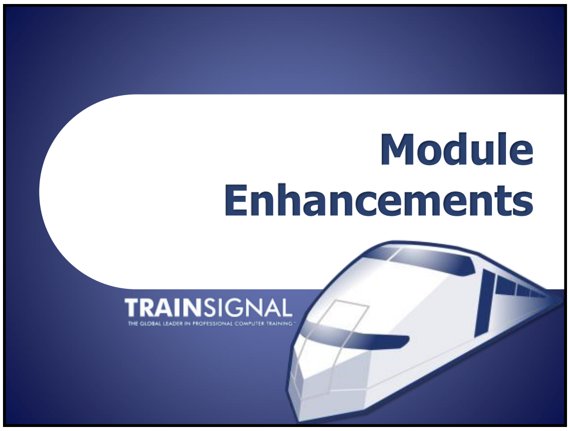


Module Enhancements

TRAINSIGNAL
THE GLOBAL LEADER IN PROFESSIONAL COMPUTER TRAINING™

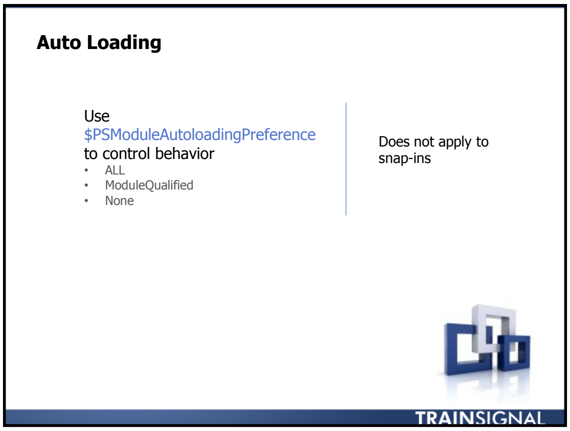
Auto loading

No longer need to import a module
(But you still can manually)

PowerShell searches modules in
$ENV:PSModulePath

If you know the cmdlet name use it

TRAINSIGNAL



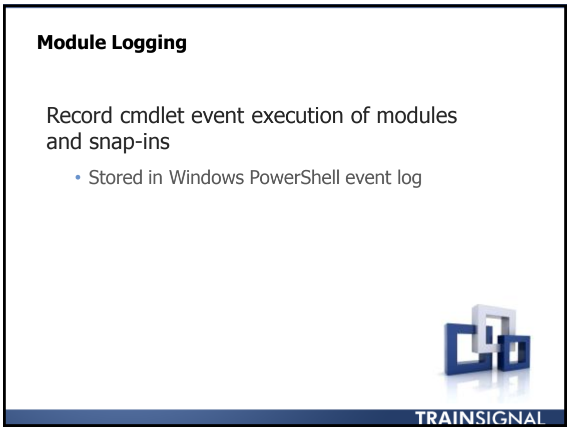
Auto Loading

Use
$PSModuleAutoloadingPreference
to control behavior

- ALL
- ModuleQualified
- None

Does not apply to snap-ins

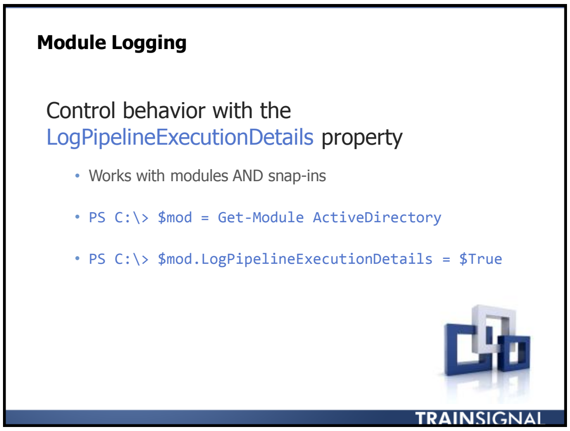
TRAINSIGNAL



Module Logging

Record cmdlet event execution of modules and snap-ins

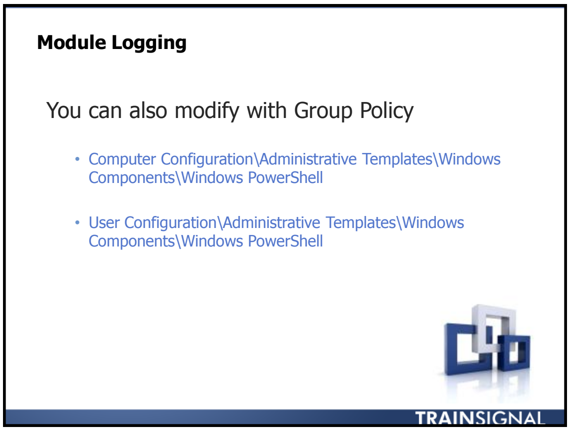
- Stored in Windows PowerShell event log



Module Logging

Control behavior with the LogPipelineExecutionDetails property

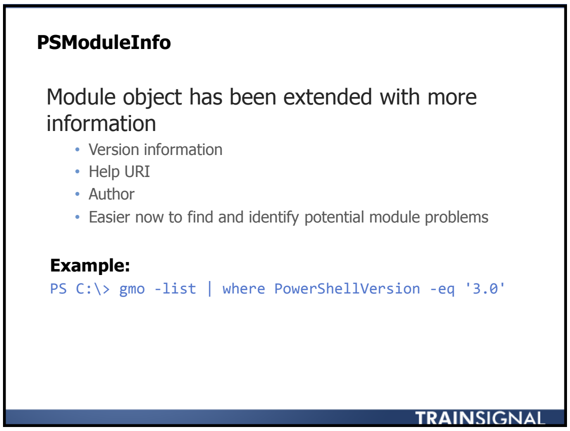
- Works with modules AND snap-ins
- `PS C:\> $mod = Get-Module ActiveDirectory`
- `PS C:\> $mod.LogPipelineExecutionDetails = $True`



Module Logging

You can also modify with Group Policy

- Computer Configuration\Administrative Templates\Windows Components\Windows PowerShell
- User Configuration\Administrative Templates\Windows Components\Windows PowerShell



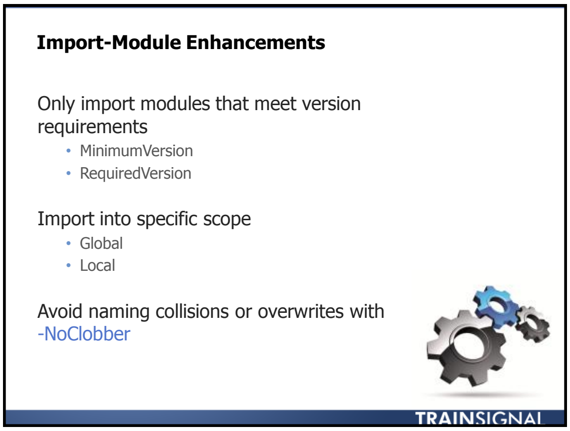
PSModuleInfo

Module object has been extended with more information

- Version information
- Help URI
- Author
- Easier now to find and identify potential module problems

Example:

```
PS C:\> gmo -list | where PowerShellVersion -eq '3.0'
```



Import-Module Enhancements

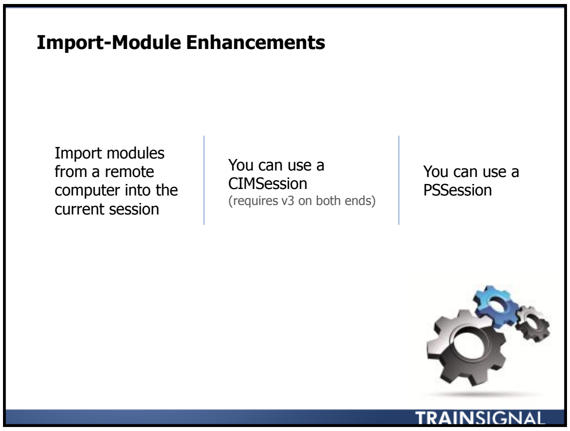
Only import modules that meet version requirements

- MinimumVersion
- RequiredVersion

Import into specific scope

- Global
- Local

Avoid naming collisions or overwrites with -NoClobber

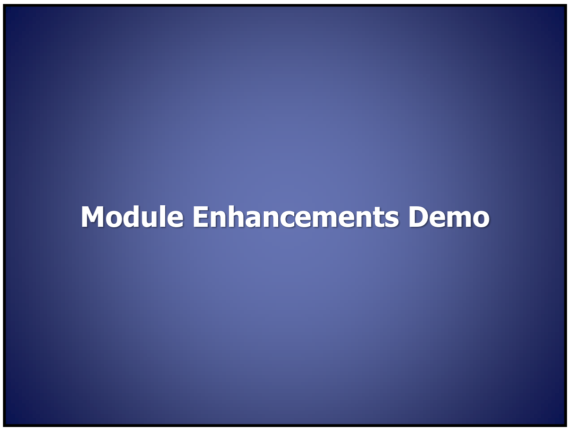


Import-Module Enhancements

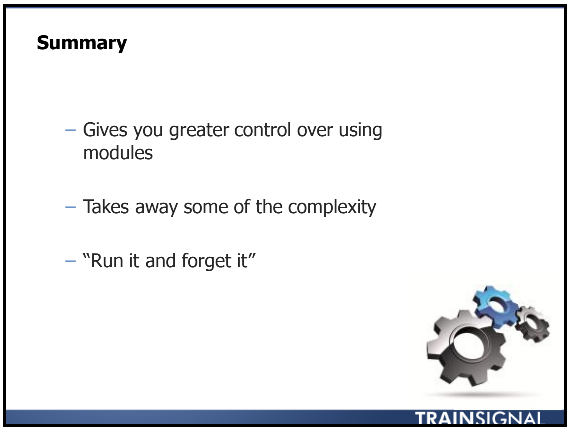
Import modules from a remote computer into the current session

You can use a CIMSession (requires v3 on both ends)

You can use a PSSession



Module Enhancements Demo



Summary

- Gives you greater control over using modules
- Takes away some of the complexity
- "Run it and forget it"