PowerShell Get-Help on command 'Wait-Debugger'

PS C:\Users\wahid> Get-Help Wait-Debugger

```
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).

---------- 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
    }
}
```

REMARKS

To see the examples, type: "get-help Wait-Debugger -examples".

For more information, type: "get-help Wait-Debugger -detailed".

For technical information, type: "get-help Wait-Debugger -full".

For online help, type: "get-help Wait-Debugger -online"