

Reading in an XML File

If the Export-Clixml cmdlet enables you to save data as an XML file, then what do you suppose the Import-Clixml cmdlet lets you do? OK, so maybe it wasn't all that tough after all. At any rate, you're right: Import-Clixml will import an XML file that was saved using Export-Clixml and then rebuilds the Windows PowerShell object for you. For example, suppose you use the Get-ChildItem cmdlet to retrieve a collection of files found in the folder C:\Scripts; you then use the Export-Clixml cmdlet to save that information as an XML file:

```
[crayon-6a9d828cb0aac962549592/]
```

Any time you feel like it you can retrieve that data using Import-Clixml; simply call the cmdlet followed by the name of the file to import. For example, this command retrieves the data and stores it in a variable named \$A:

```
[crayon-6a9d828cb0ab5657721095/]
```

So what will \$A be equal to? It will be equal to the collection of files that were in C:\Scripts at the time you originally ran Get-ChildItem. Take a look at our Windows PowerShell session:

Windows PowerShell

The best part is that \$A is not simply a collection of string data, but an actual Windows PowerShell object. How do we know that? Try piping \$A through the Sort-Object cmdlet and see what happens:

```
[crayon-6a9d828cb0ab7643853180/]
```