**Email:** meghanabadgujar07@gmail.com

**Name:** Meghana Samadhan Badgujar  
**Batch Name:** WiproNGA\_DWS\_B5\_25VID2550

**User ID:** 34737

**Batch ID:** 25VID2550

**Date:** 16th August 2025

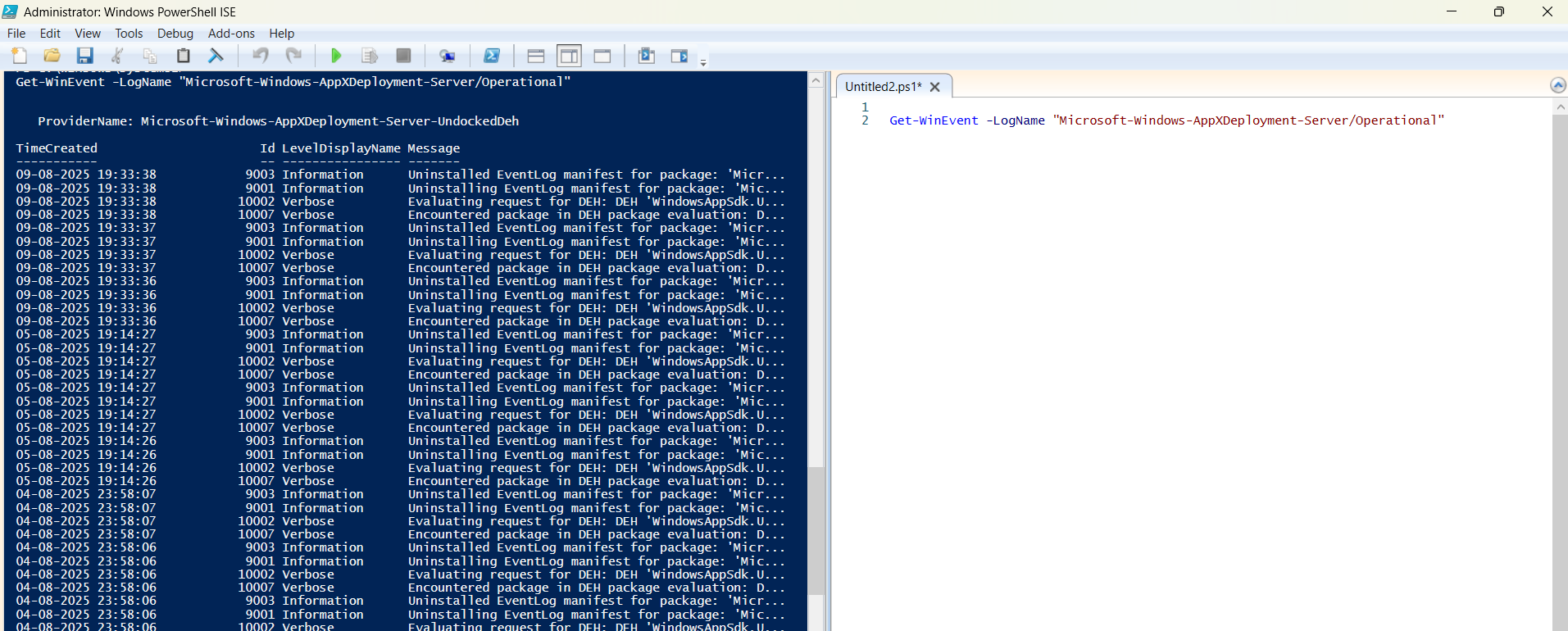
**Assignment Topic:**

* **Install location and Troubleshooting logs in MSIX.**
* **Tools for testing (ACT Issues).**
* **Hands-on analysis and fixing of compatibility problems.**
* **Troubleshooting tools (Event Viewer,Dependency Walker).**

**1. Install Location and Troubleshooting Logs in MSIX**

* **Install Location:**
  + MSIX apps are installed under:
  + C:\Program Files\WindowsApps
  + This folder is protected by Windows, requiring special permissions to access.
* **Logs:**
  + MSIX installation and deployment logs are available in:
    - **Event Viewer → Applications and Services Logs → Microsoft → Windows → AppXDeployment-Server → Operational**
  + **PowerShell commands to view logs:**

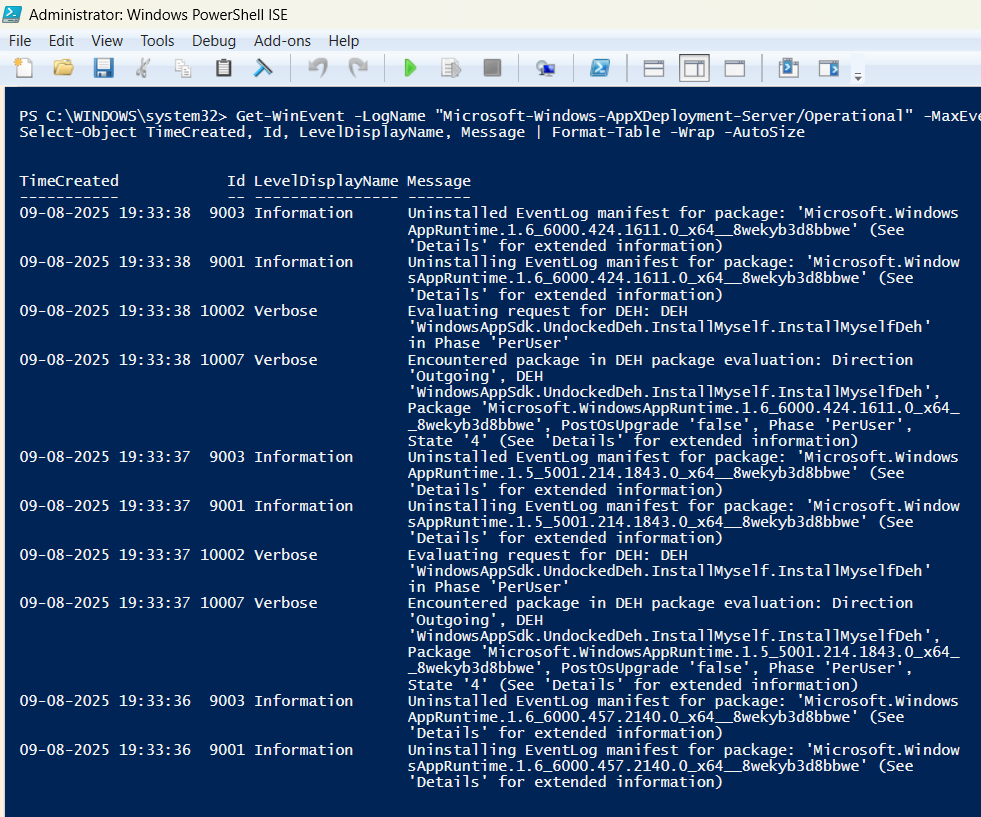
Get-WinEvent -LogName "Microsoft-Windows-AppXDeployment-Server/Operational" |



# View last 10 deployment events

Get-WinEvent -LogName "Microsoft-Windows-AppXDeployment-Server/Operational" -MaxEvents 10 |

Select-Object TimeCreated, Id, LevelDisplayName, Message | Format-Table -Wrap -AutoSize



**2. Application Compatibility Toolkit (ACT Issues)**

* + **What it is:**  
    - A Microsoft toolkit (part of ADK – Assessment & Deployment Kit) that helps test application compatibility across OS versions.
  + **Why important in App Packaging:**

- When moving old apps (Windows 7/8 → Windows 10/11), ACT detects **compatibility issues**.

- It recommends **shims (fixes)** using **Compatibility Administrator**.

* + **Common Issues ACT Detects:**

**- UAC/Privilege Issues:** App requires admin but was not designed for UAC.

**- File/Registry Virtualization:** Older apps write to C:\Windows\System32 or HKLM registry keys → blocked in newer Windows.

**- Hardcoded Paths:** Application assumes files exist in C:\Program Files instead of %ProgramFiles%.

**- Removed APIs/DLLs:** Uses old Windows APIs no longer available.

**3. Hands-on Analysis and Fixing of Compatibility Problems:**

* 1. **Event Viewer check** for app install errors.
  2. Used **PowerShell commands (Get-WinEvent)** to fetch and format logs.
  3. **Dependency Walker** installed and run to analyze missing DLL dependencies.
     + Shows all DLLs an application requires.
     + Highlights missing/incorrect versions.

**4. Troubleshooting Tools:**

* **Event Viewer**

A built-in Windows tool that records system, application, and security events in logs.

- Why important in App Packaging:

* Every install/uninstall of MSI/MSIX logs entries.
* If an app fails, Event Viewer often shows error codes, missing dependencies, or permission issues.
* Helps trace Intune / MSIX deployment failures.

- Key Logs for App Packaging:

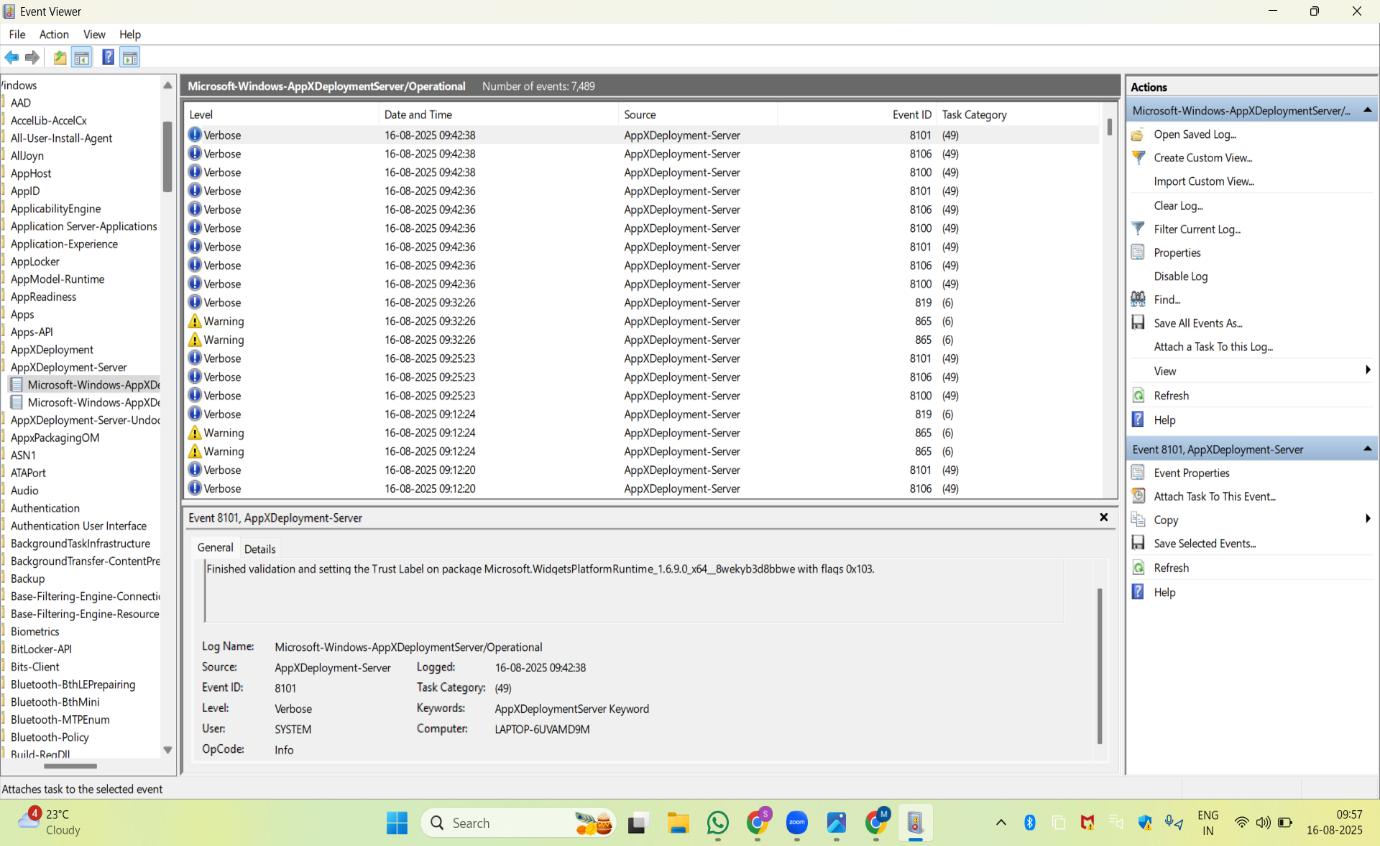
* MSIX / AppX Logs:

- Path → Applications and Services Logs → Microsoft → Windows → AppXDeployment-Server → Operational

- Shows package installation/uninstallation issues.

* MSI Logs:

- Found in Windows Logs → Application (look for MsiInstaller **events).**

****

* **Dependency Walker:**  
  A tool that scans an executable (EXE/DLL) and lists all its **dependencies (DLLs, OCX, system components)**.

**-** **Why important in App Packaging:**

* + Many older apps break on Windows 10/11 because **required DLLs are missing** or are **different versions**.

**- Helps detect:**

* + **Missing DLLs** → app won’t launch.
  + **Architecture mismatch** → 32-bit DLL called from 64-bit app.
  + **System file dependencies** → APIs deprecated in Windows 11.

**- Installation:**

