Batch Information:

- **Batch Start Date:** 2025-08-04
- **Batch Name:** WiproNGA_DWS_B5_25VID2550
- **First Name:** Aayush
- **Last Name:** Kumar
- **User ID:** 34758
- **Batch ID:** 25VID2550

Assignment

The Application Packaging process standardizes an environment and reduces administrative and support costs for companies. It helps manage the hundreds or thousands of software products installed on client computers. The process ensures a consistent, stable, and reliable environment, increases software management efficiency, mitigates security issues, and reduces administration and support costs.

The end-to-end process is composed of these five steps:

1. **Identify & Collect:** Research and gather application details and requirements.
2. **Review & Assess:** Analyze information and determine a suitable packaging approach.
3. **Package:** Create the application package according to business requirements and import it into a deployment tool.
4. **Test:** Test the packaged application through UAT (User Acceptance Testing) and a pilot program.
5. **Deploy:** Roll out the packaged application to production.

A key recommendation is to carefully plan the discovery phase, as the package is created based on the requirements and details recorded during this stage. It's also important to use containerized packaging solutions like App-V and MSIX to solve some application compatibility issues. Additionally, using the Application Model in Configuration Manager is recommended for deploying proper applications, as it provides features like dependency, supersedence, and detection methods. Phased and controlled deployments are recommended for production rollouts to mitigate risks, and a regression plan should be in place.

Windows 11, built on the foundation of Windows 10, offers a more modern user experience, enhanced security, and optimized performance. Key benefits of Windows 11 include a redesigned user interface , stronger security features like TPM 2.0 and Windows Hello , and performance improvements resulting in faster logins and wake-up times. It also introduces new features like Snap Groups for multitasking and a modernized Microsoft Store that supports Android apps.

Windows 10, in contrast, offers a more familiar interface and is known for its stability and wide compatibility with a vast array of applications and hardware.

For "App Packs," most applications should work on both operating systems, but it's important to check for compatibility issues. While Windows 11 generally offers better performance, it can vary depending on the specific application and hardware. Ultimately, Windows 11 provides a more modern, secure, and feature-rich experience, but the choice between the two depends on a user's specific needs and preferences.

In MSI (Windows Installer), the context defines the level of access a process or component has within the Windows operating system. The main difference lies in whether the installation runs under the user's profile (User Context) or with elevated system privileges (System Context). There are also situations where actions may require Admin privileges, even if not running directly in a System Context. Understanding these contexts is critical for successfully deploying software with MSI.

User Context

- **Definition:** Runs under the currently logged-in user's credentials and within their user profile.
- **Access:** Can access files and settings specific to the user's profile but generally does not have full system-wide access.
- **Best for:** User-specific applications and tasks that do not require system-wide changes.

System Context

- **Definition:** Runs with elevated privileges, often as the SYSTEM user, and has full system-wide access.
- **Access:** Can access all files and system resources, including those outside the user's profile.
- **Best for:** System-wide installations, critical system policies, and scenarios requiring full control.

Admin Context

- **Definition:** Not a distinct context like User or System, but many MSI installations that require system-wide changes need Admin privileges to run.
- **Access:** Requires the user to have Admin privileges to perform necessary system changes.
- **Best for:** Installations that modify system files, services, or other resources that require elevated permissions.

To assign a logon script to a local user's profile in Windows Server 2003, you must be logged in as an Administrator or a member of the Administrators group. This script will run when the local user logs on to the computer, but not when they log on to a domain.

The steps to assign a logon script are:

1. Open
   Computer Management by clicking **Start**, pointing to **Administrative Tools**, and selecting **Computer Management**.
2. In the console tree, expand
   Local Users and Groups and then click **Users**.
3. Right-click the desired user account in the right pane and select
   Properties.
4. Click the
   Profile tab.
5. In the
   Logon script box, type the file name of the logon script. If the script is in a subfolder of the default logon script path, you should include the relative path. For example, if `Startup.bat` is in `\\ComputerName\Netlogon\FolderName`, you would type `FolderName\Startup.bat`.
6. Click
   Apply, and then click **OK**.

Local logon scripts must be stored in a shared folder named

`Netlogon` or in a subfolder within it. The default location for these scripts is

`Systemroot\System32\Repl\Imports\Scripts`, but this folder is not created by default during a new installation of Windows. Therefore, you must create and share this folder using the

`Netlogon` share name. Alternatively, you can store the logon script in any folder the user can access during logon and then share that folder.