедуры.....	48
Передача значений параметров с помощью ключевых слов <i>ByRef</i> и <i>ByVal</i> .....	49
Классы .....	49
Получение свойств обработчика языка .....	51

Управление свойствами объектов.....	52
Создание методов .....	54
Использование инструкции <i>With</i> .....	54
Интерактивная работа скриптов .....	55
Ввод информации с помощью функции <i>InputBox()</i> .....	55
Отображение информации с помощью функции <i>MsgBox()</i> .....	56
Определение вида иконки и кнопок диалогового окна .....	57
Определение возвращаемого значения функции <i>MsgBox()</i> .....	59
<b>Глава 3. Windows Script Host .....</b>	<b>61</b>
Сервер сценариев Windows Script .....	61
Установка Windows Script Host 5.6.....	61
Запуск сценариев WSH из командной строки .....	61
Возможности WSH-сценариев .....	63
Объектная модель WSH.....	63
Объект <i>WScript</i> .....	64
Свойства объекта <i>WScript</i> .....	64
Свойство <i>Arguments</i> .....	65
Свойства <i>FullName, Name, Path, Version</i> .....	67
Свойства <i>StdErr, StdIn, StdOut</i> .....	67
Методы объекта <i>WScript</i> .....	67
Метод <i>CreateObject(ObjectID, Prefix)</i> .....	68
Метод <i>GetObject(Object, ObjectID, Prefix)</i> .....	68
Метод <i>DisconnetObject(ObjectID)</i> .....	69
Объект <i>WshArguments</i> .....	69
Объект <i>WshShell</i> .....	70
Метод <i>AppActivate</i> .....	71
Управление ярлыками методом <i>CreateShortcut</i> .....	72
Объект <i>WshShortcut</i> .....	72
Создание ярлыка.....	73
Чтение/изменение свойств ярлыка .....	74
Удаление ярлыка .....	75
Объект <i>WshURLShortcut</i> .....	75
Объект <i>Wscript.Shell</i> .....	76
Метод <i>Environment</i> .....	76
Операции с переменными окружения.....	79
Создание переменных окружения с помощью сценария .....	80
Удаление переменных окружения с помощью сценария.....	80
Метод <i>ExpandEnvironmentStrings</i> .....	80
Метод <i>LogEvent</i> .....	81

Метод <i>PopUp</i> .....	81
Определение возвращаемого значения методом <i>PopUp</i> .....	82
Метод <i>Run</i> .....	83
Метод <i>SendKeys</i> .....	85
Метод <i>SpecialFolders</i> .....	86
Объект <i>WshNetwork</i> .....	88
Методы <i>AddWindowsPrinterConnection</i> и <i>AddPrinterConnection</i> .....	89
Обработка ошибок .....	90
Метод <i>RemovePrinterConnection</i> .....	91
Метод <i>EnumPrinterConnections</i> .....	91
Метод <i>SetDefaultPrinter</i> .....	92
Метод <i>MapNetworkDrive</i> .....	92
Метод <i>EnumNetworkDrives</i> .....	93
Метод <i>RemoveNetworkDrive</i> .....	94
<b>Глава 4. На пути от VBScript к .NET.....</b>	<b>95</b>
Переход от VBScript к ASP .....	95
Настройка IIS для ASP .....	96
Переход от VBScript и ASP к ASP.NET .....	99
Установка Visual Studio .....	99
Microsoft FRAMEWORK .....	100
Управление доверительными отношениями в ASP.NET.....	101
Изменения в синтаксисе .....	104
Файловая структура.....	104
Исполнение ASP-файлов в ASP.NET.....	105
Оператор <i>Option Explicit</i> .....	106
VBScript, ASP .....	106
ASP.NET.....	106
Преобразование типов данных.....	107
Использование скобок при передаче параметров подпрограммам и методам.....	108
Операторы <i>SET</i> и <i>LET</i> .....	109
Поддержка многопоточных компонентов.....	109
Обработка ошибок в ASP.NET .....	110
<b>Глава 5. Программное управление реестром .....</b>	<b>111</b>
Основы построения реестра .....	112
Редакторы реестра.....	113
Программное управление реестром .....	114
Редактирование реестра сценарием на базе командной строки.....	114
Редактирование реестра с помощью WSH.....	115

Редактирование реестра с помощью KIXTart.....	117
Групповые политики.....	120
Административные шаблоны. Синтаксис.....	120
Комментарии.....	121
Строки.....	121
CLASS.....	122
CATEGORY.....	122
POLICY.....	124
Программирование интерфейса политик безопасности.....	125
CHECKBOX.....	127
COMBOBOX.....	128
DROPDOWNLIST.....	129
EDITTEXT.....	130
LISTBOX.....	130
NUMERIC.....	131
TEXT.....	132
Практика использования административных шаблонов.....	132
Внедрение административных шаблонов.....	134
Удаление административного шаблона.....	135
<b>Глава 6. Управление файловой системой.....</b>	<b>137</b>
Команды ввода/вывода (FSO).....	137
Работа с дисками. Набор <i>Drive</i> .....	138
Работа с папками и файлами. Наборы <i>Folders</i> и <i>Files</i> .....	140
Формирование списка вложенных объектов в папке.....	141
Чтение характеристик файлов.....	144
Изменение атрибутов файлов.....	146
Операции над файлами и папками.....	147
Управление текстовыми файлами.....	149
Управление правами доступа на файлы и папки.....	152
Библиотека ADsSecurity.dll.....	152
Организация доступа к библиотеке.....	152
Управление правами на папку.....	154
Основные операции: просмотр, добавление, удаление.....	158
Просмотр списка доступных объектов.....	159
Добавление объекта.....	161
Удаление существующих объектов.....	163
<b>Глава 7. Основы программирования Active Directory.....</b>	<b>165</b>
Active Directory изнутри.....	165
DNS и Active Directory.....	166

Архитектура службы каталогов Active Directory .....	168
Объектная модель ADSI .....	170
Провайдеры ADSI .....	171
<b>Глава 8. Программное управление ADSI: WinNT .....</b>	<b>173</b>
Объектная модель провайдера WinNT .....	173
Класс <i>Domain</i> .....	177
Определение доступных доменов .....	177
Чтение параметров класса <i>Domain</i> .....	178
Обновление параметров класса <i>Domain</i> .....	179
Перечисление объектов класса <i>Domain</i> .....	179
Создание, переименование и удаление объектов в домене .....	180
Создание объектов .....	181
Удаление объектов .....	182
Переименование объектов .....	183
Подкласс <i>User</i> .....	183
Манипулирование пользовательскими флагами функцией <i>UserFlags()</i> .....	184
Подкласс <i>Group</i> .....	187
Взаимосвязь учетных записей пользователей и групп .....	187
Добавление и удаление учетной записи пользователя в группу .....	187
Получение списка всех учетных записей, входящих в группу .....	188
Просмотр списка групп .....	188
Класс <i>Computer</i> .....	189
Подклассы <i>PrintQueue</i> , <i>PrintJob</i> , <i>PrintJobsCollection</i> .....	189
Управление принтерами и очередями принтеров .....	189
Управление принтером .....	190
Просмотр состояния принтера .....	191
Чтение свойств заданий в очереди принтера .....	192
Управление очередью печати .....	193
Класс <i>Service</i> .....	194
Подкласс <i>FileService</i> и <i>FileShare</i> .....	194
Чтение свойств совместно используемых ресурсов .....	194
Программное создание и удаление совместно используемого ресурса .....	195
Подкласс <i>Service</i> .....	196
Перечисление и чтение свойств служб на выбранном компьютере .....	196
Связывание служб на выбранном компьютере .....	197
<b>Глава 9. Программное управление ADSI: LDAP .....</b>	<b>199</b>
Структура объектной модели провайдера LDAP .....	199
Имена LDAP URL .....	199
Развернутая форма записи .....	200

Сокращенная форма записи .....	201
Определение имени домена. Обзор способов .....	201
Объекты Active Directory.....	202
Поиск объектов в AD .....	203
Поиск всех учетных записей пользователей в домене.....	203
Объектная модель провайдера LDAP .....	204
Active Directory Viewer (Microsoft) .....	204
Просмотр и редактирование объектной модели программой ADV в режиме <i>ObjectViewer</i> .....	205
LDAP Browser 2.6. (Softerra).....	207
Различия провайдеров LDAP и WinNT .....	211
Выполнение команд и программ от имени определенного пользователя .....	211
Выполнение сценариев от имени конкретного пользователя .....	212
Импорт и экспорт данных из AD.....	213
LDIF-файлы .....	214
CSVDE-файлы.....	215
Действия над объектами.....	215
Создание объекта.....	215
Удаление объекта .....	217
Перемещение объектов .....	218
Чтение атрибутов объектов .....	219
Изменение атрибутов объекта.....	220
Поиск объекта .....	220
<b>Глава 10. Microsoft Windows Management Instrument .....</b>	<b>221</b>
Средства управления WMI.....	222
Внутреннее устройство WMI .....	223
Язык запросов WQL.....	224
Безопасность и WMI .....	225
Имперсонация .....	225
Аутентификация .....	226
Привилегии .....	227
Способы доступа к объектам WMI (VBScript).....	232
Продукты, использующие WMI.....	234
SMS 2003 SP1 .....	234
MOM 2005 SP1 .....	236
<b>Глава 11. Сценарий регистрации пользователей в сети .....</b>	<b>237</b>
KIXTart.....	237
Системные требования.....	237
Комплект поставки KIXTart .....	238

Установка KIXTart .....	238
Синтаксис KIX32 .....	238
Запуск KIX32 .....	239
Режим отладки сценариев.....	239
Синтаксис KIXTart .....	239
Адаптация листингов VBScript и WSH к KIXTart.....	240
Задачи, решаемые сценарием.....	241
Решение задачи инвентаризации .....	241
Сбор информации об аппаратном обеспечении с помощью WMI .....	242
Сбор информации об учетной записи пользователя с помощью Active Directory .....	247
Формирование файла отчета в формате XML .....	248
Экспорт данных в SQL.....	250
Подключение к базе данных .....	250
Операции с SQL-таблицей .....	251
Решение задачи подключения сетевых принтеров .....	252
Соглашение об именах групп безопасности и принтеров .....	252
Предварительная настройка принтера и Active Directory .....	253
Формирование списка принтеров, которые необходимо подключить пользователю.....	255
Формирование списка сетевых принтеров, подключенных пользователю .....	258
Приведение списков принтеров в соответствие .....	259
Решение задачи подключения сетевых дисков .....	260
Определение членства в соответствующих группах безопасности.....	262
Чтение данных из AD.....	262
Отключение сетевых дисков .....	264
Подключение необходимых сетевых дисков.....	264
Корректировка описаний дисков в папке Мой компьютер .....	266
Переименование описаний сетевых дисков для Windows XP.....	267
Переименование описаний сетевых дисков для Windows 2000 .....	268
Визуализация работы сценария KIXTart .....	273
Визуализация работы скрипта с помощью стандартных диалоговых окон.....	273
Визуализация работы скрипта с помощью DLL-библиотеки .....	274
KIXForms .....	274
KIXTart Script Packager.....	274
Визуализация работы скрипта с помощью DHTML .....	275
Синтаксис KIXWin.....	276
Основы формирования файла DHTML.....	276
Внедрение скрипта в эксплуатацию.....	278

<b>Глава 12. Подготовка рабочей станции к функционированию в сети.....</b>	<b>283</b>
Клонирование жестких дисков .....	283
Автоматизация процесса установки: ОС .....	286
Обзор инсталляторов .....	286
Пакетная установка ПО.....	286
Windows Installer.....	287
Взгляд изнутри: файл msixec.exe .....	290
Параметры командной строки для Msiexec.....	290
Inno Setup.....	292
Nullsoft Scriptable Install System (NSIS).....	293
Wise Installer.....	295
Создание дистрибутива .....	295
Подготовка файловой структуры будущего диска.....	295
Создание файла ответов для установки Windows .....	296
Файловая структура дистрибутивного диска Windows .....	297
Механизм инсталляции быстрых исправлений.....	299
Подготовка дистрибутива Windows .....	299
Интеграция пакета исправлений Service Pack в Windows .....	299
Установка пакетов быстрых исправлений в автоматическом режиме.....	300
Принципы именования hotfix .....	300
Копирование hotfix из Интернета.....	300
Интеграция hotfix в дистрибутив Windows .....	302
Ключи, используемые при установке hotfix.....	302
Установка MUI в автоматическом режиме .....	303
Интеграция драйверов в дистрибутив.....	303
Автоматизация процесса установки: ПО .....	303
Установка антивируса: Norton Antivirus .....	303
Установка навигатора: WinCMD .....	304
Автоматическая установка архиватора: WinRar 3.x .....	304
Установка программы записи CD/DVD: Nero Burning Rom 6.3.0.x .....	305
Установка Adobe Acrobat Reader 6 .....	306
Автоматическая установка Microsoft Office 2003 и необходимых дополнений.....	306
Способы автоматической установки Office .....	306
Подготовка дистрибутива Office .....	307
Создание загрузочного диска .....	309
Создание файла-образа диска .....	309
Тестирование ISO-файла .....	311
Запись файла-образа на диск .....	312

<b>ПРИЛОЖЕНИЯ .....</b>	<b>315</b>
<b>Приложение 1. Управление сетевой печатью.....</b>	<b>317</b>
Сетевой принтер .....	317
Сервер печати .....	318
Соглашение об именах .....	318
Установка и настройка сетевого принтера .....	319
Описание установки принтера .....	322
<b>Приложение 2. Ошибки выполнения сценариев в WSH.....</b>	<b>329</b>
<b>Приложение 3. Объектная модель провайдера WinNT .....</b>	<b>331</b>
objectClass <i>Domain</i> .....	331
objectClass <i>User</i> .....	332
objectClass <i>Group</i> .....	334
objectClass <i>Computer</i> .....	334
objectClass <i>PrintQueue</i> .....	335
objectClass <i>PrintJob</i> .....	335
objectClass <i>FileService</i> .....	336
objectClass <i>FileShare</i> .....	336
objectClass <i>Service</i> .....	337
<b>Приложение 4. Таблица ASCII (American Standard Code for Information Interchange).....</b>	<b>338</b>
Назначение специализированных символов .....	339
Форматирование .....	339
Передача данных .....	340
Разделительные знаки при передаче информации .....	340
Другие символы .....	341
<b>Приложение 5. Файл ответов Winnt.sif.....</b>	<b>342</b>
<b>Приложение 6. Файл ответов Install.inf.....</b>	<b>344</b>
<b>Предметный указатель .....</b>	<b>347</b>

*С любовью посвящается Елене Ходовой*

## Благодарности

Создание этой книги — труд не одного дня. Я хочу выразить особую благодарность Зыкову Филиппу за ценные идеи, Алексеенко Александру за дельные советы, Тачилину Юрию за неоценимую помощь и всему "Центр-ИТ" за огромное терпение во время тестирования созданных мною скриптов.

Мои благодарности сотрудникам издательства "БХВ-Петербург", в особенности Шишигину Игорю и Капалыгиной Екатерине за их вклад в редактировании плана и текста книги.

Отдельное спасибо Ходовой Елене — первому читателю-редактору этой книги перед ее печатью, за массу дельных предложений и поддержку при работе над книгой.



# Предисловие

Человек полетит, опираясь на силу разума,  
а не на силу мышц...

*Н. Е. Жуковский*

Современные компьютерные сети становятся все сложнее и сложнее, а значит, управлять ими вручную все труднее. В больших сетях ошибки обслуживающего персонала обходятся очень дорого, поэтому необходимо свести влияние человеческого фактора к минимуму, максимально повысив производительность труда. Одним из решений является автоматизация функций ИТ-службы с помощью различных скриптов.

Книга задумывалась как подробное руководство-справочник по автоматизированию различных процессов в сети. Она имеет выраженную практическую направленность и написана на основе собственного опыта. Примеры в этой книге наглядно иллюстрируют возможные способы решения поставленной задачи, описываются достоинства и недостатки каждого из них. Прочитав эту книгу, вы узнаете, как создать файл ответов для автоматической установки операционной системы Windows и Microsoft Office, обеспечить автоматическую установку большинства приложений.

Научитесь разрабатывать интеллектуальные сценарии регистрации пользователей в сети, управляющие подключением сетевых ресурсов (дисков и принтеров). Скрипт подключит пользователю только те ресурсы, с которыми он работает. Узнаете, как формировать отчеты, содержащие в себе информацию о регистрирующемся в сети пользователе на основе данных из Active Directory. А также сохранять информацию об аппаратной и программной конфигурации рабочей станции на основе данных из реестра, WMI и в XML-файле, SQL-базе.

Книга будет полезна не только опытным администраторам и программистам, но и всем сотрудникам службы технической поддержки, которые хотят повысить эффективность своей работы, автоматизировав часть возложенных на них задач.



# Введение

Начиная свой путь со специалиста системной поддержки, автор старался автоматизировать рутинные действия — установку операционной системы, различного программного обеспечения, настройку рабочих станций для того, чтобы сэкономить время на выполнение повторяющихся операций и повысить результативность своей работы. Все это позволило снизить ТСО.

Приобретя опыт в создании файлов-ответов и несложных скриптов, автор приступил к созданию сценариев для регистрации пользователей в сети. Идея, лежащая в основе сценария, — создать для каждого пользователя персональное окружение, подключив только необходимые для его работы ресурсы и убрать все лишние. Одновременно, осуществляя при загрузке мониторинг настроек и незаметно для пользователя изменять их. Создание скрипта была очень интересной и плодотворной работой целого коллектива. Параллельно разрабатывались различные административные сценарии по управлению Active Directory.

Основная проблема, с которой столкнулся автор, — отсутствие различной справочной информации, подробного описания объектных моделей, наглядных примеров. Сейчас появилось достаточно много книг — сборников рецептов, в которых можно найти часто используемые решения и воспользоваться ими. Однако иногда необходимо создавать сложные скрипты. Для этого нужно обладать хорошими теоретическими знаниями и понимать идею, заложенную в основе той или иной объектной модели. В большинстве книг теоретический блок как таковой отсутствовал и рассматривался лишь узкий круг вопросов и практических решений. Поэтому родилась идея создания такой книги, которая содержала бы не только конкретные скриптовые рецепты, но и теоретические основы создания скриптов и не требовала обращения к другим источникам информации.

Каждая глава написана на основе личного опыта, снабжена характерными примерами.

В книге вы найдете:

- информацию об основных подходах объектно-ориентированного программирования;
- обзор основных скриптовых языков (VBScript, WSH, KIXTart) и рекомендации по выбору языка для решения поставленной задачи;
- основы программного управления реестром с помощью различных скриптовых языков;
- основы программирования Active Directory, описание объектных моделей основных провайдеров — WinNT и LDAP;
- основы программирования Windows Management Instrument (WMI);
- способы создания интеллектуального сценария регистрации пользователей в сети;
- методы работы с XML, чтения и записи данных в SQL;
- принципы создания файлов ответов для обеспечения автоматической установки ОС;
- обзор существующих инсталляторов с точки зрения автоматизации процесса установки ПО.

Здесь есть ответы на эти и многие другие вопросы, а с помощью приведенной теории можно пополнить свои знания, разрешить оставшиеся вопросы и сделать еще один шаг в программировании скриптов.

Книга состоит из 12 глав.

*Глава 1 "Основы программирования сценариев"* содержит базовые знания по объектно-ориентированному программированию; обзор систем исчислений, используемых в программировании.

Первый шаг создания скрипта — анализ поставленной задачи и выбор оптимального для ее решения языка программирования (VBScript, WSH, KIXTart). В первой главе читатель сможет ознакомиться с возможностями основных существующих скриптовых языков программирования, выбрать наиболее подходящий язык для решения поставленной перед ним задачи.

*Глава 2 "Visual Basic Script Edition"* содержит описание не только базовых возможностей одного из самых используемых скриптовых языков — VBScript, но и дополнительных возможностей: обработку ошибок, классы и т. д.

*Глава 3 "Windows Script Host"* познакомит читателя с основами программирования сценариев для Microsoft Windows Script Host 2.0. Научит управлять се-

тевыми ресурсами, ярлыками программ, осуществлять запуск сценариев из командной строки, управлять процессами ОС, спецпапками, получать доступ к различным OLE-объектам, работать с системным реестром.

*Глава 4 "На пути от VBScript к .NET"* рассказывает о практике перехода с VBScript на более мощные языки — ASP и ASP.NET, на базе полученных знаний во второй и третьих главах.

*Глава 5 "Программное управление реестром"* целиком посвящена реестру и групповым политикам, которые органично дополняют его возможности. Существует несколько скриптовых языков, в том числе VBScript, WSH, KIXTart, скрипты на базе командной строки. В главе описано не только, что такое системный реестр и как создавать REG-файлы, но и рассматриваются приемы работы, характерные для каждого из приведенных скриптовых языков программирования, проиллюстрированные наглядными примерами.

*Глава 6 "Управление файловой системой"* посвящена управлению файловой системой из VBScript, ASP и ASP.NET. Прочитав ее, читатель сможет не только осуществлять различные операции с файлами и папками, но и программно управлять их безопасностью.

*Глава 7 "Основы программирования Active Directory"* содержит обзор воспринимаемых Active Directory форматов имен, доступных провайдеров. Описана объектная модель ADSI.

*Глава 8 "Программное управление ADSI: WINNT"* и *глава 9 "Программное управление ADSI: LDAP"* познакомит читателя с основными принципами программирования Active Directory. Здесь приведены описания объектных моделей провайдеров WINNT и LDAP, рассмотрены различные способы доступа к ним и характерные примеры.

*Глава 10 "Microsoft Windows Management Instrument"* посвящена решению задачи накопления и сохранения информации об аппаратной конфигурации рабочей станции с помощью Microsoft Windows Management Instrument (WMI).

*Глава 11 "Сценарий регистрации пользователей в сети"* — пожалуй, самая большая глава, в которой используются все накопленные знания в этой книге: чтение данных из реестра, использование WMI, получение доступа к Active Directory, запись данных в SQL-таблицы и многое другое. Сценарии формируются на базе специально созданного для этих целей скриптового языка KIXTart.

*Глава 12 "Подготовка рабочей станции к функционированию в сети"* расскажет читателю о клонировании жестких дисков и подготовке рабочей станции к работе в сети. Вы узнаете много нового об инсталляторах и научи-

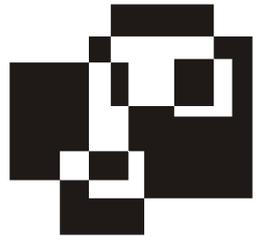
тесь автоматизировать установку различных программ, в том числе Microsoft Office, WinRar, Nero и т. д. Сможете создавать новые файлы ответа, позволяющие обеспечить автоматическую установку операционной системы.

В *приложениях* приведена различная справочная информация, не вошедшая в главы.

Создавая эту книгу, автор надеялся, что она станет настольной.

## ОТВЕТСТВЕННОСТЬ

В книге описан очень мощный инструментарий по управлению серверов и настройке программного обеспечения. Все скрипты перед введением в эксплуатацию настоятельно рекомендуется многократно и всесторонне протестировать в тестовой зоне. Перед внедрением в эксплуатацию следует делать резервные копии. В неумелых руках сценарии могут привести к необратимым последствиям.



## Глава 1

# Основы программирования сценариев

Создание различных сценариев базируется на знании основ объектно-ориентированного программирования, синтаксисе языка, на котором он создается, используемых систем исчисления и их взаимодействии. Первый шаг в создании сценария — выбор оптимального языка программирования. От того, насколько правильно сделан этот шаг, будет зависеть не только удобство создания скрипта программистом, но и скорость его работы и надежность. Повысить скорость создания скрипта можно, выбрав один из предлагаемых редакторов, который обеспечит контроль вводимой информации, позволит сделать пошаговую его трассировку.

## Программирование с использованием объектов

### Понятие объекта в программировании

Примерами *объектов* реального мира могут быть автомобиль, дом, книга, стол. У каждого объекта есть характеризующие его *свойства*. Например, у автомобиля — цвет, габариты, вес и т. д. Число свойств зависит от объекта. Над объектом реального мира можно совершать различные действия: книгу — читать, машину — водить, одежду — носить, стирать и т. д.

В мире разработки программного обеспечения в программах также можно использовать *объектно-ориентированный подход* при определении объектов, которые имеют набор свойств и над которыми можно производить определенные действия. Рассмотрим простой объект — диалоговое окно (рис. 1.1). Его можно представить как объект, поскольку оно содержит данные (информацию) и совершает действие (выводит информацию). Свойствами объекта

"диалоговое окно" являются его положение на рабочем столе, цвет фона, размер шрифта заголовка и т. д.

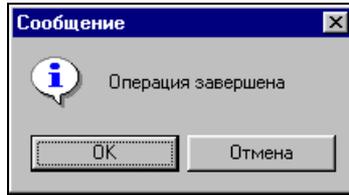


Рис. 1.1. Пример диалогового окна

Идея построения объектной модели основана на том, что объект может содержать другие объекты, так называемые "дочерние" объекты.

Приведенное в качестве примера диалоговое окно имеет две кнопки (**ОК** и **Отмена**) и иконку знака информации. Эти элементы диалогового окна являются его *субобъектами* ("дочерние" объекты).

Рассмотрим в качестве объекта текстовый редактор. У него есть данные (документы, слова, предложения, числа и т. д.), над которыми можно выполнять действия (открывать, закрывать, сохранять документ, вставлять и удалять текст и т. д.).

Для определения структуры объектов в рамках приложения рекомендуется использовать иерархический подход. Объект `Текстовый редактор` в качестве субобъектов содержит `Документы`, которые включают в себя `Абзацы`, состоящие из `Предложений`, которые в свою очередь состоят из `Слов`, а `Слова` из `Букв`. В программе можно сослаться на эту иерархию следующим образом (рис. 1.2):

`Приложение . Документ . Абзац . Предложение . Слово . Буква .`



Рис. 1.2. Пример объектной модели

Имена объектов выстроены в иерархическом порядке и разделены точками. Левее всех находится "родительский" объект, в данном случае Приложение. Зная иерархию объектов, можно обращаться к любому из них, определяя его позицию в иерархии, известной также как *объектная модель* (object model).

## Наборы

Показанная цепочка субобъектов представляет собой простейшую иерархическую модель. При этом документы состоят из целого набора абзацев, абзацы из набора предложений, а предложение из набора слов и т. д. (рис. 1.3).

*Набор* (collection) — это совокупность субобъектов одного типа. Чтобы получить доступ к одному из элементов набора, нужно указать номер необходимого элемента:

Приложение . Документы (0) . Абзацы (5) . Предложения (2) . Слова (9) . Буквы (3) .

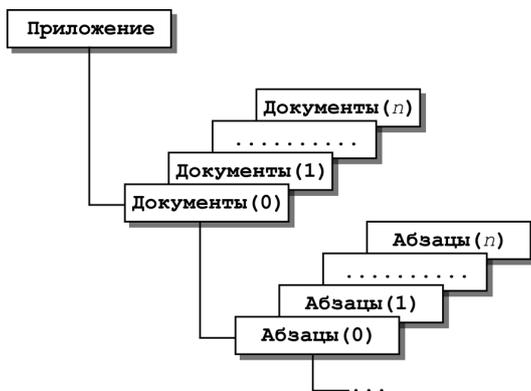


Рис. 1.3. Пример набора объектов

Нумерация элементов начинается с нуля, а имена объектов в данном случае употребляются во множественном числе. Объект Документы является набором объектов Документ. Первый член набора Документы (первый Документ), обозначен как Документы (0) .

## Методы

*Метод* представляет собой вызов процедуры или функции, ассоциированной с конкретным объектом. Свойства в свою очередь являются переменными, принадлежащими объектам.

Преимущество применения объектов в том, что программисту не нужно учитывать внутренних принципов работы. Все, что необходимо знать — это имя объекта и какие свойства и методы он поддерживает.

Например, для вывода простого диалогового окна на экран можно записать оператор на VBScript (см. рис. 1.1) — листинг 1.1.

#### Листинг 1.1. Диалоговое окно

```
Set WshShell=CreateObject("Wscript.Shell")
result=WshShell.Popup("Операция завершена", 5, "Сообщение", 65)
Wscript.Echo result
```

В данном случае объектом является `Wscript`, а `Echo` — именем метода.

## Обзор систем исчислений, используемых в программировании

Программистам и системным администраторам часто приходится иметь дело с данными, записанными в различных системах исчисления: десятичной, двоичной и шестнадцатеричной, переводить данные из одной системы в другую. Рассмотрим каждую из этих систем исчисления и методику пересчета чисел между ними.

### Десятичная система исчисления

В десятичной системе исчисления знаменателем является число 10. Для задания значений используют числа от 0 до 9. Цифры, суммируясь, умножаются справа налево на увеличивающуюся степень десяти  $10^0, 10^1, 10^2, \dots, 10^n$ .

Так, число 856 представлено как  $8 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 = 8 \cdot 100 + 5 \cdot 10 + 6 \cdot 1$ .

Десятичная система исчисления является самой удобной системой для использования в повседневной жизни, однако она малоприменима в компьютерной системе, в которой используется 0 (нет сигнала) и 1 (есть сигнал). В компьютерном мире обычно применяется двоичная система исчисления.

### Двоичная система исчисления

Знаменателем двоичной системы является число 2. Значение формируется с помощью двух цифр — нуля и единицы. Соответственно, число 100110 в десятичной системе выглядит следующим образом:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 0 + 0 + 4 + 2 + 0 = 38.$$

Переведем число 50 из десятичной в двоичную систему исчисления с помощью табл. 1.1. Найдем в ней наибольшее из меньших чисел — это число 32, которому соответствует пятый разряд. Сразу можно сказать, что двоичное число будет иметь шесть цифр, первая из которых 1. Определим остаток разности сравниваемого числа с найденным, т. е.  $50 - 32 = 18$ . Поступаем аналогично, ищем наибольшее из меньших чисел для получившегося числа — это число 16, соответствующее четвертому разряду: пятым символом двоичного числа также будет единица. Разность  $18 - 16 = 2$ .

**Таблица 1.1.** Пересчет чисел из десятичной в двоичную систему исчисления

Разряд	10	9	8	7	6	5	4	3	2	1	0
Значение	1024	512	256	128	64	32	16	8	4	2	1
	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

Повторяя описанную последовательность действий, получаем, что ближайшее число — 2 соответствует первому разряду. Знак равенства сигнализирует об окончании процедуры перевода. Тем разрядам, которые не равны единице, присваиваются нули. Таким образом, числу 50 в десятичной системе соответствует число 110010 в двоичной системе исчисления.

## Шестнадцатеричная система исчисления

Преобразовывать числа в двоичном формате в десятичную систему и обратно — задача довольно сложная. Гораздо удобнее использовать шестнадцатеричную систему, которая имеет знаменатель 16. Для обозначения числа в этой системе исчисления используются цифры от 0 до 9 и буквы от А до F. Приведем в табл. 1.2 эквиваленты шестнадцатеричной, десятичной и двоичной систем.

**Таблица 1.2.** Соотношения между системами исчисления

Двоичная система	Шестнадцатеричная система	Десятичная система
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4

Таблица 1.2 (окончание)

Двоичная система	Шестнадцатеричная система	Десятеричная система
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Шестнадцатеричное число B35F в десятичной системе представляется как:

$$11 \cdot 16^3 + 3 \cdot 16^2 + 5 \cdot 16^1 + 15 \cdot 16^0 = 11 \cdot 4096 + 3 \cdot 256 + 5 \cdot 16 + 15 \cdot 1 = 106511.$$

Обратный перевод из десятичной в шестнадцатеричную систему исчисления осуществляется в соответствии с приведенным далее алгоритмом. Представим десятичное число 356 в шестнадцатеричной системе: в табл. 1.3 найдем ближайшее меньшее число и определим соответствующий этому числу разряд. Таким числом является 256, принадлежащее ко второму разряду.

Таблица 1.3. Пересчет чисел из 10-ной в 16-ную систему исчисления

Разряд	3	2	1	0
Значение	4096	256	16	1
	$16^3$	$16^2$	$16^1$	$16^0$

Найдем целую часть от отношения исходного числа и только что определенного значения ближайшего меньшего к исходному числу числа  $\frac{356}{256} = 1,390625$ . Целая часть равна единице. Поэтому множителем при  $16^2$  будет единица. Теперь определим произведение числа, соответствующего

второму разряду —  $1 \cdot 16^2 = 256$ . Найдем разность между первоначальным и получившимся числом:  $356 - 256 = 100$ . Теперь это число считаем исходным и действуем по описанному алгоритму: ближайшее меньшее число — 16, соответствующее первому разряду. Целая часть от деления исходного числа на определенное ближайшее меньшее число равна  $6 \left( \frac{100}{6} = 6,25 \right)$ . Числом, соответствующим первому разряду, является 96 ( $16 \cdot 6 = 96$ ). Разность между первоначальным и получившимся числом — 4 ( $100 - 96 = 4$ ). Новое исходное число, соответствующее последнему разряду — 4. Множителем при первом разряде является 4. Таким образом, десятичному числу 356 соответствует шестнадцатеричное число 164.

Однако существует проблема — как идентифицировать систему исчисления числа, если оно содержит в себе числа от 0 до 9. Было принято решение — добавить к шестнадцатеричным числам префикс 0x.

Рассмотрим методику преобразования чисел из шестнадцатеричной системы в двоичную и обратно на примере числа 0xB35F. Оно состоит из четырех частей. Используя табл. 1.2, находим соответствия каждому из элементов.

$$\begin{array}{cccc} \overbrace{1011}^B & \overbrace{0011}^3 & \overbrace{1010}^5 & \overbrace{1111}^F \\ 1011001101011111. \end{array}$$

Из двоичной в шестнадцатеричную систему перевод аналогичен. Если количество знаков в числе двоичного формата кратно четырем, то перед числом дописывается нужное количество нулей. Например, число 110011111 имеет 9 символов. Ближайшее кратное четырем число — двенадцать. В соответствии с этим правилом следует дописать три нуля и разбить получившееся число на части по четыре символа:

$$\begin{array}{ccc} \underbrace{0001}_1 & \underbrace{1001}_2 & \underbrace{1111}_3 \\ 000110011111. \end{array}$$

Первой части соответствует число 1, второй — 9, третьей — 6. Таким образом, числу 110010111 в двоичной системе исчисления соответствует 0x19F в шестнадцатеричной.

## Обзор языков программирования

Существует несколько языков для создания скриптов. Среди них в операционную систему Windows интегрированы:

- сценарии на базе командной строки (файлы с расширением bat, pif);
- Windows Scripting Host (WSH);

- Microsoft Java Script (JScript);
- Microsoft Visual Basic Script Edition (VBScript).

Есть также языки, специально разработанные для создания скриптов, такие как KIXtart, AutoIT, WinBatch.

Языки программирования можно условно разделить на компилируемые и интерпретируемые.

*Компилируемый язык* — это язык, программный код которого преобразуется в машинный до его выполнения. Такое преобразование необходимо выполнять только один раз. Его выполняет компилятор.

*Интерпретируемый язык* является противоположностью компилируемого. Каждый раз при выполнении сценария код необходимо интерпретировать, т. е. преобразовывать в машинный. Для выполнения этой задачи предназначен интерпретатор. При одновременном интерпретировании на сервере некоторого количества сценариев происходит падение скорости выполнения скрипта.

Все приведенные ранее языки программирования являются интерпретируемыми, за исключением AutoIT и WinBatch — это компилируемые языки.

## Сценарии на базе командной строки

Сценарии на базе командной строки появились в Disk Operating System (DOS) версии 1.0. Файлы сценариев представляют собой обычный текстовый файл с расширением bat. DOS была создана компанией Microsoft в 1981 году, совместно с IBM. В настоящее время сценарии на основе командной строки претерпели незначительные изменения по сравнению с первыми своими версиями. Ограниченное количество поддерживаемых функций (около 40) делает сценарии громоздкими. Работа сценария осуществляется с помощью командного интерпретатора. В Windows 9x командный интерпретатор представлен исполняемым файлом command.com, в Windows 2k — cmd.exe. Интерпретатор поддерживает внутренние и внешние команды. Команды, распознающиеся и выполняющиеся непосредственно интерпретатором, называются *внутренними*. Команды операционной системы, представляющие собой отдельные подпрограммы, называются *внешними*.

Внутренние команды интерпретатора:

AT, BREAK, CALL, CHCP, CHDIR, CLS, CONVERT, COPY, DATE, DEL, DIR, ECHO, EXIT, FINDSTR, FOR, FOTO, IF, LOADHIGH, MKDIR, PATH, PAUSE, PROMPT, REM, RENAME, RMDIR, SET, SHIFT, TIME, RYPE, VER, VERIFY, VOL и др.

Все остальные команды — внешние и представлены выполняемыми файлами. Запуск внешних команд осуществляется с помощью команды START.

Команды для работы с файловой системой:

CD, ATTRIB, COPY, XCOPY, DIR, MKDIR, RMDIR, DEL, DELTREE, REN, MOVE, SUBST, VOL, LABEL.

Команды для работы с сетью: NET, PING, IPCONFIG.

Сценарии на базе командной строки обладают ограниченными возможностями и с их помощью очень трудно реализовать необходимый функционал.

Пакет поддержки сценариев на базе командной строки встраивается в ОС Microsoft Windows. Его цена входит в стоимость операционной системы.

## WinBatch

В настоящее время компанией Wilson WindowsWare (<http://www.winbatch.com>) создана программа WinBatch, которая является расширенной версией сценариев на основе командной строки. Данная программа поддерживает около 400 функций, которые можно разбить логически на следующие группы:

- системное администрирование — управление учетными записями домена, восстановление и архивирование ОС, установка ПО, управление планировщиком, управление панелью управления Windows, управление почтой;
- сетевое администрирование — управление учетными записями, определение статуса сервера, архивирование системы, установка принтеров, запуск и остановка сервисов, администрирование ADSI, удаленная установка программ;
- управление данными — манипуляции с файлами, создание PDF-файлов, чтение VAR-кода с COM-порта, поиск и обработка данных, ведение журнала.

WinBatch — платная программа, поставляется в двух вариантах. Варианты отличаются друг от друга поддержкой сетевых технологий и возможностью компиляции написанного скрипта.

## JavaScript

JavaScript — интерпретируемый язык. Интерпретатор JScript интегрирован в интернет-браузеры (Internet Explorer, Opera, Netscape Navigator и др.).

Достоинство JScript — интеграция в ОС и простота программирования. Для создания сценария достаточно иметь любой текстовый редактор. JScript не обладает средствами поддержки объектно-ориентированного программирования и является объектно-базированным языком.

Основной недостаток JScript — отсутствие доступа к файловой системе компьютера. Как следствие, с помощью JScript невозможно получить доступ