



# Содержание

<b>Благодарности</b> .....	8
<b>Предисловие</b> .....	9
<b>Введение</b> .....	10
Для кого эта книга? .....	10
Почему PowerShell? .....	10
Какие задачи может решить сценарий входа в сеть? .....	11
Каков основной принцип работы сценария? .....	11
Какова структура книги? .....	11
Условные обозначения .....	13
<b>Глава 1. Установка и настройка Windows PowerShell</b> .....	14
Установка PowerShell .....	14
Среда разработки PowerShell .....	15
Панель Commands .....	16
Быстрое создание сценариев .....	16
Усовершенствованная справка по командлетам .....	18
Технология IntelliSense .....	18
Поддержка шаблонов сценариев .....	19
Первый запуск PowerShell .....	20
Запуск сценариев PowerShell .....	20
Документация по PowerShell .....	21
<b>Глава 2. Управление каталогом Active Directory</b> .....	23
Объекты Active Directory .....	23
Поддерживаемые форматы имен .....	23
Виды объектов .....	24
Идентификация объектов .....	25
Определение имени домена .....	27
Действия над объектами .....	30
Создание объекта .....	30
Копирование объекта .....	31
Удаление объекта .....	32
Перемещение объекта .....	33
Изменение свойств объекта .....	34
Переименование объекта .....	35

Поиск объектов .....	36
Получение доступа к контейнеру поиска .....	36
Формирование атрибутов поиска .....	37
Просмотр атрибутов каталога Active Directory .....	42
Active Directory Users and Computer .....	43
Microsoft ADSI Edit .....	44
Microsoft Active Directory Explorer .....	44
Softerra LDAP Browser .....	45
<b>Глава 3. Архитектура сценария .....</b>	<b>49</b>
Модульный принцип построения сценария .....	49
Компоненты сценария .....	51
Конфигурационный файл в формате XML .....	53
Правила синтаксиса XML-файла .....	53
Структура XML-файла .....	54
Сценарий на PowerShell .....	55
Работа с XML-файлом .....	55
Работа с переменными среды .....	56
Определение имени домена .....	56
Определение имени компьютера .....	57
Определение названия сценария .....	57
Определение имени пользователя в сети .....	58
Определение полного имени сотрудника .....	58
Определение названия подразделения сотрудника .....	59
Компиляция .NET-кода на лету .....	59
Компиляция C# листинга .....	60
Компиляция VB.NET листинга .....	61
Ведение журнала событий .....	62
<b>Глава 4. Задачи, решаемые сценарием</b>	
<b>регистрации .....</b>	<b>64</b>
Анализ возможностей сценария регистрации .....	64
Задача инвентаризации .....	65
Управление сетевыми ресурсами .....	65
Олицетворение в IIS .....	66
Использование IIS в сценариях регистрации .....	68
Создание журнала событий .....	71
<b>Глава 5. Задача инвентаризации .....</b>	<b>73</b>
Классификация собираемой информации .....	73
Персональная информация сотрудника .....	74
Аппаратная конфигурация рабочей станции .....	76
Установленное программное обеспечение .....	79
Чтение конфигурационного файла .....	85

Экспорт собранных данных .....	86
Экспорт в HTML-файл .....	86
Экспорт в XML-файл.....	88
Экспорт в SQL-таблицу .....	91
<b>Глава 6. Управление сетевыми ресурсами .....</b>	<b>95</b>
Типы сетевых ресурсов .....	95
Анатомия группы безопасности .....	95
Характеристики подключения ресурса .....	96
Идентификаторы групп безопасности .....	97
Работа с псевдонимами .....	99
<b>Глава 7. Управление сетевыми дисками.....</b>	<b>102</b>
Управление сетевыми дисками в Active Directory .....	102
Сценарий подключения сетевых дисков .....	103
Алгоритм работы сценария .....	104
Чтение конфигурационного файла .....	104
Определение характеристик для подключения дисков .....	105
Отключение подключенных сетевых дисков .....	107
Подключение сетевых дисков пользователю .....	107
Использование команды net use .....	108
Использование COM-объекта .....	108
Использование API-функции .....	108
Изменение описания дисков в папке Мой компьютер.....	112
<b>Глава 8. Управление сетевыми принтерами .....</b>	<b>113</b>
Управление сетевыми принтерами в Active Directory .....	113
Публикация принтеров в Active Directory.....	113
Формирование окружения в Active Directory .....	114
Идентификация принтеров .....	114
Иерархическая структура для принтера в Active Directory ....	115
Создание иерархической структуры для принтера .....	117
Настройка безопасности принтера .....	119
Требования к сценарию с точки зрения безопасности .....	120
Сценарий подключения сетевых принтеров .....	121
Алгоритм работы сценария .....	121
Чтение конфигурационного файла .....	122
Формирование списка сетевых принтеров.....	123
Определение принтера по умолчанию.....	123
Удаление всех подключенных сетевых принтеров .....	124
Подключение принтеров .....	125
Назначение принтера по умолчанию .....	127
<b>Глава 9. Подключение баз 1С.....</b>	<b>129</b>
Архитектура файловой структуры 1С клиента .....	129

Структура файла ibases.v8i.....	129
Структура файла 1CEStart.cfg .....	131
Структура файла 1cv8strt.pfl.....	131
Сценарий подключения баз 1C.....	132
Алгоритм работы сценария .....	133
Чтение характеристик конфигурационного файла .....	133
Определение характеристик для подключения баз 1C.....	134
Очистка существующего списка баз.....	136

## **Глава 10. Централизованное управление**

<b>ярылками .....</b>	<b>137</b>
Архитектура решения .....	137
Архитектура клиентской части.....	138
Архитектура серверной части.....	139
Архитектура реестра на компьютере клиента .....	140
Сценарий регистрации пользователя в сети .....	142
Разработка веб-приложения .....	143
Создание и настройка проекта в Visual Studio.....	143
Алгоритм работы веб-страницы .....	145
Определение значений внешних параметров.....	145
Определение имени компьютера .....	147
Определение имени пользователя .....	147
Определение имени домена .....	147
Чтение данных из конфигурационного файла.....	149
Чтение данных из каталога Active Directory .....	150
Подготовка данных для записи в реестр.....	152
Управление удаленным реестром .....	153

## **Глава 11. Внедрение сценария .....**

Сопоставление сценария учетной записи пользователю .....	159
Политика безопасности как средство запуска сценария .....	159
Управление запуском сценария через свойства учетной записи пользователя .....	162
Формирование точки входа в сценарий.....	163
Командный файл.....	163
VBScript-сценарий .....	165
Групповое изменение сценария регистрации в Active Directory ...	169

## **Глава 12. Распределенная файловая система .....**

Файловая система обмена данными .....	170
Требования к файловой системе обмена.....	170
Архитектура файловой системы .....	171
Масштабируемость файловой системы .....	172
Защита файловой системы от изменений пользователем .....	172
Управление видимостью данных в файловой системе.....	173

<b>Глава 13. Управление безопасностью</b>	175
Программное управление безопасностью	179
Отключение наследования	180
Удаление ACE-элементов	181
Создание ACE-элементов	184
Хранилище настроек безопасности	185
<b>Глава 14. Управление ABE</b>	187
Что такое ABE?	187
Управление ABE в графическом интерфейсе	187
Программное управление ABE	188
Управление ABE в PowerShell	192
Создание удаленного сеанса с сервером	192
Интеграция кода в PowerShell	194
<b>Приложения</b>	195
Приложение 1. Синтаксис командного файла	195
Команда call	195
Команда echo	196
Команда endlocal	196
Команда for	197
Команда goto	197
Команда if	198
Команда setlocal	199
Команда pause	200
Команда rem	200
Команда shift	201
Приложение 2. Обязательные атрибуты объектов Active Directory	201
Приложение 3. Основные объекты Active Directory	203
Пользователь	203
Группа безопасности	210
Контейнер	212
Компьютер	212
Приложение 4. PowerShell и другие языки программирования	217
Компиляция кода на DOT.NET в PowerShell	218
Передача параметров из PowerShell в вставку DOT.NET	219
Вызов API-функций из DOT.NET-вставки в PowerShell	221



*С любовью к моему сыну Матвею Коробко и жене Елене Ходовой*

## **Благодарности**

Эта книга – результат почти десятилетнего труда. Я хочу выразить свою благодарность Филиппу Зыкову, Александру Алексеенко, Денису Шестопалову, Юрию Тачилину, принимавшим активное участие в создании данной книги.

Отдельное спасибо Галине Положевец – главному редактору журнала «Системный администратор». Благодаря ей эта книга вышла в свет. О



## Предисловие

Работа системного администратора и специалиста системной поддержки во многих случаях сводится к выполнению рутинных действий, которые ежедневно отнимают много рабочего времени. Ошибка специалиста системной поддержки в большинстве случаев увеличит время обработки заявки и вызовет недовольство сотрудника предприятия. Ошибка системного администратора может иметь гораздо более серьезные последствия. Иногда она может быть фатальной.

С появлением PowerShell множество технологий получило вторую жизнь. Не являются исключением и хорошо забытые сценарии регистрации в сети. Используя различные приемы администрирования, знания по внутреннему устройству реестра и т. д., можно создать интеллектуальный сценарий, который позволит восстановить рабочее окружение пользователя независимо от его местонахождения на предприятии. Обеспечение одинакового набора сетевых дисков с различным содержанием для каждого сотрудника, поддержание актуального списка сетевых принтеров и актуального списка сетевых баз 1С – это лишь начало списка тех возможностей, которые может предоставить сценарий, значительно снизив при этом нагрузку на службу специалистов системной поддержки.

Прочитав эту книгу, читатель сможет самостоятельно разработать сценарий регистрации в сети, обеспечив необходимый функционал. Его сложность зависит от потребностей администратора и уровня знаний PowerShell и DOT.NET.

Автор надеется, что данная книга будет полезна не только системным администраторам, специалистам системной поддержки, но и программистам, разрабатывающим серьезные системы.



# Введение

## Для кого эта книга?

Книга, которую вы держите в руках, поможет вам разработать собственный сценарий регистрации пользователей в сети. Для чтения книги вам не требуется высокого уровня знаний PowerShell – достаточно знать основы этого языка и обладать базовыми знаниями по администрированию сетей. Чтобы помочь читателю ориентироваться в книге, в начале каждой главы приведено ее краткое содержание.

## Почему PowerShell?

Microsoft Windows PowerShell – это бесплатная оболочка командной строки и язык создания сценариев для операционных систем Windows. Первая версия PowerShell появилась 14 ноября 2006 года. Причина появления нового языка (новой оболочки) программирования проста. Microsoft постепенно приводит все языки программирования к общему знаменателю, а именно к библиотекам .NET Framework. Все компилируемые языки, объединенные в Visual Studio, поддерживают .NET Framework и традиционно COM-объекты. Все интерпретируемые языки, такие как VBScript и Jscript, поддерживают только COM-объекты, поэтому значительная часть функционала оказывается нереализованной. Именно поэтому и появился новый интерпретируемый язык, который поддерживает работу как с COM-объектами, так и с .NET Framework.

PowerShell, в отличие от KIXtart (<http://kixtart.org>), не является специально разработанным языком для создания сценариев регистрации входа в сети, и некоторые функции можно реализовать только с помощью функций библиотек .NET Framework.

KIXtart, активно использовавшийся до настоящего времени для создания сценариев входа в сеть, изжил себя. Новые версии содержат лишь обновления к вышедшим операционным системам, в нем не реализована поддержка NET Framework.



## Какие задачи может решить сценарий входа в сеть?

С помощью сценария регистрации пользователей в сети можно решить множество различных задач:

- ❑ *инвентаризация*. Включает в себя сбор информации о регистрирующемся в сети пользователе, рабочей станции и экспорте данных в файл отчета или базу данных;
- ❑ *подключение сетевых ресурсов*. Сетевые ресурсы – диски, принтеры, приложения, подключаются к компьютеру пользователя в автоматическом режиме. Каждому пользователю сопоставляется индивидуальный набор ресурсов;
- ❑ *подключение баз 1С 8х*. Список баз формируется в автоматическом режиме индивидуально для каждого пользователя на основе заданных параметров, при этом список подключенных баз недоступен пользователю для изменений;
- ❑ *настройка рабочей станции*. Объем изменяемых параметров зависит от корпоративной политики предприятия.

## Каков основной принцип работы сценария?

Работа сценария основана на работе со свойствами объектов каталога Active Directory. Членство учетной записи пользователя в группе безопасности дает право на использование ресурса. Сценарий лишь подключает этот ресурс к компьютеру пользователя, используя различные точки монтирования. Правила, по которым осуществляется подключение, описаны в свойствах объектов (группах безопасности) каталога Active Directory.

## Какова структура книги?

Книга состоит из 14 глав.

*Глава 1 «Установка и настройка Windows PowerShell»* познакомит читателя с описанием процедуры установки, удаления Windows PowerShell. Рассказывается об особенностях настройки оболочки для выполнения настройки.

*Глава 2 «Управление каталогом Active Directory»* целиком посвящена описанию основных объектов каталога Active Directory и действиям, выполняемым над ними. Приведены характерные примеры.

В главе 3 «Архитектура сценария» рассмотрена архитектура сценария, построенного по модульному принципу. Описаны общая структура сценария, его компоненты. Особое внимание уделено механизму настройки скриптов с помощью конфигурационного файла.

Глава 4 «Задачи, решаемые сценарием регистрации» позволит узнать, какие задачи решаются с помощью сценария регистрации пользователей сети. Рассказано о журналировании работы сценариев.

Глава 5 «Задача инвентаризации» расскажет читателю о процедуре сбора, обработки и сохранения данных, касаемых не только аппаратной конфигурации рабочей станции, с которой совершается вход в сеть, но и информации об учетной записи пользователя, характеристики установленного программного обеспечения и др.

В главе 6 «Управление сетевых ресурсов» обсуждаются основные принципы, заложенные в сценарий подключения сетевых ресурсов независимо от его типа. Не важно, диск это, принтер, ярлык или какой-либо другой ресурс.

Глава 7 «Управление сетевыми принтерами» включает в себя не только описание методики подключения сетевых дисков на основе данных, хранящихся в каталоге Active Directory, но и информацию о том, как изменить описание диска в папке *Мой компьютер*.

Глава 8 «Управление сетевыми принтерами». В этой главе рассказано о подключении сетевых принтеров с помощью сценария регистрации в сети. Особое внимание уделено создаваемой структуре данных в каталоге Active Directory, обеспечивающей работу скрипта.

Глава 9 «Подключение баз 1С». В этой главе описан механизм динамического подключения баз данных 1С. Рассказано о внутренней файловой структуре создаваемых файлов и их содержании на рабочей станции пользователя.

Глава 10 «Централизованное управление ярлыками». В этой главе подробно описана процедура разработки веб-сайта в Visual Studio, создающего ярлыки в папке *Мой компьютер* в зависимости от членства учетной записи в соответствующей группе безопасности.

Глава 11 «Внедрение сценария». В этой главе описано, как правильно внедрить сценарий регистрации пользователей в сеть и избежать досадных ошибок. Рассказано о сценариях, позволяющих быстро внедрить сценарий в эксплуатацию.

Глава 12 «Распределенная файловая система». В этой главе рассмотрены архитектура и правила построения распределенной файловой системы, управление которой осуществляется с помощью сценария регистрации пользователей в сети.

*Глава 13 «Управление безопасностью».* В этой главе рассказано об основных элементах, из которых складывается управление файловой системой. На основе изложенных принципов работает автоматическая система управления распределенной файловой системы, ресурсы которой подключает сценарий регистрации пользователей в сети по оговоренным правилам.

*Глава 14 «Управление ABE».* В этой главе рассказано, как с помощью Access Based Enumeration (ABE) в сетевой папке скрыть от глаз пользователя данные, на которые он не имеет прав доступа. Рассмотрено как интерфейсное, так и программное решение данной задачи.

Помимо основного материала, в книге читатель найдет ряд приложений, в которых собрана различная справочная информация, которая поможет успешно создать сценарий регистрации пользователей в сети.

## Условные обозначения

В некоторых случаях в сценариях должны присутствовать длинные строки, которые не удастся разместить в строку в печатном издании. Такие строки в листингах разбиты на две или несколько строк, а в начале перенесенной строки стоит знак ¶:

---

Очень длинная строка, которую никак невозможно разместить в одной  
¶ строке книги, но в сценарии нельзя разбивать

---



# Глава 1

## УСТАНОВКА И НАСТРОЙКА WINDOWS POWERSHELL

*В этой главе кратко описывается процедура установки и удаления Windows PowerShell. Рассказывается об особенностях настройки оболочки для выполнения настройки.*

### Установка PowerShell

Для каждой версии операционной системы, каждой платформы существует индивидуальная сборка оболочки PowerShell. Дополнительно почти для каждой ее версии присутствует русификация либо в виде локализованной версии, либо в виде отдельного многоязыкового пакета (Multilingual User Interface – MUI). Подробно обо всех возможных вариантах – в табл. 1.1.

**Таблица 1.1. Дистрибутивы PowerShell**

Операционная система	Платформа	Статья KB	Язык	Файл
Windows XP	x86	926139	Англ.	WindowsXP-KB926139-x86-ENU.exe
			Рус.	WindowsXP-KB926139-x86-RUS.exe
		926141	MUI	WindowsXP-KB926141-v2-x86-ENU.exe
	x64	926140	Англ.	WindowsXP-KB926140-x64-ENU.exe
			Рус.	WindowsXP-KB926140-x64-RUS.exe
		926141	MUI	WindowsServer2003.WindowsXP-KB926141-x64-ENU.exe
Windows Vista	x86	928439	–	Windows6.0-KB928439-x86.msu
	x64			Windows6.0-KB928439-x64.msu
Windows 2003	x86	926139	Англ.	WindowsServer2003-KB926139-x86-ENU.exe
			Рус.	WindowsServer2003-KB926139-x86-RUS.exe
		926141	MUI	WindowsServer2003-KB926141-x86-ENU.exe

**Таблица 1.1. Дистрибутивы PowerShell (окончание)**

Операционная система	Платформа	Статья KB	Язык	Файл
	x64	926139	Англ.	WindowsServer2003-KB926139-x64-ENU.exe
			Рус.	WindowsServer2003-KB926139-x64-RUS.exe
		926141	MUI	WindowsServer2003.WindowsXP-KB926141-x64-ENU.exe
Windows 7	x86, x64	Интегрирован в операционную систему		
Windows 8	x86, x64	Интегрирован в операционную систему		

В зависимости от используемой версии операционной системы необходимо установить соответствующее обновление (см. табл. 1.1). Для операционной системы Windows 7 или Windows 8 – выбрать соответствующий компонент из стандартного комплекта операционной системы, если он не установлен.

Windows PowerShell, любой из выбранных версий, устанавливается в папку %SystemRoot%\System32\WindowsPowerShell\v1.0. Для запуска оболочки используется файл powershell.exe.

## Среда разработки PowerShell

Начиная с Windows PowerShell CTP 2.0, в комплект дистрибутива включено приложение для разработки Windows PowerShell Integrated Script Environment (ISE), которое запускается с помощью файла powershell\_ise.exe.

В новой, третьей версии Windows PowerShell оболочка претерпела значительные изменения. Цель, поставленная разработчиками, – сделать интерфейс максимально удобным для разработчиков, как, например, в Visual Studio, – успешно достигнута. Рассмотрим основные нововведения:

- ❑ панель **Commands** (рис. 1.1). Предназначена для оперативного получения справочной информации по командам и быстрому их созданию;
- ❑ поддержка технологии **IntelliSense**, позволяющая выводить подсказку в момент написания сценария в режиме реального времени. Именно эта система дает возможность в Visual Studio быстро и просто создавать сценарии;
- ❑ поддержка шаблонов сценариев (**snippet**).

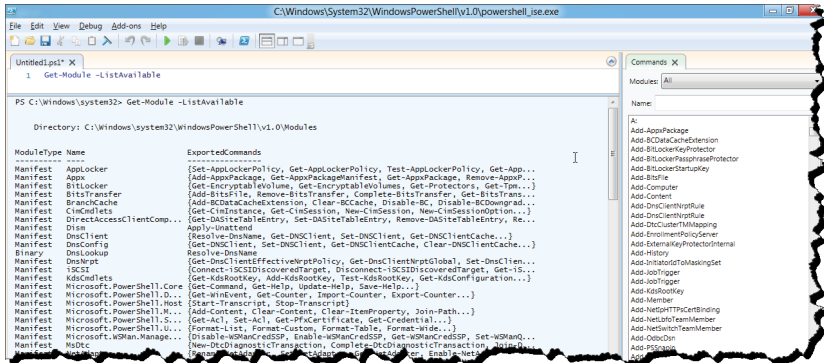


Рис. 1.1. Внешний вид Windows PowerShell ISE 3.0

## Панель **Commands**

К трем панелям, имевшимся в предыдущей версии PowerShell, добавилась вертикальная панель **Commands**, расположившаяся с правой стороны относительно трех ранее существовавших в предыдущей версии горизонтальных панелей. В ней отображается список доступных командлетов, сгруппированных по модулям.

Непосредственно из самой панели с командлетами можно выполнять следующие действия:

- ❑ **Run (запустить)**. Для большинства командлетов реализована возможность определения параметров запуска прямо в этой панели. Обязательные параметры отмечены звездочкой;
- ❑ **Insert (поместить)**. Помещает выбранный командлет в окно консоли. Причины, по которым оно не помещается в панель, где ранее был установлен курсор, или непосредственно в сценарий, не понятны;
- ❑ **Copy (скопировать)**. Помещает командлет в буфер обмена.

Здесь же сосредоточены элементы управления отображения самой панели.

## Быстрое создание сценариев

Основное назначение панели – упростить разработку новых сценариев. Рассмотрим простую задачу: необходимо вывести список всех файлов с расширением EXE, находящихся в папке c:\windows.

Очевидно, что для ее решения используется команда `dir` или соответствующий этому псевдониму командлет `Get-ChildItem`.

Если известно, к какому модулю относится командлет, то необходимо выбрать нужный модуль (1) (рис. 1.2) из списка, затем установить курсор непосредственно на командлет (3) или набрать его название в поле **Name** (2). В процессе набора список будет автоматически сужаться. Если вы не помните название модуля, то с помощью значения фильтра **Modules** выведете все командлеты, выбрав значение **All** (1).

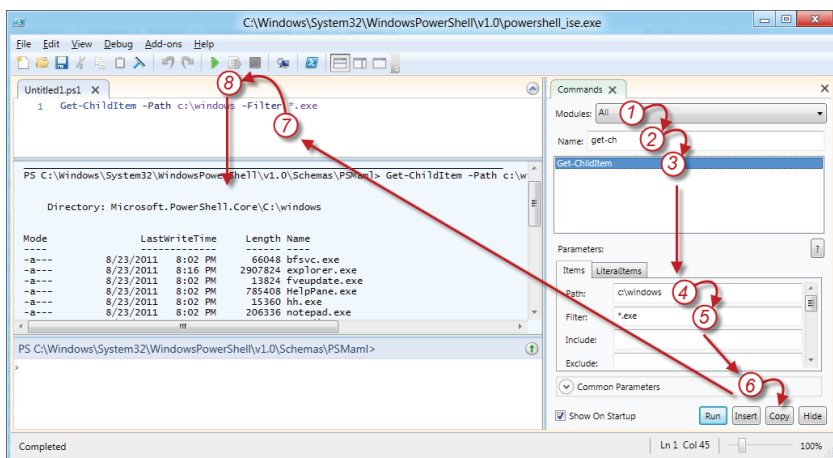


Рис. 1.2. Быстрое создание командлетов

После того как курсор установлен на нужном командлете, ниже в панели **Commands** отображается группа характеристик **Parameters**, определяющих его параметры запуска. В соответствии с условиями задачи определим значения для **Path** (4) и **Filter** (5). Затем скопируем сформированный текст запуска командлета в буфер обмена, нажав на кнопку **Copy** (6). Вставим скопированные данные в сценарий. Для этого установим курсор в окно сценария и нажмем комбинацию кнопок **Shift+Ins** (7). Запустим сценарий, воспользовавшись соответствующей кнопкой (8) или нажав на кнопку **F5**. Результат будет отображен в соответствующем окне.

### Замечание

Все стандартные аргументы запуска командлетов, такие как `Debug`, `Error Action` и др., сгруппированы в разделе **Common Parameters**.

## Усовершенствованная справка по командлетам

Кроме традиционного вызова справки с помощью командлета `get-help`, в новой панели *Commands* появилась возможность оперативного доступа к справочной информации, разработан интерфейс для настройки выводимой информации.

Получить справку по интересующему вас командлету очень просто. Выбрав из списка нужный командлет, необходимо нажать на кнопку с вопросительным знаком (?) (рис. 1.3). Автоматически сгенерируется диалоговое окно, содержащее информацию, выводимую командлетом, — `get-help`. Кроме того, в нем реализованы механизм поиска данных в выведенной справке и возможность настройки вывода данных. При нажатии на кнопку *Settings* появится диалоговое окно, в котором можно определить список отображаемых разделов справки и настройки системы поиска. По умолчанию отображаются все секции справки.

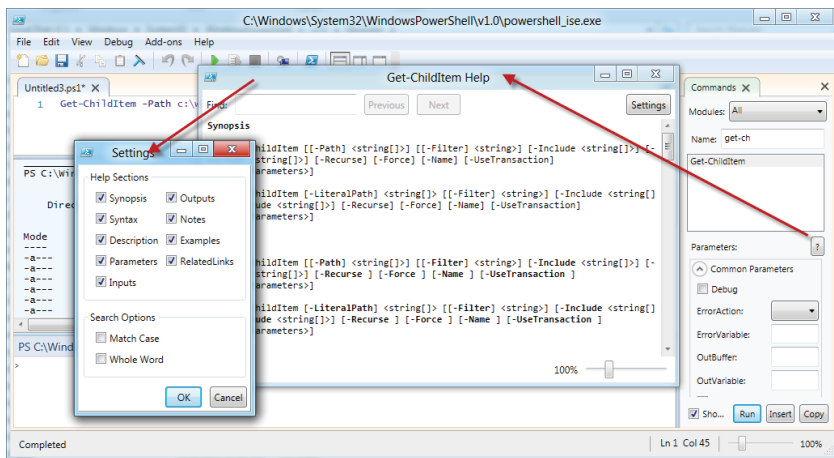


Рис. 1.3. Оперативное получение справочной информации по командлетам

## Технология IntelliSense

Технология **IntelliSense** хорошо известна программистам по Microsoft Visual Studio. Сегодня она наконец-то появилась в Windows PowerShell. Если в предыдущих версиях для быстрого ввода имени



The screenshot shows the Windows PowerShell ISE interface. The top pane displays the command `Get-ACL [-Path <string>] [-Audit] [-AllCimAccessPolicies] [-Filter <string>] [-Include <string>] [-Exclude <string>] [-UseTransaction] [-CommonParameters]`. The bottom pane shows the output of the command, which is a list of ACL entries for the `PS C:\Users\>` directory. The output includes the path, input object, literal path, audit, all cim access policies, filter, include, exclude, and use transaction properties.

Аналогичная ситуация – с подсказкой названий атрибутов командлета. Как только после названия команды вводится дефис, сразу отображается список используемых в нем параметров (см. рис. 1.4).

В новой версии реализована возможность создания своих шаблонов сценария, которые позволяют формировать конечный сценарий из уже готовых фрагментов. По умолчанию в PowerShell СТР 3 всего 12 шаблонов для условных операций, например `If`, `For`, `Switch`, и для пользовательских функций, имеющих различное количество аргументов.

Для вызова списка шаблонов достаточно в меню **Edit** выбрать **Start Snippets** (рис. 1.5) или воспользоваться клавиатурным сокращением **Ctrl+J**.

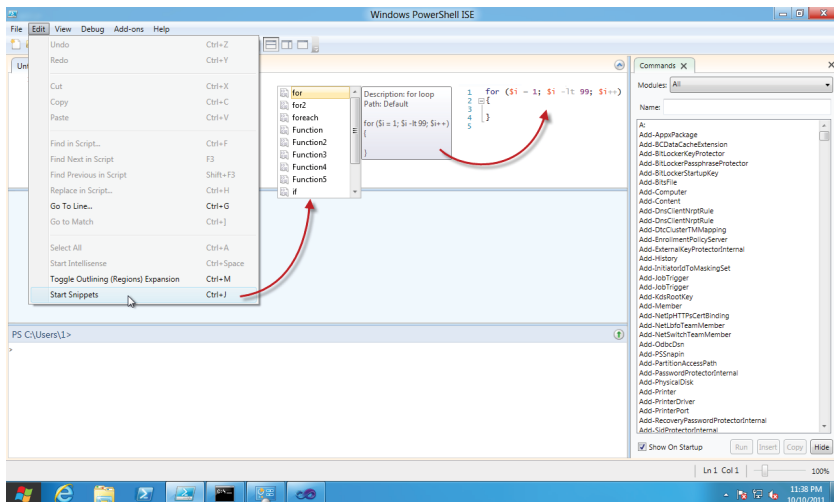


Рис. 1.5. Использование шаблонов сценариев

## Первый запуск PowerShell

Для запуска PowerShell необходимо в меню **Пуск** ⇨ **Программы** ⇨ **Windows PowerShell 1.0** кликнуть по ярлычку **Windows PowerShell**. В результате выполнения команды появится диалоговое окно, очень похожее на оболочку CMD.

## Запуск сценариев PowerShell

По умолчанию выполнение скриптов Windows PowerShell отключено. Установлен режим *Restricted*, позволяющий работать с консолью только в интерактивном варианте. Существуют четыре уровня безопасности запуска сценариев:

- ❑ **AllSigned** – все файлы, в том числе и на локальной машине, содержащие сценарии на языке PowerShell, должны иметь цифровую подпись. При запуске файла оболочка задаст вопрос о том, доверяете ли вы данному издателю и можно ли доверять в дальнейшем файлам с этой подписью;
- ❑ **RemoteSigned** – все сценарии, находящиеся в сети; должны быть подписаны. Файлы, запускаемые с локальной машины; могут не иметь цифровой подписи;
- ❑ **Restricted** (по умолчанию) – выполнение сценариев запрещено;

- ❑ **Unrestricted** – все сценарии, запускаемые с локального диска или из сети, могут не иметь цифровой подписи. Запуск файлов из сети сопровождается соответствующим предупреждением. Для его подавления необходимо в свойствах файла выбрать *Unblock* (разблокировать).

Уровень безопасности задается с помощью параметра реестра `ExecutionPolicy`. Значение параметра соответствует названию уровня безопасности (листинг 1.1).

### Листинг 1.1. Настройка параметра безопасности PowerShell по умолчанию

---

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell1\Shell\Ids\
                                                                    ↗ Microsoft.PowerShell]
"ExecutionPolicy"="Restricted"
```

---

Управление уровнем безопасности исполнения сценариев средствами PowerShell осуществляется с помощью двух командлетов – `Get-ExecutionPolicy` и `Set-ExecutionPolicy`. Командлет `Get-ExecutionPolicy` предназначен для просмотра установленного уровня безопасности (по умолчанию `Restricted`), а командлет `Set-ExecutionPolicy` – для изменения текущего уровня безопасности. В отличие от предыдущего командлета, при использовании данного командлета в качестве аргумента необходимо указать новый уровень безопасности. Недостаток данного метода заключается в том, что пользователь, выполняющий данные командлеты, должен обладать административными привилегиями.

Из всех существующих уровней безопасности поставленным условиям удовлетворяет один из двух:

- ❑ **RemoteSigned**. Файл сценария должен иметь электронную подпись;
- ❑ **Unrestricted**. Файлы могут быть не подписаны.

Использование первого способа позволяет повысить уровень безопасности выполнения различных сценариев в сети, по сравнению со вторым способом.

## Документация по PowerShell

Пакет документации можно скачать с официального сайта Microsoft со страницы *Windows PowerShell 1.0 Documentation Pack* (<http://>

[www.microsoft.com/downloads/details.aspx?familyid=B4720B00-9A66-430F-BD56-EC48BFCA154F&displaylang=en](http://www.microsoft.com/downloads/details.aspx?familyid=B4720B00-9A66-430F-BD56-EC48BFCA154F&displaylang=en)). В комплект входят четыре документа, краткое описание которых приведено в табл. 1.2.

**Таблица 1.2. Содержание пакета Windows PowerShell 1.0 Documentation Pack**

Название документа (англ.)	Название документа (в локализации для России)	Имя файла	Краткое описание
Windows PowerShell Getting Started Guide	Знакомство с Windows PowerShell	GettingStarted.rtf	Рассказывается о целях и задачах PowerShell, основах построения, о навигации и работе оболочки
Windows PowerShell Primer	Введение в Windows PowerShell	userguide.rtf	Рассказано о приемах программирования, о функциях и командах, доступных в PowerShell
Windows PowerShell Language Quick Reference	Краткий обзор языка Windows PowerShell	quadfold.rtf	Описаны основы синтаксиса языка (типы данных, комментарии, поддерживаемые циклы и др.)
Windows PowerShell V1.0 Release Notes	Примечания к выпуску Windows PowerShell v 1.0 (для .NET Framework 2.0 RTM)	releasenotes.rtf	Описаны различия версий PowerShell, PowerShell RC1, PowerShell RC2

Еще один важный документ, который обязательно пригодится программистам, знающим VBScript и желающим перейти на PowerShell, – *Translating VBScript to Windows PowerShell*. Этот документ входит в пакет *Windows PowerShell Week Script Samples* (<https://www.microsoft.com/downloads/details.aspx?FamilyId=264CE487-1D36-4466-BD8B-23A7F1FA967E&displaylang=en>). После установки пакета из файла PowerShellWeekSamples.exe в указанной для установки папке будут присутствовать четыре папки и текстовый файл с детальным описанием их содержания. Стоит отметить, что первые три папки содержат примеры на PowerShell, которые будут очень полезны. В последней папке – VBStoPSGuide – находится файл vbscript\_to\_powershell.doc, содержащий описание перехода от VBScript к PowerShell.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)