

Introduction to Regular Expressions

A regular expression is a way of describing data by what it looks like:

- \\chi-fp01\public
- 172.16.30.203Globomantics\jeff

PowerShell supports the full .NET Regular Expression library

Regular expressions can be case sensitive

Full coverage of regular expressions is outside the scope of this course

TRAINSIGNAL

Regular Expression Operators Wildcard - Like - NotLike Regular Expression - Match - NotMatch True or False results TRAINSIGNAL

Wildcard Matching

Compare strings

Comparison string uses wildcard characters
Useful for simple matching

```
PS C:\> "Powershell" -like "*shell"
True
PS C:\> "Powershell" -notlike "cmd*"
True
```

TRAINSIGNAL

Matching

Can match on simple strings
Can match on a regular expression pattern
Matches "float" unless anchored
Matching creates a MatchInfo object
Match stops at first non-matching character

```
PS C:\> "Powershell" -match "shell"
True
PS C:\> "Powershell" -notmatch "cmd"
True
```

TRAINSIGNAL

Regular Expression Basics

Matches on character sets

Match on any single character .

Match on a single set of characters [abc]

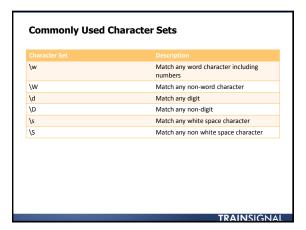
Match on a range of characters [m-z]

Anchor starting at the beginning ^

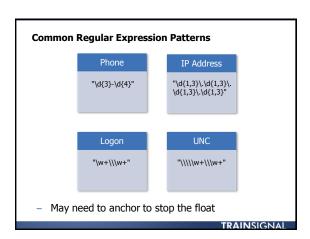
Anchor from the end \$

The regular expression escape character \

TRAINSIGNAL



Quantifier	Description
+	Match repeating instances of preceding character
*	Match zero or more instances of preceding character
?	Match 0 or 1 instances of preceding character
{n}	Match exactly n number of preceding characters
{n,}	Match at least n number of preceding characters
{n,m}	Match at least n number of preceding characters and no more than m



Introducing the Regex Object Create with the [regex] type accelerator Find multiple matches Is case sensitive PS C:\> [regex]\$rx = "^\\\CHI-\w+\d{2}\\\w+\$" PS C:\> \$rx.match(\$name)

TRAINSIGNAL

Regex in Action

Groups : {\\CHI-FP01\public}
Success : True
Captures : {\\CHI-FP01\public}
Index : 0
Length : 17
Value : \\CHI-FP01\public

Select-String Find text in strings and files Use simple or complex patterns Can be case-sensitive Can find multiple matches Can display context Writes a MatchInfo object to the pipeline PS C:\> dir c:\work*.txt | select-string "globomantics" | select filename TRAINSIGNAL **Select-String in Action**

Lab

- 1. Create a regular expression pattern for a phone number and test it with a variety of numbers, good and bad.
- 2. Create a text file of phone numbers and use Select-String to find all numbers that match.
- 3. Repeat #2 but find all numbers that do not match.

TRAINSIGNAL