

7 Reasons Why your Installations May Fail on Windows Vista (And How You Can Fix Them)

By Stefan Krueger

Author: www.installsite.org

Windows Installer is not just a technology but also a set of rules and best practices. While these rules have been specified in the SDK for several years, many of them haven't been enforced. While ignoring these rules didn't break your setups in the past, installations that don't follow best practices may eventually fail on Windows Vista™. This article alerts you to the most common pitfalls in setup authoring and helps you troubleshoot your installations to be compatible with Windows Vista.

Below is a list of problems that cause setups that seemingly worked on Windows XP to now fail on Windows Vista. In most cases, the problem arises due to **User Account Control (UAC)** restrictions.

On Windows Vista, even members of the Administrators group work with restricted permissions most of the time. If elevated rights are required for a task, UAC prompts users for their consent. In the past, such elevation often took place in corporate deployment scenarios, where an administrator advertised a setup that a restricted user could then install as a "blessed" setup. But with UAC on Windows Vista, this elevation becomes standard for all interactive installations.

1) Property values aren't carried over from the UI sequence to the Execute sequence.

Symptoms: Property values that you set, or that a users enters, in the User Interface sequence are lost or reset to their default values when your setup switches to the Execute sequence. In the log file you see a message like:

Ignoring Disallowed Property PROPERTYNAME

Cause: This problem occurs if a privilege elevation takes place when the setup switches to the Execute sequence. Before Windows Vista this mainly affected corporate deployment scenarios, but with UAC this is now the default for all MSI installations.

Solution: The properties that you want to pass from the UI sequence to the Execute sequence must be listed as being secure. To do this, add a property called SecureCustomProperties to the Property Manager and enter a list of your properties (delimited by semicolons) as its value (for example, "EDITPROP;RADIOPROP"). For more information, see the documentation in the [Platform SDK](#).

2) Custom Actions fail because of missing permissions

Symptom: Custom actions scheduled for immediate or deferred execution fail with error 1721.

Cause: On Windows Vista, even administrators have restricted permissions by default. Windows Vista prompts for privilege elevation if needed. However, only custom actions that are running in the Local System account get elevated correctly, while custom actions that impersonate as the current user do not.

Solution: Schedule the custom action for "deferred execution in system context", which is equivalent to setting the `msidbCustomActionTypeNoImpersonate` bit in the CustomAction table.

3) Custom Action still fails because of missing privileges

Symptom: Custom action fails even though it is set to run "Deferred in System Context" (`msidbCustomActionTypeNoImpersonate`).

Cause: In Windows Vista, NT services can explicitly have only the necessary privileges. The Windows Installer engine, running in the Local System account on Windows Vista, has lesser permissions than it had on previous Windows versions. For instance, the Windows Installer service does not have the permission `SeBackupPrivilege`, which is required to call functions such as `RegLoadKey`. Custom actions inherit the privileges from Windows Installer and thus fail if they try to call this function.

Solution: There is currently no solution. For more information please see [Christopher Painter's Blog](#) and the [Vista Compatibility Team Blog](#).

4) Setup works when using setup.exe but fails when directly launching the .msi file

Symptom: When you launch your setup by double clicking the .msi file or running `msiexec` it fails, but if you launch it using `setup.exe` your setup succeeds.

Cause: Windows Vista detects that `setup.exe` is an install program that will probably require elevated rights, and therefore displays the UAC prompt. As a result, the UI sequence will run with elevated privileges. However, if you launch the .msi file directly the UAC prompt will happen later, when your setup switches from the UI sequence to the Execute sequence. In this case the UI sequence runs with restricted permissions.

Solution: Custom actions that modify the target system (and therefore need elevated rights) should be scheduled as deferred in the system context and placed in the Execute sequence between `InstallInitialize` and `InstallFinalize`. Modifying the system in the UI sequence has never been good practice. Often, this was used to run child installs without using the deprecated "nested install" actions. The "InstallShield MSDE 2000 Object for NT Systems" is an example of this, and it will fail on Windows Vista.

5) Setup fails in silent mode

Symptom: Your setup fails if run in silent mode, e.g., using `msiexec /q`, but succeeds in Full, Reduced, and Basic UI modes.

Cause: In silent mode Windows Installer does not display the UAC elevation prompt, so the entire setup runs with restricted permissions.

Solution: If running in one of the other UI modes is not feasible, use the Elevated Command

Prompt to launch your setup, or use a management tool such as SMS, Tivoli, Altiris, etc., to deploy your setup.

6) Setup works when run by the built-in Administrator account but fails for other users of the Administrators group

Cause: The built-in account named "Administrator" — which is disabled by default — is treated in a special way. UAC restrictions do not apply to this special account, but they do apply for all other members of the Administrators group.

Solution: Find out why your setup fails if UAC restrictions apply; maybe it is one of the problems listed above, and fix it.

7) Setup packaged as single EXE or web download fails

Symptom: hile your setup executable unpacks or downloads, it fails with a message similar to: Unable to save file C:\Windows\Downloaded Installations\...

Cause: Because of UAC, even members of the Administrators group do not have write access to locations such as the Windows folder and the Program Files folder.

Solution: Change the location of the folder for the extracted or downloaded setup files from "[WindowsFolder]Downloaded Installations" to "[LocalAppDataFolder]Downloaded Installations" (without the quotes).

About the Author

Stefan Krueger is a freelance setup consultant and runs InstallSite.org, a forum where installation developers share resources and information with peers. Krueger has been recognized by Microsoft as a Windows Installer [MVP](#) (Most Valuable Professional).

Krueger welcomes any [feedback](#) on this article.