



Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Wait-Debugger'

PS:\>Get-HELP Wait-Debugger -Full

NAME

Wait-Debugger

SYNOPSIS

Stops a script in the debugger before running the next statement in the script.

SYNTAX

Wait-Debugger [<CommonParameters>]

DESCRIPTION

Stops the PowerShell script execution engine at the point immediately after the `Wait-Debugger` cmdlet and waits for a debugger to be attached. This is similar to

using `Enable-RunspaceDebug -BreakAll` in a DSC resource but breaks at a specific point in the script.

> [!CAUTION] > Make sure you remove the `Wait-Debugger` lines after you are done. A running script appears to be > hung when it is stopped at a `Wait-Debugger`.

PARAMETERS

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkId=113216>).

INPUTS

None

You can't pipe objects to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

----- Example 1: Insert breakpoint for debugging -----

```
[DscResource()]
class FileResource
{
    [DscProperty(Key)]
    [string] $Path

    [DscProperty(Mandatory)]
    [Ensure] $Ensure
```

```
[DscProperty(Mandatory)]
```

```
[string] $SourcePath
```

```
[DscProperty(NotConfigurable)]
```

```
[Nullable[datetime]] $CreationTime
```

```
[void] Set()
```

```
{
```

```
$fileExists = $this.TestFilePath($this.Path)
```

```
if ($this.ensure -eq [Ensure]::Present)
```

```
{
```

```
if (! $fileExists)
```

```
{
```

```
$this.CopyFile()
```

```
}
```

```
}
```

```
else
```

```
{
```

```
if ($fileExists)
```

```
{
```

```
Write-Verbose -Message "Deleting the file $($this.Path)"
```

```
Remove-Item -LiteralPath $this.Path -Force
```

```
}
```

```
}
```

```
}
```

```
[bool] Test()
```

```
{
```

```
$present = Test-Path -LiteralPath $this.Path
```

```
if ($this.Ensure -eq [Ensure]::Present)
```

```
{
```

```

        return $present
    }

else
{
    return (! $present)
}

}

[FileResource] Get()
{
    $present = Test-Path -Path $this.Path

    if ($present)
    {
        $file = Get-ChildItem -LiteralPath $this.Path
        $this.CreationTime = $file.CreationTime
        $this.Ensure = [Ensure]::Present
    }
    else
    {
        $this.CreationTime = $null
        $this.Ensure = [Ensure]::Absent
    }

    return $this
}

[void] CopyFile()
{
    # Testing only - Remove before deployment!
    Wait-Debugger

    if (! (Test-Path -LiteralPath $this.SourcePath))

```

```
{  
    throw "SourcePath $($this.SourcePath) is not found."  
}  
  
if (Test-Path -LiteralPath $this.Path -PathType Container)  
{  
    throw "Path $($this.Path) is a directory path"  
}  
  
Write-Verbose "Copying $($this.SourcePath) to $($this.Path)"  
  
Copy-Item -LiteralPath $this.SourcePath -Destination $this.Path -Force  
}  
}
```

RELATED LINKS

Online

Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/wait-debugger?view=powershell-5.1&WT.mc_id=ps-gethelp
Enable-DscDebug