

Difference between User, Admin & System Context

Contexts are crucial for correctly deploying software using MSI.

There are 3 types of contexts:

1) User Context

- It runs under currently logged-in user within their user profile.
- It can access files and settings specific to the user profile but it doesn't have full system-wide access.
- It's best used for user-specific applications, customizations, and tasks that don't require system-wide changes.

2) System Context

- It runs with elevated privileges, often as the system user with full system-wide access.
- It has access to all files and system resources including those outside the user's profile.
- It's best for system-wide installations and scenarios where full control is needed.

3) Admin Context

- Admin is a person who has full access to the system but isn't the user.
- It can access the installations required to have Admin privileges to run the MSI and perform the necessary system changes.
- It's best for installations that modify system files, services or other resources that require elevated permissions.

Logon Script

Logon Script is a script that runs automatically as the user logs in.

Ways to use logon script with Active Setup effectively:

1) Leverage Active Setup in MSI Packages

- It allows you to run specific actions like copying files, updating registry keys or executing scripts during user logon process.
- Include Active Setup within our MSI package to trigger these actions whenever a user logs in, ensuring user-specific data is available.
- It is used to copy configuration files from per-machine location to the user's AppData folder during logon

2) Create and Assign Logon Scripts

- These scripts can be batch files, PowerShell scripts or even other scripting languages like VBScript.
- A script copies user-specific files from a shared network location to the user's profile directory during logon.
- Logon scripts can be assigned to individual users or to groups of users via Group Policy.

3) Consider Deployment Strategies

- In Group Policy we can deploy logon scripts and assign them to specific organizational units (OU) or user accounts.
- In Software Distribution we can use Group Policy Software Distribution to deploy MSI packages, including those that utilize logon scripts or Active Setup.
- Choose a scripting language suitable for your needs. Batch files are simpler, while PowerShell offers more advanced capabilities.

4) Example Scenario: Copying User Settings Files

Scenario: An application needs to store user-specific settings files in the user's AppData folder, but these files need to be available immediately upon logon.

Solution:

1. **MSI Package:** Include an Active Setup entry that triggers a logon script during user logon.
2. **Logon Script:** Create a script (e.g., a batch file) that copies the application's settings files from a shared network location (e.g., \\server\netlogon\MyApplication) to the user's AppData folder (%AppData%\MyApplication).
3. **Deployment:** Deploy the MSI package and the associated logon script using Group Policy or Software Distribution.

5) Best Practices

- **Error Handling:** Incorporate error handling into your logon scripts to gracefully handle potential issues.
- **Security:** Ensure scripts are secure, especially when dealing with sensitive data or file paths.
- **Testing:** Thoroughly test your scripts and deployment process to ensure they work as expected in your environment.
- **Documentation:** Document your scripts, deployment procedures and any related configurations for easy maintenance 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.

2) 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.

3) 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.

4) 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.

5) 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.

6) 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.

7) 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.

2) 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.

3) 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.