

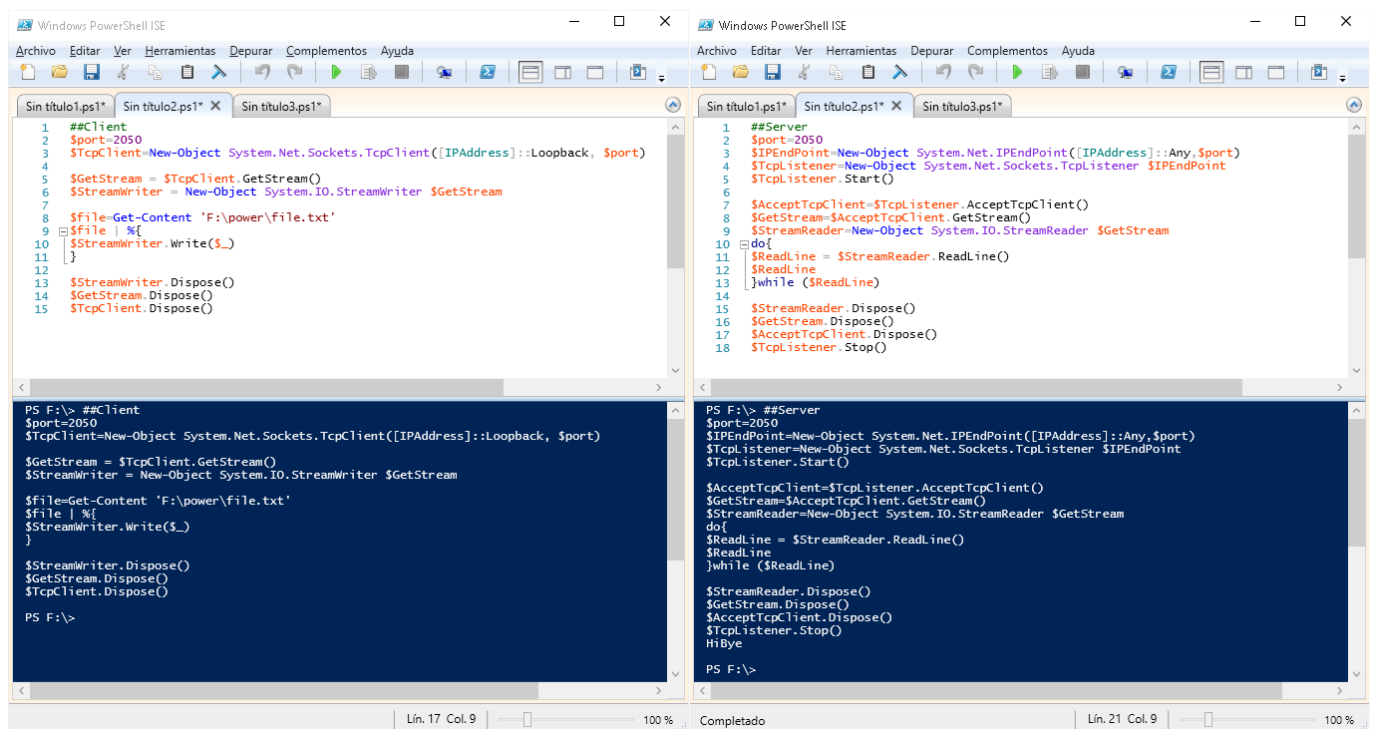
Transfer content file between server and client using PowerShell (Sockets TCP)

Server

[crayon-6a9d5fdd3903c745409310/]

Client

[crayon-6a9d5fdd39042747535817/]



The image displays two side-by-side screenshots of the Windows PowerShell ISE (Integrated Scripting Environment) interface. The left window shows the code for a client script, and the right window shows the code for a server script. Both scripts are written in PowerShell and use the .NET Framework's Socket classes to establish a TCP connection and transfer a file.

Client Script (Left Window):

```
1 ##Client
2 $port=2050
3 $TcpClient=New-Object System.Net.Sockets.TcpClient([IPAddress]::Loopback, $port)
4
5 $GetStream = $TcpClient.GetStream()
6 $StreamWriter = New-Object System.IO.StreamWriter $GetStream
7
8 $File=Get-Content 'F:\power\file.txt'
9 $File | %{
10     $StreamWriter.Write($_)
11 }
12
13 $StreamWriter.Dispose()
14 $GetStream.Dispose()
15 $TcpClient.Dispose()
```

Server Script (Right Window):

```
1 ##Server
2 $port=2050
3 $IPEndPoint=New-Object System.Net.IPEndPoint([IPAddress]::Any, $port)
4 $TcpListener=New-Object System.Net.Sockets.TcpListener $IPEndPoint
5 $TcpListener.Start()
6
7 $AcceptTcpClient=$TcpListener.AcceptTcpClient()
8 $GetStream=$AcceptTcpClient.GetStream()
9 $StreamReader=New-Object System.IO.StreamReader $GetStream
10
11 do{
12     $ReadLine = $StreamReader.ReadLine()
13     $ReadLine
14 }while ($ReadLine)
15
16 $StreamReader.Dispose()
17 $GetStream.Dispose()
18 $AcceptTcpClient.Dispose()
19 $TcpListener.Stop()
20 HiBye
```