Batch Information:
Batch Start Date: 2025-08-04
Batch Name: WiproNGA_DWS_B5_25VID2550
Name: Sooraj B
Email: rajsooraj318@gmail.com
User ID: 34758
Batch ID: 25VID2550

Assignment

1)Application Packaging Process?

Think of this process as preparing a neat and complete software delivery box. It begins with Identify & Collect, where we analyze the application's functionality, installation method, dependencies, and user environment needs. We gather everything from install switches to special configurations.
Next is Packaging, where we create the actual installer using tools like Advanced Installer or InstallShield. This package is designed to install the app in a standard, repeatable way, regardless of the system.
Once the package is built, we move to Quality Assurance (QA). Here, a senior engineer reviews the package for compliance, functionality, and errors before it reaches end users.
After internal checks, the package undergoes User Acceptance Testing (UAT). Real users test it in real environments, making sure it works as expected. Any issues here are noted and fixed before moving ahead.
Finally, we reach Deployment, where the app is rolled out in phases across systems using SCCM, Intune, or other tools. Phased rollout ensures we catch any issues early without disrupting the entire organization.
This structured, step-by-step process ensures applications work perfectly, install smoothly, and require minimal support later.

2)Windows 10 vs Windows 11?

While Windows 10 and Windows 11 share many core packaging principles, Windows 11 introduces more modern frameworks and enhancements for app deployment. The focus in Windows 11 shifts toward better support for MSIX packaging, stronger integration with cloud-native tools like Microsoft Intune, and increased security via hardware-based requirements (like TPM 2.0).
Logon scripts and legacy Active Setup still function, but are being gradually replaced by modern MDM policies and richer Intune-based configuration profiles.
UWP and Desktop Bridge applications receive more attention in Windows 11, and

compatibility testing becomes more important due to more frequent updates in the Windows servicing model.

Overall, packaging for Windows 11 emphasizes modern deployment, automation, and security readiness—while still maintaining backward compatibility with most Windows 10 packaging techniques.

With Windows 11, many packaging behaviors remain similar, but newer UWP and MSIX improvements are emphasized. Deployment tools like Intune now offer more native features, and modern policies replace some legacy scripting methods. Compatibility testing is essential as Windows 11 evolves faster than previous versions.

3) MSI (Window Installer) Contexts ?

When deploying MSI packages, understanding the context in which they run is crucial. The context determines how and where the application installs, what permissions it has, and who can use the app after installation.

1. User Context – This installs the application only for the logged-in user. It uses the user's own privileges, meaning it can't perform system-wide changes like writing to protected folders or modifying machine-level registry keys. It's commonly used for lightweight tools or user-specific apps.
2. System Context – Here, the package installs for all users on the device. It runs with elevated (system-level) permissions, making it suitable for apps that need access to Program Files, system registry entries, or scheduled tasks. SCCM deployments often use this context.
3. Admin Context – This is a subset of the system context, but it's initiated manually by a user with administrative rights. It's often used when packaging scripts or third-party tools need to perform deeper customizations, like driver installs or service creation.

4) The Steps to assign a logon script are ?

1. Open Computer Management from the Start menu under Administrative Tools.
2. Go to Local Users and Groups > Users.
3. Right-click the desired user and choose Properties.
4. Click the Profile tab.
5. In the Logon Script box, type the script name (e.g., startup.bat).
6. Make sure the script is stored in the Net logon shared folder.

7. Click Apply and then OK to save.

SUMMARY

MSI Application Packaging is all about creating neat, ready-to-deploy software bundles that install cleanly across systems without manual effort. MSIX takes it a step further with more modern, container-based deployment.
The process flows like this: first, we understand the app (Discovery), then build the package (Packaging), test it (QA), validate it with users (UAT), and finally push it live (Deployment). We automate tasks, set up user-specific settings using logon scripts or Active Setup, and deploy smartly with tools like SCCM or Intune.
Different install contexts—User, System, Admin—decide how and where the app gets installed. And with Windows 11 pushing newer methods, keeping up with compatibility and policy changes is key.
In short, great packaging means fewer issues, better security, and a smooth experience for everyone involved—from IT teams to end users.