How User Account Control works

By officedocspr5

Archived: 2026-04-06 00:27:26 UTC

User Account Control (UAC) is a key part of Windows security. UAC reduces the risk of malware by limiting the ability of malicious code to execute with administrator privileges. This article describes how UAC works and how it interacts with the end-users.

With UAC, each application that requires the *administrator access token* must prompt the end user for consent. The only exception is the relationship that exists between parent and child processes. Child processes inherit the user's access token from the parent process. Both the parent and child processes, however, must have the same *integrity level*.

Windows protects processes by marking their integrity levels. Integrity levels are measurements of trust:

- A *high integrity application* is one that performs tasks that modify system data, such as a disk partitioning application
- A *low integrity application* is one that performs tasks that could potentially compromise the operating system, like as a Web browser

Applications with lower integrity levels can't modify data in applications with higher integrity levels. When a standard user attempts to run an app that requires an administrator access token, UAC requires that the user provides valid administrator credentials.

To better understand how this process works, let's take a closer look at the Windows sign in process.

The following diagram shows how the sign in process for an administrator differs from the sign in process for a standard user.


Diagram that describes the UAC Windows sign-in process.

By default, both standard and administrator users access resources and execute apps in the security context of a standard user.
When a user signs in, the system creates an access token for that user. The access token contains information about the level of access that the user is granted, including specific security identifiers (SIDs) and Windows privileges.

When an administrator logs on, two separate access tokens are created for the user: a *standard user access token* and an *administrator access token*. The standard user access token:

- Contains the same user-specific information as the administrator access token, but the administrative Windows privileges and SIDs are removed
- Is used to start applications that don't perform administrative tasks (standard user apps)

- Is used to display the desktop by executing the process *explorer.exe*. Explorer.exe is the parent process from which all other user-initiated processes inherit their access token. As a result, all apps run as a standard user unless a user provides consent or credentials to approve an app to use a full administrative access token

A user that is a member of the Administrators group can sign in, browse the Web, and read e-mail while using a standard user access token. When the administrator needs to perform a task that requires the administrator access token, Windows automatically prompts the user for approval. This prompt is called an *elevation prompt*, and its behavior can be configured via policy or registry.

When UAC is enabled, the user experience for standard users is different from administrator users. The recommended and more secure method of running Windows, is to ensure your primary user account is a standard user. Running as a standard user helps to maximize security for a managed environment. With the built-in UAC elevation component, standard users can easily perform an administrative task by entering valid credentials for a local administrator account.

The default, built-in UAC elevation component for standard users is the *credential prompt*.

The alternative to running as a standard user is to run as an administrator in *Admin Approval Mode*. With the built-in UAC elevation component, members of the local Administrators group can easily perform an administrative task by providing approval.

The default, built-in UAC elevation component for an administrator account in Admin Approval Mode is called the *consent prompt*.

The credential prompt is presented when a standard user attempts to perform a task that requires a user's administrative access token. Administrators can also be required to provide their credentials by setting the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting value to **Prompt for credentials**.

 Screenshot showing the UAC credential prompt.

The consent prompt is presented when a user attempts to perform a task that requires a user's administrative access token.

 Screenshot showing the UAC consent prompt.

The UAC elevation prompts are color-coded to be app-specific, enabling for easier identification of an application's potential security risk. When an app attempts to run with an administrator's full access token, Windows first analyzes the executable file to determine its publisher. Apps are first separated into three categories based on the file's publisher:

- Windows
- Publisher verified (signed)
- Publisher not verified (unsigned)

The elevation prompt color-coding is as follows:

- Gray background: The application is a Windows administrative app, such as a Control Panel item, or an application signed by a verified publisher
   Screenshot showing the UAC credential prompt with a signed executable.
- Yellow background: the application is unsigned or signed but isn't trusted
   Screenshot showing the UAC consent prompt with an unsigned executable.

Some Control Panel items, such as **Date and Time**, contain a combination of administrator and standard user operations. Standard users can view the clock and change the time zone, but a full administrator access token is required to change the local system time. The following is a screenshot of the **Date and Time** Control Panel item.

 Screenshot showing the UAC Shield Icon in Date and Time Properties.

The shield icon on the **Change date and time...** button indicates that the process requires a full administrator access token.

The elevation process is further secured by directing the prompt to the *secure desktop*. The consent and credential prompts are displayed on the secure desktop by default. Only Windows processes can access the secure desktop. For higher levels of security, we recommend keeping the **User Account Control: Switch to the secure desktop when prompting for elevation** policy setting enabled.

When an executable file requests elevation, the *interactive desktop*, also called the *user desktop*, is switched to the secure desktop. The secure desktop dims the user desktop and displays an elevation prompt that must be responded to before continuing. When the user selects **Yes** or **No**, the desktop switches back to the user desktop.

Note

Starting in **Windows Server 2019**, it's not possible to paste the content of the clipboard on the secure desktop. This behavior is the same as the currently supported Windows client OS versions.

Malware can present an imitation of the secure desktop, but when the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting is set to **Prompt for consent**, the malware doesn't gain elevation if the user selects **Yes** on the imitation. If the policy setting is set to **Prompt for credentials**, malware imitating the credential prompt might be able to gather the credentials from the user. However, the malware doesn't gain elevated privilege and the system has other protections that mitigate malware from taking control of the user interface even with a harvested password.

While malware could present an imitation of the secure desktop, this issue can't occur unless a user previously installed the malware on the PC. Because processes requiring an administrator access token can't silently install when UAC is enabled, the user must explicitly provide consent by selecting **Yes** or by providing administrator credentials. The specific behavior of the UAC elevation prompt is dependent upon security policies.
