

# Formatting Data in PowerShell

---

FORMATTING WITH POWERSHELL



**Matt Allford**

@mattallford [www.mattallford.com](http://www.mattallford.com)





# Prerequisites

**PowerShell  
language familiarity**

**PowerShell help  
system**

**Running and piping  
commands**





# Course Scenario



**Dan Stanton**

Desktop support  
engineer for  
Globomantics



**Anna Fisher**

Globomantics  
engineering manager





# Course Overview



## Formatting with PowerShell

- Understanding and exploring PowerShell's format system
- Formatting to table, list, and wide
- Formatting views
- Formatting gotchas

## Working with “Out” Cmdlets





# Exercise Files and Demo Environment

---





# Exercise Files

[Table of contents](#) [Description](#) [Transcript](#) **[Exercise files](#)** [Discussion](#) [Related Courses](#)

These exercise files are intended to provide you with the assets you need to create a video-based hands-on experience. With the exercise files, you can follow along with the author and re-create the same solution on your computer.

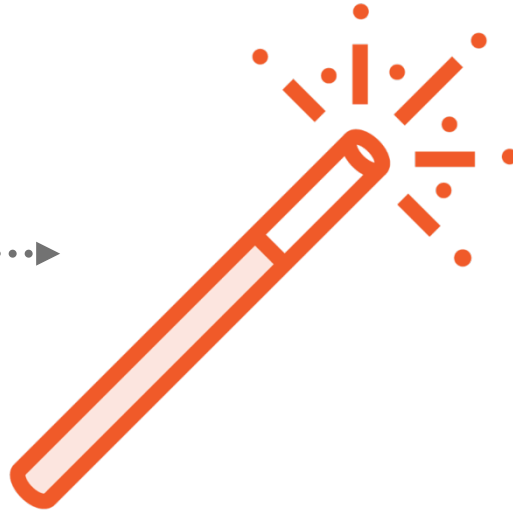
[Download exercise files](#)





# PowerShell Format System

Get-Process



```
PS C:\> get-process
```

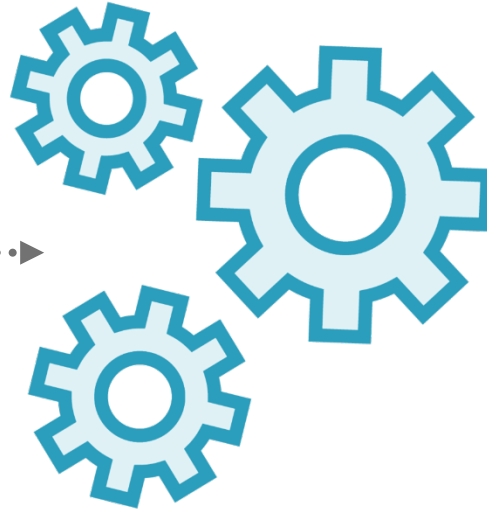
Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
196	14	2884	2852		16836	0	aesm_service
260	13	2980	1760		3304	0	afcdpsrv
286	22	4244	6020		3016	0	AppleMobileDeviceService
552	30	20128	23660	1.38	15076	1	ApplicationFrameHost
498	18	24236	15668	1,341.31	3316	0	audiodg
305	17	4616	20732	0.13	26316	1	backgroundTaskHost
239	16	27116	54556	0.36	1148	1	brave
310	22	104972	88828	137.19	6412	1	brave
307	22	104208	88996	140.25	6424	1	brave
304	20	39848	6856	5.48	6676	1	brave
273	19	42232	65564	32.36	6872	1	brave
702	72	87264	44348	436.25	7124	1	brave
577	31	176964	208280	566.92	8788	1	brave
302	20	66928	110864	3.56	9556	1	brave
238	16	30240	40008	5.89	9664	1	brave
362	21	55488	101176	3.00	9808	1	brave
267	18	43404	52236	46.03	10164	1	brave
282	20	119036	99004	115.36	10616	1	brave
391	23	61128	17704	15.97	12556	1	brave





# PowerShell Format System

Get-Process



PowerShell Format  
System

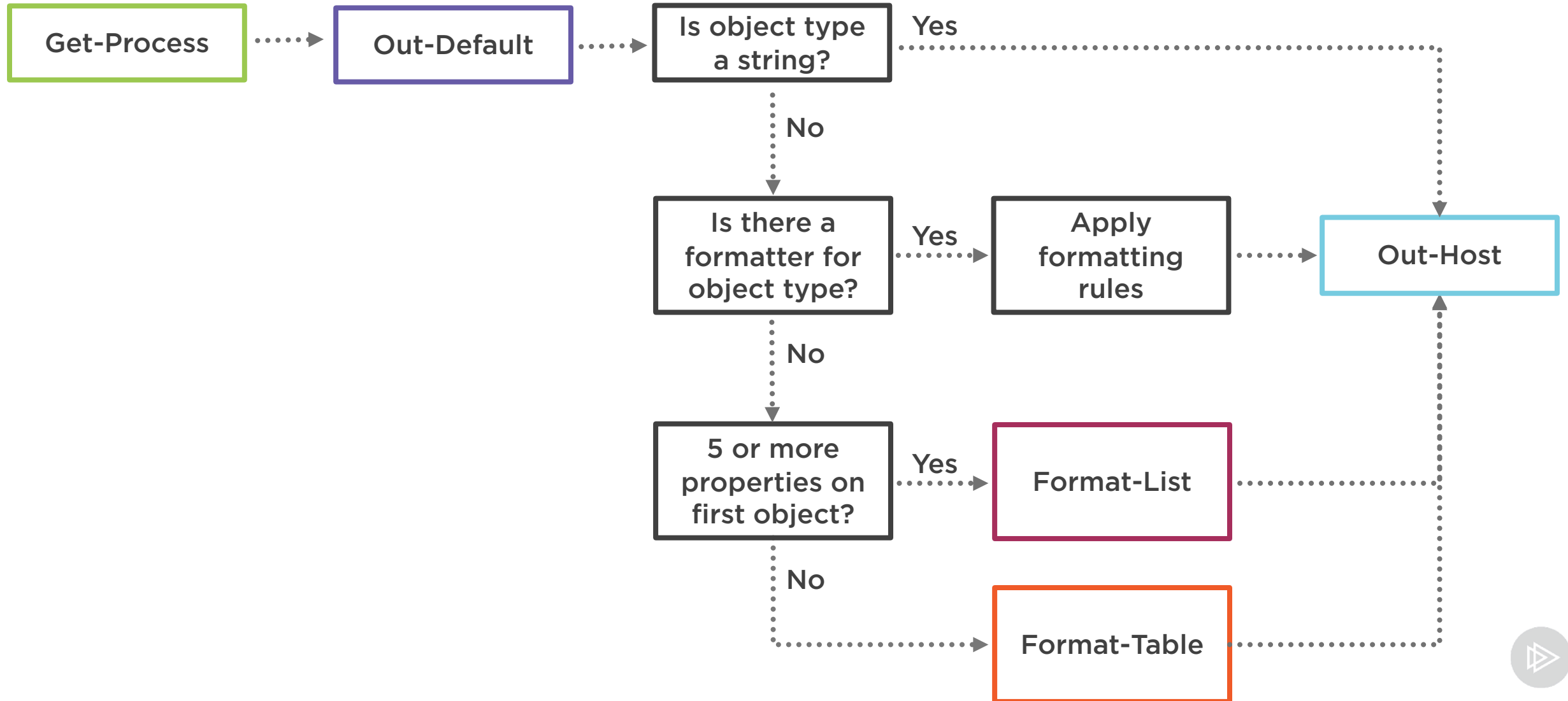
```
PS C:\> get-process
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
196	14	2884	2852		16836	0	aesm_service
260	13	2980	1760		3304	0	afcdpsrv
286	22	4244	6020		3016	0	AppleMobileDeviceService
552	30	20128	23660	1.38	15076	1	ApplicationFrameHost
498	18	24236	15668	1,341.31	3316	0	audiodg
305	17	4616	20732	0.13	26316	1	backgroundTaskHost
239	16	27116	54556	0.36	1148	1	brave
310	22	104972	88828	137.19	6412	1	brave
307	22	104208	88996	140.25	6424	1	brave
304	20	39848	6856	5.48	6676	1	brave
273	19	42232	65564	32.36	6872	1	brave
702	72	87264	44348	436.25	7124	1	brave
577	31	176964	208280	566.92	8788	1	brave
302	20	66928	110864	3.56	9556	1	brave
238	16	30240	40008	5.89	9664	1	brave
362	21	55488	101176	3.00	9808	1	brave
267	18	43404	52236	46.03	10164	1	brave
282	20	119036	99004	115.36	10616	1	brave
391	23	61128	17704	15.97	12556	1	brave





# Default Formatting Workflow





# Registered Views

---





# Format Files for Windows PowerShell

```
Get-ChildItem $PSHome -Filter *format.ps1xml | Select-Object name
```

```
Certificate.format.ps1xml
```

```
Diagnostics.Format.ps1xml
```

```
DotNetTypes.format.ps1xml
```

```
Event.Format.ps1xml
```

```
FileSystem.format.ps1xml
```

```
Help.format.ps1xml
```

```
HelpV3.format.ps1xml
```

```
...
```





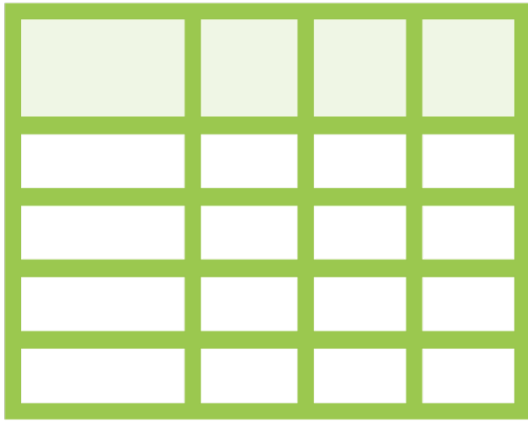
# Get-FormatData

Gets the formatting data in the current session. You can use this cmdlet to examine the formatting data. Then, you can use the **Export-FormatData** cmdlet to serialize the objects, convert them to XML, and save them in Format.ps1xml files.





# Formatting Cmdlets



Format-Table



Format-List



Format-Wide



# Formatting Commonalities

**Autosize**

**Grouping**

**Views**

**InputObject**





Format commands just  
format data. They do NOT  
display data.





# Demo



## Exploring PowerShell's format system





# Demo



## Exploring Format-Table





# Demo



## Exploring Format-List





# Demo



## Exploring Format-Wide





# Demo

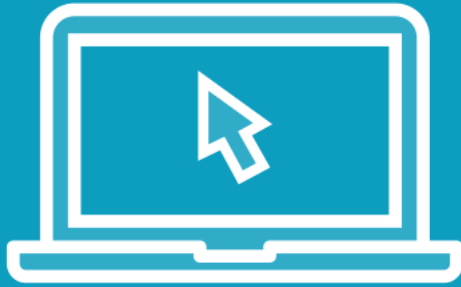


## Using formatting views





# Demo



## Formatting caveats





Do not use format cmdlets  
inside of scripts or  
functions.





# Summary



PowerShell formats well, out of the box

The formatting system is complex, but don't let that worry you

Formatting to tables, lists, and wide

Format cmdlets format, “out” cmdlets output formatted data

Leave formatting until last

- Unless piping to an “out” cmdlet





Up Next:

Working with “Out” Cmdlets

---

