

## LogonUserA function (winbase.h) - Win32 apps

By GrantMeStrength

Archived: 2026-04-06 02:05:05 UTC

The **LogonUser** function attempts to log a user on to the local computer. The local computer is the computer from which **LogonUser** was called. You cannot use **LogonUser** to log on to a remote computer. You specify the user with a user name and domain and [authenticate](#) the user with a [plaintext](#) password. If the function succeeds, you receive a handle to a token that represents the logged-on user. You can then use this token handle to impersonate the specified user or, in most cases, to create a [process](#) that runs in the context of the specified user.

```
BOOL LogonUserA(  
    [in]          LPCSTR  lpszUsername,  
    [in, optional] LPCSTR  lpszDomain,  
    [in, optional] LPCSTR  lpszPassword,  
    [in]          DWORD   dwLogonType,  
    [in]          DWORD   dwLogonProvider,  
    [out]         PHANDLE phToken  
);
```

[in] lpszUsername

A pointer to a null-terminated string that specifies the name of the user. This is the name of the user account to log on to. If you use the [user principal name](#) (UPN) format, *User@DNSDomainName*, the *lpszDomain* parameter must be **NULL**.

[in, optional] lpszDomain

A pointer to a null-terminated string that specifies the name of the domain or server whose account database contains the *lpszUsername* account. If this parameter is **NULL**, the user name must be specified in UPN format. If this parameter is ".", the function validates the account by using only the local account database.

[in, optional] lpszPassword

A pointer to a null-terminated string that specifies the plaintext password for the user account specified by *lpszUsername*. When you have finished using the password, clear the password from memory by calling the [SecureZeroMemory](#) function. For more information about protecting passwords, see [Handling Passwords](#).

[in] dwLogonType

The type of logon operation to perform. This parameter can be one of the following values, defined in Winbase.h.

Value	Meaning
-------	---------

<p><b>LOGON32_LOGON_BATCH</b></p>	<p>This logon type is intended for batch servers, where processes may be executing on behalf of a user without their direct intervention. This type is also for higher performance servers that process many plaintext authentication attempts at a time, such as mail or web servers.</p>
<p><b>LOGON32_LOGON_INTERACTIVE</b></p>	<p>This logon type is intended for users who will be interactively using the computer, such as a user being logged on by a <a href="#">terminal</a> server, remote shell, or similar process. This logon type has the additional expense of caching logon information for disconnected operations; therefore, it is inappropriate for some client/server applications, such as a mail server.</p>
<p><b>LOGON32_LOGON_NETWORK</b></p>	<p>This logon type is intended for high performance servers to authenticate plaintext passwords. The <b>LogonUser</b> function does not cache credentials for this logon type.</p>
<p><b>LOGON32_LOGON_NETWORK_CLEARTEXT</b></p>	<p>This logon type preserves the name and password in the <a href="#">authentication package</a>, which allows the server to make connections to other network servers while impersonating the client. A server can accept plaintext credentials from a client, call <b>LogonUser</b>, verify that the user can access the system across the network, and still communicate with other servers.</p>
<p><b>LOGON32_LOGON_NEW_CREDENTIALS</b></p>	<p>This logon type allows the caller to clone its current token and specify new credentials for outbound connections. The new logon session has the same local identifier but uses different credentials for other network connections.</p> <p>This logon type is supported only by the LOGON32_PROVIDER_WINNT50 logon provider.</p> <p>Note: As of January 2023, it is not possible to use the LOGON32_LOGON_NEW_CREDENTIALS logon type with a Group Managed Service Account (gMSA).</p>

<b>LOGON32_LOGON_SERVICE</b>	Indicates a service-type logon. The account provided must have the service privilege enabled.
<b>LOGON32_LOGON_UNLOCK</b>	GINAs are no longer supported.  <b>Windows Server 2003 and Windows XP:</b> This logon type is for <a href="#">GINA</a> DLLs that log on users who will be interactively using the computer. This logon type can generate a unique audit record that shows when the workstation was unlocked.

[in] dwLogonProvider

Specifies the logon provider. This parameter can be one of the following values.

Value	Meaning
<b>LOGON32_PROVIDER_DEFAULT</b>	Use the standard logon provider for the system. The default <a href="#">security provider</a> is negotiate, unless you pass <b>NULL</b> for the domain name and the user name is not in UPN format. In this case, the default provider is NTLM.
<b>LOGON32_PROVIDER_WINNT50</b>	Use the negotiate logon provider.
<b>LOGON32_PROVIDER_WINNT40</b>	Use the NTLM logon provider.

[out] phToken

A pointer to a handle variable that receives a handle to a token that represents the specified user.

You can use the returned handle in calls to the [ImpersonateLoggedOnUser](#) function.

In most cases, the returned handle is a [primary token](#) that you can use in calls to the [CreateProcessAsUser](#) function. However, if you specify the LOGON32\_LOGON\_NETWORK flag, **LogonUser** returns an [impersonation token](#) that you cannot use in **CreateProcessAsUser** unless you call [DuplicateTokenEx](#) to convert it to a primary token.

When you no longer need this handle, close it by calling the [CloseHandle](#) function.

If the function succeeds, the function returns nonzero.

If the function fails, it returns zero. To get extended error information, call [GetLastError](#).

The LOGON32\_LOGON\_NETWORK logon type is fastest, but it has the following limitations:

- The function returns an [impersonation token](#), not a primary token. You cannot use this token directly in the [CreateProcessAsUser](#) function. However, you can call the [DuplicateTokenEx](#) function to convert the token to a primary token, and then use it in **CreateProcessAsUser**.
- If you convert the token to a primary token and use it in [CreateProcessAsUser](#) to start a process, the new process cannot access other network resources, such as remote servers or printers, through the redirector. An exception is that if the network resource is not access controlled, then the new process will be able to access it.

The SE\_TCB\_NAME privilege is not required for this function unless you are logging onto a Passport account.

The account specified by *lpszUsername*, must have the necessary account rights. For example, to log on a user with the LOGON32\_LOGON\_INTERACTIVE flag, the user (or a group to which the user belongs) must have the SE\_INTERACTIVE\_LOGON\_NAME account right. For a list of the account rights that affect the various logon operations, see [Account Rights Constants](#).

A user is considered logged on if at least one token exists. If you call [CreateProcessAsUser](#) and then close the token, the system considers the user as still logged on until the process (and all child processes) have ended.

If the **LogonUser** call is successful, the system notifies network providers that the logon occurred by calling the provider's [NPLogonNotify](#) entry-point function.

You can generate a LocalService token by using the following code.

```
LogonUser(L"LocalService", L"NT AUTHORITY", NULL, LOGON32_LOGON_SERVICE, LOGON32_PROVIDER_DEFAULT, &hToken)
```

Note

The winbase.h header defines LogonUser as an alias that automatically selects the ANSI or Unicode version of this function based on the definition of the UNICODE preprocessor constant. Mixing usage of the encoding-neutral alias with code that is not encoding-neutral can lead to mismatches that result in compilation or runtime errors. For more information, see [Conventions for Function Prototypes](#).

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

[Client/Server Access Control](#)

[Client/Server Access Control Functions](#)

[CloseHandle](#)

[CreateProcessAsUser](#)

[DuplicateTokenEx](#)

[ImpersonateLoggedOnUser](#)

---

Source: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa378184\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378184(v=vs.85).aspx)