

Troubleshooting Failed Background Jobs



Liam Cleary

CEO / MICROSOFT MVP / MICROSOFT CERTIFIED TRAINER

@shareplicity www.shareplicity.com | @helloitsliam www.helloitsliam.com



Overview



Investigating Failed Jobs

- Review the State
- Check the Reason

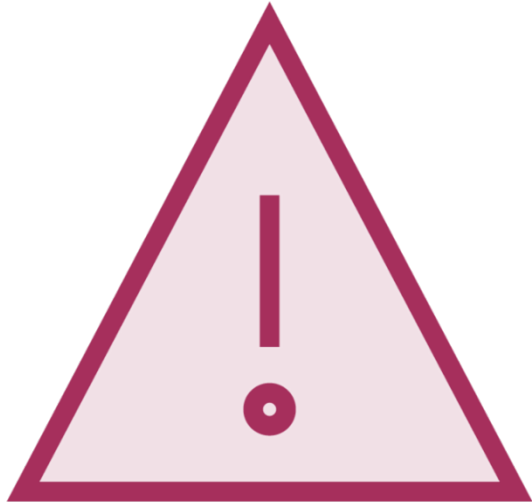
Using the "Debug-Job" Command



Investigating Failed Jobs



Background Jobs



Status



Reason



Retrieving Status and Reason

Start Job that will Fail

```
Start-Job -ScriptBlock { Get-EventLog -LogName Nothing }
```

Get Failed Job and Child Job Details

```
Get-Job -IncludeChildJob
```

Get Failed Child Job

```
Get-Job -ChildJobState Failed
```



Retrieving Status and Reason

Retrieve the Failed Job State Information

```
Get-Job -Id 1 | Select -ExpandProperty JobStateInfo
```

Retrieve the Reason from the Job State Information

```
Get-Job -Id 1 | `
    Select -ExpandProperty JobStateInfo `
    | Select -ExpandProperty Reason
```



Demo



Investigate Failed Jobs

- Review the Status
- Examine the Reason



Using the "Debug-Job" Command



Debug-Job

Debug scripts that are running within jobs. Debug PowerShell Workflow jobs, background jobs, and jobs running in remote sessions



Debug Background Jobs



Debug Remote
Background Jobs



Debug Long Running
Jobs



Debug Jobs that use
"Wait-Debugger"



Retrieving Status and Reason

Create Job using "Wait-Debugger"

```
$scriptBlock = {`  
    $var = 10;  
    Wait-Debugger;  
    $var;  
    $var + 1;  
}
```

Start Running Job

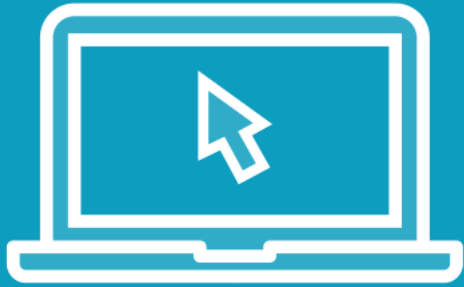
```
$job = Start-Job -ScriptBlock $scriptBlock
```

Debug the Running Job

```
Debug-Job -job $job
```



Demo



Debug a Background Job



Summary



Investigated failed jobs by first reviewing the State, then the Reason

Walked through how to use the "Debug-Job" command



Up Next:

Executing Background Jobs on Remote
Computers

