Thursday , 22 May 2025 (Summary)

MSI Contexts

In MSI, **context** refers to the level of access an installation or action has in the Windows operating system. There are three main contexts to understand:

1. User Context

Definition: Runs under the current user's profile and credentials.
Access: Only to user-specific files and settings.
Best for: User-specific applications ,Customizations, Tasks that don’t require system-level changes

---

2. System Context

Definition: Runs with elevated privileges as the SYSTEM user.
Access: Full access to all system files and settings.
Best for: System-wide installations, Enforcing system policies, Critical updates or configurations

---

3. Admin Context (Implied)

Definition: Not a separate context, but involves requiring **Admin rights** during installation.
Access: Needed to perform system-level changes.
Best for: Installing services, Modifying system files or settings, Any operation needing Admin approval

---

User Context = Limited to user's profile
System Context = Full system access
Admin Context = Requires Admin privileges for system-wide changes

---

Logon Scripts to populate User Profile Data in msi application packaging

1. Active Setup in MSI Packages

Purpose: Executes actions (e.g., copying files, modifying registry) during user login.
Use Case: Ensures required files/configs are added to the user's profile after installation.
Example: Copying a config file to %AppData% during user logon.

2. Logon Scripts

Script Types: Batch files, PowerShell, VBScript.
Purpose: Executes user-specific actions (e.g., copying files) at login.
Example: Copy files from \\server\netlogon\MyApp to the user's profile folder.

3. Deployment Strategies

Group Policy: Assign logon scripts to users or Organizational Units (OUs).
Software Distribution: Deploy MSI and scripts via GPO Software Distribution.
Script Choice: Use PowerShell for advanced tasks; batch for simpler needs.

4. Example Use Case

Scenario: An app needs to store user-specific settings in %AppData%.
Solution:
Add Active Setup to MSI to trigger logon action.
Use a logon script to copy required files to the user’s profile.
Deploy the MSI + script via Group Policy.

5. Best Practices

Error Handling: Handle issues (e.g., missing paths, network issues) in scripts.
Security: Avoid exposing sensitive data.
Testing: Validate scripts in various user environments.
Keep proper notes for future maintenance.

Windows 11 benefits Over Windows 10

Enhanced security and performance, especially in task scheduling and registry virtualization. Improved compatibility with modern packaging tools (like MSIX). User profile management has better roaming and syncing capabilities across devices. Versioning Active Setup and Versioning in Windows Purpose of Active Setup: Active Setup is a Windows feature that enables applications to perform user-specific configuration during user login.

Versioning

Active Setup and Versioning in Windows Purpose of Active Setup:
Active Setup is a Windows feature that enables applications to perform user-specific configuration during user login. It works by comparing version values in two registry hives: HKLM(HKEYLOCALMACHINE)–Storesthemasterconfiguration: applicationnameStubPath It works by comparing version values in two registry hives: HKLM

(HKEY_LOCAL_MACHINE) – Stores the master configuration: application name, StubPath, and Version. HKCU (HKEY_CURRENT_USER) – Stores user-specific configuration data.