

Инструменты для автоматизации работы в Microsoft Windows

Шестаков А.П.

Требования к средствам автоматизации

- работа в разных версиях операционной системы (желательно во всех) без установки какого-либо дополнительного программного обеспечения;
- интеграция с командной строкой (непосредственное выполнение вводимых с клавиатуры команд);
- согласованный и непротиворечивый синтаксис команд и утилит;
- наличие подробной встроенной справки по командам с примерами использования;
- возможность выполнения сценариев, составленных на простом для изучения языке;
- возможность использования всех технологий, поддерживаемых операционной системой.

Оболочка командной строки

- реализация директивной парадигмы программирования;
- выполнение в режиме построчной интерпретации;
- наличие управляющих конструкций;
- поддержка нескольких видов циклов (в том числе специальных циклов для обработки текстовых файлов);
- наличие оператора присваивания (установки значения переменной);
- возможность использования внешних программ (команд) операционной системы в качестве операторов и обработки их кодов возврата;
- наличие нетипизированных переменных, которые декларируются первым упоминанием (значения переменных могут интерпретироваться как числа и использоваться в выражениях целочисленной арифметики).

Сервер сценариев Windows Script Host (WSH)

- VBScript и JScript – это полноценные алгоритмические языки, имеющие встроенные функции и методы для обработки символьных строк, выполнения математических операций, обработки исключительных ситуаций и т.д.; кроме того, для написания сценариев WSH может использоваться любой другой язык сценариев (например, широко распространенный в Unix-системах Perl), для которого установлен соответствующий модуль поддержки.
- WSH поддерживает несколько собственных объектов, свойства и методы которых позволяют решать некоторые часто возникающие повседневные задачи администратора операционной системы: работа с сетевыми ресурсами, переменными среды, системным реестром, ярлыками и специальными папками Windows, запуск и управление работой других приложений.
- из сценариев WSH можно обращаться к службам любых приложений-серверов автоматизации (например, программ из пакета Microsoft Office), которые регистрируют в операционной системе свои объекты.
- сценарии WSH позволяют работать с объектами информационной модели Windows Management Instrumentation (WMI), обеспечивающей программный интерфейс управления всеми компонентами операционной модели, а также с объектами службы каталогов Active Directory Service Interface.

Командная оболочка Microsoft PowerShell

- применение командной строки в качестве основного интерфейса администрирования;
- реализация модели ObjectFlow (элементом обмена информации является объект);
- переработка существующих команд, утилит и оболочки;
- интеграция командной строки, объектов COM, WMI и .NET;
- работа с произвольными источниками данных в командной строке по принципу файловой системы.

PowerShell одновременно является и оболочкой командной строки (пользователь работает в интерактивном режиме) и средой выполнения сценариев, которые пишутся на специальном языке PowerShell.

ОБОЛОЧКА КОМАНДНОЙ СТРОКИ CMD.EXE

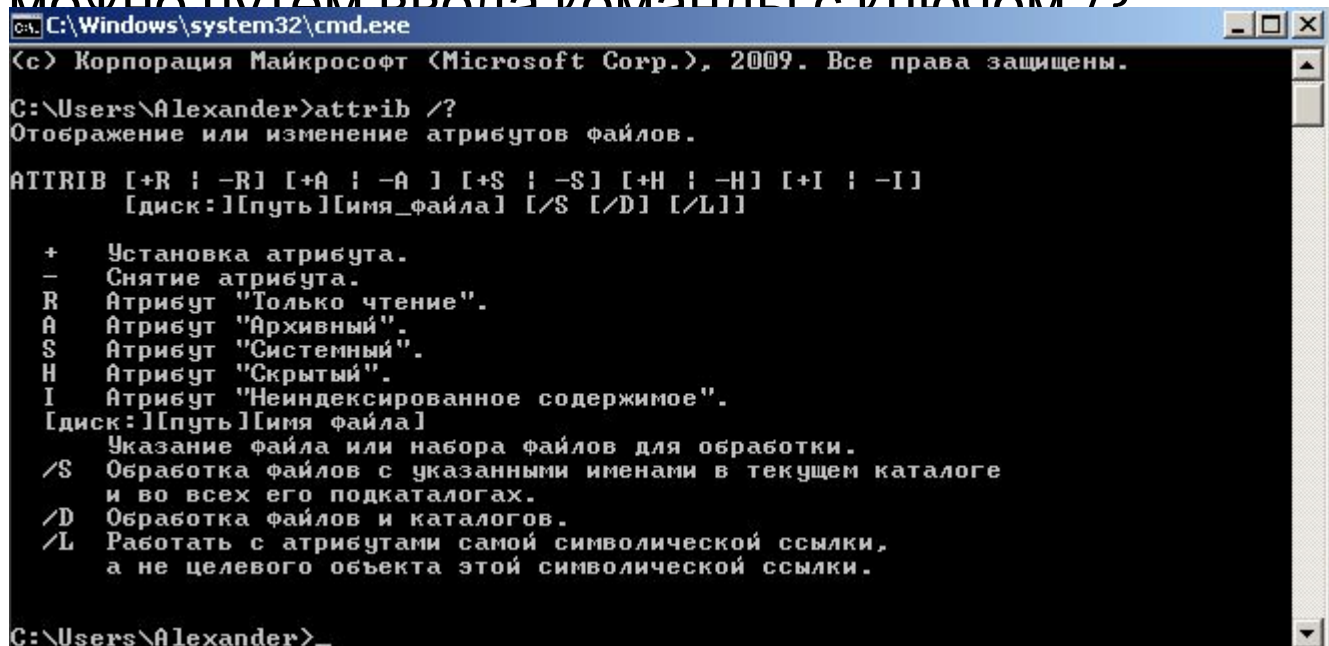
Внутренние и внешние команды

Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором — такие команды называются внутренними (например, COPY или DIR) Другие команды операционной системы представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и Cmd.exe, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, MORE или XCOPY).

Для того, чтобы выполнить команду, после *приглашения командной строки* (например, C:\>) вводится имя этой команды (регистр не важен), ее параметры и ключи (если они необходимы) и нажимается клавиша <Enter>.

Справка по командам

Многие команды Windows имеют большое количество дополнительных параметров и ключей, запомнить которые зачастую бывает трудно. Большинство команд снабжено встроенной справкой, в которой кратко описываются назначение и синтаксис данной команды. Получить доступ к такой справке можно путем ввода команды с ключом /?



```
C:\Windows\system32\cmd.exe
(C) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\Alexander>attrib /?
Отображение или изменение атрибутов файлов.

ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [+I | -I]
      [диск:] [путь] [имя_файла] [/S [/D] [/L]]

+      Установка атрибута.
-      Снятие атрибута.
R      Атрибут "Только чтение".
A      Атрибут "Архивный".
S      Атрибут "Системный".
H      Атрибут "Скрытый".
I      Атрибут "Неиндексированное содержимое".
[диск:] [путь] [имя файла]
      Указание файла или набора файлов для обработки.
/S     Обработка файлов с указанными именами в текущем каталоге
      и во всех его подкаталогах.
/D     Обработка файлов и каталогов.
/L     Работать с атрибутами самой символической ссылки,
      а не целевого объекта этой символической ссылки.

C:\Users\Alexander>
```


Для некоторых команд текст встроенной справки может быть довольно большим и не уместиться на одном экране. В этом случае помощь можно выводить последовательно по одному экрану с помощью команды MORE и символа конвейеризации |, например:

```
XCOPY /? | MORE
```

В этом случае после заполнения очередного экрана вывод помощи будет прерываться до нажатия любой клавиши. Кроме того, используя символы перенаправления вывода > и >>, можно текст, выводимый на экран, направить в текстовый файл для дальнейшего просмотра. Например, для вывода текста справки к команде XCOPY в текстовый файл xcopy.txt, используется следующая команда:

```
XCOPY /? > XCOPY.TXT
```

Перечень основных команд

- **ATTRIB** Просмотр и модификация свойств файлов.
- **CALL** Вызывает один пакетный файл из другого, а также может передавать входные аргументы.
- **CHDIR** Выводит название или перемещает к другой папке.
- **CHKDSK** Диагностика накопителя на ошибки.
- **COLOR** Изменяет и устанавливает основной фон окна и самих шрифтов.
- **COMP** Показывает различия и сравнивает содержимое двух файлов.
- **COPY** Создаёт копию файла или файлов и помещает их в указанное место.
- **DATE** Показывает или устанавливает текущую дату.
- **DEL** Уничтожает один или несколько файлов сразу.
- **DIR** Показывает имена файлов и папок с их датой создания, находящихся в текущей или указанной в параметрах папки .
- **ECHO** Выводит текстовую информацию и меняет режим отображения команд на экране.

Перечень основных команд

- **ERASE** Уничтожает файл или файлы.
- **EXIT** Прекращает работу программы командной строки
- **FIND** Осуществляет поиск текстовой строки в файлах либо в одном файле.
- **FORMAT** Форматирование накопителя для работы с Windows.
- **HELP** Выводит все данные о существующих командах Windows.
- **LABEL** Создает, модифицирует и уничтожает метки тома для накопителей.
- **MKDIR** Создает пустой каталог.
- **MODE** Настраивает системные устройства.
- **MORE** Последовательно выводит информацию по блокам размером в один экран
- **MOVE** Перемещает файлы из одного места в другое.
- **PATH** Выводит или задает полный путь к исполняемым файлам.
- **PAUSE** Прекращает выполнение команд командной строки и выводит информирующий текст.
- **PRINT** Выводит на печать содержимое текстового файла.

Перечень основных команд

- **REN** Меняет название как файлов так и папок.
- **REPLACE** Меняет файлы местами.
- **RMDIR** Уничтожает каталог.
- **TIME** Устанавливает и выводит системное время.
- **TREE** Выводит директории накопителя в удобном наглядном виде.
- **TYPE** Выводит содержимое текстовых файлов.
- **VER** Выводит краткие сведения о версии Windows.
- **VERIFY** Проверяет на ошибки записи файла на накопителе.
- **XCOPY** Создает копию файлов.

Условное выполнение и группировка команд

В командной строке Windows можно использовать специальные символы, которые позволяют вводить несколько команд одновременно и управлять работой команд в зависимости от результатов их выполнения. С помощью таких символов условной обработки можно содержание небольшого пакетного файла записать в одной строке и выполнить полученную составную команду.

- & - можно разделить несколько утилит в одной командной строке, при этом они будут выполняться друг за другом
- ^ - позволяет использовать командные символы как текст, то есть при этом происходит игнорирование значения специальных символов
- && - запускает команду, стоящую за ним в командной строке, только в том случае, если команда, стоящая перед, была выполнена успешно
- || - запускает команду, стоящую за этими символами, только в том случае, если команда, идущая перед ними, не была успешно выполнена

Язык оболочки командной строки

Язык оболочки командной строки (*shell language*) в *Windows* реализован в виде командных (или пакетных) файлов. Командный *файл* в *Windows* — это обычный *текстовый файл* с расширением `bat` или `cmd`, в котором записаны допустимые команды операционной системы (как внешние, так и внутренние), а также некоторые дополнительные инструкции и ключевые слова, придающие командным файлам некоторое сходство с алгоритмическими языками программирования.

Использование параметров командной строки

При запуске пакетных файлов в командной строке можно указывать произвольное число параметров, значения которых можно использовать внутри файла. Это позволяет, например, применять один и тот же командный файл для выполнения команд с различными параметрами.

Для доступа из командного файла к параметрам командной строки применяются символы %0, %1, ..., %9 или %*. При этом вместо %0 подставляется имя выполняемого пакетного файла, вместо %1, %2, ..., %9 — значения первых девяти параметров командной строки соответственно, а вместо %* — все аргументы. Если в командной строке при вызове пакетного файла задано меньше девяти параметров, то "лишние" переменные из %1 – %9 замещаются пустыми строками.

Переменные окружения

Внутри командных файлов можно работать с так называемыми **переменными среды** (или **переменными окружения**), каждая из которых хранится в оперативной памяти, имеет свое уникальное имя, а ее значением является строка. Стандартные переменные среды автоматически инициализируются в процессе загрузки операционной системы. Такими переменными являются, например, WINDIR, которая определяет расположение каталога Windows, TEMP, которая определяет путь к каталогу для хранения временных файлов Windows или PATH, в которой хранится системный путь (путь поиска), то есть список каталогов, в которых система должна искать выполняемые файлы или файлы совместного доступа (например, динамические библиотеки). Кроме того, в командных файлах с помощью команды SET можно объявлять собственные переменные среды.

Для получения значения определенной переменной среды нужно имя этой переменной заключить в символы %.

Вызов внешних командных файлов

Для того, чтобы вызвать внешний командный файл с последующим возвратом в первоначальный файл, нужно использовать специальную команду

CALL файл

Операторы перехода

Командный файл может содержать метки и команды GOTO перехода к этим меткам. Любая строка, начинающаяся с двоеточия :, воспринимается при обработке командного файла как метка. Имя метки задается набором символов, следующих за двоеточием до первого пробела или конца строки.

Операторы условия

С помощью команды IF ... ELSE (ключевое слово ELSE может отсутствовать) в пакетных файлах можно выполнять обработку условий нескольких типов. При этом если заданное после IF условное выражение принимает истинное значение, система выполняет следующую за условием команду (или несколько команд, заключенных в круглые скобки), в противном случае выполняется команда (или несколько команд в скобках), следующие за ключевым словом ELSE.

Операторы условия

1. Первый тип условия используется обычно для проверки значения переменной. Для этого применяются два варианта синтаксиса команды IF:

- **IF [NOT] строка1==строка2 команда1 [ELSE команда2]**

(квадратные скобки указывают на необязательность заключенных в них параметров) или

- **IF [/I] [NOT] строка1 оператор_сравнения строка2 команда**

Ключ /I, если он указан, задает сравнение текстовых строк без учета регистра.

Операторы сравнения в IF

Оператор	Значение
EQ	Равно
NEQ	Не равно
LSS	Меньше
LEQ	Меньше или равно
GTR	Больше
GEQ	Больше или равно

2. проверка существования заданного файла:

- **IF [NOT] EXIST файл команда1 [ELSE команда2]**

3. проверка наличия в системе определенной переменной среды:

- **IF DEFINED переменная команда1 [ELSE команда2]**

4. проверка кода завершения (кода выхода) предыдущей команды:

- **IF [NOT] ERRORLEVEL число команда1 [ELSE команда2]**

Организация циклов

В командных файлах для организации циклов используются несколько разновидностей оператора FOR, которые обеспечивают следующие функции:

- выполнение заданной команды для всех элементов указанного множества;
- выполнение заданной команды для всех подходящих имен файлов;
- выполнение заданной команды для всех подходящих имен каталогов;
- выполнение заданной команды для определенного каталога, а также всех его подкаталогов;
- получение последовательности чисел с заданными началом, концом и шагом приращения;
- чтение и обработка строк из текстового файла;
- обработка строк вывода определенной команды.

- **FOR %%переменная IN (множество) DO команда [параметры]**

В случае, если набор содержит подстановочные знаки, то команда выполняется для всех подходящих имен каталогов, а не имен файлов.

- **FOR /D %%переменная IN (набор) DO команда [параметры]**

Ключ /L позволяет реализовать с помощью команды FOR арифметический цикл, в этом случае синтаксис имеет следующий вид:

- **FOR /L %%переменная IN (начало, шаг, конец) DO команда [параметры]**

Самые мощные возможности имеет команда FOR с ключом /F:

- **FOR /F ["ключи"] %%переменная IN (набор) DO команда [параметры]**

Здесь параметр набор содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбиении его на отдельные строки текста и выделении из каждой строки заданного числа подстрок. Затем найденная подстрока используется в качестве значения переменной при выполнении основного тела цикла (заданной команды).

По умолчанию ключ /F выделяет из каждой строки файла первое слово, очищенное от окружающих его пробелов. Пустые строки в файле пропускаются. Необязательный параметр "ключи" служит для переопределения заданных по умолчанию правил обработки строк.

Ключ	Описание
<i>EOL=C</i>	Определение символа комментариев в начале строки (допускается задание только одного символа)
<i>SKIP=N</i>	Число пропускаемых при обработке строк в начале файла
<i>DELIMS=XXX</i>	Определение набора разделителей для замены заданных по умолчанию пробела и знака табуляции
<i>TOKENS=X, Y, M-N</i>	Определение номеров подстрок, выделяемых из каждой строки файла и передаваемых для выполнения в тело цикла