Read in a Comma-Separated Values File

The Import-Csv cmdlet provides a way for you to read in data from a comma-separated values file (CSV) and then display that data in tabular format within the Windows PowerShell console. For example, suppose you have a file named C:\Scripts\Test.txt, a file that contains the following data:

[crayon-6a9d78768e317714367165/]

You say you'd like to view that data on-screen, and in a table, to boot? Then just call Import-Csv followed by the path to the file you want to open:

[crayon-6a9d78768e31d581732255/]

In turn, Windows PowerShell will give you output that looks like this:

[crayon-6a9d78768e320587838626/]

That's cool. But if you really want to have some fun – well, good point: if you really want to have some fun you'd probably do something other than import CSV files. Let's put it this way: if you want to have some relative fun, pipe the imported data to the Where-Object cmdlet. For example, this command retrieves only those employees who work for the Finance department:

[crayon-6a9d78768e322950747123/]

Here's what the resulting dataset looks like:

[crayon-6a9d78768e323681877983/]

Granted, the right side of the pipeline is bit cryptic looking. However, it's not all that hard to interpret. The $_

is simply Windows PowerShell notation for the current object; the current object, of course, is the text file C:\Scripts\Test.txt. The .department is standard «dot» notation; this is no different than referring to a WMI property by saying something like objItem.Name. All we're saying is that, for the text file in question, we want to check the value of the Department property (i.e., the Department field). If the field is equal to (-eq) Finance we want to see it; if it's not equal to Finance then we don't want to see it.
Suppose we wanted to see the records for all employees except those from the Finance Department. In that case we'd use this command, with -ne meaning «not equal to»:

[crayon-6a9d78768e325989789667/]

That command returns the following data:

[crayon-6a9d78768e326288685155/]

Can you have more than one item in a where clause? Sure: just combine the two criteria with -and or -or. For example, this command returns only accountants who work in the Finance department:

[crayon-6a9d78768e328054444862/]

What does that mean? That means you'll get back information like this:

[crayon-6a9d78768e32a146433797/]

Meanwhile, this command returns a list of employees who either work in the Research department or are accountants:

[crayon-6a9d78768e32b731922553/]

You can probably guess what information we get back when we run that command:

[crayon-6a9d78768e32d239245212/]