Thursday, 22 May 2025 6:13 PM

MSI Contexts

1. User Context:

Runs under the currently logged-in user's credentials and within their user profile. Can access files and settings specific to indivisual user and don't require system-wide changes.

2. System Context:

Runs with elevated privileges, often as the system user, with full system-wide access. Has access to all files and system resources, including those outside the user's profile.

3. Admin Context (Implicit in some cases):

Although not a distinct context like User or System, many MSI installations that require system-wide changes (like installing a program that runs as a service) might require Admin privileges.

Active Setup and logon scripts within MSI application packaging:

1. Leverage Active Setup in MSI Packages

Purpose: Active Setup allows running custom actions (e.g., copy files, update registry, run scripts) at user login.

How it Works: By embedding Active Setup in an MSI, actions are triggered when users log in, ensuring per-user setup.

2. Create and Assign Logon Scripts

Batch files, PowerShell, VBScript, etc.

Copy user-specific files from a shared network location to %AppData% during login.

3. Consider Deployment Strategies

Assign logon scripts to OUs or user accounts.
Software Distribution: Deploy MSI packages using tools like Group Policy Software Distribution.
Batch is simple; PowerShell provides more advanced capabilities.

4. Example: Copying User Settings Files

An app needs user-specific settings in %AppData% on first login.

1. Add Active Setup in MSI to trigger a logon script.
2. Create a logon script to copy files from a network share to %AppData%.
3. Use Group Policy or Software Distribution to deploy the MSI and script.

5. Best Practices

Error Handling: Handle issues like network problems in scripts.
Security: Secure scripts, especially when accessing sensitive data.
Testing: Test deployment processes thoroughly.
Documentation: Keep scripts and deployment procedures well documented.

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.

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.

.