**Вступ до PowerShell**

Windows PowerShell – це нова командна оболонка Windows, розроблена в першу чергу для системних адміністраторів. Вона включає інтерактивний командний рядок і середовище виконання сценаріїв, які можна використовувати разом або окремо.

На відміну від більшості оболонок, які приймають і повертають текст, оболонка Windows PowerShell, розроблена на основі середовища CRL .NET і платформи .NET Framework, приймає і повертає об'єкти .NET. Це фундаментальна зміна робить можливими абсолютно нові засоби і методи адміністрування і конфігурації систем Windows.

У Windows PowerShell реалізована нова концепція командлетів – простих, вузько спеціалізованих засобів командного рядка, вбудованих в оболонку. Командлети можна використовувати і окремо, проте по-справжньому їх достоїнства виявляються тоді, коли ці прості засоби використовуються в комбінації один з одним для вирішення складних завдань. Windows PowerShell включає більше ста основних командлетов, до того можна створювати власні командлети і обмінюватися ними з іншими користувачами.  
 Як і багато інших оболонок, Windows PowerShell забезпечує доступ до файлової системи на комп'ютері. Крім того, до складу оболонки Windows PowerShell входять постачальники, що дозволяють настільки ж легко працювати з іншими сховищами даних, такими як реєстр і сховища сертифікатів цифрових підписів.

Windows PowerShell відрізняється від інших оболонок такими ознаками:

• Windows PowerShell опрацьвує не текст, а об'єкти платформи .NET;

• Windows PowerShell включає в себе велику кількість команд, що мають поліпшений інтерфейс;

• всі команди оболонки опрацьовуються одним синтаксичним аналізатором;

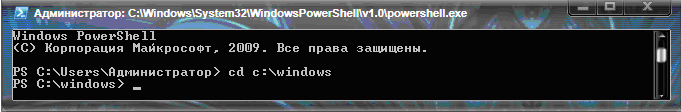
• в оболонці Windows PowerShell можна використовувати традиційні засоби Windows, такі як Net, SC і Reg.exe.

Навігація у файловій системі

Все люди, работающие с компьютером, ежедневно выполняют множество обращений к файловой системе, читая и сохраняя файлы, создавая каталоги. Всі люди, що працюють з комп'ютером, щодня виконують безліч звернень до файлової системи, читаючи і зберігаючи файли, створюючи каталоги,копируя файлы из одного места в другое и т. п. В данной главе мы научимся манипулировать объектами файловой системы с помощью средств оболочки PowerShell. копіюючи файли з одного місця в інше і т.п.

Рассмотрим наиболее часто выполняемые операции при работе с файлами и каталогами. Розглянемо операції, які найбільш часто виконуються при роботі з файлами та каталогами.

Навигация в файловой системе В оболочке cmd.exe смена текущего каталога производится с помощью команды cd. В оболонці cmd.exe зміна поточного каталога відбувається за допомогою команди cd.В PowerShell команда cd имеет тот же смысл, при этом она является стандартным псевдонимом командлета set-Location. У PowerShell команда cd має таке ж значення, при цьому вона є стандартним псевдонімом командлета Set-Location. Например, следующая команда делает текущим каталог [c:\windows](file:///C:\windovvs) : Наприклад, наступна команда робить поточним каталог [c: \ windows](file:///C:\windovvs) :

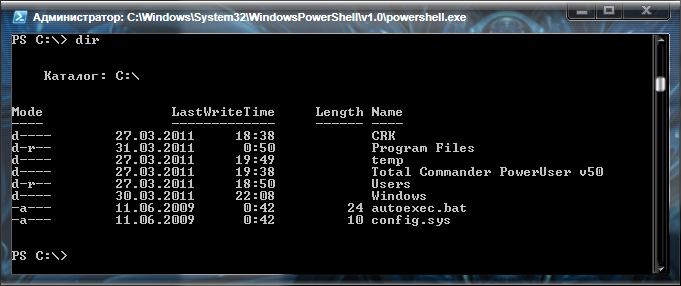


Как и в оболочке cmd.exe. Як і в оболонці cmd.exe, в качестве пути можно указывать символы ..в якості шляху можна вказувати символи .. (для перехода в родительский каталог) и \ (для перехода в корневой каталог текущего диска). (для переходу в батьківський каталог) і \ (для переходу в кореневий каталог поточного диска).

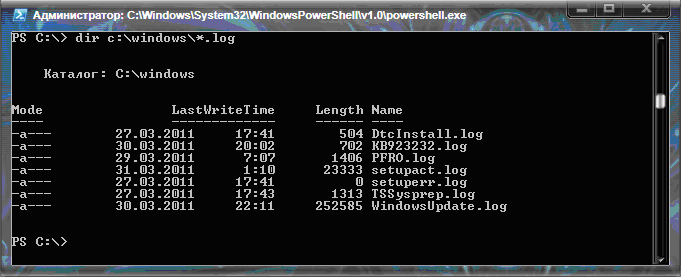
**При этом нужно учитывать следующий нюанс.**

**Получение списка файлов и каталогов** Отримання списку файлів і каталогів

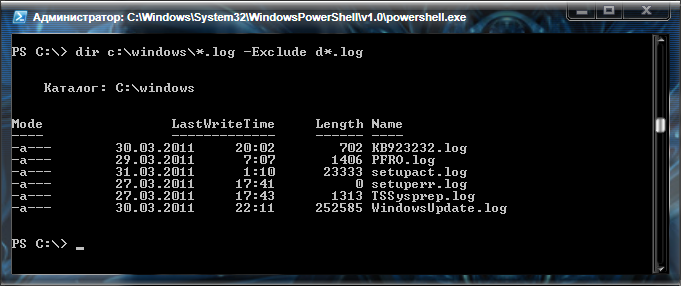
Напомним, что в оболочке cmd.cxe список файлов и каталогов формируется с помощью внутренней команды dir, которая имеет множество ключей, позволяющих, например, выводить только файлы с определенными атрибутами, обрабатывать вложенные подкаталоги, сортировать список по различным полям и т.д. В оболонці cmd.cxe список файлів і каталогів формується за допомогою внутрішньої команди dir, яка має безліч ключів, котрі дозволяють, наприклад, виводити тільки файли з певними атрибутами, обробляти вкладені підкаталоги, сортувати список за різними полях і т.д. В PowerShell також можна використовувати команду dir, яка виконує роль коменлетВ PowerShell также можно использовать команду dir, которая является здесь псевдонимом командлета Get-childitem. У PowerShell також можна використовувати команду dir, яка є тут псевдонімом командлета Get-ChildItem. Якщо виконати цю команду без параметрів, то буде виведено вміст поточного каталога: Если выполнить эту команду без параметров, то будет выведено содержимое текущего каталога: Якщо виконати цю команду без параметрів, то буде виведено вміст поточного каталогу

PS С:\> dirВ пути, который указывается для команды dir.

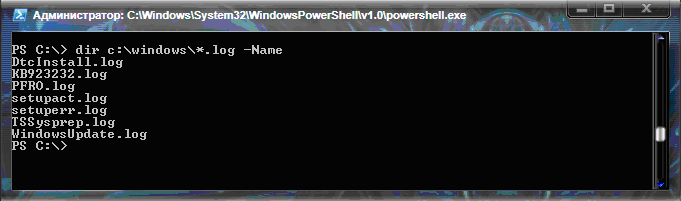
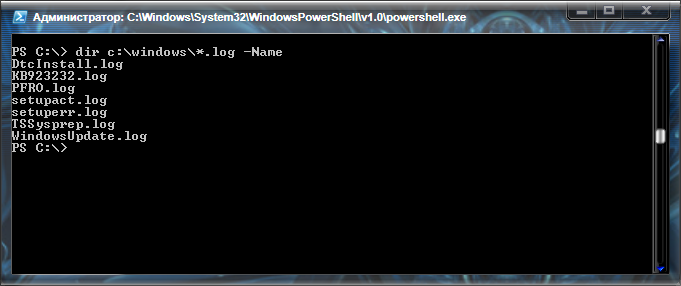
В шляху, який вказується для команди dirможно применять подстановочные знаки. можна використовувати символи підстановки. Наприклад, наступна команда виведе всі файли з розширенням log із каталогу c:\windows:Например, следующая команда выведет все файлы с расшире-1ием log из каталога [c:\windows](file:///C:\windo\vs) : Наприклад, наступна команда виведе всі файли з розширенням log з каталогу [c: \ windows](file:///C:\windo\vs) :



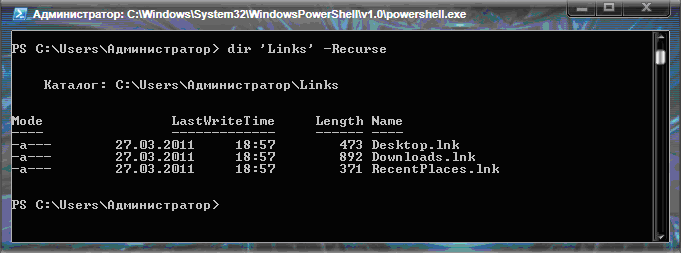
Параметр -Exclude позволяет задать маску файлов, которые не будут обрабатываться командой dir.Параметр -Exclude дозволяє задати маску файлів, які не будуть оброблятися командою dir. Например, следующая команда выведет все файлы с расширением log из каталога [c:\vvindows](file:///C:\vvindows) , кроме тех, чье имя начинается на букву d: Наприклад, наступна команда виведе всі файли з розширенням log з каталогу [c: \ windows](file:///C:\vvindows) , крім тих, чиє ім'я починається на букву d: Наприклад, наступна команда виведе всі файли з розширенням log із каталогу c:\windows, крім тих, ім’я яких починається на букву d:

Параметр -Name позволяет выводить на экран только имена файлов (таким образом, этот параметр является аналогом ключа /ь команды dir из cmd.exe).

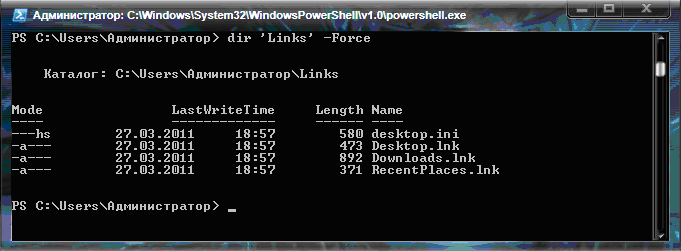
Параметр -Name дозволяє виводити на екран тільки імена файлів (таким чином, цей параметр є аналогом ключа /b команди dir з cmd.exe). Наприклад:

например: наприклад:PS C:\> dir c:\windows\\*.log -Name 0.log

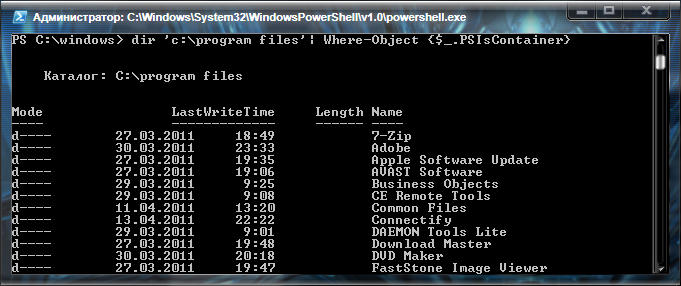
Параметр -Recurse включает режим рекурсии, при котором командлет dir отображает не только содержимое указанного каталога, но и всех его подкаталогов: Параметр -Recurse включає режим рекурсії, при якому командлет dir відображає не тільки вміст зазначеного каталогу, але і всіх його підкаталогів:

PS C:\> dir 'Documents and Settings' -Recurse1о умолчанию командлет dir не "видит" скрытые файлы.

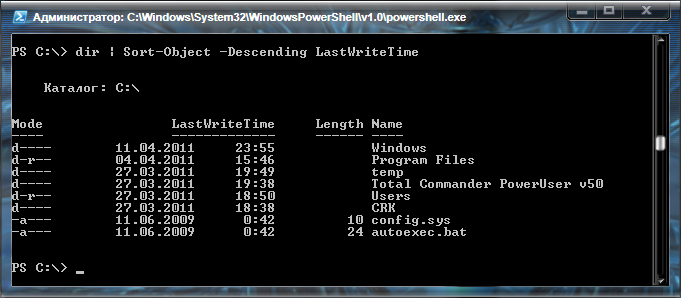
За замовчуванням командлет dir не "бачить" приховані файли. Щоб включити такі файли до списку потрібно вказати параметр -Force:

Если необходимо такие файлы также включать в список, то нужно указать параметр -Force: Якщо необхідно такі файли також включати до списку, то потрібно вказати параметр-Force:

Иногда бывает нужно сформировать список, состоящий только из подкаталогов или только из файлов.Іноді буває потрібно сформувати список, що складається тільки з підкаталогів або тільки із файлів. Иногда бывает нужно сформировать список, состоящий только из подкаталогов или только из файлов.В оболонці cmd.exe подібну фільтрацію в команді dir дозволяє виконувати ключ /a, що задає обмеження на атрибути елементів, на які діє команда(наприклад, команда dir /a:d відобразить всі підкаталоги поточного типу каталогу). У PowerShell побудувати список підкаталогів можна шляхом фільтрації об’єктів за допомогою комедлета Where-Object по значенню логічної властивості PSIsContainer(якщо об’єкт відповідає каталогу файлової системи, то ця властивість рівна $True, якщо ж об’єкт відповідає файлу, то властивість PSIsContainer рівна $False). Наступна команда виведе список підкаталогів каталогу с:\program files:

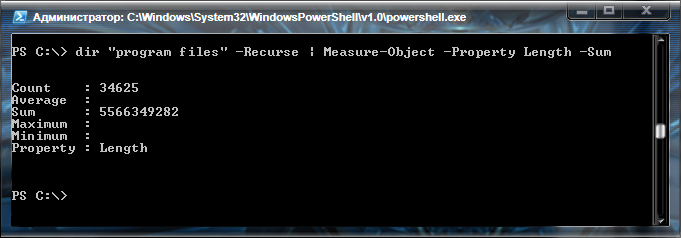
В оболочке cmd.exe подобную фильтрацию в команде dir позволяет выполнять ключ /а, задающий ограничения на атриб)ты элементов, на которые действует команда (например, команда dir /a:d отобразит все подкаталоги текущего каталога). В оболонці cmd.exe подібну фільтрацію в команді dir дозволяє виконувати ключ / а, що задає обмеження на атріб) ти елементів, на які діє команда (наприклад, команда dir / a: d відобразить всі підкаталоги поточного каталогу). В PowerShell построить список подкаталогов можно путем фильтрации объектов с помощью комапдлста Where-Object по значению логического свойства psisContainer (если объект соответствует каталогу файловой системы, то данное свойство равно STrue, если объект соответствует файлу, то свойство PsisContainer равно SFalse). У PowerShell побудувати список підкаталогів можна шляхом фільтрації об'єктів за допомогою комапдлста Where-Object за значенням логічного властивості psisContainer (якщо об'єкт відповідає каталогу файлової системи, то ця властивість одно STrue, якщо об'єкт відповідає файлу, то властивість PsisContainer одно SFalse). Следующая команда выведет список подкаталогов каталога [c:\program](file:///C:\program) files: Наступна команда виведе список підкаталогів каталогу [c: \ Program](file:///C:\program) Files:

ч\* За сортування списку файлів і каталогів в команді dir оболонки cmd.exe відповідає ключ /о. Нариклад, команда dir /o:-d видасть список файлів у поточному каталозі, впорядкований за часом останнього звернення до них. У PowerShell сортування списку можна виконати за будь-якою властивістю відповідних об’єктів за допомогою командлета Sort-Object, наприклад:

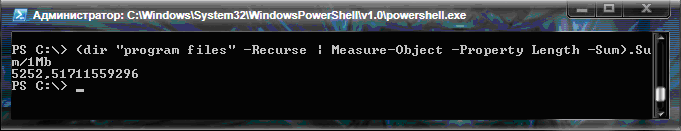


**Определение размера каталогов** Визначення розміру каталогів

юбому каталогу файловой системы в PovverShell соответствует объект типа ystem.io.Directoryinfo, который не имеет свойства, в котором хранился ы размер этого каталога. Будь-якому каталогу файлової системи в PowerShell відповідає об'єкт типу System.IO.DirectoryInfo, який не має властивості, в якій б зберігався розмір цього каталогу. Тому для визначення розміру каталогу потрібно підрахувати загальний розмір файлів, записаних в даному каталозі і всіх його підкаталогах. Зробити це можна шляхом сумування за допомогою командлета Measure-Object значень властивості Length об’єктів-файлів, які постачаються по конвеєру комадлетом dir (Get-ChildItem). Наприклад, наступна команда показує розмір в байтах каталогу [c:\program](file:///C:\program) files (властивість Sum) і кількість файлів та підкаталогів в цьому каталозі (властивість Count):

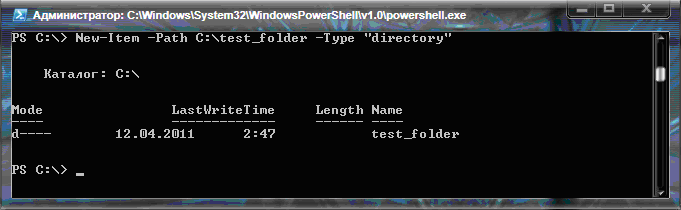
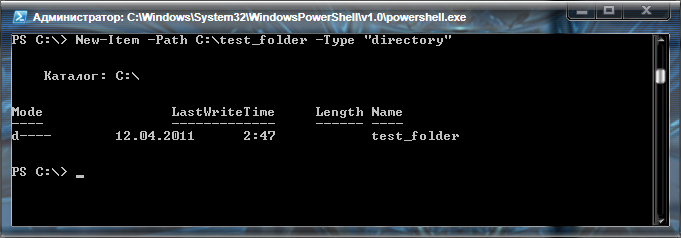


Например, следующая оманда показывает размер в байтах каталога [c:\program](file:///C:\program) files (свойство Sum) поставляе-ых по конвейеру командлетом dir (Get-childitem). пос-их по конвеєру командлетів dir (Get-childitem). Например, следующая оманда показывает размер в байтах каталога [c:\program](file:///C:\program) files (свойство Sum) Наприклад, наступна оманда показує розмір в байтах каталогу [c: \ Program](file:///C:\program) Files (властивість Sum) ели нужно определить размер в мегабайтах или гигабайтах, то значение войства Sum следует разделить на соответствующую константу-суффикс 1Mb или 1Gb): Якщо потрібно визначити розмір в мегабайтах або гігабайтах, то значення властивості Sum слід розділити на відповідну константу-суфікс (1Mb або 1Gb):

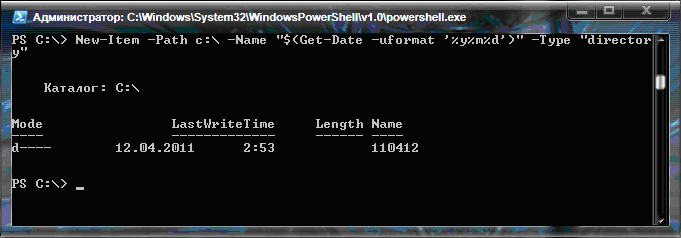


С:\> (dir " [c:\program](file:///C:\program) files" -Recurse | Measure-Object -Property Length -Sum).Sum/IMb >ак видим, результат содержит 11 знаков после запятой. Создание файлов и каталогов Определение размера каталогов Створення файлів і каталогів

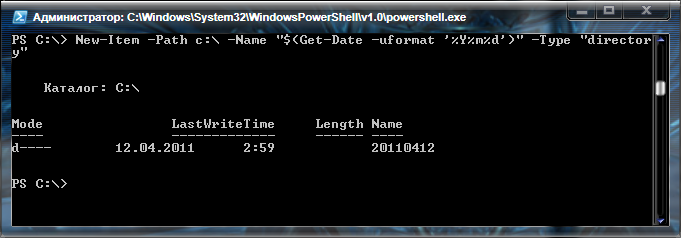
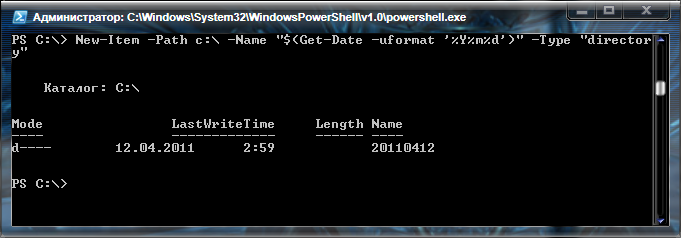
Создать новый файл или каталог в PowerShell позволяет командлет New-item. Створити новий файл або каталог в PowerShell дозволяє командлет New-Item. Путь к создаваемому элементу указывается в виде значения параметра -Path, а в качестве значения параметра -itemType указывается "directory", если нужно создать каталог, и "file", если нужно создать файл. Шлях до створюваного елементу вказується у вигляді значення параметра -Path, а в якості значення параметра -ItemType вказується "directory", якщо потрібно створити каталог, і "file", якщо потрібно створити файл. Например, следующая команда создает на диске с: каталог с именем testfolder: Наприклад, наступна команда створює на диску c: каталог з ім'ям test\_folder:

PS C:\> New-Item -Path [c:\test\_folder](file:///C:\test_folder) -Type "directory" 

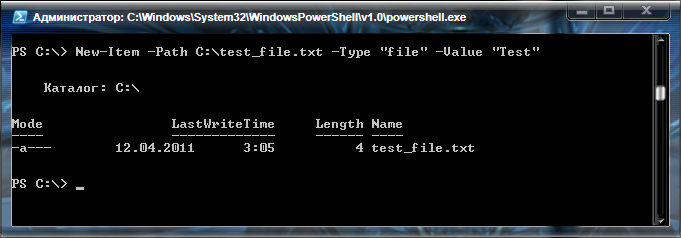
Каталог: Microsoft.PowerShell.Core\FileSystera::C:\ Иногда бывает нужно создавать каталоги, в имени которых присутствовала бы текущая дата в определенном формате. Іноді буває потрібно створити каталоги, в імені яких була б присутня поточна дата в певному форматі. Необхідне текстове представлення дати може бути отримано за допомогою параметра -uformat командлета Get-Date. Наприклад, наступна команда створить на диску с: каталог з ім'ям, яке співпадає з поточною датою в форматі РРММДД:

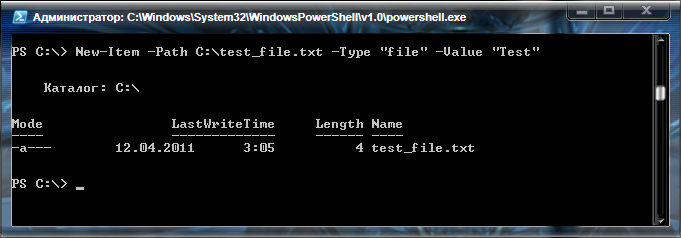
PS C:\> New-Item -Path c:\ -Name "$(Get-Date -uformat '%y%m%d')" -Type "directory"Если в номере года нужно указывать четыре цифры, то следует изменить спецификатор формата на '%T%mtd':

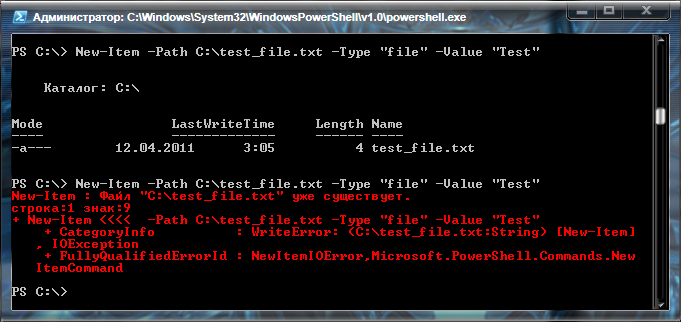
Якщо в номері року потрібно вказувати чотири цифри, то слід змінити специфікатор формату на '%Y%m%d'':

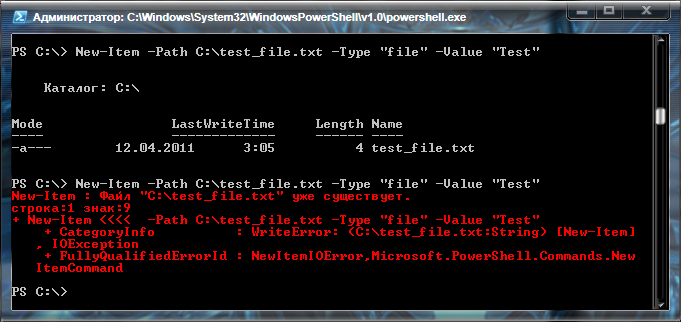


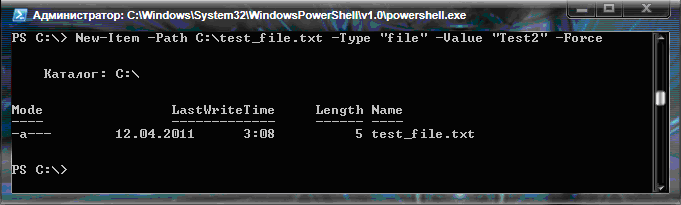
Глава 11 Работа с файловой системой 229 При создании файла в него сразу можно записать строку, указав ее в качестве значения параметра-value, например: При створенні файлу в нього відразу можна записати рядок, зазначивши його у якості значення параметра -Value, наприклад:



Если попытаться назвать создаваемый файл именем уже существующего Якщо спробувати назвати створюваний файл назвою вже існуючого файла, то возникнет ошибка: файлу, то виникне помилка:

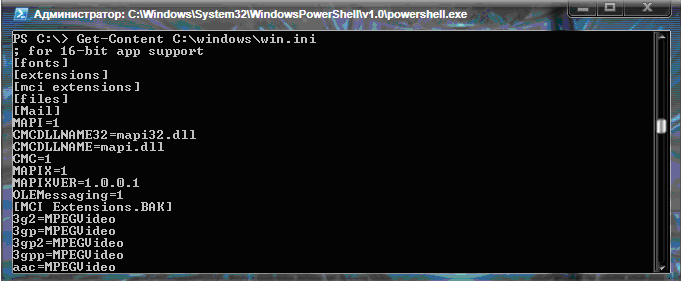


Для перезаписи существующего файла при создании нужно указать параметрДля перезапису існуючого файлу при створенні потрібно вказати параметр-Force: -Force:



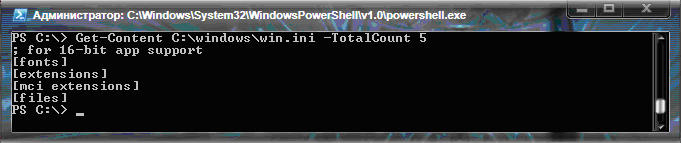
**Создание файлов и каталогов Определение размера каталогов** Читання та перегляд вмісту файлів

В оболочке cmd.exe имеется команда type, которая выводит содержимое тек-, стового файла на экран. В оболонці cmd.exe існує команда type, яка виводить вміст текстовго файлу на екран. В PowerShell команда type є псевдонімом В PowerShell команда type является псевдонимом У PowerShell команда type є псевдонімом командлета командлета Get-Content (другие псевдонимы этого же командлета -- cat | и дс), предназначенного для построчного считывания содержимого гекстово-г го файла с возвращением объекта для каждой строки (при этом строки I отображаются на экране).Get-Content (інші псевдоніми цього ж командлета - cat і gc), призначеного для порядкового зчитування вмісту текстового файлу з поверненням об'єкта для кожного рядка (при цьому рядки відображаються на екрані). НаприкладНапример: Наприклад:

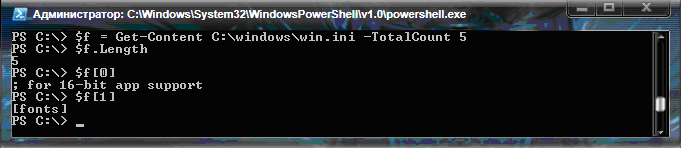


Параметр -Encoding командлета Get-Сontent дозволяє явно вказувати кодування файлу для коректної обробки його вмісту. Допустимые значения данного параметра: Unicode, Byte, BigEndianUnicode, UTF8, UTF7, Ascii. Допустимі значення даного параметра: Unicode, Byte, BigEndianUnicode, UTF8, UTF7, Ascii.

По умолчанию командлет Get-Content считывает все строки из файла; их количество можно ограничить с помощью параметра -TotalCount. За замовчуванням командлет Get-Content зчитує всі рядки з файлу; їх кількість можна обмежити за допомогою параметра -TotalCount. Например. Наприклад. следующая команда считывает первые пять строк из файла [c:\windows\win.ini](file:///C:\windows\win.ini) : наступна команда зчитує перші п’ять рядків із файлу [c:\windows\win.ini](file:///C:\windows\win.ini):



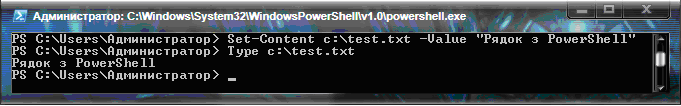
Результат выполнения комаидлета Get-Content можно сохранять в переменной. Результат виконання командлета Get-Content можна зберігати в змінній, обращаясь в дальнейшем к ней, как к массиву строк, например: звертаючись надалі до неї, як до масиву рядків, наприклад:



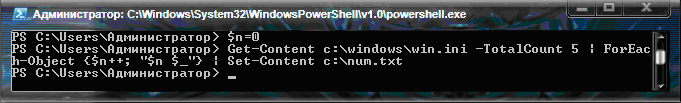
5 Разумеется, возвращаемый .массив строк можно сразу передавать по конвейеру для обработки другими командлетами, скажем, для какого-либо преобразования и записи в другой файл (соответствующий пример приведен в следующем разделе). Зрозуміло, що масив рядків можна відразу передавати по конвеєру для обробки іншими командлетами, скажімо, для якого-небудь перетворення та запису в інший файл.

Запис файлів

Ранее в главе 5 мы уже рассматривали возможность записи данных во внешние файлы с помощью операторов перенаправления (> и >>) и комаидлета Out-File. Щоб записати в файл текстову інформацію (без додаткового форматування) краще скористатися командлетом Set-Content. Данные для записи в файл могут задаваться в качестве значения параметра -value.Дані для запису в файл можуть задаватися в якості значення параметра -value. Например, следующая команда записывает в файл [c:\test.txt](file:///C:/test.txt) строку "Строка из PowerShell": Наприклад, наступна команда записує у файл [c:\test.txt](file:///C:/test.txt) наступне "Рядок з PowerShell". Перевіримо відразу вміст файлу:

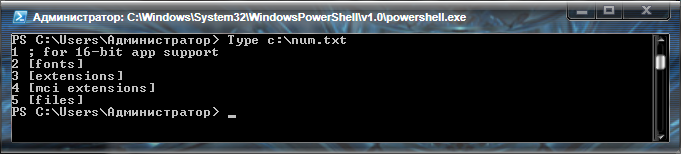


Массив записываемых строк может приниматься по конвейеру от других командлетов. Масив рядків може прийматися по конвеєру від інших командлетів. В качестве примера мы считаем пять первых строк из файла [c:\windovvs\win.ini](file:///C:/windovvs/win.ini) и запишем их в файл [c:\iiiim.txt](file:///C:/iiiim.txt) . В якості приклада зчитаємо п'ять перших рядків з файлу [c:\windows\win.ini](file:///C:/windovvs/win.ini) і запишемо їх в файл [c:\num.txt](file:///C:/iiiim.txt), ддобавив перед каждой строкой ее порядковый номер.одавши перед кожним рядком його порядковий номер. Для этого сначала обнулим переменную $п (счетчик строк): Для цього спочатку занулим змінну $n(лічильник рядків) і PS С:\> $п=0Теперь выполним следующий конвейер комавиконаємо наступний конвеєр команд:



Командлет Get-Content в этом конвейере считывает строки из файла [c:\windo\vs\win.ini](file:///C:/windo/vs/win.ini) . Командлет Get-Content в цьому конвеєрі зчитує рядки з файлу [c:\windows\win.ini](file:///C:/windo/vs/win.ini), которые по очереди обрабатываются командлетом ForEach-object.які по черзі обробляються командлетом ForEach-Оbject. Для каждой обрабатываемой строки значение переменной $п увеличивается на единицу, после чего в выходной поток помешается значение. Для кожного оброблюваного рядка значення змінної $n збільшується на одиницю, після чого у вихідний потік поміщається значення, яке вычисляемое при расширении строки "$п $\_".обчислюється при розширенні рядка "$n $\_". Затем полученная эока передается по конвейеру командлету Set-Content, который добавляет ее в файл [c:\num.txt](file:///C:/num.txt) . Потім отриманий рядок передається по конвеєру командлетом Set-Content, який додає його у файл [c:\num.txt](file:///C:/num.txt) .

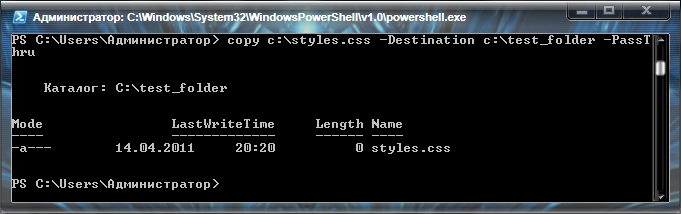
Убедимся, что в файле [c:\num.txt](file:///C:/num.txt) записаны нужные строки: PS C:\> type [c:\num.txt](file:///C:/num.txt) | ; for 16-bit app support Переконаємося, що у файлі [c:\num.txt](file:///C:/num.txt) записані потрібні рядки:



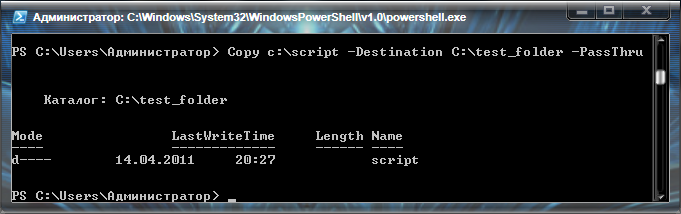
Если файл, в который производится запись, уже существует, то командлет Set-Content заменит его содержимое.Якщо файл, в який проводиться запис, вже існує, то командлет Set-Content замінить його вміст. В случае, когда требуется добавить строки в конец существующего файла, следует воспользоваться командлетом Add-Content. У випадку, коли потрібно додати рядки в кінець існуючого файлу, слід скористатися командлетом Add-Content.

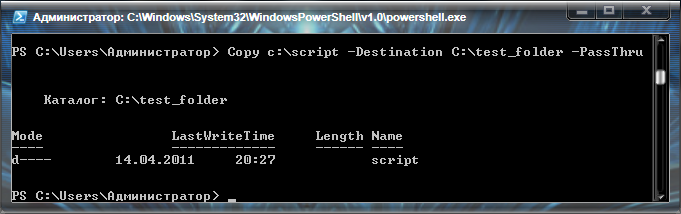
Копирование файлов и каталоговКопіювання файлів і каталогів

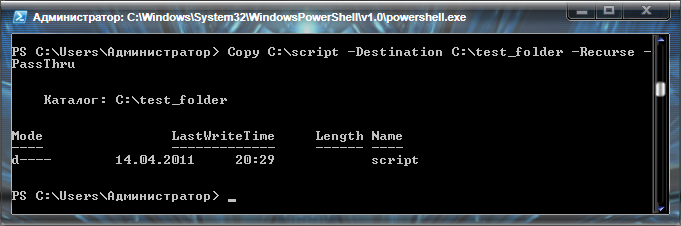
В PowerShell копирование файлов и каталогов осуществляется командлетом Copy-item, имеющим псевдоним сору. У PowerShell копіювання файлів і каталогів здійснюється командлетом Copy-Item, який має псевдонім copy. Путь к копируемым файлам при этом указывается в качестве значения параметра -Path (данный параметр используется по умолчанию), а путь к целевому каталогу, в который нужно скопировать файлы, задается значением параметра -Destination. Шлях до копійованих файлів при цьому вказується в якості значення параметра -Path(даний параметр використовується за замовчуванням), а шлях до цільового каталогу, в який потрібно скопіювати файли, задається значенням параметра -Destination. Например. Наприклад, следующая команда скопирует файл styles.css с корневого каталога диска с: в каталог [c:\test\_folder](file:///C:/test_folder) : наступна команда скопіює файл styles.css з кореневого каталогу диска с: в каталог [c:\test\_folder](file:///C:/test_folder). Відразу ж переглянемо результат виконання команди копіювання за допомогою параметра -PassThru:

 PS C:\> copy [c:\styles.css](file:///C:/styles.css) -Destination [c:\test\_folder](file:///C:/test_folder) -PassThru

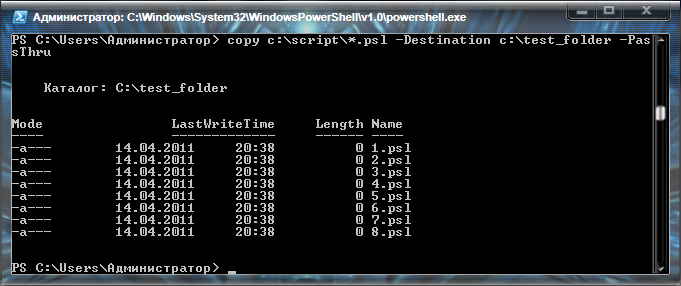
Если путь к копируемым объектам указывает на каталог, то по умолчанию будет скопирован только этот каталог без своего содержимого (этим Power-Shell отличается от большинства других оболочек, в том числе от cmd.exe). Якщо шлях до об'єктів, які копіюються, вказує на каталог, то по замовчуванню буде скопійований тільки цей каталог без свого вмісту (цим Power-Shell відрізняється від більшості інших оболонок, у тому числі від cmd.exe). Например: Наприклад:



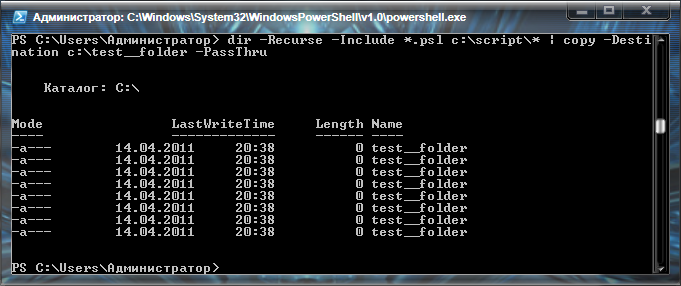
Параметр -Recurse позволяет копировать содержимое вложенных каталогов. Параметр -Recurse дозволяє копіювати вміст вкладених каталогів. например: наприклад:

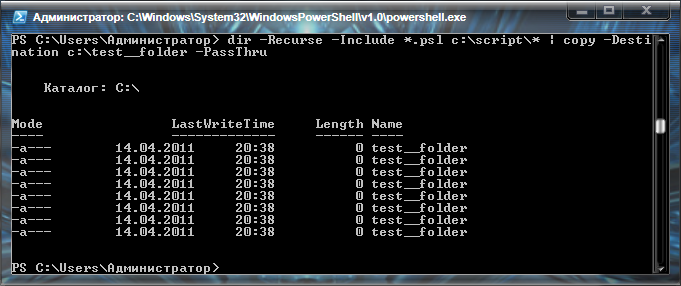


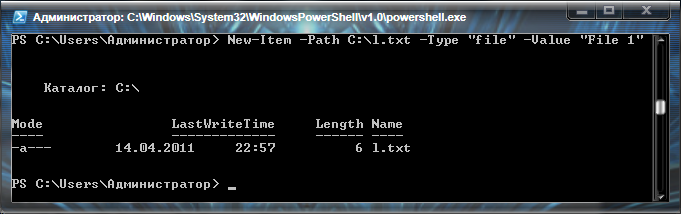
\*ожмо копировать не все файлы т каталога, а только соответствующие определенной маске. Можна копіювати не всі файли із каталогу, а тільки відповідні певній масці. При этом маску можно укачан, внутри пути для копирования или в качестве значения параметра -include. При цьому маску можна вказати всередині шляху для копіювання або в якості значення параметра -Іnclude. Например, следующая команда копирует все файлы с расширением psl нч каталога [c:\scripl](file:///C:/scripl) в папку ■Atestfoldcr: Наприклад, наступна команда копіює всі файли з розширенням psl из каталогу [c:\scrip](file:///C:/scripl)t в папку c:\test\_foldеr:

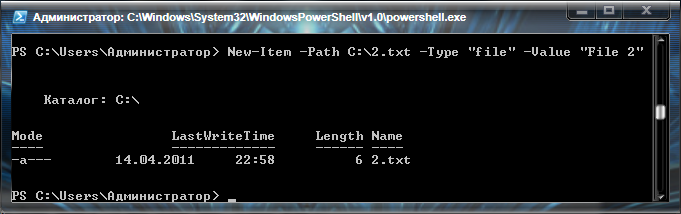
)днако если необходимо скопировать файлы по маске из всех вложенных подкаталогов, то одним командлетом Copy-item обойтись не удастся.

Однак якщо необхідно скопіювати файли по масці з усіх вкладених підкаталогів, то одним командлетом Copy-Іtem обійтися не вдасться. Предварительно требуемые файлы нужно получить командлетом Get-Childitem (dir). Попередньо необхідні файли потрібно отримати командлетом Get-ChildІtem(dir),а затем передать их командлету Copy-item по конвейеру. а потім передати їх командлету Copy-Іtem по конвеєру. Например. Наприклад,следующая команда копирует все файлы с расширением psi из каталога [c:\script](file:///C:/script) и всех его подкаталогов в папку [c:\tcst\_folder](file:///C:/tcst_folder) : наступна команда копіює всі файли з розширенням psl з каталогу [c:\script](file:///C:/script) і всіх його підкаталогів в папку [c:\test\_folder](file:///C:/tcst_folder) :

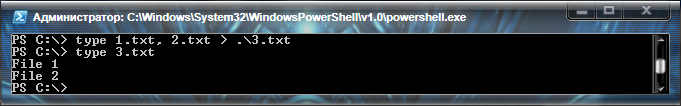


Команда copy оболочки cmd.exe позволяла объединять несколько файлов вместо (конкатенация файлов).Команда copy оболонки cmd.exe дозволяла об'єднувати декілька файлів замість(конкатенації файлів). В PowerShell объединить файлы можно с помощью комаидлета Get-Content (псевдоним type) и перенаправления вывода в результирующий файл. У PowerShell об'єднати файли можна за допомогою командлета Get-Content(псевдонім type) і перенаправлення виводу в результуючий файл. Рассмотрим пример. Розглянемо приклад. Создадим файлы l.txt и 2.txt: Створимо файли l.txt і 2.txt:





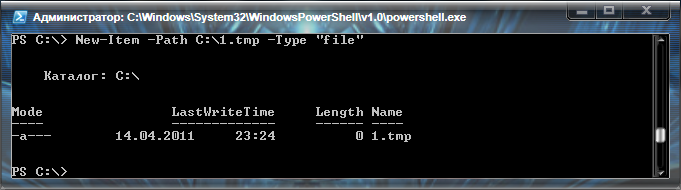
Следующая команда объединяет файлы 1 .txt и 2.txt в файл 3.txT. Об'єднаємо файли 1.txt і 2.txt в файл 3.txt і перевіримо його вміст:



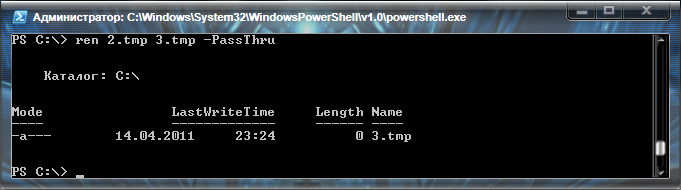
Как видим, конкатенация файлов 1 .txt и 2.txt выполнена успешно.Як бачимо, конкатенація файлів 1.txt і 2.txt виконана успішно.

Перейменування і переміщення файлів і каталогів

Переименовать файл или каталог можно с помощью команллета Rename- .Перейменувати файл або каталог можна за допомогою команллета Rename-Item (псевдоним геп).псевдонім ren). Значение параметра -Path этого команллета задает путь к элементам для переименования, а значение параметра -NewName-- новое имя. Значення параметра -Path цього команллета задає шлях до елементів для перейменування, а значення параметра -NewName - нове ім'я. Имена этих параметров можно опускать (в этом случае первым должно Імена цих параметрів можна опускати (у цьому випадку першим має [Называться значение параметра -Path).вказуватися значення параметра -Path). Например, создадим файл [c:\l.lmp](file:///C:/l.lmp) переименуем его в файл 2.tmp: Наприклад, створимо файл [c:\l.tmp](file:///C:/l.lmp) перейменуємо його у файл 2.tmp:

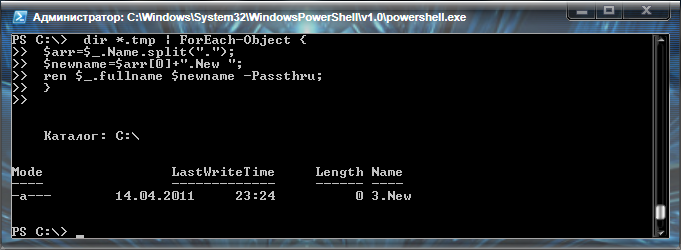


Для того щоб побачити результат дії команллета Rename-Itemнужно казать параметр -PassThru: потрібно вказати параметр -PassThru:

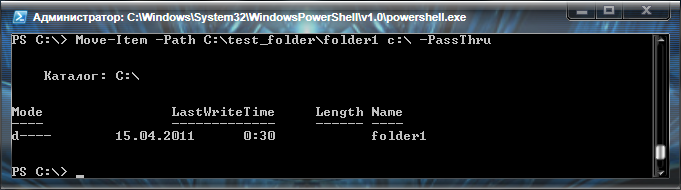


отличие от команды ren оболочки cmd.exe командлет Rename-item не поддерживает подстановочные знаки в значении параметров -Path или -Newr; 1оэтому для переименования группы файлов придется выполнить несколько дополнительных действий. На відміну від команди ren оболонки cmd.exe командлет Rename-item не підтримує символи підстановки у значенні параметрів -Path або –NewName. Тому для перейменування групи файлів доведеться виконати кілька додаткових дій. В качестве примера мы переименуем все файлы ! На приклад ми перейменуємо всі файли з расширением imp на диске с:, назначив им новое расширение new.розширенням tmp на диску с:, призначивши їм нове розширення new.

1Я этого вначале с помощью команллета Get-childltem (dir) формиру-:я коллекция нужных файлов, каждый элемент которой обрабатываетсядлдл Для цього спочатку за допомогою команллета Get-СhildІtem(dir) формується колекція потрібних файлів, кожен елемент якої обробляється комапдлетом ForEach-object.командлетом ForEach-Оbject. Для текущего файла с расширением Imp выделяется его имя без расширения (после вычисления выражения $arr=S\_.Name.split (".") имя файла будет храниться в первом элементе массива Sarr). Для поточного файлу з розширенням tmp виділяється його ім'я без розширення (після обчислення виразу $arr = S\_.Name.split (".") ім'я файлу буде зберігатися в першому елементі масиву $arr), составляется новое имя файла ($newname=$arr [0]+".new") и выполняется командлет Rename-Item (ren): складається нове ім'я файлу ($newname=$arr[0]+".new") і виконується командлет Rename-Item(ren):



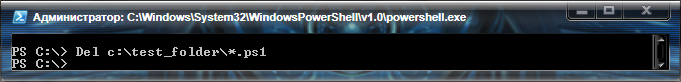
Командлет Rename-item позволяет лишь переименовывать файлы или каталоги, а не перемещать их. Командлет Rename-Item дозволяє лише перейменовувати файли або каталоги, а не переміщувати їх. Если нужно переместить файл или каталог в другую папку, то следует воспользоваться командлетом Move-item (псевдоним move). Якщо потрібно перемістити файл або каталог в іншу папку, то слід скористатися командлетом Move-Item (псевдонім move). Значение параметра -Path данного командлета задает путь к файлам или каталогам для перемещения (в этом пути допускается использование подстановочных знаков), а значение параметра -Destination -- путь к каталогу, куда будут перемещены эти файлы или каталоги. Значення параметра -Path даного командлету задає шлях до файлів або каталогів для переміщення (в цьому шляху допускається використання символів підствновки), а значення параметра -Destination - шлях до каталогу, куди будуть переміщені ці файли чи каталоги. Результат перемещения можно увидеть на экране, указав параметр -PassThru. Результат переміщення можна побачити на екрані, вказавши параметр -PassThru. Например, следующая команда переносит в корневой каталог диска с: каталог [c:\test\_loIder\folderl](file:///C:/test_loIder/folderl) со всем его содержимым: Наприклад, наступна команда переносить у кореневий каталог диска с: каталог [c:\test\_folder\folder](file:///C:/test_loIder/folderl)1 з усім його вмістом:



Удаление файлов и каталогов Видалення файлів і каталогів

Удалять объекты файловой системы можно с помощью командлета Remove-Item (псевдоним del). Видаляти об'єкти файлової системи можна за допомогою командлета Remove-Item(псевдонім del). Значення параметра –Path цього командлета задає шлях до файлів та каталогів, які потрібно видалити(ім’я параметра в команді можна не вказувати). В шляху допускаються символи підстановки. Крім того, команд лет Remove-Item має параметр –Include, значення якого задає файли, на які буде діяти команда, і параметр –Exclude, який задає файли-винятки, які видалятись не будуть.

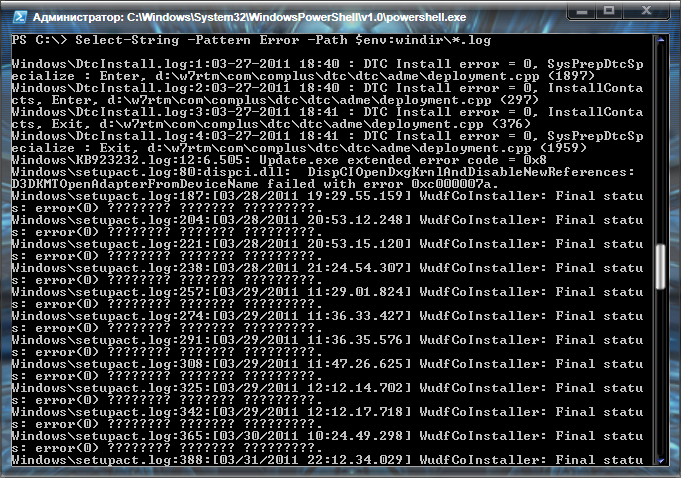
Наприклад, наступна команда видалить всі файли з розширенням psl в каталозі c:\test\_folder:

****

Пошук тесту в файлах

В приниципі для пошуку тексту в файлі можна зчитати його вміст в масив з допомогою командлета Get-Content і примінити до них один із операторів порівняння. Однак в PowerShell існує командлет Select-String, спеціально призначений для пошуку рядків текста в одному або декількох файлах(аналогічною утилітою в cmd.exe являється findstr).

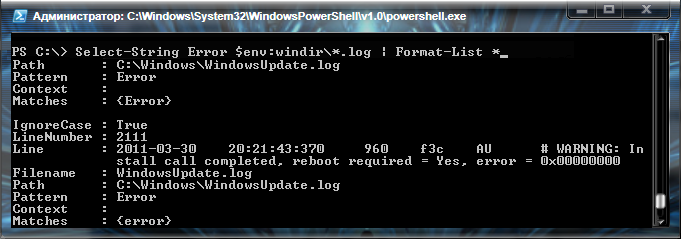
Наприклад, наступна команда шукає слово Error в усіх файлах з розширенням log Windows(імена параметрів –Pattern і –Path можна опустити, не забуваючи, що шаблон пошуку повинен стояти після командлета першим):



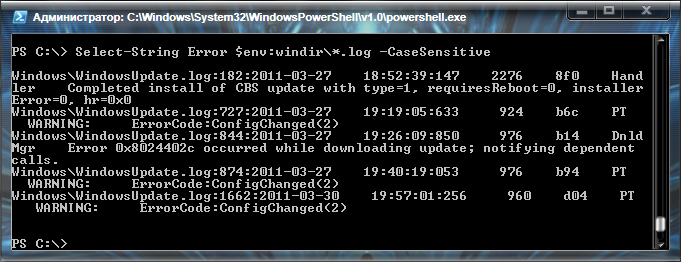
Як бачимо, на екрані по замовчуванню виводиться:

* шлях до файлу, що перевіряється;
* номер рядка, в якому знайдено співпадання;
* сам рядок.

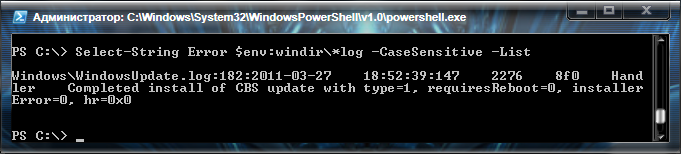
Насправді результат пошуку є об’єктом типу Microsoft.PowerShell.Commands.MathchInfo, який містить властивість Path(шлях до файлу), Pattern(шаблон, по якому відбувається пошук), LineNumber(номер знайденого рядка в файлі), Line(вміст знайденого рядку) та інше:



Якщо при пошуку потрібно врахувати регістр символів, то слід вказати параметр –CaseSensitive:

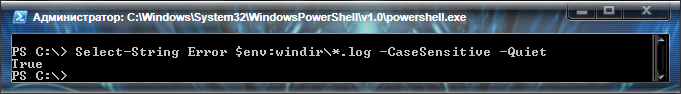


За замовчуванням командлет Select-String знаходить в файлах всі відповідності для рядків. У випадку необхідності можна обмежитись лише першою знайденою відповідність –List, наприклад:

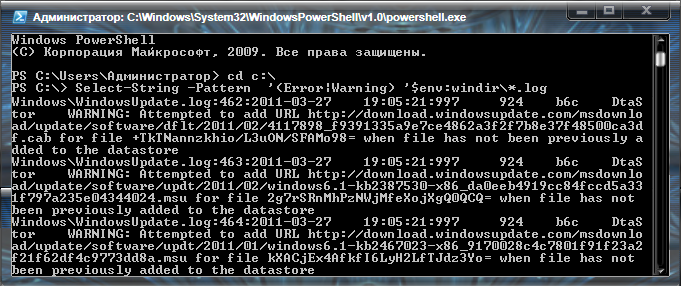


В результаті ми бачимо, що для кожного файлу на екран виводиться лише по одному рядку, який містить слово Error.

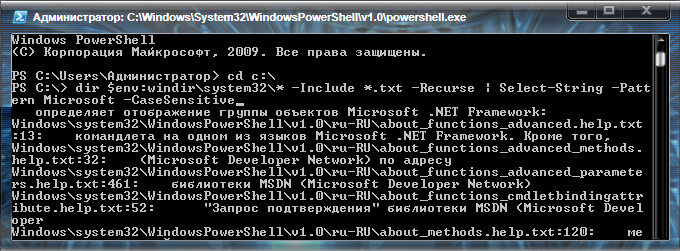
В тому випадку, коли потрібно визначити сам факт наявності потрібного рядка в файлах і неважливо, на якій позиції знаходиться цей рядок, то можна використовувати параметр –Quiet. Командлет повертає $True, якщо пошук завершено успішно, і $False в протилежному випадку:



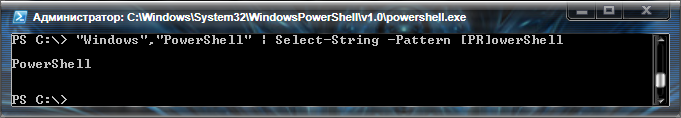
В якості шаблону для пошуку можна використовувати регулярні вирази. Наприклад, наступна команда буде шукати в файлах з розширенням log в системному каталозі Windows рядки, які містять слова Error або Warning:



Файли, в яких відбувається пошук, можуть не лише вказуватися в якості значення параметра –Path, але і надходити командлету Select-String по конвеєру. Наприклад, наступна команда шукає слово Microsoft в усіх файлах з розширенням txt, які зберігаються в каталозі Windows\System32 і в усіх його підкаталогах:



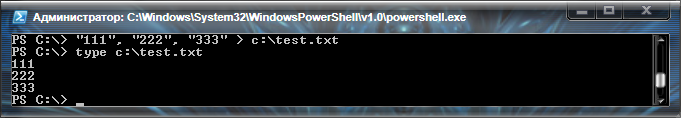
Також по конвеєру можна передавати масив рядків, в якому буде відбуватись пошук, наприклад:



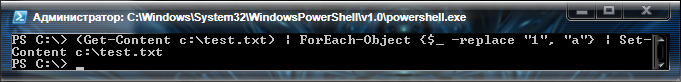
Заміна тексту в файлах

Заміна тексту в рядках здійснюється оператором –replace, який підтримує регулярні вирази. Тому для заміни тексту в файлі можна зчитати його вміст в масив рядків за допомогою командлета Get-Content, обробити потрібні рядки оператором –replace і зберегти результат в тому ж або у іншому файлі, використовуючи командлет Set-Content.

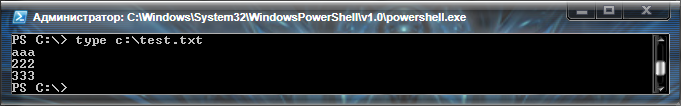
Розглянем простий приклад. Створимо файл c:\test.txt із трьох рядків:

****

Тепер замінимо в цьому файлі всі символи “1” на “а”:



Перша команда в конвеєрі взята в дужки, для того щоб команд лет Set-Content почав працювати лише після того, як командлет Get-Content повністю прочитає файл(в протилежному випадку може виникнути помилка при одночасному звертанні до файлу с:\test.txt при зчитуванні і записі). Переконаємся, що заміна відбулась успішно:



# Література

# Введение в Windows PowerShell/ А.Попов. – БХВ-Петергург, 2009. – 464с

# Руководство по сценариям Windows PowerShell/ Уилсон Эд; пер. С англ. Розовского М. – Эком, 2009. – 704с

# Windows PowerShell 2.0. Справочник администратора/ Уильям Р.Станек – Русская Редакция, 2010. – 416с

4. [http://uk.wikipedia.org](http://uk.wikipedia.org/)