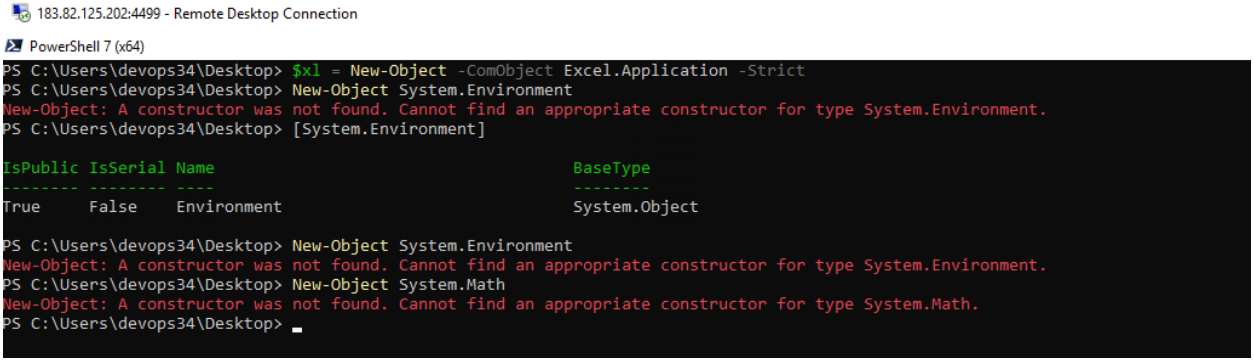
Using Static Classes and Methods

Not all .NET Framework classes can be created by using `New-Object`. For example, if you try to create a **System.Environment** or a **System.Math** object with `New-Object`, you will get the following error messages:

PowerShellCopy
```
New-Object System.Environment
```
PowerShellCopy
```
New-Object System.Math
```
OutputCopy



These errors occur because there is no way to create a new object from these classes. These classes are reference libraries of methods and properties that do not change state. You don't need to create them, you simply use them. Classes and methods such as these are called **static classes** because they are not created, destroyed, or changed. To make this clear we will provide examples that use static classes.

Getting Environment Data with System.Environment

Usually, the first step in working with an object in Windows PowerShell is to use Get-Member to find out what members it contains. With static classes, the process is a little different because the actual class is not an object.

Referring to the Static System.Environment Class

You can refer to a static class by surrounding the class name with square brackets. For example, you can refer to **System.Environment** by typing the name within brackets. Doing so displays some generic type information:

PowerShellCopy
```
[System.Environment]
```

As we mentioned previously, Windows PowerShell automatically prepends '**System.**' to type names when you use `New-Object`. The same thing happens when using a bracketed type name, so you can specify **[System.Environment]** as **[Environment]**.

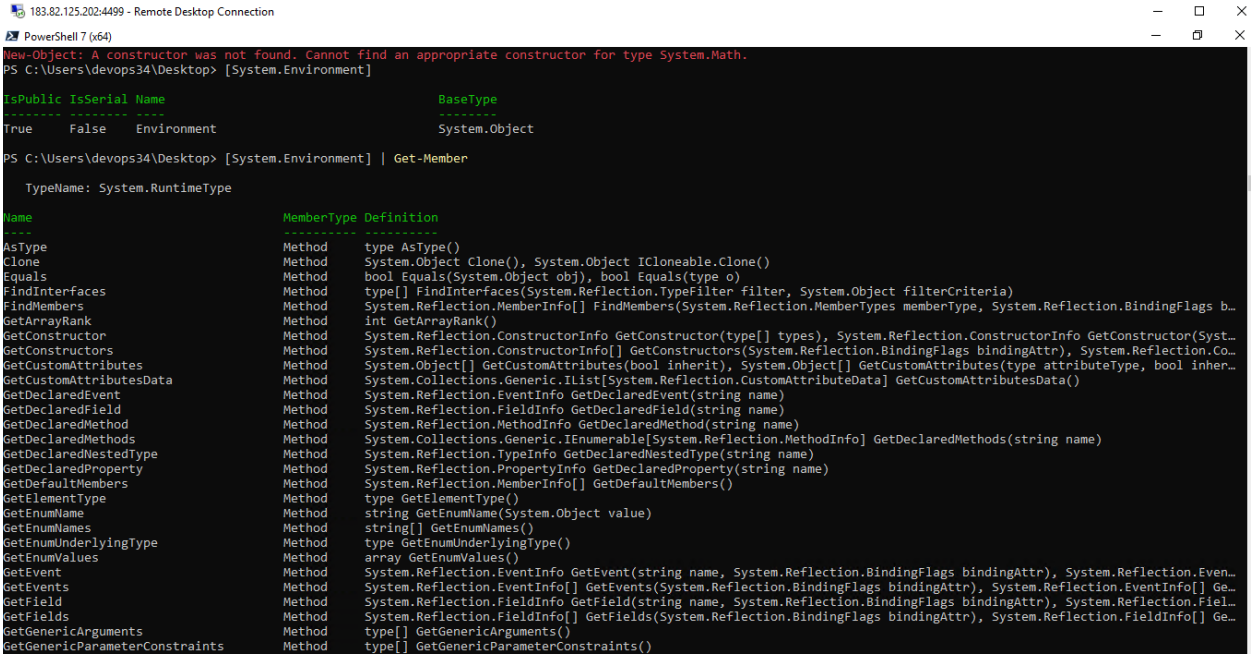
The **System.Environment** class contains general information about the working environment for the current process, which is `powershell.exe` when working within Windows PowerShell.

If you try to view details of this class by typing **[System.Environment] | Get-Member**, the object type is reported as being **System.RuntimeType** , not **System.Environment**:

PowerShellCopy

```
[System.Environment] | Get-Member
```

Output :



PowerShellCopy

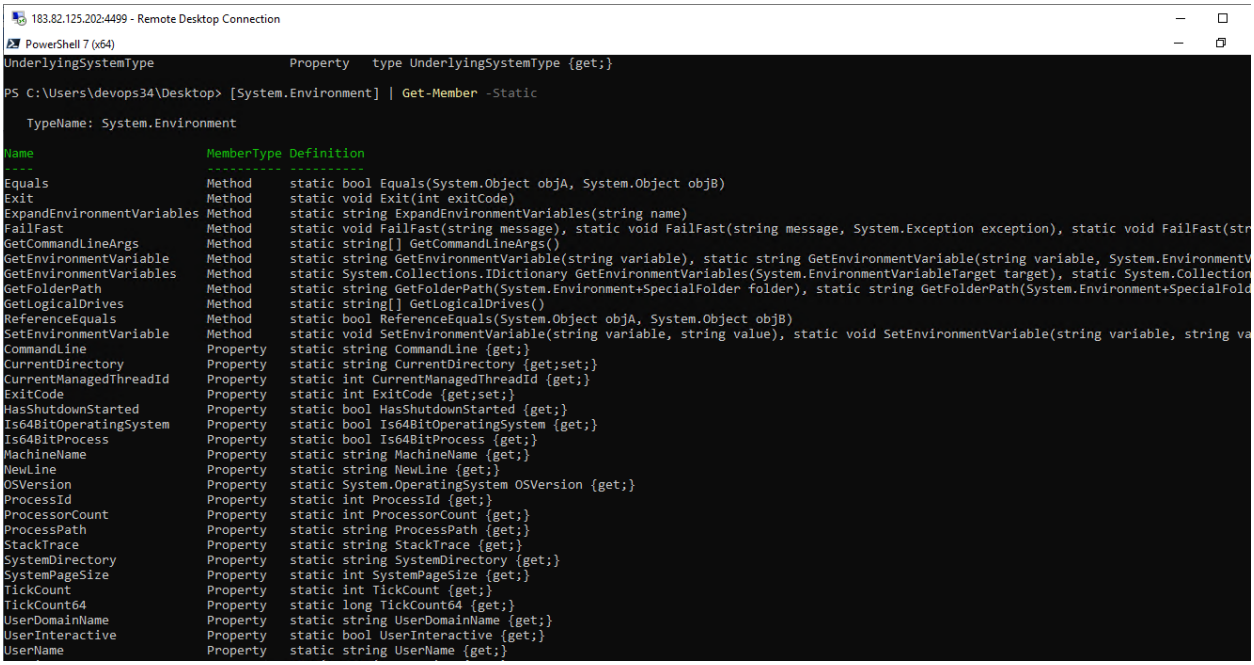
```
[System.Environment] | Get-Member
```

To view static members with Get-Member, specify the **Static** parameter:

PowerShellCopy

```
[System.Environment] | Get-Member -Static
```

OutputCopy



Displaying Static Properties of System.Environment

The properties of System.Environment are also static, and must be specified in a different way than normal properties. We use `::` to indicate to Windows PowerShell that we want to work with a static method or property. To see the command that was used to launch Windows PowerShell, we check the **CommandLine** property by typing:

PowerShellCopy
```
[System.Environment]::Commandline
```

To check the operating system version, display the OSVersion property by typing:

PowerShellCopy
```
[System.Environment]::OSVersion
```

We can check whether the computer is in the process of shutting down by displaying the **HasShutdownStarted** property:

PowerShellCopy
```
[System.Environment]::HasShutdownStarted
```
OutputCopy: