



We use optional cookies to improve your experience on our websites, such as through social media connections, and to display personalized advertising based on your online activity. If you reject optional cookies, only cookies necessary to provide you the services will be used. You may change your selection by clicking "Manage Cookies" at the bottom of the page. [Privacy Statement](#) [Third-Party Cookies](#)

Accept

Reject

Manage cookies



# CredUIPromptForCredentialsAfunction (wincred.h)

Article • 02/09/2023

Feedback

## In this article

- Syntax
- Parameters
- Return value
- Remarks
- Show 2 more

The **CredUIPromptForCredentials** function creates and displays a configurable dialog box that accepts credentials information from a user.

Applications that target Windows Vista or Windows Server 2008 should call [CredUIPromptForWindowsCredentials](#) instead of this function, for the following reasons:

- [CredUIPromptForWindowsCredentials](#) is consistent with the current Windows user interface.
- [CredUIPromptForWindowsCredentials](#) is more extensible, allowing integration of additional authentication mechanisms such as biometrics and smart cards.
- [CredUIPromptForWindowsCredentials](#) is compliant with the Common Criteria specification.

## Syntax

```
CREDUIAPI DWORD CredUIPromptForCredentialsA(  
    [in, optional] PCREDUI_INFOA pUiInfo,  
    [in]           PCSTR          pszTargetName,  
    [in]           PCtxtHandle    pContext,  
    [in, optional] DWORD          dwAuthError,  
    [in, out]      PSTR           pszUserName,  
    [in]           ULONG          ulUserNameBufferSize,  
    [in, out]      PSTR           pszPassword,  
    [in]           ULONG          ulPasswordBufferSize,  
    [in, out]      BOOL           *save,  
    [in]           DWORD          dwFlags  
);
```

## Parameters

[in, optional] pUiInfo

A pointer to a [CREDUI\\_INFO](#) structure that contains information for customizing the appearance of the dialog box.

[in] pszTargetName

A pointer to a null-terminated string that contains the name of the target for the credentials, typically a server name. For Distributed File System (DFS) connections, this string is of the form *ServerName\ShareName*. This parameter is used to identify target information when storing and retrieving credentials.

[in] pContext

This parameter is reserved for future use. It must be **NULL**.

[in, optional] dwAuthError

Specifies why the credential dialog box is needed. A caller can pass this Windows error parameter, returned by another authentication call, to allow the dialog box to accommodate certain errors. For example, if the password expired status code is passed, the dialog box could prompt the user to change the password on the account.

[in, out] pszUserName

A pointer to a null-terminated string that contains the user name for the credentials. If a nonzero-length string is passed, the *UserName* option of the dialog box is prefilled with the string. In the case of credentials other than *UserName/Password*, a marshaled format of the credential can be passed in. This string is created by calling [CredMarshalCredential](#).

This function copies the user-supplied name to this buffer, copying a maximum of *ulUserNameMaxChars* characters. This format can be converted to *UserName/Password* format by using [CredUIParseUsername](#). A marshaled format can be passed directly to a [security support provider](#) (SSP).

If the CREDUI\_FLAGS\_DO\_NOT\_PERSIST flag is not specified, the value returned in this parameter is of a form that should not be inspected, printed, or persisted other than passing it to [CredUIParseUsername](#). The subsequent results of **CredUIParseUsername** can be passed only to a

client-side authentication function such as [WNetAddConnection](#) or an SSP function.

If no domain or server is specified as part of this parameter, the value of the **pszTargetName** parameter is used as the domain to form a *DomainName\UserName* pair. On output, this parameter receives a string that contains that *DomainName\UserName* pair.

[in] ulUserNameBufferSize

The maximum number of characters that can be copied to *pszUserName* including the terminating null character.

**Note** CREDUI\_MAX\_USERNAME\_LENGTH does not include the terminating null character.

[in, out] pszPassword

A pointer to a null-terminated string that contains the password for the credentials. If a nonzero-length string is specified for *pszPassword*, the password option of the dialog box will be prefilled with the string.

This function copies the user-supplied password to this buffer, copying a maximum of *ulPasswordMaxChars* characters. If the CREDUI\_FLAGS\_DO\_NOT\_PERSIST flag is not specified, the value returned in this parameter is of a form that should not be inspected, printed, or persisted other than passing it to a client-side authentication function such as [WNetAddConnection](#) or an SSP function.

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] ulPasswordBufferSize

The maximum number of characters that can be copied to *pszPassword* including the terminating null character.

**Note** CREDUI\_MAX\_PASSWORD\_LENGTH does not include the terminating null character.

[in, out] *save*

A pointer to a **BOOL** that specifies the initial state of the **Save** check box and receives the state of the **Save** check box after the user has responded to the dialog box. If this value is not **NULL** and **CredUIPromptForCredentials** returns **NO\_ERROR**, then *pfSave* returns the state of the **Save** check box when the user chose **OK** in the dialog box.

If the **CREDUI\_FLAGS\_PERSIST** flag is specified, the **Save** check box is not displayed, but is considered to be selected.

If the **CREDUI\_FLAGS\_DO\_NOT\_PERSIST** flag is specified and **CREDUI\_FLAGS\_SHOW\_SAVE\_CHECK\_BOX** is not specified, the **Save** check box is not displayed, but is considered to be cleared.

An application that needs to use CredUI to prompt the user for credentials, but does not need the credential management services provided by the credential manager, can use *pfSave* to receive the state of the **Save** check box after the user closes the dialog box. To do this, the caller must specify **CREDUI\_FLAGS\_DO\_NOT\_PERSIST** and **CREDUI\_FLAGS\_SHOW\_SAVE\_CHECK\_BOX** in *dwFlags*. When **CREDUI\_FLAGS\_DO\_NOT\_PERSIST** and **CREDUI\_FLAGS\_SHOW\_SAVE\_CHECK\_BOX** are set, the application is responsible for examining *\*pfSave* after the function returns, and if *\*pfSave* is **TRUE**, then the application must take the appropriate action to save the user credentials within the resources of the application.

[in] *dwFlags*

A **DWORD** value that specifies special behavior for this function. This value can be a bitwise-**OR** combination of zero or more of the following values.

 Expand table


Value	Meaning
CREDUI_FLAGS_ALWAYS_SHOW_UI 0x00080	Specifies that a user interface is shown even if the credentials are returned from an existing credential manager. This is only if CREDUI_FLAGS_GENERIC_CREDENTIALS is also specified.
CREDUI_FLAGS_COMPLETE_USERNAME 0x00800	Populate the combo box with a user name.
CREDUI_FLAGS_DO_NOT_PERSIST 0x00002	Do not store credentials in the credential boxes. You can pass CREDUI_FLAGS_SHOW_USERNAME with this flag to display the username box only, and the result of the <i>pfSave</i> output parameter is ignored.
CREDUI_FLAGS_EXCLUDE_CERTIFICATES 0x00008	Populate the combo box with name/password only. Do not include certificates or smart card boxes.
CREDUI_FLAGS_EXPECT_CONFIRMATION 0x20000	Specifies that the caller expects the <a href="#">CredUIConfirmCredentials</a> function to determine whether the credentials are actually valid. This mechanism ensures that credentials that are not valid are not saved in the credential manager. Specifies that the caller expects confirmation unless CREDUI_FLAGS_DO_NOT_PERSIST is specified.

CREDUI_FLAGS_GENERIC_CREDENTIALS 0x40000	Consider the credentials user to be generic credentials.
CREDUI_FLAGS_INCORRECT_PASSWORD 0x00001	Notify the user of insufficient password by displaying the "Logo balloon" tip.
CREDUI_FLAGS_KEEP_USERNAME 0x100000	Do not allow the user to change the supplied user name.
CREDUI_FLAGS_PASSWORD_ONLY_OK 0x00200	Populate the combo box with password only. Do not allow other characters to be entered.
CREDUI_FLAGS_PERSIST 0x01000	Do not show the <b>Save</b> checkbox if a credential is saved as the default and shown and selected.
CREDUI_FLAGS_REQUEST_ADMINISTRATOR 0x00004	Populate the combo box with administrators only. <b>Windows 10 Anniversary Edition:</b> This flag will filter for known Administrator accounts.
CREDUI_FLAGS_REQUIRE_CERTIFICATE 0x00010	Populate the combo box with certificates and smart cards only. Do not allow a name to be entered.
CREDUI_FLAGS_REQUIRE_SMARTCARD 0x00100	Populate the combo box with smart cards only. Do not allow a name to be entered.
CREDUI_FLAGS_SERVER_CREDENTIAL 0x04000	This flag is meaningful only when a matching credential to populate the combo box, should authentication flag is specified, wildcard characters should not be matched. It has no effect when writing a credential. Credentials created by the system create credentials that contain wildcard characters. Any found will be matched explicitly by the user or programmatically, as has been made connection is made.

CREDUI_FLAGS_SHOW_SAVE_CHECK_BOX 0x00040	If the check box is select check box and return <b>TI</b> output parameter, othe <b>FALSE</b> . Check box uses 1 by default.
CREDUI_FLAGS_USERNAME_TARGET_CREDENTIALS 0x80000	The credential is a "runa <i>TargetName</i> parameter of the command or pro is used for prompting p
CREDUI_FLAGS_VALIDATE_USERNAME 0x00400	Check that the user nar

## Return value

The return value is a **DWORD** and can be one of the following values.

 Expand table

Value	Description
ERROR_CANCELLED	User chose <b>Cancel</b> . The <i>pszUserName</i> and <i>pszPassword</i> parameters have not changed.
ERROR_INVALID_FLAGS	This status is returned for any of the flag configurations that are not valid.
ERROR_INVALID_PARAMETER	<p>Either <i>pszTargetName</i> is <b>NULL</b>, the empty string, or longer than CREDUI_MAX_DOMAIN_LENGTH, or <i>pUiInfo</i> is not <b>NULL</b> and the <a href="#">CredUI_INFO</a> structure pointed to did not meet one of the following requirements:</p> <ul style="list-style-type: none"><li>• The <b>cbSize</b> member must be one.</li><li>• If the <b>hbmBanner</b> member is not <b>NULL</b>, it must be of type OBJ_BITMAP.</li></ul>



	<ul style="list-style-type: none"><li>• If the <b>pszMessageText</b> member is not <b>NULL</b>, it must not be greater than <b>CREDUI_MAX_MESSAGE_LENGTH</b>.</li><li>• If the <b>pszCaptionText</b> member is not <b>NULL</b>, it must not be greater than <b>CREDUI_MAX_CAPTION_LENGTH</b>.</li></ul>
<b>ERROR_NO_SUCH_LOGON_SESSION</b>	The credential manager cannot be used. Typically, this error is handled by calling <a href="#">CredUIPromptForCredentials</a> and passing in the <b>CREDUI_FLAGS_DO_NOT_PERSIST</b> flag.
<b>NO_ERROR</b>	User chose <b>OK</b> . The <i>pszUserName</i> , <i>pszPassword</i> , and <i>pfSave</i> parameters will return the values documented earlier.

## Remarks

The **CredUIPromptForCredentials** function creates and displays an application modal dialog box. After the dialog box is displayed and until it is closed by the user or application, no other window in the application can become active.

The flags **CREDUI\_FLAGS\_REQUIRE\_SMARTCARD**, **CREDUI\_FLAGS\_REQUIRE\_CERTIFICATE**, and **CREDUI\_FLAGS\_EXCLUDE\_CERTIFICATE** are mutually exclusive.

If **CREDUI\_FLAGS\_DO\_NOT\_PERSIST** is specified, either *pszTargetName* must be specified or *uiInfo->pszMessageText* and *uiInfo->pszCaption* must be specified.

The flags **CREDUI\_FLAG\_USERNAME\_TARGET\_CREDENTIALS** and **CREDUI\_FLAGS\_GENERIC\_CREDENTIALS** are mutually exclusive. If neither is specified, the credential is a domain credential.

An X509 certificate must have an [enhanced key usage](#) (EKU) value of `szOID_KP_SMARTCARD_LOGON` (1.3.6.1.4.1.311.20.2.2) to be displayed.

**Windows XP:** This EKU value is not required to display X509 certificates.

If `CREDUI_FLAGS_GENERIC_CREDENTIALS` is not specified or `CREDUI_FLAGS_COMPLETE_USERNAME` is specified, the typed name is *syntax checked*. Syntax checking applies the same rules as applied by [CredUIParseUserName](#). If the typed name is not valid, the user is prompted for a valid one. If the domain portion of the typed name is missing, one will be supplied based on the target name.

If `CREDUI_FLAGS_GENERIC_CREDENTIALS` is specified and `CREDUI_FLAGS_VALIDATE_USERNAME` is also specified, the typed name is syntax checked. If the typed name is not valid, the user is prompted for a valid one.

If `CREDUI_FLAGS_GENERIC_CREDENTIALS` is specified and neither `CREDUI_FLAGS_COMPLETE_USERNAME` nor `CREDUI_FLAGS_VALIDATE_USERNAME` is specified, the typed name is not syntax checked in any way.

If neither `CREDUI_FLAGS_PERSIST` nor `CREDUI_FLAGS_DO_NOT_PERSIST` is set, the **Save** check box is shown, and it controls whether the credential is saved.

### Calling Modes

- The caller will attempt to access the target resource, call `credui` (passing a description of the target resource and the failure status), call [CredUIParseUserName](#), access the target resource again, and then call [CredUIConfirmCredentials](#).
- The caller can prompt for credentials without accessing any resources by passing `CREDUI_FLAGS_DO_NOT_PERSIST`.
- For generic credentials, there is no authentication package. Therefore, the application needs to access the credential to do the

authentication. Prompt for credentials, not passing CREDUI\_FLAGS\_ALWAYS\_SHOW\_UI before the first authentication. The user interface will appear only if there is no credential in the credential manager. On all subsequent messages from within the application, CREDUI\_FLAGS\_ALWAYS\_SHOW\_UI will be passed because the credential in the credential manager is clearly not valid for that resource.

### Target Information

Target Information is information about the location of the resource to be accessed. For a list of all potential target names for a resource, call [CredGetTargetInfo](#). **CredGetTargetInfo** returns information that was cached by the Negotiate, NTLM, or Kerberos authentication package when one of those packages was used to authenticate to the named target. **CredGetTargetInfo** returns some or all of the following names for the target:

- NetBIOS server name of the computer
- DNS server name of the computer
- NetBIOS domain name of the domain the computer belongs to
- DNS domain name of the domain the computer belongs to
- DNS tree name of the tree the computer belongs to
- Name of the package that collected the information

Any piece of this information can be missing if the information does not apply to the target computer. For instance, a computer that is a member of a workgroup does not have a NetBIOS domain name.

Credentials are stored in the credential manager based on target name. Each target name is stored as generally as possible without colliding with credentials already stored in the credential manager. Because credentials are stored by target name, a particular user can only have one credential per target stored in the credential manager.

#### Note

The wincred.h header defines CredUIPromptForCredentials as an alias which 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 not encoding-neutral can lead to mismatches that result in compilation or runtime errors. For more information, see [Conventions for Function Prototypes](#).

## Requirements

 Expand table

Requirement	Value
Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	wincred.h
Library	Credui.lib
DLL	Credui.dll

## See also

- [CredGetTargetInfo](#)
- [CredMarshalCredential](#)
- [CredUIConfirmCredentials](#)
- [CredUIParseUserName](#)
- [CredUIPromptForWindowsCredentials](#)

CredUI\_INFO

WNetAddConnection

---


## Feedback

Was this page helpful?



 Yes

 No

[Provide product feedback](#)  | [Get help at Microsoft Q&A](#)

 English (United States)

  Your Privacy Choices


 Theme 

[Manage cookies](#)


[Previous Versions](#)

[Blog](#) 

[Contribute](#)

[Privacy](#) 

[Terms of Use](#)

[Trademarks](#) 

© Microsoft 2024