

WIPRO ASSIGNMENT – 1

- **Submitted By:** Anuj Bose
- **User ID:** 34757
- **Batch Name:** WiproNGA_DWS_B5_25VID2550
- **Date:** 5th August, 2025

What is Application Packaging ?

Application Packaging is the process that deals with the actual creation of the application packages. Application packaging is the process of customizing and getting applications ready for easy deployment across an organization's systems. It provides a consistent, secure, and manageable environment for users and IT teams.

Need of Application Packaging

Modern organizations use hundreds or even thousands of software applications, each with unique installation and configuration needs, making management complex. With Windows 10 and the Windows as a Service model, this complexity increases. Application Packaging helps simplify this by easing the load on IT teams and enhancing the end-user experience.

Benefits of Application Packaging

Application Packaging has a large number of benefits. Some of them are:

- It ensures consistent, stable and reliable standard environment.
- It streamlines the software deployment and increases the efficiency of the software management.
- It mitigates security issues.
- It decreases the risk of business disruption.

- It helps reduce the ongoing administration and support costs.

Steps Involved in Application Packaging

- **Identify and Collect:** Research to get application details and requirements.
- **Review and Assess:** Analyze the gathered information and determine the suitable packaging approach.
- **Package:** Package applications inline business requirements and import to deployment tool.
- **Test:** Tests packaged applications: UAT, Pilot.
- **Deploy:** Rolls out the packaged application to production.

End To End Application Packaging Process

- **Application Discovery:** Application Discovery is the initial step where the application source file is validated and its functionality within the organization's environment is verified. During this phase, all necessary requirements and details are gathered. Some applications may need minimal discovery, while others might require vendor support if issues arise due to compatibility or configuration problems.
- **Application Packaging:** Application Packaging involves creating the software package using the information collected during the Discovery stage. The

process varies based on the chosen packaging format, and best practices can be followed from available resources.

- **User Acceptance Testing:** Application UAT (User Acceptance Testing) is the step before deploying the packaged application to production. It involves validating the package to ensure it functions correctly and matches the behavior of the original application tested during the Discovery phase.

Difference between Windows 10 and Windows 11

Windows 11 is built on Windows 10 but offers a more modern user experience with better performance, enhanced security, and a cleaner interface. Both support most applications, but Windows 11 generally provides a smoother overall experience.

Windows 11

- **Modern Interface:** Redesigned Start Menu and Taskbar with rounded corners and a cleaner look.
- **Better Security:** Includes TPM 2.0 and Windows Hello for stronger, biometric-based protection.
- **Faster Performance:** Quicker logins, browsing, and wake-up times.
- **Updated Microsoft Store:** Supports Android apps through the Amazon Appstore.

- **Improved Multitasking:** New features like Snap Layouts and Snap Groups for better window management.
- **Built-in AI Assistant:** Windows Copilot helps with tasks and boosts productivity.
- **Enhanced Gaming:** Features like DirectX 12 Ultimate and DirectStorage improve graphics and speed.
- **Efficient Updates:** Smaller and faster updates reduce downtime.

Windows 10

- **Familiar Interface:** Easy to use for those already comfortable with Windows.
- **Wide Compatibility:** Supports a large range of applications and hardware.
- **Stable and Reliable:** Known for consistent performance and fewer issues.
- **Cost-Effective:** Often more affordable, especially for upgrades from older Windows versions.

Handling Scheduled Tasks within a Windows MSI Installation Package

Handling Scheduled Tasks within a Windows MSI Installation Package automates actions after the deployment such as starting services, running scripts or

executing specific time or events, making it a powerful tool for post-installation automation.

Difference between User Context, System Context and Admin Context

In MSI installations, context refers to the access level a process has. It can run in User Context (under the user's profile) or System Context (with full system privileges). Some actions may also require Admin privileges, even if not using System Context.

User Context: Runs under the logged-in user's profile with limited access, suitable for user-specific apps and settings that don't need system-wide changes.

System Context: Runs with full system-wide access as the system user, ideal for system-wide installations and tasks that require complete control.

Admin Context: Not a separate context, but some MSI installations need Admin privileges to make system-wide changes like installing services or modifying system files.

How to assign Logon Scripts ?

Logon scripts, especially when used in conjunction with Active Setup, can be a powerful way to populate user profile data within MSI application packages. Here's how you can use them effectively:

1. Active Setup in MSI

- Runs actions (like copying files or updating registry keys) during user logon.
- Ensures user-specific data is applied automatically.
- Common use: Copy files from machine-level folders to %AppData%.

2. Logon Scripts

- Scripts (batch, PowerShell, or VBScript) run at logon to apply user configurations.
- Can be assigned via Group Policy to users or groups.
- Example: Copy user-specific files from a shared network location to the user profile.

3. Deployment Strategies

- Use **Group Policy** to deploy scripts to organizational units or accounts.
- Use **Software Distribution** to deploy MSI packages with Active Setup or logon scripts.
- Choose scripting languages based on complexity—batch for simplicity, PowerShell for advanced tasks.

4. Example Use Case

- A program needs config files in %AppData% on logon.

- Solution:
 1. Add Active Setup to the MSI.
 2. Create a logon script to copy files.
 3. Deploy using Group Policy or software tools.

5. Best Practices

- Add error handling to scripts.
- Secure scripts, especially when dealing with sensitive data.
- Test thoroughly in your environment.
- Keep clear documentation for future updates or troubleshooting.