WinVerifyTrust function (wintrust.h) - Win32 apps

By GrantMeStrength

Archived: 2026-04-06 00:05:31 UTC

The **WinVerifyTrust** function performs a trust verification action on a specified object. The function passes the inquiry to a trust provider that supports the action identifier, if one exists.

For certificate verification, use the CertGetCertificateChain and CertVerifyCertificateChainPolicy functions.

```
 LONG WinVerifyTrust(
  [in] HWND   hwnd,
  [in] GUID   *pgActionID,
  [in] LPVOID pWVTData
);
```

`[in] hwnd`

Optional handle to a caller window. A trust provider can use this value to determine whether it can interact with the user. However, trust providers typically perform verification actions without input from the user.

This parameter can be one of the following values.

Value	Meaning
INVALID_HANDLE_VALUE	There is no interactive user. The trust provider performs the verification action without the user's assistance.
Zero	The trust provider can use the interactive desktop to display its user interface.
A valid window handle	A trust provider can treat any value other than INVALID_HANDLE_VALUE or zero as a valid window handle that it can use to interact with the user.

`[in] pgActionID`

A pointer to a **GUID** structure that identifies an action and the trust provider that supports that action. This value indicates the type of verification action to be performed on the structure pointed to by *pWinTrustData*.

The WinTrust service is designed to work with trust providers implemented by third parties. Each trust provider provides its own unique set of action identifiers. For information about the action identifiers supported by a trust provider, see the documentation for that trust provider.

For example, Microsoft provides a Software Publisher Trust Provider that can establish the trustworthiness of software being downloaded from the Internet or some other public network. The Software Publisher Trust Provider supports the following action identifiers. These constants are defined in Softpub.h.

Value	Meaning
DRIVER_ACTION_VERIFY	Verify the authenticity of a Windows Hardware Quality Labs (WHQL) signed driver. This is an Authenticode add-on policy provider.
HTTPSPROV_ACTION	Verify an SSL/TLS connection established by WinINet.
OFFICESIGN_ACTION_VERIFY	This Action ID is not supported. Verify the authenticity of a structured storage file by using the Microsoft Office Authenticode add-on policy provider. **Windows Server 2003 and Windows XP:** This Action ID is supported.
WINTRUST_ACTION_GENERIC_CHAIN_VERIFY	Verify certificate chains created from any object type. A callback is provided to implement the final chain policy by using the chain context for each signer and counter signer.
WINTRUST_ACTION_GENERIC_VERIFY_V2	Verify a file or object using the Authenticode policy provider.
WINTRUST_ACTION_TRUSTPROVIDER_TEST	Write the CRYPT_PROVIDER_DATA structure to a file after calling the Authenticode policy provider.

`[in] pWVTData`

A pointer that, when cast as a WINTRUST_DATA structure, contains information that the trust provider needs to process the specified action identifier. Typically, the structure includes information that identifies the object that the trust provider must evaluate.

The format of the structure depends on the action identifier. For information about the data required for a specific action identifier, see the documentation for the trust provider that supports that action.

If the trust provider verifies that the subject is trusted for the specified action, the return value is zero. No other value besides zero should be considered a successful return.

If the trust provider does not verify that the subject is trusted for the specified action, the function returns a status code from the trust provider.

Note The return value is a **LONG**, not an **HRESULT** as previously documented. Do not use **HRESULT** macros such as **SUCCEEDED** to determine whether the function succeeded. Instead, check the return value for equality to zero.

For example, a trust provider might indicate that the subject is not trusted, or is trusted but with limitations or warnings. The return value can be a trust-provider-specific value described in the documentation for an individual trust provider, or it can be one of the following error codes.

Return code	Description
TRUST_E_SUBJECT_NOT_TRUSTED	The subject failed the specified verification action. Most trust providers return a more detailed error code that describes the reason for the failure. **Note** The **TRUST_E_SUBJECT_NOT_TRUSTED** return code may be returned depending on the value of the **EnableCertPaddingCheck** registry key under **HKLM\Software\Microsoft\Cryptography\Wintrust\Config**. If **EnableCertPaddingCheck** is set to "1", then an additional check is performed to verify that the **WIN_CERTIFICATE** structure does not contain extraneous information. The check validates that there is no non-zero data beyond the PKCS #7 structure. For more information, please refer to the following security advisory: http://technet.microsoft.com/security/advisory/2915720#section1.
TRUST_E_PROVIDER_UNKNOWN	The trust provider is not recognized on this system.
TRUST_E_ACTION_UNKNOWN	The trust provider does not support the specified action.
TRUST_E_SUBJECT_FORM_UNKNOWN	The trust provider does not support the form specified for the subject.

The **WinVerifyTrust** function enables applications to invoke a trust provider to verify that a specified object satisfies the criteria of a specified verification operation. The *pgActionID* parameter identifies the verification operation, and the *pWinTrustData* parameter identifies the object whose trust is to be verified. A trust provider is a DLL registered with the operating system. A call to **WinVerifyTrust** forwards that call to the registered trust provider, if there is one, that supports that specified action identifier.

For example, the Software Publisher Trust Provider can verify that an executable image file comes from a trusted software publisher and that the code in the file has not been modified since it was signed. In this case, the

pWinTrustData parameter specifies the name of the file and the type of file, such as a Microsoft Portable Executable image file.

Each trust provider supports a specific set of actions that it can evaluate. Each action has a GUID that identifies it. A trust provider can support any number of action identifiers, but two trust providers cannot support the same action identifier.

For an example that demonstrates how to use this function to verify the signature of a portable executable (PE) file, see Example C Program: Verifying the Signature of a PE File.

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