The PSAppDeployToolkit (PSADT) offers robust logging capabilities, including the ability to log individual commands within your deployment scripts. This is achieved through the use of the Write-Log function, which is a core component of the toolkit.

How to Log Individual Commands:

- Utilize Write-Log: The simplest and most effective way to log individual commands or specific actions within your PSADT script is by using the Write-Log function. This function allows you to write custom messages to the main PSADT log file.

```
## Example of logging a message
Write-Log "Starting the installation of Application X."

## Example of logging a command execution
Execute-Process -Path "msiexec.exe" -Parameters "/i
application.msi /qn"
Write-Log "MSI installation command executed."

## Example of logging a status update
If (Test-Path -Path "C:\Program Files\Application X") {
    Write-Log "Application X installation successful."
} Else {
    Write-Log -Message "Application X installation failed."
-Severity "Error"
}
```

- **Specify Log File (for specific commands like MSI):** For certain PSADT functions, such as Execute-MSI, you can specify a dedicated log file for that specific operation using the -LogName parameter. This creates a separate log file in the default PSADT log directory (e.g., C:\Windows\Logs\Software) alongside the main toolkit log.

```
Execute-MSI -Action 'Install' -Path "$InstallPath\MyApp.msi"
-Parameters "/qb" -LogName "MyAppInstall.log"
```

- **Leverage PowerShell's Built-in Logging:** For commands that are not directly handled by PSADT functions but are standard PowerShell commands, you can still leverage PowerShell's built-in logging mechanisms like Start-Transcript and Stop-Transcript to capture command output and details. While this creates a separate transcript file, it can be useful for detailed debugging of specific PowerShell operations within your PSADT script.

By incorporating Write-Log throughout your script, you can create a detailed and informative log of your deployment process, aiding in troubleshooting and verification.