


# Windows Data Types

05/31/2018 • 19 minutes to read • 

## In this article

### [Requirements](#)

The data types supported by Windows are used to define function return values, function and message parameters, and structure members. They define the size and meaning of these elements. For more information about the underlying C/C++ data types, see [Data Type Ranges](#).

The following table contains the following types: character, integer, Boolean, pointer, and handle. The character, integer, and Boolean types are common to most C compilers. Most of the pointer-type names begin with a prefix of P or LP. Handles refer to a resource that has been loaded into memory.

For more information about handling 64-bit integers, see [Large Integers](#).

Data type	Description
<b>APIENTRY</b>	The calling convention for system functions. This type is declared in WinDef.h as follows: <pre>#define APIENTRY WINAPI</pre>
<b>ATOM</b>	An atom. For more information, see <a href="#">About Atom Tables</a> . This type is declared in WinDef.h as follows: <pre>typedef WORD ATOM;</pre>
<b>BOOL</b>	A Boolean variable (should be <b>TRUE</b> or <b>FALSE</b> ). This type is declared in WinDef.h as follows: <pre>typedef int BOOL;</pre>
<b>BOOLEAN</b>	A Boolean variable (should be <b>TRUE</b> or <b>FALSE</b> ). This type is declared in WinNT.h as follows: <pre>typedef BYTE BOOLEAN;</pre>
<b>BYTE</b>	A byte (8 bits). This type is declared in WinDef.h as follows: <pre>typedef unsigned char BYTE;</pre>
<b>CALLBACK</b>	The calling convention for callback functions. This type is declared in WinDef.h as follows: <pre>#define CALLBACK __stdcall</pre> <p><b>CALLBACK</b>, <b>WINAPI</b>, and <b>APIENTRY</b> are all used to define functions with the <code>__stdcall</code> calling convention. Most functions in the Windows API are declared using <b>WINAPI</b>. You may wish to use <b>CALLBACK</b> for the callback functions that you implement to help identify the function as a callback function.</p>
<b>CCHAR</b>	An 8-bit Windows (ANSI) character. This type is declared in WinNT.h as follows: <pre>typedef char CCHAR;</pre>
<b>CHAR</b>	An 8-bit Windows (ANSI) character. For more information, see <a href="#">Character Sets Used By Fonts</a> . This type is declared in WinNT.h as follows: <pre>typedef char CHAR;</pre>

Data type	Description
<b>COLORREF</b>	The red, green, blue (RGB) color value (32 bits). See <a href="#">COLORREF</a> for information on this type. This type is declared in WinDef.h as follows: <pre>typedef DWORD COLORREF;</pre>
<b>CONST</b>	A variable whose value is to remain constant during execution. This type is declared in WinDef.h as follows: <pre>#define CONST const</pre>
<b>DWORD</b>	A 32-bit unsigned integer. The range is 0 through 4294967295 decimal. This type is declared in IntSafe.h as follows: <pre>typedef unsigned long DWORD;</pre>
<b>DWORDLONG</b>	A 64-bit unsigned integer. The range is 0 through 18446744073709551615 decimal. This type is declared in IntSafe.h as follows: <pre>typedef unsigned __int64 DWORDLONG;</pre>
<b>DWORD_PTR</b>	An unsigned long type for pointer precision. Use when casting a pointer to a long type to perform pointer arithmetic. (Also commonly used for general 32-bit parameters that have been extended to 64 bits in 64-bit Windows.) This type is declared in BaseTsd.h as follows: <pre>typedef ULONG_PTR DWORD_PTR;</pre>
<b>DWORD32</b>	A 32-bit unsigned integer. This type is declared in BaseTsd.h as follows: <pre>typedef unsigned int DWORD32;</pre>
<b>DWORD64</b>	A 64-bit unsigned integer. This type is declared in BaseTsd.h as follows: <pre>typedef unsigned __int64 DWORD64;</pre>
<b>FLOAT</b>	A floating-point variable. This type is declared in WinDef.h as follows: <pre>typedef float FLOAT;</pre>
<b>HACCEL</b>	A handle to an <a href="#">accelerator table</a> . This type is declared in WinDef.h as follows: <pre>typedef HANDLE HACCEL;</pre>
<b>HALF_PTR</b>	Half the size of a pointer. Use within a structure that contains a pointer and two small fields. This type is declared in BaseTsd.h as follows:  <b>C++</b> <pre>#ifdef _WIN64     typedef int HALF_PTR; #else     typedef short HALF_PTR; #endif</pre>
<b>HANDLE</b>	A handle to an object.

Data type	Description
	<p>This type is declared in WinNT.h as follows:</p> <pre data-bbox="462 220 683 241">typedef PVOID HANDLE;</pre>
<b>HBITMAP</b>	<p>A handle to a <a href="#">bitmap</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 411 703 432">typedef HANDLE HBITMAP;</pre>
<b>HBRUSH</b>	<p>A handle to a <a href="#">brush</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 602 695 623">typedef HANDLE HBRUSH;</pre>
<b>HCOLORSPACE</b>	<p>A handle to a <a href="#">color space</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 793 748 814">typedef HANDLE HCOLORSPACE;</pre>
<b>HCONV</b>	<p>A handle to a dynamic data exchange (DDE) conversation.</p> <p>This type is declared in Ddeml.h as follows:</p> <pre data-bbox="462 984 683 1005">typedef HANDLE HCONV;</pre>
<b>HCONVLIST</b>	<p>A handle to a DDE conversation list.</p> <p>This type is declared in Ddeml.h as follows:</p> <pre data-bbox="462 1176 724 1197">typedef HANDLE HCONVLIST;</pre>
<b>HCURSOR</b>	<p>A handle to a <a href="#">cursor</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 1367 695 1388">typedef HICON HCURSOR;</pre>
<b>HDC</b>	<p>A handle to a <a href="#">device context</a> (DC).</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 1558 662 1579">typedef HANDLE HDC;</pre>
<b>HDDEDATA</b>	<p>A handle to DDE data.</p> <p>This type is declared in Ddeml.h as follows:</p> <pre data-bbox="462 1749 716 1770">typedef HANDLE HDDEDATA;</pre>
<b>HDESK</b>	<p>A handle to a <a href="#">desktop</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 1940 683 1961">typedef HANDLE HDESK;</pre>

Data type	Description
<b>HDROP</b>	<p>A handle to an internal drop structure.</p> <p>This type is declared in ShellApi.h as follows:</p> <pre>typedef HANDLE HDROP;</pre>
<b>HDWP</b>	<p>A handle to a deferred window position structure.</p> <p>This type is declared in WinUser.h as follows:</p> <pre>typedef HANDLE HDWP;</pre>
<b>HENHMETAFILE</b>	<p>A handle to an <a href="#">enhanced metafile</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HENHMETAFILE;</pre>
<b>HFILE</b>	<p>A handle to a file opened by <a href="#">OpenFile</a>, not <a href="#">CreateFile</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef int HFILE;</pre>
<b>HFONT</b>	<p>A handle to a <a href="#">font</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HFONT;</pre>
<b>HGDIOBJ</b>	<p>A handle to a GDI object.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HGDIOBJ;</pre>
<b>HGLOBAL</b>	<p>A handle to a global memory block.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HGLOBAL;</pre>
<b>HHOOK</b>	<p>A handle to a <a href="#">hook</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HHOOK;</pre>
<b>HICON</b>	<p>A handle to an <a href="#">icon</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HICON;</pre>
<b>HINSTANCE</b>	<p>A handle to an instance. This is the base address of the module in memory.</p> <p><b>HMODULE</b> and <b>HINSTANCE</b> are the same today, but represented different things in 16-bit Windows.</p> <p>This type is declared in WinDef.h as follows:</p>

Data type	Description
	<pre>typedef HANDLE HINSTANCE;</pre>
<b>HKEY</b>	<p>A handle to a registry key.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HKEY;</pre>
<b>HKL</b>	<p>An input locale identifier.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HKL;</pre>
<b>HLOCAL</b>	<p>A handle to a local memory block.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HLOCAL;</pre>
<b>HMENU</b>	<p>A handle to a <a href="#">menu</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HMENU;</pre>
<b>HMETAFILE</b>	<p>A handle to a <a href="#">metafile</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HMETAFILE;</pre>
<b>HMODULE</b>	<p>A handle to a module. This is the base address of the module in memory.</p> <p><b>HMODULE</b> and <b>HINSTANCE</b> are the same in current versions of Windows, but represented different things in 16-bit Windows.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HINSTANCE HMODULE;</pre>
<b>HMONITOR</b>	<p>A handle to a display monitor.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>if(WINVER &gt;= 0x0500) typedef HANDLE HMONITOR;</pre>
<b>HPALETTE</b>	<p>A handle to a palette.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HPALETTE;</pre>
<b>HPEN</b>	<p>A handle to a <a href="#">pen</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HPEN;</pre>

Data type	Description
<b>HRESULT</b>	<p>The return codes used by COM interfaces. For more information, see <a href="#">Structure of the COM Error Codes</a>. To test an <b>HRESULT</b> value, use the <a href="#">FAILED</a> and <a href="#">SUCCEEDED</a> macros.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef LONG HRESULT;</pre>
<b>HRGN</b>	<p>A handle to a <a href="#">region</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HRGN;</pre>
<b>HRSRC</b>	<p>A handle to a resource.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HRSRC;</pre>
<b>HSZ</b>	<p>A handle to a DDE string.</p> <p>This type is declared in Ddeml.h as follows:</p> <pre>typedef HANDLE HSZ;</pre>
<b>HWINSTA</b>	<p>A handle to a <a href="#">window station</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HWINSTA;</pre>
<b>HWND</b>	<p>A handle to a <a href="#">window</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE HWND;</pre>
<b>INT</b>	<p>A 32-bit signed integer. The range is -2147483648 through 2147483647 decimal.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef int INT;</pre>
<b>INT_PTR</b>	<p>A signed integer type for pointer precision. Use when casting a pointer to an integer to perform pointer arithmetic.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#if defined(_WIN64)     typedef __int64 INT_PTR; #else     typedef int INT_PTR; #endif</pre>

 Copy

```
#if defined(_WIN64)
    typedef __int64 INT_PTR;
#else
    typedef int INT_PTR;
#endif
```

Data type	Description
<b>INT8</b>	<p>An 8-bit signed integer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef signed char INT8;</pre>
<b>INT16</b>	<p>A 16-bit signed integer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef signed short INT16;</pre>
<b>INT32</b>	<p>A 32-bit signed integer. The range is -2147483648 through 2147483647 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef signed int INT32;</pre>
<b>INT64</b>	<p>A 64-bit signed integer. The range is -9223372036854775808 through 9223372036854775807 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef signed __int64 INT64;</pre>
<b>LANGID</b>	<p>A language identifier. For more information, see <a href="#">Language Identifiers</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef WORD LANGID;</pre>
<b>LCID</b>	<p>A locale identifier. For more information, see <a href="#">Locale Identifiers</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef DWORD LCID;</pre>
<b>LCTYPE</b>	<p>A locale information type. For a list, see <a href="#">Locale Information Constants</a>.</p> <p>This type is declared in WinNls.h as follows:</p> <pre>typedef DWORD LCTYPE;</pre>
<b>LGRPID</b>	<p>A language group identifier. For a list, see <a href="#">EnumLanguageGroupLocales</a>.</p> <p>This type is declared in WinNls.h as follows:</p> <pre>typedef DWORD LGRPID;</pre>
<b>LONG</b>	<p>A 32-bit signed integer. The range is -2147483648 through 2147483647 decimal.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef long LONG;</pre>
<b>LONGLONG</b>	<p>A 64-bit signed integer. The range is -9223372036854775808 through 9223372036854775807 decimal.</p> <p>This type is declared in WinNT.h as follows:</p> <pre><b>C++</b></pre>

Data type	Description
<b>C++</b>	<pre data-bbox="513 323 831 468">#if !defined(_M_IX86)     typedef __int64 LONGLONG; #else     typedef double LONGLONG; #endif</pre>
<b>LONG_PTR</b>	<p data-bbox="462 562 1414 617">A signed long type for pointer precision. Use when casting a pointer to a long to perform pointer arithmetic.</p> <p data-bbox="462 653 894 678">This type is declared in BaseTsd.h as follows:</p>
<b>C++</b>	<pre data-bbox="513 877 831 1022">#if defined(_WIN64)     typedef __int64 LONG_PTR; #else     typedef long LONG_PTR; #endif</pre>
<b>LONG32</b>	<p data-bbox="462 1117 1252 1142">A 32-bit signed integer. The range is -2147483648 through 2147483647 decimal.</p> <p data-bbox="462 1173 894 1199">This type is declared in BaseTsd.h as follows:</p> <pre data-bbox="462 1236 732 1262">typedef signed int LONG32;</pre>
<b>LONG64</b>	<p data-bbox="462 1308 1471 1333">A 64-bit signed integer. The range is -9223372036854775808 through 9223372036854775807 decimal.</p> <p data-bbox="462 1365 894 1390">This type is declared in BaseTsd.h as follows:</p> <pre data-bbox="462 1428 699 1453">typedef __int64 LONG64;</pre>
<b>LPARAM</b>	<p data-bbox="462 1499 678 1524">A message parameter.</p> <p data-bbox="462 1556 894 1581">This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 1619 711 1644">typedef LONG_PTR LPARAM;</pre>
<b>LPBOOL</b>	<p data-bbox="462 1690 667 1715">A pointer to a <a href="#">BOOL</a>.</p> <p data-bbox="462 1747 894 1772">This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 1810 721 1835">typedef BOOL far *LPBOOL;</pre>
<b>LPBYTE</b>	<p data-bbox="462 1881 659 1906">A pointer to a <a href="#">BYTE</a>.</p> <p data-bbox="462 1938 894 1963">This type is declared in WinDef.h as follows:</p> <pre data-bbox="462 2001 721 2026">typedef BYTE far *LPBYTE;</pre>



Data type	Description
<b>LPCOLORREF</b>	<p>A pointer to a <a href="#">COLORREF</a> value.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef DWORD *LPCOLORREF;</pre>
<b>LPCSTR</b>	<p>A pointer to a constant null-terminated string of 8-bit Windows (ANSI) characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef __nullterminated CONST CHAR *LPCSTR;</pre>
<b>LPCTSTR</b>	<p>An <a href="#">LPCWSTR</a> if <b>UNICODE</b> is defined, an <a href="#">LPCSTR</a> otherwise. For more information, see <a href="#">Windows Data Types for Strings</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#ifdef UNICODE typedef LPCWSTR LPCTSTR; #else typedef LPCSTR LPCTSTR; #endif</pre>
<b>LPCVOID</b>	<p>A pointer to a constant of any type.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef CONST void *LPCVOID;</pre>
<b>LPCWSTR</b>	<p>A pointer to a constant null-terminated string of 16-bit Unicode characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CONST WCHAR *LPCWSTR;</pre>
<b>LPDWORD</b>	<p>A pointer to a <a href="#">DWORD</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef DWORD *LPDWORD;</pre>
<b>LPHANDLE</b>	<p>A pointer to a <a href="#">HANDLE</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HANDLE *LPHANDLE;</pre>
<b>LPINT</b>	<p>A pointer to an <a href="#">INT</a>.</p> <p>This type is declared in WinDef.h as follows:</p>

Data type	Description
	<pre>typedef int *LPINT;</pre>
<b>LPLONG</b>	<p>A pointer to a <a href="#">LONG</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef long *LPLONG;</pre>
<b>LPSTR</b>	<p>A pointer to a null-terminated string of 8-bit Windows (ANSI) characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CHAR *LPSTR;</pre>
<b>LPTSTR</b>	<p>An <a href="#">LPWSTR</a> if <b>UNICODE</b> is defined, an <a href="#">LPSTR</a> otherwise. For more information, see <a href="#">Windows Data Types for Strings</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#ifdef UNICODE typedef LPWSTR LPTSTR; #else typedef LPSTR LPTSTR; #endif</pre>
<b>LPVOID</b>	<p>A pointer to any type.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef void *LPVOID;</pre>
<b>LPWORD</b>	<p>A pointer to a <a href="#">WORD</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef WORD *LPWORD;</pre>
<b>LPWSTR</b>	<p>A pointer to a null-terminated string of 16-bit Unicode characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef WCHAR *LPWSTR;</pre>
<b>LRESULT</b>	<p>Signed result of message processing.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef LONG_PTR LRESULT;</pre>
<b>PBOOL</b>	<p>A pointer to a <a href="#">BOOL</a>.</p>

Data type	Description
	<p>This type is declared in WinDef.h as follows:</p> <pre>typedef BOOL *PBOOL;</pre>
<b>PBOOLEAN</b>	<p>A pointer to a <a href="#">BOOLEAN</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef BOOLEAN *PBOOLEAN;</pre>
<b>PBYTE</b>	<p>A pointer to a <a href="#">BYTE</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef BYTE *PBYTE;</pre>
<b>PCHAR</b>	<p>A pointer to a <a href="#">CHAR</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CHAR *PCHAR;</pre>
<b>PCSTR</b>	<p>A pointer to a constant null-terminated string of 8-bit Windows (ANSI) characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CONST CHAR *PCSTR;</pre>
<b>PCTSTR</b>	<p>A <a href="#">PCWSTR</a> if <b>UNICODE</b> is defined, a <a href="#">PCSTR</a> otherwise. For more information, see <a href="#">Windows Data Types for Strings</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#ifndef UNICODE typedef LPCWSTR PCTSTR; #else typedef LPCSTR PCTSTR; #endif</pre>
<b>PCWSTR</b>	<p>A pointer to a constant null-terminated string of 16-bit Unicode characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CONST WCHAR *PCWSTR;</pre>
<b>PDWORD</b>	<p>A pointer to a <a href="#">DWORD</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef DWORD *PDWORD;</pre>

Data type	Description
<b>PDWORDLONG</b>	<p>A pointer to a <a href="#">DWORDLONG</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef DWORDLONG *PDWORDLONG;</pre>
<b>PDWORD_PTR</b>	<p>A pointer to a <a href="#">DWORD_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef DWORD_PTR *PDWORD_PTR;</pre>
<b>PDWORD32</b>	<p>A pointer to a <a href="#">DWORD32</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef DWORD32 *PDWORD32;</pre>
<b>PDWORD64</b>	<p>A pointer to a <a href="#">DWORD64</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef DWORD64 *PDWORD64;</pre>
<b>PFLOAT</b>	<p>A pointer to a <a href="#">FLOAT</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef FLOAT *PFLOAT;</pre>
<b>PHALF_PTR</b>	<p>A pointer to a <a href="#">HALF_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#ifdef _WIN64 typedef HALF_PTR *PHALF_PTR; #else typedef HALF_PTR *PHALF_PTR; #endif</pre>
<b>PHANDLE</b>	<p>A pointer to a <a href="#">HANDLE</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef HANDLE *PHANDLE;</pre>
<b>PHKEY</b>	<p>A pointer to an <a href="#">HKEY</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef HKEY *PHKEY;</pre>
<b>PINT</b>	<p>A pointer to an <a href="#">INT</a>.</p>

Data type	Description
	<p>This type is declared in WinDef.h as follows:</p> <pre>typedef int *PINT;</pre>
<b>PINT_PTR</b>	<p>A pointer to an <a href="#">INT_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef INT_PTR *PINT_PTR;</pre>
<b>PINT8</b>	<p>A pointer to an <a href="#">INT8</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef INT8 *PINT8;</pre>
<b>PINT16</b>	<p>A pointer to an <a href="#">INT16</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef INT16 *PINT16;</pre>
<b>PINT32</b>	<p>A pointer to an <a href="#">INT32</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef INT32 *PINT32;</pre>
<b>PINT64</b>	<p>A pointer to an <a href="#">INT64</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef INT64 *PINT64;</pre>
<b>PLCID</b>	<p>A pointer to an <a href="#">LCID</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef PDWORD PLCID;</pre>
<b>PLONG</b>	<p>A pointer to a <a href="#">LONG</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef LONG *PLONG;</pre>
<b>PLONGLONG</b>	<p>A pointer to a <a href="#">LONGLONG</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef LONGLONG *PLONGLONG;</pre>
<b>PLONG_PTR</b>	<p>A pointer to a <a href="#">LONG_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef LONG_PTR *PLONG_PTR;</pre>

Data type	Description
<b>PLONG32</b>	<p>A pointer to a <a href="#">LONG32</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef LONG32 *PLONG32;</pre>
<b>PLONG64</b>	<p>A pointer to a <a href="#">LONG64</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef LONG64 *PLONG64;</pre>
<b>POINTER_32</b>	<p>A 32-bit pointer. On a 32-bit system, this is a native pointer. On a 64-bit system, this is a truncated 64-bit pointer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#if defined(_WIN64) #define POINTER_32 __ptr32 #else #define POINTER_32 #endif</pre>
<b>POINTER_64</b>	<p>A 64-bit pointer. On a 64-bit system, this is a native pointer. On a 32-bit system, this is a sign-extended 32-bit pointer.</p> <p>Note that it is not safe to assume the state of the high pointer bit.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#if (_MSC_VER &gt;= 1300) #define POINTER_64 __ptr64 #else #define POINTER_64 #endif</pre>
<b>POINTER_SIGNED</b>	<p>A signed pointer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>#define POINTER_SIGNED __sptr</pre>
<b>POINTER_UNSIGNED</b>	<p>An unsigned pointer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>#define POINTER_UNSIGNED __uptr</pre>

Data type	Description
<b>PSHORT</b>	<p>A pointer to a <a href="#">SHORT</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef SHORT *PSHORT;</pre>
<b>PSIZE_T</b>	<p>A pointer to a <a href="#">SIZE_T</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef SIZE_T *PSIZE_T;</pre>
<b>PSSIZE_T</b>	<p>A pointer to a <a href="#">SSIZE_T</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef SSIZE_T *PSSIZE_T;</pre>
<b>PSTR</b>	<p>A pointer to a null-terminated string of 8-bit Windows (ANSI) characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef CHAR *PSTR;</pre>
<b>PTBYTE</b>	<p>A pointer to a <a href="#">TBYTE</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef TBYTE *PTBYTE;</pre>
<b>PTCHAR</b>	<p>A pointer to a <a href="#">TCHAR</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef TCHAR *PTCHAR;</pre>
<b>PTSTR</b>	<p>A <a href="#">PWSTR</a> if <b>UNICODE</b> is defined, a <a href="#">PSTR</a> otherwise. For more information, see <a href="#">Windows Data Types for Strings</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#ifndef UNICODE typedef LPWSTR PTSTR; #else typedef LPSTR PTSTR; #endif</pre>
<b>PUCHAR</b>	<p>A pointer to a <a href="#">UCHAR</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef UCHAR *PUCHAR;</pre>

Data type	Description
<b>PUHALF_PTR</b>	<p>A pointer to a <a href="#">UHALF_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#ifdef _WIN64     typedef UHALF_PTR *PUHALF_PTR; #else     typedef UHALF_PTR *PUHALF_PTR; #endif</pre>
<b>PUINT</b>	<p>A pointer to a <a href="#">UINT</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef UINT *PUINT;</pre>
<b>PUINT_PTR</b>	<p>A pointer to a <a href="#">UINT_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef UINT_PTR *PUINT_PTR;</pre>
<b>PUINT8</b>	<p>A pointer to a <a href="#">UINT8</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef UINT8 *PUINT8;</pre>
<b>PUINT16</b>	<p>A pointer to a <a href="#">UINT16</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef UINT16 *PUINT16;</pre>
<b>PUINT32</b>	<p>A pointer to a <a href="#">UINT32</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef UINT32 *PUINT32;</pre>
<b>PUINT64</b>	<p>A pointer to a <a href="#">UINT64</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef UINT64 *PUINT64;</pre>
<b>PULONG</b>	<p>A pointer to a <a href="#">ULONG</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef ULONG *PULONG;</pre>
<b>PULONGLONG</b>	<p>A pointer to a <a href="#">ULONGLONG</a>.</p>



Data type	Description
	<p>This type is declared in WinDef.h as follows:</p> <pre>typedef ULONGLONG *PULONGLONG;</pre>
<b>PULONG_PTR</b>	<p>A pointer to a <a href="#">ULONG_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef ULONG_PTR *PULONG_PTR;</pre>
<b>PULONG32</b>	<p>A pointer to a <a href="#">ULONG32</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef ULONG32 *PULONG32;</pre>
<b>PULONG64</b>	<p>A pointer to a <a href="#">ULONG64</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef ULONG64 *PULONG64;</pre>
<b>PUSHORT</b>	<p>A pointer to a <a href="#">USHORT</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef USHORT *PUSHORT;</pre>
<b>PVOID</b>	<p>A pointer to any type.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef void *PVOID;</pre>
<b>PWCHAR</b>	<p>A pointer to a <a href="#">WCHAR</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef WCHAR *PWCHAR;</pre>
<b>PWORD</b>	<p>A pointer to a <a href="#">WORD</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef WORD *PWORD;</pre>
<b>PWSTR</b>	<p>A pointer to a null-terminated string of 16-bit Unicode characters. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef WCHAR *PWSTR;</pre>
<b>QWORD</b>	<p>A 64-bit unsigned integer.</p> <p>This type is declared as follows:</p> <pre>typedef unsigned __int64 QWORD;</pre>

Data type	Description
<b>SC_HANDLE</b>	<p>A handle to a service control manager database. For more information, see <a href="#">SCM Handles</a>.</p> <p>This type is declared in WinSvc.h as follows:</p> <pre>typedef HANDLE SC_HANDLE;</pre>
<b>SC_LOCK</b>	<p>A lock to a service control manager database. For more information, see <a href="#">SCM Handles</a>.</p> <p>This type is declared in WinSvc.h as follows:</p> <pre>typedef LPVOID SC_LOCK;</pre>
<b>SERVICE_STATUS_HANDLE</b>	<p>A handle to a service status value. For more information, see <a href="#">SCM Handles</a>.</p> <p>This type is declared in WinSvc.h as follows:</p> <pre>typedef HANDLE SERVICE_STATUS_HANDLE;</pre>
<b>SHORT</b>	<p>A 16-bit integer. The range is -32768 through 32767 decimal.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef short SHORT;</pre>
<b>SIZE_T</b>	<p>The maximum number of bytes to which a pointer can point. Use for a count that must span the full range of a pointer.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef ULONG_PTR SIZE_T;</pre>
<b>SSIZE_T</b>	<p>A signed version of <a href="#">SIZE_T</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef LONG_PTR SSIZE_T;</pre>
<b>TBYTE</b>	<p>A <a href="#">WCHAR</a> if <b>UNICODE</b> is defined, a <a href="#">CHAR</a> otherwise.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#ifdef UNICODE typedef WCHAR TBYTE; #else typedef unsigned char TBYTE; #endif</pre>
<b>TCHAR</b>	<p>A <a href="#">WCHAR</a> if <b>UNICODE</b> is defined, a <a href="#">CHAR</a> otherwise.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p>

Data type	Description
	<p><b>C++</b></p> <pre>#ifndef UNICODE typedef WCHAR TCHAR; #else typedef char TCHAR; #endif</pre>
<b>UCHAR</b>	<p>An unsigned <a href="#">CHAR</a>.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef unsigned char UCHAR;</pre>
<b>UHALF_PTR</b>	<p>An unsigned <a href="#">HALF_PTR</a>. Use within a structure that contains a pointer and two small fields.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#ifndef _WIN64 typedef unsigned int UHALF_PTR; #else typedef unsigned short UHALF_PTR; #endif</pre>
<b>UINT</b>	<p>An unsigned <a href="#">INT</a>. The range is 0 through 4294967295 decimal.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef unsigned int UINT;</pre>
<b>UINT_PTR</b>	<p>An unsigned <a href="#">INT_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre>#if defined(_WIN64) typedef unsigned __int64 UINT_PTR; #else typedef unsigned int UINT_PTR; #endif</pre>
<b>UINT8</b>	<p>An unsigned <a href="#">INT8</a>.</p>

Data type	Description
	<p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned char UINT8;</pre>
<b>UINT16</b>	<p>An unsigned <a href="#">INT16</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned short UINT16;</pre>
<b>UINT32</b>	<p>An unsigned <a href="#">INT32</a>. The range is 0 through 4294967295 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned int UINT32;</pre>
<b>UINT64</b>	<p>An unsigned <a href="#">INT64</a>. The range is 0 through 18446744073709551615 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned __int 64 UINT64;</pre>
<b>ULONG</b>	<p>An unsigned <a href="#">LONG</a>. The range is 0 through 4294967295 decimal.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef unsigned long ULONG;</pre>
<b>ULONGLONG</b>	<p>A 64-bit unsigned integer. The range is 0 through 18446744073709551615 decimal.</p> <p>This type is declared in WinNT.h as follows:</p> <p><b>C++</b></p> <pre>#if !defined(_M_IX86) typedef unsigned __int64 ULONGLONG; #else typedef double ULONGLONG; #endif</pre>
<b>ULONG_PTR</b>	<p>An unsigned <a href="#">LONG_PTR</a>.</p> <p>This type is declared in BaseTsd.h as follows:</p> <p><b>C++</b></p> <pre></pre>

Data type	Description
	<b>C++</b>
	<pre>#if defined(_WIN64) typedef unsigned __int64 ULONG_PTR; #else typedef unsigned long ULONG_PTR; #endif</pre>
<b>ULONG32</b>	<p>An unsigned <a href="#">LONG32</a>. The range is 0 through 4294967295 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned int ULONG32;</pre>
<b>ULONG64</b>	<p>An unsigned <a href="#">LONG64</a>. The range is 0 through 18446744073709551615 decimal.</p> <p>This type is declared in BaseTsd.h as follows:</p> <pre>typedef unsigned __int64 ULONG64;</pre>
<b>UNICODE_STRING</b>	<p>A Unicode string.</p> <p>This type is declared in Winternl.h as follows:</p> <b>C++</b>
	<pre>typedef struct _UNICODE_STRING {     USHORT Length;     USHORT MaximumLength;     PWSTR Buffer; } UNICODE_STRING; typedef UNICODE_STRING *PUNICODE_STRING; typedef const UNICODE_STRING *PCUNICODE_STRING;</pre>
<b>USHORT</b>	<p>An unsigned <a href="#">SHORT</a>. The range is 0 through 65535 decimal.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef unsigned short USHORT;</pre>
<b>USN</b>	<p>An update sequence number (USN).</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef LONGLONG USN;</pre>
<b>VOID</b>	<p>Any type.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>#define VOID void</pre>

Data type	Description
<b>WCHAR</b>	<p>A 16-bit Unicode character. For more information, see <a href="#">Character Sets Used By Fonts</a>.</p> <p>This type is declared in WinNT.h as follows:</p> <pre>typedef wchar_t WCHAR;</pre>
<b>WINAPI</b>	<p>The calling convention for system functions.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>#define WINAPI __stdcall</pre> <p><b>CALLBACK</b>, <b>WINAPI</b>, and <b>APIENTRY</b> are all used to define functions with the <code>__stdcall</code> calling convention. Most functions in the Windows API are declared using <b>WINAPI</b>. You may wish to use <b>CALLBACK</b> for the callback functions that you implement to help identify the function as a callback function.</p>
<b>WORD</b>	<p>A 16-bit unsigned integer. The range is 0 through 65535 decimal.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef unsigned short WORD;</pre>
<b>LPARAM</b>	<p>A message parameter.</p> <p>This type is declared in WinDef.h as follows:</p> <pre>typedef UINT_PTR LPARAM;</pre>

## Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Header	BaseTsd.h; WinDef.h; WinNT.h

### Is this page helpful?

 Yes  No