

# Назначение PowerShell

- Windows PowerShell – это не язык скриптов или, по крайней мере, это не только язык скриптов.
- Windows PowerShell – это целый механизм, созданный для того, чтобы выполнять команды, которые решают административные задачи, такие как
  - создание нового пользовательского аккаунта
  - конфигурация сервисов
  - удаление почтовых ящиков
- PowerShell предоставляет возможность проектировать GUI интерфейсы на своей базе. Вы можете выполнять одни команды двумя путями
  - Набирать команды в консоли
  - Выбирать графические элементы, которые выполнят те же команды.

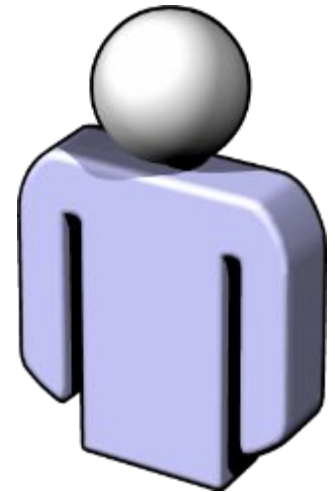
## Где используется Windows PowerShell?

- Windows PowerShell используется с большим количеством продуктов Microsoft, например:
  - Microsoft System Center Data Protection Manager
  - Microsoft System Center Operations Manager
  - Microsoft System Center Virtual Machine Manager
  - Microsoft SQL Server

**Get-Mailbox | Sort Size | Select -first 100 |  
Move-Mailbox Server2**

# Какие команды вы уже знаете?

- Например команды, которые вы бы использовали для выполнения каждой из следующих задач в оболочке `cmd` или `Unix shell`:
  - Изменить каталоги.
  - Список файлов и подкаталогов в каталоге.
  - Скопировать файл.
  - Отображение содержимого текстового файла.
  - Удалить файл.
  - Перемещение файлов.
  - Переименовать файл.
  - Создать новый каталог.



# Команды

- С Windows PowerShell можно также запускать большинство внешних команд, которые могут быть знакомы вам:
  - Ipconfig.exe
  - Ping.exe
  - Tracert.exe
  - Nslookup.exe
  - Pathping.exe
  - Net.exe (например, Net Use)
- Windows PowerShell распознает многие из имен команд, с которыми вы, вероятно, уже знакомы, в том числе Cd, Dir, Ls, Cat, Type, Mkdir, Rmdir, Rm, Del, Cp, Copy, Move и так далее. Одновременно доступны наиболее общие file-and-folder команды управления Cmd.exe (которая использует MS-DOS Синтаксис команды) и оболочки Unix.

# Иерархические хранилища

- Файловая система Windows и, если уж на то пошло, файловая система большинства компьютеров является иерархической
  - Это означает, что она состоит из отдельных ячеек, которые называются папками или директориями, которые, в свою очередь, содержат либо файлы, либо другие папки. Папки содержат подпапки, которые, опять же, содержат свои подпапки, и так далее
- Однако файловая система – это не единственная иерархическая система в Windows.
  - The registry
  - The certificate store
  - Active Directory
- Одна из фиц PowerShell – единая система работы со всеми иерархическими хранилищами.

# Один набор команд для различных хранилищ

- Возможность использования единого набора команд для навигации по хранилищам, имеющим различную форму, становится возможным благодаря функции Windows PowerShell, которая называется провайдер PSDrive или просто провайдер.
- Провайдер представляет собой разновидность адаптера, который устанавливает соединение с системой хранения и представляет ее Windows PowerShell в форме дискового накопителя.
- PowerShell поставляется с набором провайдеров:

The file system	IIS
The registry	SQL Server
The environment variables	Active Directory
The certificate store	

# Управление PSDrives

- `Get-PSDrive` - выводит все доступные хранилища. По умолчанию, этот список включает все доступные накопители и логические диски
- `New-PSDrive` - создает новое хранилище. Вы должны указать имя хранилища (без двоеточия на конце), имя провайдера и стартовую точку или путь. Тип такой стартовой точки или пути зависит от типа используемого провайдера
- `Remove-PSDrive` - уничтожает хранилище. Вы также можете удалить хранилища, присутствующие по умолчанию, например `HKCU:` или `ENV:`, однако они будут воссозданы при запуске нового экземпляра оболочки.

# Псевдонимы

- **Get-Alias** – отображение список всех псевдонимов. Также можно использовать `Dir Alias:` чтобы увидеть все содержимое папки `ALIAS: drive`.
- **New-Alias** – создание нового псевдонима. Здесь обязательно указать имя нового псевдонима, а также название команды, для которой он предназначен.
- **Del** или **Rm** – удаление псевдонима из `ALIAS: drive`.
- **Import-Alias** и **Export-Alias** – импорт и экспорт псевдонимов в файл и из файла.



## Help commandName

- С помощью некоторых параметров команды Help можно получить еще более подробную информацию:
- **-detailed** – показ более детального описания
- **-examples** – показ примеров использования
- **-full** – показ полной информации, включая детальное описание, описание каждого параметра и примеры использования
- **-online** – открытие браузера и показ описания командлетов на сайте Microsoft. На сайте может содержаться обновленная или расширенная информация, которая еще не была выпущена в служебном пакете.

# Расширение оболочки

- Командлеты, присутствующие по умолчанию в оболочке, не являются единственными доступными для вас командлетами. Microsoft, так же, как и сторонние разработчики программного обеспечения, могут создавать дополнительные командлеты и провайдеры PowerShell, и предоставлять их вам в виде оснасток или модулей. Управление оснастками осуществляется с помощью набора командлетов, название которых включает существительное PowerShell:
  - Get-PSSnapin
  - Add-PSSnapin
  - Remove-PSSnapin
  - Import-Module
  - Remove-Module
  - Get-Module

**Get-PSSnapin –registered**  
**Get-Module –list**

# Конвейеризация (Piping)

- Акт передачи выходных данных одного командлета во входные данные другого командлета называется конвейеризацией. Возможно, в других оболочках вам уже приходилось использовать конвейер. Например, это стандартная команда в Cmd.exe:

```
dir | more
```

- Здесь выходные данные команды Dir перенаправляются во входные данные команды More, которая создает постраничное отображение выходных данных.
- Конвейер нашел широкое применение в Windows PowerShell. Весьма распространенным явлением здесь является строка из полдюжины командлетов, связанных между собой конвейером. Данные переходят из одного командлета в другой, при этом они постепенно уточняются, детализируются и превращаются именно в ту информацию, которая вам требуется.

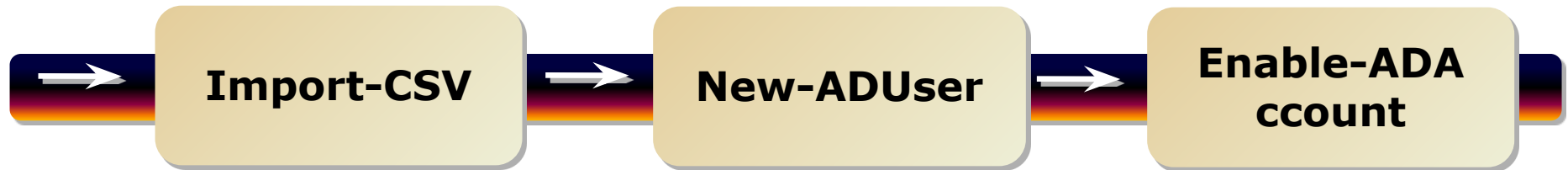
# Вывод командлетов

- Основные командлеты для модификации вывода
  - Format-Table, имеет alias Ft
  - Format-List, имеет alias Fl
  - Format-Wide, имеет alias Fw

**Get-Service | Format-List**  
**Get-Process | Fw**

# Конвейер

- 



# Терминология

- командлет Get-Process помещает элементы процесса в конвейер. Более официальным термином здесь будет слово «объект», т.е. вы можете сказать, что командлет Get-Process помещает объекты процесса в конвейер.
- Эти объекты имеют атрибуты. Для процесса атрибутами могут быть имя, ID, объем занимаемой памяти и.т.д. Формальным названием атрибута будет термин «свойство». Другими словами, вы можете сказать, что объект процесса обладает свойством имени, свойством ID, и.т.д.

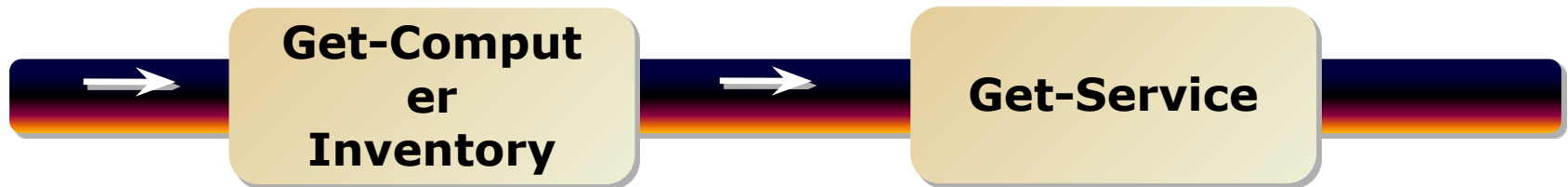
<b>Pid</b>	<b>Name</b>	<b>Image</b>
---	-----	-----
<b>092</b>	<b>Notepad</b>	<b>notepad.exe</b>
<b>098</b>	<b>Windows Paint</b>	<b>mspaint.exe</b>
<b>112</b>	<b>Calculator</b>	<b>calc.exe</b>
<b>164</b>	<b>Windows PowerShell</b>	<b>powershell.exe</b>

# Свойства объектов

- PowerShell командлеты возвращают объекты
- PowerShell командлеты используют объекты как входные данные
- Командлеты можно заставить использовать свойства входящих объектов в качестве параметров

**Get-Service -computerName SEA-SRV2**

- Но параметр можно получить из свойства, переданного другим командлетом



# Перенаправления ввода-вывода и форматирование

- Имеется ряд Out- командлетов
  - Out-Gridview
  - Out-Printer
  - Out-File
  - Out-Host
  - Out-Null





# Изучение свойств

- The Get-Member cmdlet...

**Get-EventLog Security -newest 10 | Get-Member  
Get-Process | Format-List \*  
Get-Service | Get-Member | Out-GridView**