Summary of MSI Application Packaging and Windows Tools

Difference Between User, Admin, and System Context in MSI

- 1. User Context

- **Runs under**: Logged-in user's credentials.
- **Access**: Limited to user profile directories and settings.
- **Use Case**: Ideal for installing user-specific apps or making non-system changes.

---

2. System Context

- **Runs under**: SYSTEM account (high privilege).
- **Access**: Full system-level access across the OS.
- **Use Case**: Required for system-wide installs, services, and policies.

---

3. Admin Context (Implied)

- **Definition**: Not a separate context, but many MSI operations **require admin rights** to run.
- **Access**: Elevated permission needed to modify system files/services.
- **Use Case**: Installing apps that alter shared or protected system areas.

Context	Access Level	Typical Usage
User Context	Limited to user profile	User-level apps or settings
System Context	Full OS access (SYSTEM user)	System-wide apps, services, policies
Admin Context	Requires admin privileges	MSI installs with elevated system changes

Logon Scripts to Populate User Profile Data in MSI Application Packaging

Purpose

Logon scripts, when paired with **Active Setup**, help populate **user-specific data** (e.g., config files, registry settings) during user logon — especially important for MSI deployments that install in **system context**.

1. Using Active Setup in MSI Packages

- **What it does**: Triggers actions (copy files, update registry, run scripts) during a user's first logon.
- **How**: Embed Active Setup keys in the MSI to run post-installation actions under the user profile.
- **Example**: Copy config files from a machine-level location to %AppData%.

---

2. Creating Logon Scripts

- **Script Types**: Batch files, PowerShell scripts, VBScript.
- **Function**: Perform tasks like copying files to user directories or setting user-specific registry keys.
- **Assignment**: Use **Group Policy** or assign directly to user accounts.

---

3. Deployment Strategies

- **Group Policy (GPO)**:
  - Assign logon scripts to OUs or user groups.
  - Use Software Installation feature to deploy MSI + scripts.
- **Choose the right scripting language**:
  - Batch: Simple tasks.
  - PowerShell: Complex logic, error handling.

---

4. Example Scenario

Goal: Deploy app with user-specific settings.

- **MSI Package** includes Active Setup entry.
- **Logon Script** copies settings from a network share to %AppData%\MyApplication.

- **Deployed via** Group Policy or Software Distribution.

---

5. Best Practices

- Include **error handling** (e.g., for network failures).
- Ensure **security** (avoid exposing sensitive data in scripts).
- **Test thoroughly** across user profiles and machines.
- **Document** the scripts, logic, and deployment steps for maintainability.

Windows 11 vs Windows 10: Key Differences and Considerations

Windows 11 – Key Benefits

- **Modern UI**: Sleek, updated design with centered Start Menu, rounded corners, and cleaner layout.
- **Stronger Security**: Built-in support for **TPM 2.0**, **Windows Hello**, and better zero-trust posture.
- **Performance Boosts**: Faster boot, web browsing, and wake-up times.
- **Enhanced Multitasking**: Features like **Snap Layouts** and **Snap Groups** help manage multiple windows efficiently.
- **AI Integration**: **Windows Copilot** offers real-time assistance and productivity tools.
- **Gaming Upgrades**: Support for **DirectX 12 Ultimate** and **DirectStorage** for better graphics and load times.
- **New Microsoft Store**: Supports Android apps via Amazon Appstore.
- **Optimized Updates**: Smaller, faster updates reduce downtime.

Windows 10 – Key Benefits

- **Familiar Interface**: Comfortable for users accustomed to older versions.
- **Extensive Compatibility**: Supports a vast range of legacy hardware and applications.
- **Proven Stability**: Long-term reliability with fewer changes to the user experience.
- **Cost-Effective**: Often cheaper, especially for upgrading older systems.

Considerations for App Pack Users

- **App Compatibility**: Most apps run on both OS versions, but legacy apps might perform more reliably on Windows 10.
- **Performance**: Windows 11 is more optimized but requires newer hardware to see full benefits.
- **Security**: Windows 11 provides better out-of-the-box protection—important for enterprise environments.
- **New Features**: Windows 11 adds tools like Copilot and Snap Assist that boost productivity.

Using Windows Tools for Debugging (Sysinternals Utilities)

These tools are vital for IT professionals involved in system **diagnostics**, **administration**, and **security monitoring**:

1. Autologon

- **Purpose**: Automates Windows logins without prompting the user.
- **How**: Stores login credentials in the registry.
- **Use Case**: Ideal for test environments or kiosks/headless systems.

---

2. Process Explorer

- **Purpose**: An advanced Task Manager replacement.
- **How**: Displays detailed info about processes, memory, handles, and loaded DLLs.
- **Use Case**: Detect malware, troubleshoot resource usage, inspect file locks.

---

3. PsExec

- **Purpose**: Executes commands and applications **remotely**.
- **How**: Enables command-line access to other systems over the network.
- **Use Case**: Remote administration, patching, silent software installs.

---

4. PSTools Suite

- **Purpose**: A set of command-line utilities for system and network management.
- **Components**: Includes tools like PsLoggedOn, PsList, PsFile, etc.
- **Use Case**: Administer both local and remote systems from the command line.

---

5. RegMon (Registry Monitor)

- **Purpose**: Monitors and logs **real-time registry activity**.
- **How**: Captures registry reads, writes, and deletes.
- **Use Case**: Registry troubleshooting, malware detection, policy enforcement.

---

6. Sysmon (System Monitor)

- **Purpose**: Logs detailed system activity for security auditing.
- **What It Tracks**: Process creation, network connections, file changes.
- **Use Case**: Threat hunting, incident response, and digital forensics.

---

7. Whois (non-Sysinternals)

- **Purpose**: Queries WHOIS databases for domain/IP registration info.
- **Use Case**: Network analysis, identifying domain ownership, threat intelligence.

Active Setup Versioning to Ensure It Runs Each Time During a Fresh Install

What is Active Setup?

Active Setup is a Windows feature that allows applications to perform **user-specific configuration** during user logon. It compares **registry versions** between system-wide and user-specific keys to determine if setup tasks need to run.

Key Registry Paths

- **HKLM (HKEY_LOCAL_MACHINE)**:

    - Stores the **master** configuration for Active Setup.
    - Includes:
        - DisplayName (app name),
        - StubPath (command to run),
        - Version (version number).
- **HKCU (HKEY_CURRENT_USER)**:
    - Stores **per-user** configuration after Active Setup runs.

How It Works

- On user login, Windows compares the **HKLM version** with the **HKCU version**.
- If HKCU is **missing** or the version is **lower** than HKLM:
    - Windows **executes the StubPath command** (e.g., an MSI installer).
    - The **HKCU key is updated** to match the HKLM version.

Example

registry

```
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{GUID}
@ = "MyApp"
StubPath = "msiexec.exe /qb /i C:\Path\To\MyInstall.msi"
Version = "1,0,0"
```

- If HKCU is empty or has an older version, Active Setup runs the MSI during the next user logon.

Fresh Install Strategy

To **force Active Setup to run on new logins**:

- **Increment the "Version" value** in the **HKLM key** (e.g., from "1,0,0" to "1,0,1").
- This ensures the setup command runs again even for users who previously skipped or partially completed the configuration.