**Distinction Between User, Admin, and System Contexts:**

Understanding these contexts is essential for properly deploying software using MSI installers. There are three primary execution contexts:

1. **User Context**

* Operates under the currently logged-in user's profile.
* Has access only to the user's personal files and settings, without elevated permissions across the system.
* Ideal for deploying applications or customizations meant solely for individual users and that don’t require administrative rights.

1. **System Context**

* Executes with high-level privileges, typically as the system account, allowing unrestricted access to the entire operating system.
* Can interact with files and resources beyond a specific user's environment.
* Best suited for installations that affect all users on a device or require full system-level modifications.

1. **Admin Context**

* Refers to a user with administrative rights who isn't necessarily the active user session.
* Needed for installations that demand admin-level privileges to modify protected system areas or configurations.
* Suitable for setting up applications or services that require changes to core system components or shared directories.

### **Logon Script Overview**

A **logon script** is a file that executes automatically when a user signs in to their computer session. It's a valuable tool for automating tasks and customizing user environments. Below are effective ways to use logon scripts alongside **Active Setup** in enterprise deployments:

### **1) Utilizing Active Setup Within MSI Installers**

* Active Setup enables the execution of specific actions during user logon—such as copying files, updating registry settings, or launching scripts.
* Embed Active Setup components within your MSI package to ensure these actions occur every time a user logs in, delivering personalized configurations.
* A common use case includes transferring configuration files from machine-level directories to a user’s **AppData** folder at logon.

### **2) Defining and Applying Logon Scripts**

* Scripts can be written in **batch**, **PowerShell**, or **VBScript**, depending on your environment and complexity.
* These scripts typically move user-specific files from a central network share to the individual’s profile path when they log in.
* You can assign logon scripts to single users or user groups using **Group Policy** settings.

### **3) Planning Deployment Approaches**

* Use **Group Policy Objects (GPOs)** to deploy logon scripts across specific **Organizational Units (OUs)** or user accounts.
* Through Group Policy Software Deployment, MSI packages can be pushed to systems—these packages may include both Active Setup and logon scripts.
* Choose the right scripting format: batch files are straightforward, while PowerShell offers greater flexibility and functionality.

### **4) Sample Use Case – Delivering User Preferences**

**Scenario:**  
An application requires that user-specific configuration files be present in the **%AppData%** directory right after login.

**Solution:**

1. **MSI Integration**: Embed an Active Setup entry that calls a logon script when the user signs in.
2. **Logon Script**: Build a script (e.g., batch file) that copies required settings from a shared drive (e.g., \\server\netlogon\MyApplication) to %AppData%\MyApplication.
3. **Deployment**: Roll out the MSI and script using Group Policy or a centralized software distribution tool.

### **5) Recommended Best Practices**

* **Error Handling**: Add proper error-checking mechanisms to avoid silent failures during script execution.
* **Security Considerations**: Protect sensitive file paths or credentials if used within scripts.
* **Validation**: Test your scripts and deployment procedures in a staging environment before going live.
* **Documentation**: Keep a detailed record of script logic, deployment methods, and any dependencies for smoother management and troubleshooting.

**Windows 11 VS Windows 10**

Windows 11 generally provides a better overall experience due to its optimized performance, improved security features, and streamlined interface.

**Windows 11 Benefits**

* Improved User Interface
* Enhanced Security
* Performance Improvements
* Modernized Microsoft Store
* Improved Multi-tasking
* Integrated AI Assistant
* Enhanced Gaming Experience
* Optimized Update Process

**Windows 10 Benefits**

* Familiar Interface
* Wide Compatibility
* Stability
* Cost-Effective

**Considerations for an App-Pack**

* App Compatibility
* Performance
* Security

**Using Windows Tools for Debugging**

1. **Autologon**

* It automates the login process on a Windows system.
* It's a GUI tool that configures the Windows registry to automatically log on a specified user with provided credentials.
* It is useful for headless systems or automated testing environments.

1. **Process Explorer**

* It is a powerful tool for viewing and managing running processes.
* It provides detailed information about processes including memory usage, handles and open files.
* It is essential for troubleshooting process-related issues, identifying resource bottlenecks and investigating malware.

1. **PsExec**

* It is a powerful tool for remote execution of commands and programs.
* It allows administrators to run applications on a remote computer as if they were running locally.
* It is useful for remote system management, patching, and troubleshooting.

1. **PSTools**

* It is a collection of command-line tools for system administration and troubleshooting.
* It includes tools like PsLoggedOn, PsFile and PsList.
* It provides a wide range of administrative capabilities for local and remote systems.

1. **RegMon**

* It monitors registry access and changes in real-time.
* It tracks all registry activity including reads, writes and deletes.
* It helps troubleshoot registry-related issues, identify rogue applications and investigate security vulnerabilities.

1. **Sysmon**

* It is a Windows system service and driver that monitors and logs system activity.
* It provides detailed information about process creations, network connections and file access changes.
* It is essential for security monitoring, intrusion detection and forensic analysis.

1. **Whois**

* It is a command-line tool used to retrieve information about domain names and IP addresses.
* It queries a Whois database to retrieve registration details.
* It is useful for network troubleshooting, identifying domain owners and checking domain availability.

**Active Setup Versioning to ensure it runs each time during Fresh Install**

To ensure Active Setup runs during a fresh install, increment the **"Version"** value in the **HKLM (HKEY\_LOCAL\_MACHINE)** registry key. This forces the Active Setup process to compare the **HKLM** version with the **HKCU (HKEY\_CURRENT\_USER)** version and execute the **"StubPath"** command when a user logs in.

1. **Active Setup and Versioning**

* Active Setup is a Windows mechanism that allows an application to perform user-specific configuration upon user login.
* It works by comparing versions in the HKLM and HKCU registry hives.

1. **HKLM vs HKCU**

* **HKLM:** Stores the master Active Setup data such as application name, StubPath and Version.
* **HKCU:** Stores the user-specific Active Setup data which is populated based on the HKLM data during logon.

1. **Incrementing the Version**

* If the version in HKLM is higher than the version in HKCU, Active Setup will execute the command specified in the "StubPath" value and update the HKCU version.