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.

Основные возможности

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 в PowerShell

Windows PowerShell создает журнал событий Windows с именем «Windows PowerShell» для записи событий Windows PowerShell, это классический журнал событий, в котором не используется технология Windows Eventing. Этот журнал можно просмотреть в средстве просмотра событий или с помощью командлетов, получающих события, например, Get-EventLog командлета. По умолчанию события ядра и поставщика Windows PowerShell записываются в журнал событий, но вы можете использовать переменные предпочтений журнала событий, чтобы настроить журнал событий. Например, вы можете добавить события о командах Windows PowerShell.

Командлет Get-WinEvent получает события из журналов событий, включая классические журналы, такие как журналы системы и приложений. Командлет получает данные из журналов событий, созданных технологией журнала событий Windows, представленной в Windows Vista, и событиями в файлах журналов, созданных трассировкой событий Windows (ETW). По умолчанию Get-WinEvent возвращает сведения о событии в порядке последней до самой старой.

Get-WinEvent выводит список журналов событий и поставщиков журналов событий. Чтобы прервать команду, нажмите клавиши CTRL+C. События можно получить из выбранных журналов или из журналов, созданных выбранными поставщиками событий. Кроме того, можно объединять события из нескольких источников в одну команду. Get-WinEvent позволяет фильтровать события с помощью запросов XPath, структурированных XML-запросов и хэш-запросов таблиц.

Если вы не используете PowerShell от имени администратора, могут появиться сообщения об ошибках, которые не могут получить сведения о журнале.

Получение классического журнала установки

Эта команда получает объект EventLogConfiguration, представляющий классический журнал установки**.** Объект содержит сведения о журнале, такие как размер файла, поставщик, путь к файлу и включение журнала.

Командлет Get-WinEvent использует параметр ListLog для указания журнала Setap . Объект отправляется по конвейеру командлету Format-List. Format-List использует параметр Property со звездочкой (*) для отображения каждого свойства: `Get-WinEvent -ListLog Setup | Format-List -Property *`

Получение классического журнала безопасности

Эта команда получает объект EventLogConfiguration**,** представляющий классический журнал безопасности**.**

Командлет Get-WinEvent использует параметр ListLog для указания журнала Security.

Отличие cmd от Power Shell

PowerShell	Командная строка
Он поддерживает как пакетные, так и командлеты PowerShell.	На CMD следует запускать только пакетные команды.
Он также поддерживает создание псевдонимов для командлетов, чтобы пользователи могли легко перемещаться между функциями.	CMD не может создавать псевдонимы команд.
У вас есть свобода передавать выходные данные командлета другому командлету.	Вы никогда не сможете передать результат одной команды другой.
PowerShell возвращает результат в виде объекта.	CMD возвращает только текст.
Мы можем запустить последовательность командлетов в сценарии.	В cmd вторая команда будет запущена, если первая команда будет завершена.
Вы можете использовать команду Help, чтобы с легкостью получить информацию о любых командлетах.	В CMD нет такой опции помощи.

PowerShell	Командная строка
PowerShell имеет интегрированную среду сценариев (ISE).	Это просто интерфейс командной строки.
Вы можете получить доступ к библиотекам программирования, потому что они построены на основе .NET Common Language Runtime (CLR).	Нет доступа к библиотекам
Вы можете напрямую интегрировать его с WMI.	Если вам нужна интеграция с WMI, вам потребуются внешние плагины.
С облачными продуктами Microsoft проще подключиться.	CMD не может подключиться к продуктам Microsoft Online.
Он поддерживает системы Linux.	CMD не поддерживает системы Linux.
Вы можете запускать все типы программ с помощью PowerShell.	ssCMD предназначен только для программ консольного типа.