PowerShell: возможности, журналы. Отличие cmd от Power Shell. Команда просмотра списка доступных журналов событий.

PowerShell — это кроссплатформенное решение для автоматизации задач, которое включает оболочку командной строки, скриптовый язык и платформу управления конфигурацией. **PowerShell** поддерживается **в Windows**, Linux и macOS. Оболочка командной строки.

Windows PowerShell позволяет системным администраторам автоматизировать большинство рутинных задач. С ее помощью можно менять настройки, останавливать и запускать сервисы, а также производить обслуживание большинства установленных приложений. Воспринимать синее окошко как еще один интерпретатор команд было бы неправильно. Такой подход не отражает сути предложенных корпорацией Microsoft инноваций. На самом деле возможности Windows PowerShell гораздо шире. Попробуем разобраться, чем решение Microsoft отличается от более привычных нам средств.

Основные возможности

Разумеется Windows PowerShell — это в первую очередь командная оболочка с языком сценариев, изначально созданная на основе платформы .NET Framework, а позднее — на .NET Core. В отличие от принимающих и возвращающих текстовые данные оболочек, Windows PowerShell работает с классами .NET, у которых есть свойства и методы. PowerShell позволяет выполнять обычные команды, а также дает доступ к объектам COM, WMI и ADSI. В ней используются различные хранилища, вроде файловой системы или реестра Windows, для доступа к которым созданы т.н. поставщики (providers). Стоит отметить возможность встраивания исполняемых компонентов PowerShell в другие приложения для реализации различных операций, в т.ч. через графический интерфейс. Верно и обратное: многие приложения для Windows предоставляют доступ к своим интерфейсам управления через PowerShell.

Windows PowerShell позволяет:

- Менять настройки операционной системы;
- Управлять службами и процессами;
- Настраивать роли и компоненты сервера;
- Устанавливать программное обеспечение;
- Управлять установленным ПО через специальные интерфейсы;
- Встраивать исполняемые компоненты в сторонние программы;
- Создавать сценарии для автоматизации задач администрирования;
- Работать с файловой системой, реестром Windows, хранилищем сертификатов и т.д.

Журналы PowerShell и характер записываемой в них информации.

Журнал Приложение содержит записи связанные с программами на вашем компьютере. В журнал пишется когда программа была запущена, если запускалась с ошибкоу, то тут это тоже будет отражено.

Журналы аудита для Windows 365 содержат записи действий, приводящих к изменениям на облачном компьютере. Действия создания, обновления (редактирования), удаления, назначения и удаленные действия приводят к созданию событий аудита, которые администраторы могут просматривать для большинства действий облачных компьютеров, выполняющихся через Graph. По умолчанию аудит включен для всех клиентов. Отключить его невозможно. Пользователи со следующими разрешениями могут просматривать журналы аудита:

- Глобальный администратор
- Администратор службы Intune
- администраторы, которым назначена роль Intune с разрешениями **Данные аудита — чтение**.

Пункт Установка, в него записывает Windows логи о том что и когда устанавливалось Например программы или обновления.

Самый важный журнал Это система. Сюда записывается все самое нужное и важное. Например у вас был синий экран bsod, и данные сообщения что тут заносятся помогут вам определить его причину.

При вводе команды в командной строке PowerShell сохраняет команду в журнале команд. Команды в журнале можно использовать в качестве записи о работе. Кроме того, вы можете отозвать и выполнить команды из журнала команд.
PowerShell имеет два разных поставщика журналов: встроенный журнал и журнал, управляемый модулем **PSReadLine**. Журналы управляются отдельно, но обе истории доступны в сеансах, где **загружается PSReadLine**.

Использование журнала PSReadLine
Журнал PSReadLine отслеживает команды, используемые во всех сеансах PowerShell. Журнал записывается в центральный файл на узел. Этот файл журнала доступен для всех сеансов и содержит весь прошлый журнал. Журнал не удаляется при завершении сеанса. Кроме того, этот журнал не может управляться командлетами*-History. Дополнительные сведения см. в about_PSReadLine.

Использование встроенного журнала сеансов
Встроенный журнал отслеживает только команды, используемые в текущем сеансе. Журнал недоступен для других сеансов и удаляется после окончания сеанса.
Командлеты журнала
PowerShell содержит набор командлетов, управляющих журналом команд.

Командлет	Псевдоним	Описание
`Get-History`	`h`	Возвращает журнал команд.
`Invoke-History`	`r`	Выполняет команду в журнале команд.
`Add-History`		Добавляет команду в журнал команд.
`Clear-History`	`clhy`	Удаляет команды из журнала команд.

Сочетания клавиш для управления журналом
В консоли PowerShell для управления журналом команд можно использовать следующие сочетания клавиш.

- UpArrow — отображает предыдущую команду.
- DownArrow — отображает следующую команду.
- F7 — отображает журнал команд.
- ESC — чтобы скрыть журнал.
- F8 — находит команду. Введите один или несколько символов, а затем нажмите клавишу F8. Снова нажмите клавишу F8 для следующего экземпляра.
- F9 — поиск команды по идентификатору журнала. Введите идентификатор журнала и нажмите клавишу F9. Нажмите клавишу F7, чтобы найти идентификатор.
- #<string>Вкладка —поиск журнала *<string>* и возврат последнего совпадения. Если вы несколько раз нажимаете клавишу TAB, она циклически проходит по соответствующим элементам в журнале.

Примечание
Эти ключевые привязки реализуются ведущим приложением консоли. Другие приложения, такие как Visual Studio Code или Терминал Windows, могут иметь разные привязки ключей. Привязки можно переопределить модулем PSReadLine. PSReadLine загружается автоматически при запуске сеанса PowerShell. При загрузке PSReadLine F7 и F9 не привязаны к какой-либо функции. PSReadLine не предоставляет эквивалентную функциональность.

MaximumHistoryCount

Переменная $MaximumHistoryCount предпочтения определяет максимальное количество команд, которые PowerShell сохраняет в журнале команд. Значение по умолчанию — 4096.
Например, следующая команда снижает $MaximumHistoryCount до 100 команд:
PowerShellКопировать
$MaximumHistoryCount = 100
Чтобы применить этот параметр, перезапустите PowerShell.
Чтобы сохранить новое значение переменной для всех сеансов PowerShell, добавьте инструкцию присваивания в профиль PowerShell.

Порядок команд в журнале

Команды добавляются в журнал после завершения выполнения команды, а не после ввода команды. Если выполнение команд занимает некоторое время или если команды выполняются во вложенной строке, команды могут оказаться неупорядоченными в журнале. Команды, выполняемые во вложенной строке, выполняются только при выходе из уровня запроса.

Отличие cmd от Power Shell

Главное отличие между PowerShell и CMD является то, что PowerShell — это мощный интерфейс командной строки и среда сценариев, которая позволяет выполнять сложные сценарии для простого и эффективного выполнения задач администрирования системы Windows, а CMD — интерфейс командной строки, который предоставляет пользователю набор команд для взаимодействия с компьютером для выполнения задач.

Windows PowerShell, командная строка и терминал обладают аналогичной функциональностью, что и специализированные инструменты. Способ выполнения команды в Windows PowerShell и в окне PowerShell терминала Windows одинаков. Аналогично, выполнение команды в выделенном окне CMD работает одинаково с ее запуском в командной строке терминала Windows.

Среди инструментов командной строки, основанных на задачах, PowerShell и Command Prompt полезны для автоматизации задач системного администрирования, но их функциональные возможности и возможности немного отличаются. Оболочка командной строки основана на тех же принципах, что и DOS, и была представлена вместе с Windows NT. PowerShell, с другой стороны, представляет собой оболочку командной строки на основе задач и язык сценариев, основанный на .Net framework, который в основном используется для пакетной обработки и управления системой.

PowerShell и Command Prompt - это инструменты командной строки, которые позволяют писать сценарии и пакетные файлы для выполнения различных задач системного администрирования. Оба интерфейса работают по-разному, хотя кажется, что команда dir работает одинаково на любом из них. По сравнению с PowerShell командная строка предлагает ограниченные административные возможности, тогда как PowerShell предлагает расширенную среду командной оболочки и больше возможностей. Хотя функции PowerShell более сложны, чем функции традиционной командной строки, они все еще довольно мощные.

Команда в PowerShell для просмотра списка доступных журналов событий

Команда/ Get-EventLog - возвращает события в журнале событий или список журналов событий на локальном компьютере или удаленных компьютерах.

Командлет Get-EventLog получает события и журналы событий с локальных и удаленных компьютеров. По умолчанию Get-EventLog получает журналы с локального компьютера. Чтобы получить журналы с удаленных компьютеров, используйте параметр ComputerName .

Для поиска событий можно использовать Get-EventLog параметры и значения свойств. Командлет получает события, соответствующие указанным значениям свойств.

Командлеты PowerShell, содержащие EventLog существительное, работают только в классических журналах событий Windows, таких как приложение, система или безопасность. Чтобы получить журналы, использующие технологию журнала событий Windows в Windows Vista и более поздних версиях Windows, используйте Get-WinEvent.

Примечание
Get-EventLog использует API Win32, который не рекомендуется. Результаты могут быть не точными. Get-WinEvent Вместо этого используйте командлет.