

# ImpersonateLoggedOnUser function (securitybaseapi.h) - Win32 apps

By GrantMeStrength

Archived: 2026-04-05 21:07:40 UTC

The **ImpersonateLoggedOnUser** function lets the calling thread impersonate the [security context](#) of a logged-on user. The user is represented by a token handle.

```
BOOL ImpersonateLoggedOnUser(  
    [in] HANDLE hToken  
);
```

[in] hToken

A handle to a primary or impersonation [access token](#) that represents a logged-on user. This can be a token handle returned by a call to [LogonUser](#), [CreateRestrictedToken](#), [DuplicateToken](#), [DuplicateTokenEx](#), [OpenProcessToken](#), or [OpenThreadToken](#) functions. If *hToken* is a handle to a [primary token](#), the token must have **TOKEN\_QUERY** and **TOKEN\_DUPLICATE** access. If *hToken* is a handle to an [impersonation token](#), the token must have **TOKEN\_QUERY** and **TOKEN\_IMPERSONATE** access.

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call [GetLastError](#).

The impersonation lasts until the thread exits or until it calls [RevertToSelf](#).

The calling thread does not need to have any particular [privileges](#) to call **ImpersonateLoggedOnUser**.

If the call to **ImpersonateLoggedOnUser** fails, the client connection is not impersonated and the client request is made in the security context of the process. If the process is running as a highly privileged account, such as LocalSystem, or as a member of an administrative group, the user may be able to perform actions they would otherwise be disallowed. Therefore, it is important to always check the return value of the call, and if it fails, raise an error; do not continue execution of the client request.

All impersonate functions, including **ImpersonateLoggedOnUser** allow the requested impersonation if one of the following is true:

- The caller has the **SeImpersonatePrivilege** privilege.
- A process (or another process in the caller's logon session) created the token using explicit credentials through [LogonUser](#) or [LsaLogonUser](#) function.
- The authenticated identity is same as the caller.

## Important

The token must have an impersonation level of **SecurityImpersonation** or higher for impersonation to succeed. Tokens with **SecurityIdentification** or **SecurityAnonymous** levels cannot be used for impersonation, even if the caller has **SeImpersonatePrivilege**. **SecurityIdentification** tokens allow identity verification and ACL checks but do not permit impersonation.

The behavior varies based on the token's impersonation level:

- **SecurityAnonymous**: The server cannot obtain client identity information and cannot impersonate the client.
- **SecurityIdentification**: The server can obtain the client's identity and perform access validation, but cannot impersonate the client. This is the default level for many scenarios.
- **SecurityImpersonation**: The server can impersonate the client's security context on the local system. This is the minimum level required for **ImpersonateLoggedOnUser** to succeed.
- **SecurityDelegation**: The server can impersonate the client on remote systems as well as locally.

**Windows XP with SP1 and earlier:** The **SeImpersonatePrivilege** privilege is not supported.

For more information about impersonation, see [Client Impersonation](#).

Requirement	Value
Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	securitybaseapi.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

[Client/Server Access Control Functions](#)

[Client/Server Access Control Overview](#)

[CreateProcessAsUser](#)

[CreateRestrictedToken](#)

[DuplicateToken](#)

[DuplicateTokenEx](#)

[LogonUser](#)

[OpenProcessToken](#)

[OpenThreadToken](#)

[RevertToSelf](#)

---

Source: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa378612\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378612(v=vs.85).aspx)