**PowerShell App Deployment Toolkit (PSADT)**

**1. Introduction**

The **PowerShell App Deployment Toolkit (PSADT)** is an open-source framework designed to simplify the process of deploying applications in Windows environments.  
It provides a standard, consistent, and flexible way to package and deploy applications using **PowerShell scripting**, with features such as:

* User-friendly deployment experience
* Standardized logging
* Easy integration with **SCCM/MECM** or **Intune**
* Built-in functions for install/uninstall actions
* Handling application prerequisites and dependencies

**2. Features**

* **Standardized Deployment Interface** – Displays consistent installation UI messages to end users.
* **Pre-built Functions** – Includes functions like Show-InstallationPrompt, Show-InstallationWelcome, and Execute-MSI.
* **Logging** – Automatically creates detailed logs for troubleshooting.
* **Silent Deployment** – Supports completely silent installs for enterprise environments.
* **Custom Actions** – Allows execution of scripts before or after installation.
* **Supports Both MSI and EXE** installers.

**3. Why Use PSADT?**

* Saves time compared to writing deployment scripts from scratch.
* Reduces human error in deployments.
* Easy to train IT teams since all scripts follow a similar structure.
* Handles common deployment needs (e.g., stopping processes, checking if an app is running, installing prerequisites).

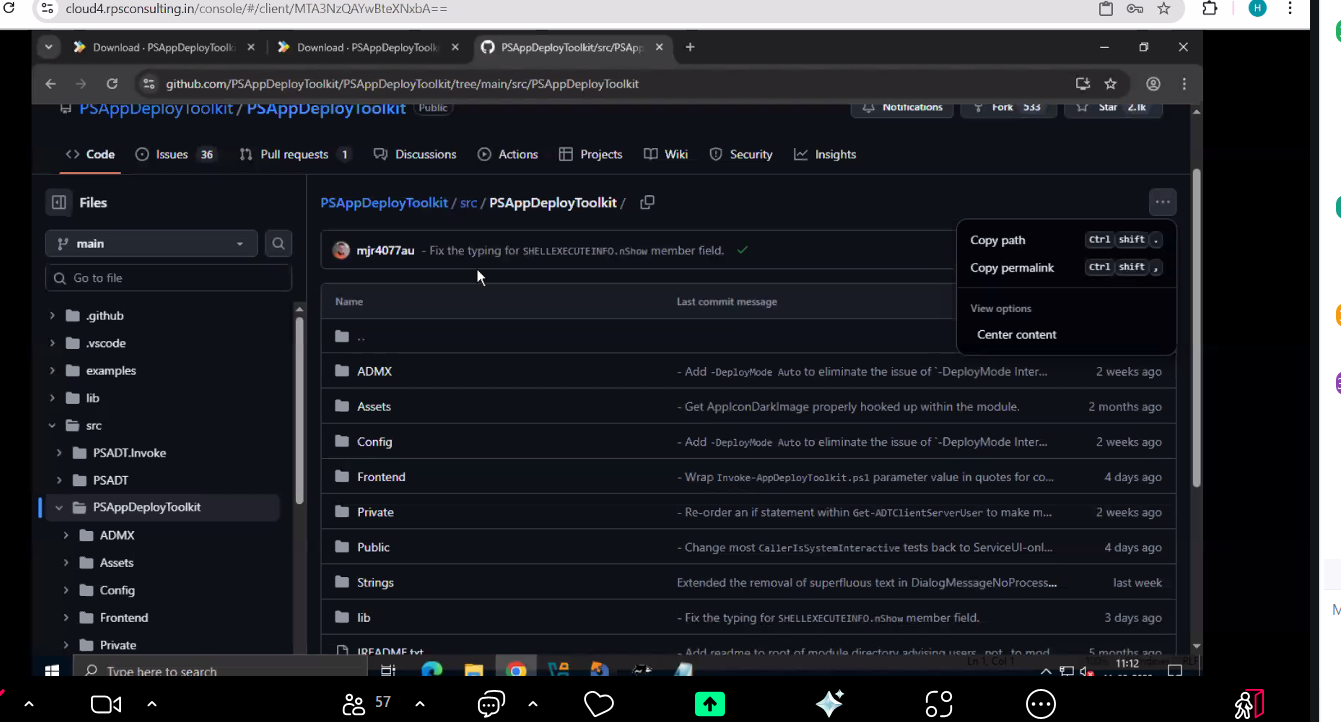
**4. Prerequisites**

Before installing and using PSADT, ensure:

* **Windows OS** (Windows 7 or later for clients, Windows Server 2012 or later for servers)
* **PowerShell 3.0 or higher**
* Administrator privileges
* An archive tool like **7-Zip** or **WinRAR** (for extracting PSADT ZIP file)

**5. Installation Process (Step-by-Step)**

**Step 1: Download PSADT**

1. Go to the official GitHub page:  
   🔗 <https://github.com/PSAppDeployToolkit/PSAppDeployToolkit>
2. Click the **Code** button and select **Download ZIP**.
3. Save the ZIP file to a preferred location on your PC.

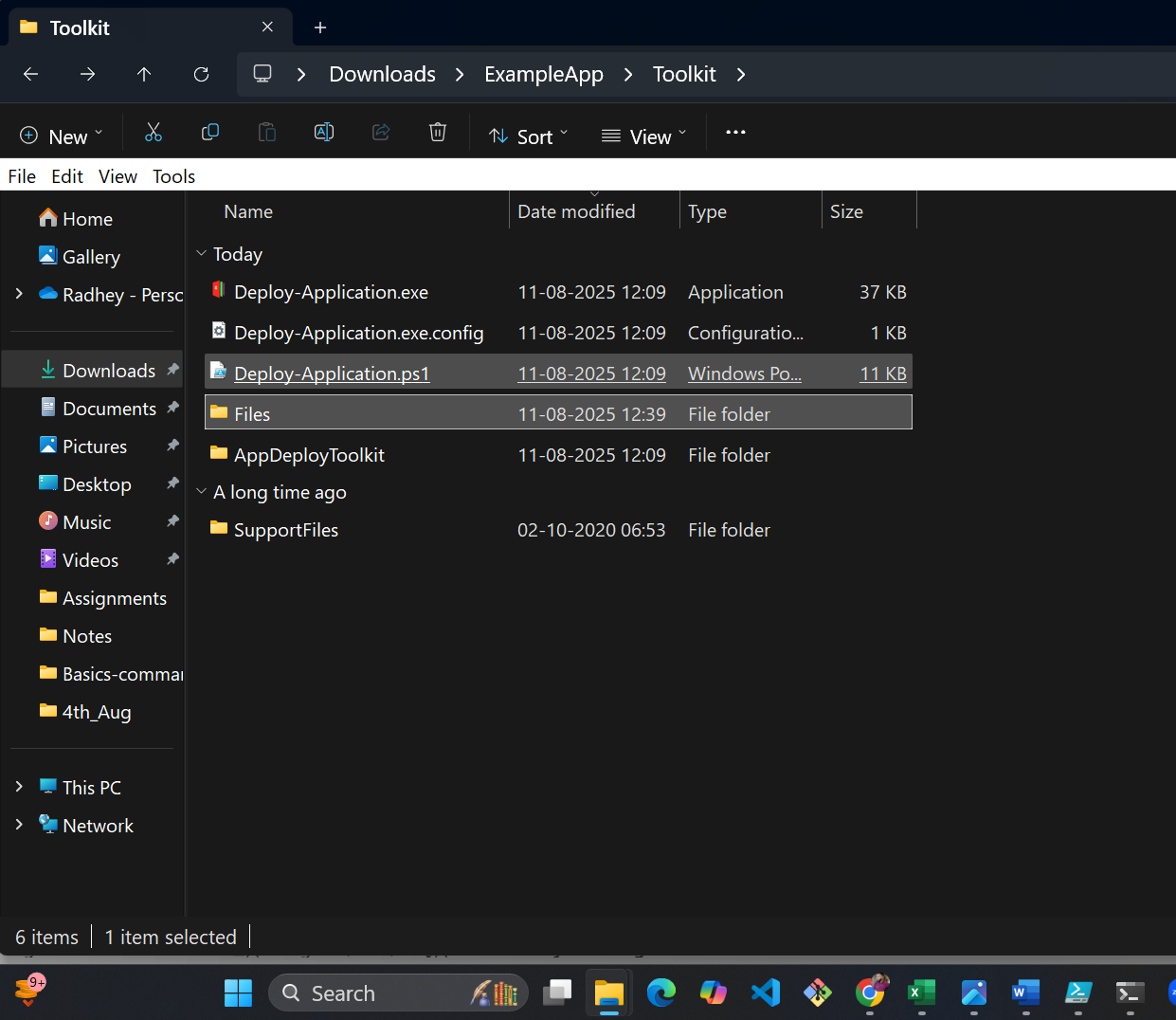
**Step 2: Extract the Toolkit**

1. Right-click on the downloaded ZIP file.
2. Choose **Extract All** (or use 7-Zip/WinRAR).
3. Extract to a folder, e.g., C:\PSADT.

**Step 3: Understand the Folder Structure**

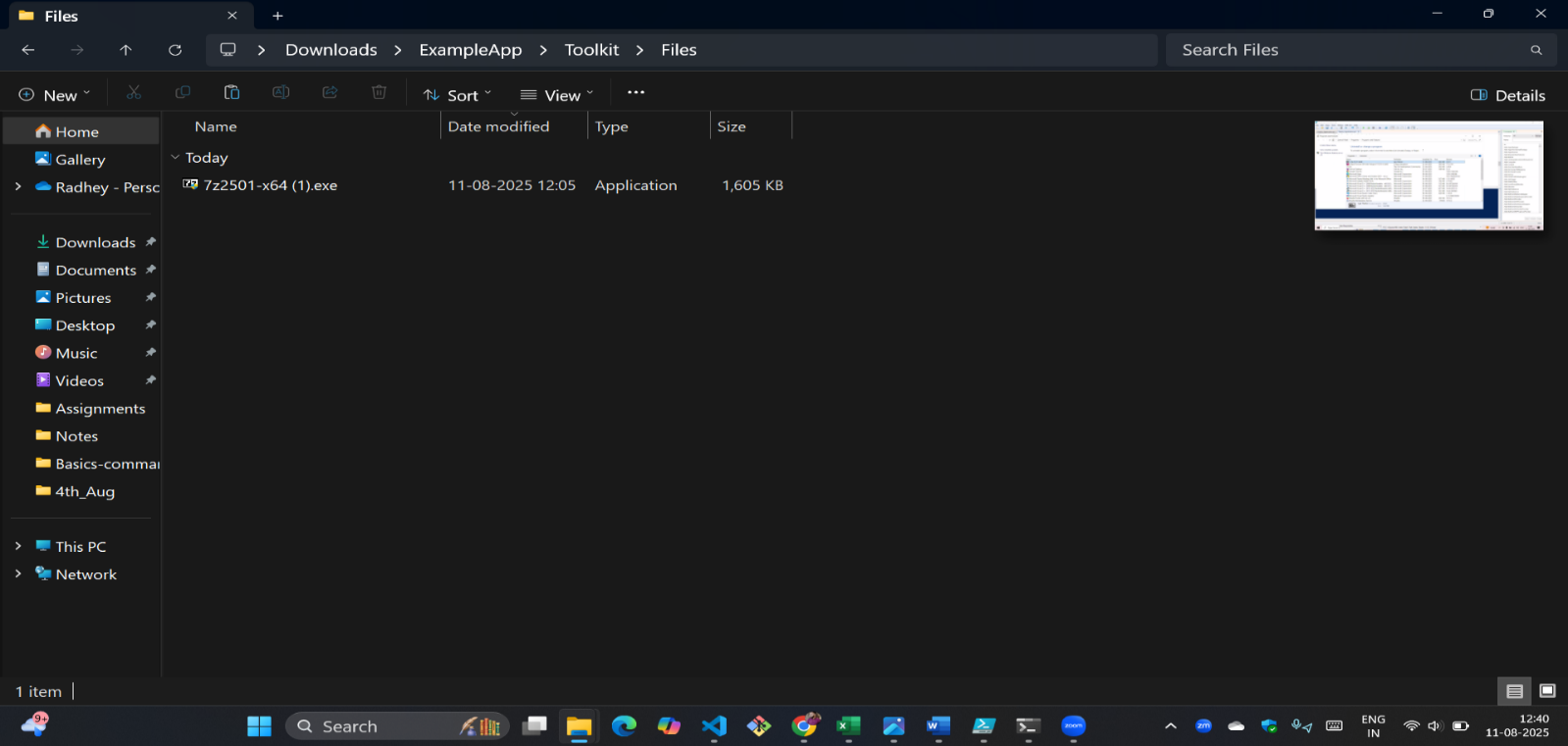
Inside the extracted folder, you will find:

* **Files**
  + Deploy-Application.ps1 → Main script for deployment logic
  + AppDeployToolkitMain.ps1 → Core functions of the toolkit
  + Deploy-Application.exe → Optional launcher
* **Folders**
  + AppDeployToolkit → Core toolkit scripts
  + Files → Place your application installer files here
  + SupportFiles → Additional files needed for your installation
  + Logs → Installation logs will be stored here

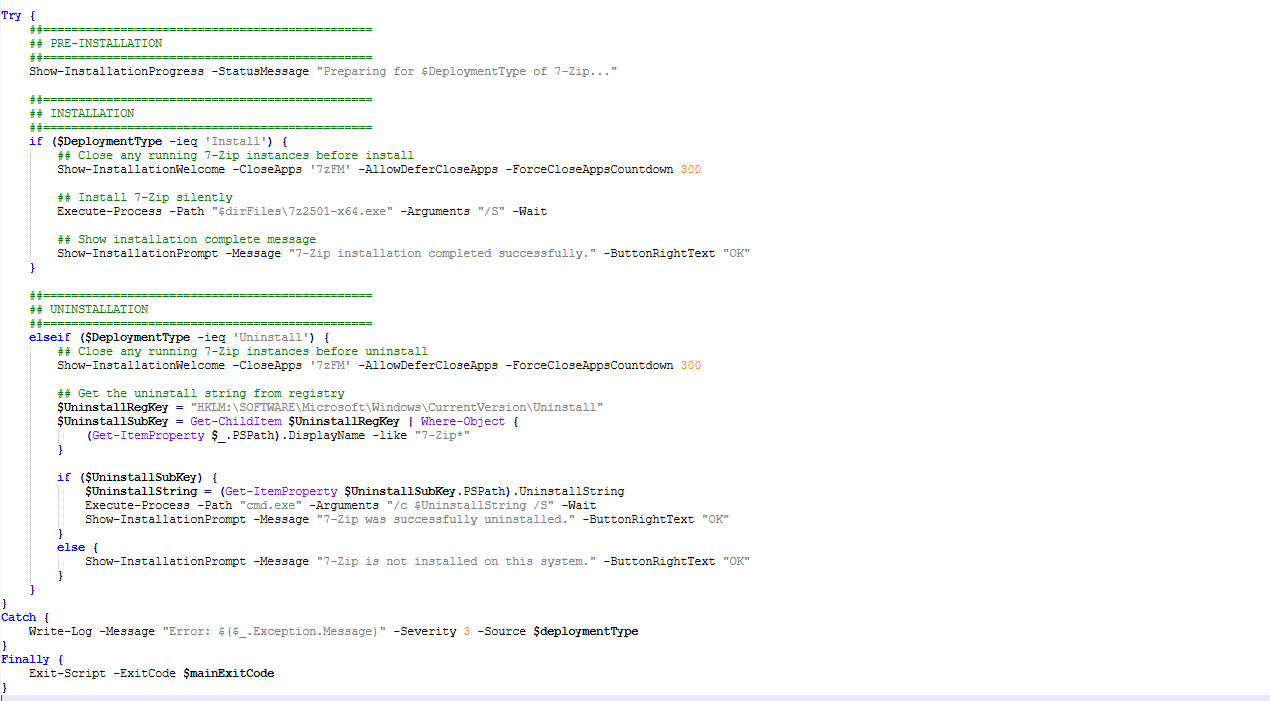


**Step 4: Place Application Files**

* Copy your application installer (.msi, .exe, etc.) into the **Files** folder.
* In our case :- .\Files\7z2501-x64.exe



**Step 5: Edit the Main Script**

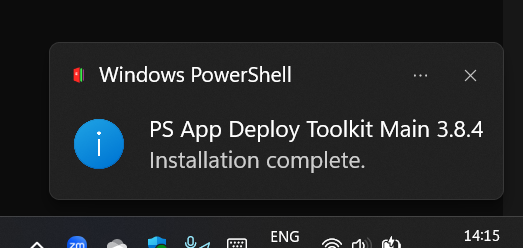
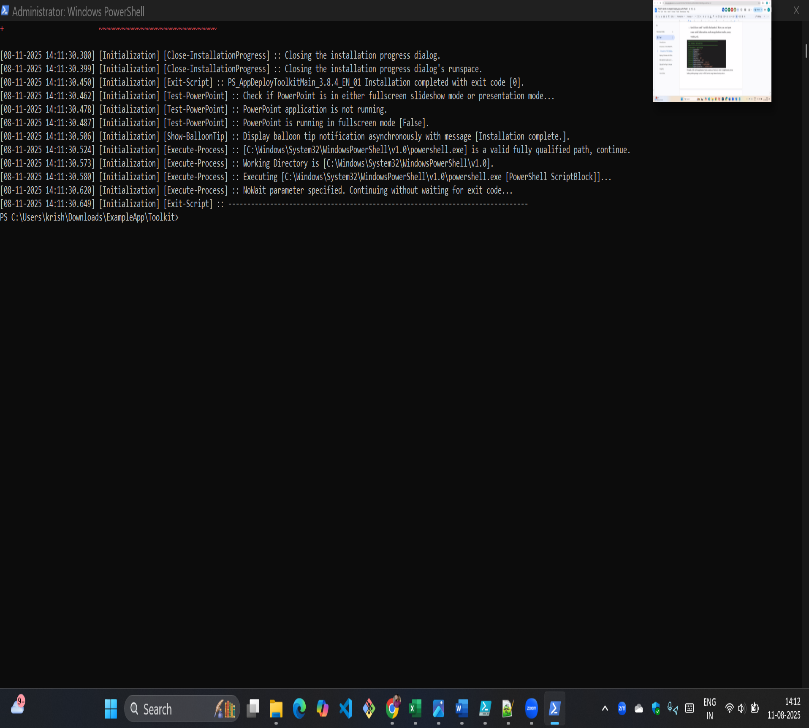
****

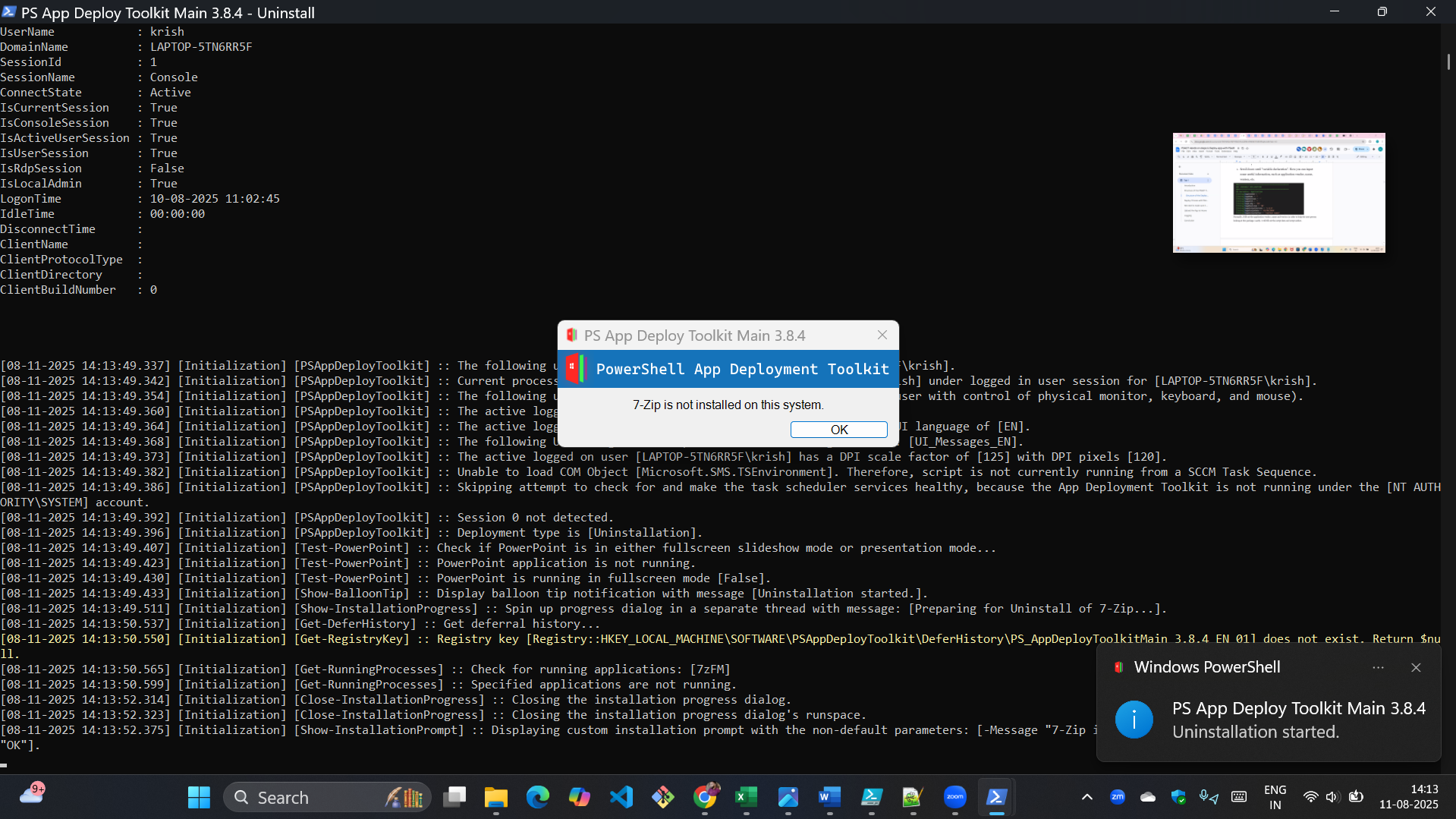
**Step 7: Verify Logs**

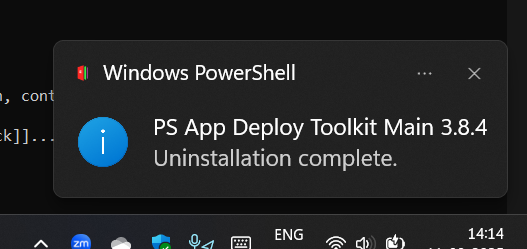
* Check the Logs folder for a log file named like:  
  Deploy-Application-<ComputerName>-<Date>-<Time>.log  
  This helps in troubleshooting.

**Run :-**

* **Installation :-** .\Deploy-Application.ps1 -DeploymentType Install -DeployMode Interactive
* **Uninstallation :-** .\Deploy-Application.ps1 -DeploymentType Uninstall -DeployMode Interactive
* **At the path :-** cd "C:\Users\krish\Downloads\ExampleApp\Toolkit"







**6. Best Practices**

* Always test on a non-production machine first.
* Keep scripts well-documented within comments.
* Use version control (like Git) for script changes.
* Use silent install switches for production deployments.
* Standardize script headers for team collaboration.

**PowerShell App Deployment Toolkit (PSADT)**

* **What it is:** An open-source toolkit for creating standardized, scripted application installers.
* **Purpose:** Helps IT admins package apps with pre/post-install actions, user prompts, logging, and error handling.
* **Use case:** You use PSADT to *prepare* an app for deployment, especially when custom install logic is needed.

**Microsoft Intune**

* **What it is:** A cloud-based endpoint management platform from Microsoft.
* **Purpose:** Distributes, installs, updates, and manages apps, settings, and policies on Windows, macOS, iOS, and Android devices.
* **Use case:** You use Intune to *deliver and manage* apps (including ones packaged with PSADT) to devices anywhere.

**Key Difference:**

* PSADT = Focuses on **how** the app installs and interacts with the user.
* Intune = Focuses on **where** and **to whom** the app is deployed.

If you want, I can make a **PSADT vs Intune table** so the differences are instantly clear. That would make it even easier to remember.

**You said:**

how they use together

**ChatGPT said:**

Here’s how **PSADT** and **Intune** work together in a simple step-by-step flow:

**1. Package the App with PSADT**

* You use **PSADT** to wrap your application installer (MSI, EXE, etc.) in a deployment script.
* In that script, you can:
  + Show messages to the user
  + Close conflicting apps
  + Install prerequisites
  + Run the main installer
  + Log the whole process
* Example: Wrapping VLC.exe with PSADT so it installs silently and logs everything.

**2. Test the PSADT Package**

* Run the Deploy-Application.ps1 (or .exe) locally to confirm:
  + It installs correctly
  + Prompts and messages work
  + Logs are created
  + No errors occur

**3. Upload to Intune**

* Compress the PSADT folder into a .intunewin file using the **Microsoft Win32 Content Prep Tool**.
* Go to **Intune admin center** → **Apps** → **Add** → **Windows app (Win32)**.
* Upload your .intunewin package.

**4. Configure Intune Deployment**

* Set:
  + **Install command** → Deploy-Application.exe
  + **Uninstall command** → Deploy-Application.exe -DeploymentType Uninstall
  + Detection rules → to check if the app is installed
  + Requirements → OS version, architecture, etc.
  + Assignments → which devices or users get the app

**5. Deploy & Manage**

* Intune pushes your PSADT-wrapped app to targeted devices.
* On the device:
  + Intune runs your PSADT script
  + PSADT handles the install process smoothly
* You can monitor success/failure in Intune logs.

💡 **In short:**

* **PSADT** = Makes the installer smart and user-friendly.
* **Intune** = Sends that smart installer to the right devices and keeps it managed.