**Windows 数据类型**

Windows 支持的数据类型用于定义函数返回值、函数和消息参数以及结构成员。 它们定义这些元素的大小和含义。 有关基础 C/C++ 数据类型的详细信息，请参阅 [数据类型范围](https://learn.microsoft.com/zh-cn/cpp/cpp/data-type-ranges)。

下表包含以下类型：character、integer、Boolean、pointer 和 handle。 字符、整数和布尔类型是大多数 C 编译器通用的。 大多数指针类型名称以 P 或 LP 前缀开头。 句柄引用已加载到内存中的资源。

有关处理 64 位整数的详细信息，请参阅 [大整数](https://learn.microsoft.com/zh-cn/windows/win32/winprog/large-integers)。

|  |  |
| --- | --- |
| **APIENTRY** | 系统函数的调用约定。 此类型在 WinDef.h 中声明，如下所示： #define APIENTRY WINAPI |
| **原子** | 原子。 有关详细信息，请参阅 [关于 Atom 表](https://learn.microsoft.com/zh-cn/windows/desktop/dataxchg/about-atom-tables)。 此类型在 WinDef.h 中声明，如下所示： typedef WORD ATOM; |
| **Bool** | 布尔变量 (应为 **TRUE** 或 **FALSE**) 。 此类型在 WinDef.h 中声明，如下所示： typedef int BOOL; |
| **布尔** | 布尔变量 (应为 **TRUE** 或 **FALSE**) 。 此类型在 WinNT.h 中声明，如下所示： typedef BYTE BOOLEAN; |
| **字节** | 字节 (8 位) 。 此类型在 WinDef.h 中声明，如下所示： typedef unsigned char BYTE; |
| **回调** | 回调函数的调用约定。 此类型在 WinDef.h 中声明，如下所示： #define CALLBACK \_\_stdcall **CALLBACK**、 **WINAPI** 和 **APIENTRY** 都用于定义具有\_\_stdcall调用约定的函数。 Windows API 中的大多数函数都是使用 **WINAPI** 声明的。 你可能希望对实现的回调函数使用 **CALLBACK** ，以帮助将函数标识为回调函数。 |
| **CCHAR** | 8 位 Windows (ANSI) 字符。 此类型在 WinNT.h 中声明，如下所示： typedef char CCHAR; |
| **字符** | 8 位 Windows (ANSI) 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。 此类型在 WinNT.h 中声明，如下所示： typedef char CHAR; |
| **COLORREF** | 红色、绿色、蓝色 (RGB) 颜色值 (32 位) 。 有关此类型的信息，请参阅 [**COLORREF**](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/colorref) 。 此类型在 WinDef.h 中声明，如下所示： typedef DWORD COLORREF; |
| **常量** | 一个变量，其值在执行期间保持不变。 此类型在 WinDef.h 中声明，如下所示： #define CONST const |
| **Dword** | 32 位无符号整数。 范围为 0 到十进制4294967295。 此类型在 IntSafe.h 中声明，如下所示： typedef unsigned long DWORD; |
| **DWORDLONG** | 64 位无符号整数。 范围为 0 到十进制18446744073709551615。 此类型在 IntSafe.h 中声明，如下所示： typedef unsigned \_\_int64 DWORDLONG; |
| **DWORD\_PTR** | 指针精度的无符号长类型。 将指针强制转换为长类型以执行指针算术时使用。 (还通常用于在 64 位 Windows 中扩展到 64 位的常规 32 位参数。) 此类型在 BaseTsd.h 中声明，如下所示： typedef ULONG\_PTR DWORD\_PTR; |
| **DWORD32** | 32 位无符号整数。 此类型在 BaseTsd.h 中声明，如下所示： typedef unsigned int DWORD32; |
| **DWORD64** | 64 位无符号整数。 此类型在 BaseTsd.h 中声明，如下所示： typedef unsigned \_\_int64 DWORD64; |
| **浮动** | 浮点变量。 此类型在 WinDef.h 中声明，如下所示： typedef float FLOAT; |
| **HACCEL** | [快捷键表](https://learn.microsoft.com/zh-cn/windows/desktop/menurc/keyboard-accelerators)的句柄。 此类型在 WinDef.h 中声明，如下所示： typedef HANDLE HACCEL; |
| **HALF\_PTR** | 指针大小的一半。 在包含指针和两个小字段的结构中使用 。 此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #ifdef \_WIN64  typedef int HALF\_PTR;  #else  typedef short HALF\_PTR;  #endif | |
| **处理** | 对象的句柄。  此类型在 WinNT.h 中声明，如下所示：  typedef PVOID HANDLE; |
| **HBITMAP** | [位图](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/bitmaps)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HBITMAP; |
| **HBRUSH** | [画笔](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/brushes)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HBRUSH; |
| **HCOLORSPACE** | [颜色空间](https://learn.microsoft.com/zh-cn/previous-versions/dd316799(v=vs.85))的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HCOLORSPACE; |
| **HCONV** | DDE) 会话 (动态数据交换的句柄。  此类型在 Ddeml.h 中声明，如下所示：  typedef HANDLE HCONV; |
| **HCONVLIST** | DDE 会话列表的句柄。  此类型在 Ddeml.h 中声明，如下所示：  typedef HANDLE HCONVLIST; |
| **HCURSOR** | [游标](https://learn.microsoft.com/zh-cn/windows/desktop/menurc/cursors)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HICON HCURSOR; |
| **HDC** | [设备上下文](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/device-context-types)的句柄 (DC) 。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HDC; |
| **HDDEDATA** | DDE 数据的句柄。  此类型在 Ddeml.h 中声明，如下所示：  typedef HANDLE HDDEDATA; |
| **HDESK** | [桌面](https://learn.microsoft.com/zh-cn/windows/desktop/winstation/desktops)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HDESK; |
| **HDROP** | 内部放置结构的句柄。  此类型在 ShellApi.h 中声明，如下所示：  typedef HANDLE HDROP; |
| **HDWP** | 延迟窗口位置结构的句柄。  此类型在 WinUser.h 中声明，如下所示：  typedef HANDLE HDWP; |
| **HENHMETAFILE** | [增强型图元文件的句柄](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/metafiles)。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HENHMETAFILE; |
| **HFILE** | 由 [**OpenFile**](https://learn.microsoft.com/zh-cn/windows/desktop/api/winbase/nf-winbase-openfile) 打开的文件的句柄，而不是 [**CreateFile**](https://learn.microsoft.com/zh-cn/windows/desktop/api/fileapi/nf-fileapi-createfilea)。  此类型在 WinDef.h 中声明，如下所示：  typedef int HFILE; |
| **HFONT** | [字体](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/about-fonts)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HFONT; |
| **HGDIOBJ** | GDI 对象的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HGDIOBJ; |
| **HGLOBAL** | 全局内存块的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HGLOBAL; |
| **HHOOK** | [挂钩](https://learn.microsoft.com/zh-cn/windows/desktop/winmsg/hooks)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HHOOK; |
| **HICON** | [图标](https://learn.microsoft.com/zh-cn/windows/desktop/menurc/icons)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HICON; |
| **HINSTANCE** | 实例的句柄。 这是内存中模块的基址。  **HMODULE** 和 **HINSTANCE** 目前相同，但在 16 位 Windows 中表示不同的内容。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HINSTANCE; |
| **HKEY** | 注册表项的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HKEY; |
| **HKL** | 输入区域设置标识符。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HKL; |
| **HLOCAL** | 本地内存块的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HLOCAL; |
| **HMENU** | [菜单](https://learn.microsoft.com/zh-cn/windows/desktop/menurc/menus)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HMENU; |
| **HMETAFILE** | [图元文件](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/metafiles)句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HMETAFILE; |
| **HMODULE** | 模块的句柄。 这是内存中模块的基址。  **HMODULE** 和 **HINSTANCE** 在当前版本的 Windows 中相同，但在 16 位 Windows 中表示不同的内容。  此类型在 WinDef.h 中声明，如下所示：  typedef HINSTANCE HMODULE; |
| **HMONITOR** | 显示监视器的句柄。  此类型在 WinDef.h 中声明，如下所示：  if(WINVER >= 0x0500) typedef HANDLE HMONITOR; |
| **HPALETTE** | 调色板的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HPALETTE; |
| **HPEN** | [笔](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/pens)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HPEN; |
| **HRESULT** | COM 接口使用的返回代码。 有关详细信息，请参阅 [COM 错误代码的结构](https://learn.microsoft.com/zh-cn/windows/desktop/com/structure-of-com-error-codes)。 若要测试 **HRESULT** 值，请使用 [**FAILED**](https://learn.microsoft.com/zh-cn/windows/desktop/api/winerror/nf-winerror-failed) 和 [**SUCCEEDED**](https://learn.microsoft.com/zh-cn/windows/desktop/api/winerror/nf-winerror-succeeded) 宏。  此类型在 WinNT.h 中声明，如下所示：  typedef LONG HRESULT; |
| **HRGN** | [区域的](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/regions)句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HRGN; |
| **HRSRC** | 资源的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HRSRC; |
| **HSZ** | DDE 字符串的句柄。  此类型在 Ddeml.h 中声明，如下所示：  typedef HANDLE HSZ; |
| **HWINSTA** | [窗口工作站](https://learn.microsoft.com/zh-cn/windows/desktop/winstation/window-stations)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE WINSTA; |
| **HWND** | [窗口](https://learn.microsoft.com/zh-cn/windows/desktop/winmsg/windows)的句柄。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE HWND; |
| **Int** | 32 位带符号整数。 范围是 -2147483648 到 2147483647 decimal。  此类型在 WinDef.h 中声明，如下所示：  typedef int INT; |
| **INT\_PTR** | 指针精度的有符号整数类型。 在将指针强制转换为整数时使用 以执行指针算术。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if defined(\_WIN64)  typedef \_\_int64 INT\_PTR;  #else  typedef int INT\_PTR;  #endif | |
| **INT8** | 8 位有符号整数。  此类型在 BaseTsd.h 中声明，如下所示：  typedef signed char INT8; |
| **INT16** | 16 位带符号整数。  此类型在 BaseTsd.h 中声明，如下所示：  typedef signed short INT16; |
| **INT32** | 32 位带符号整数。 范围是 -2147483648 到 2147483647 decimal。  此类型在 BaseTsd.h 中声明，如下所示：  typedef signed int INT32; |
| **INT64** | 64 位带符号整数。 范围是 -9223372036854775808 到 9223372036854775807 decimal。  此类型在 BaseTsd.h 中声明，如下所示：  typedef signed \_\_int64 INT64; |
| **LANGID** | 语言标识符。 有关详细信息，请参阅 [语言标识符](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/language-identifiers)。  此类型在 WinNT.h 中声明，如下所示：  typedef WORD LANGID; |
| **Lcid** | 区域设置标识符。 有关详细信息，请参阅 [区域设置标识符](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/locale-identifiers)。  此类型在 WinNT.h 中声明，如下所示：  typedef DWORD LCID; |
| **LCTYPE** | 区域设置信息类型。 有关列表，请参阅 [区域设置信息常量](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/locale-information-constants)。  此类型在 WinNls.h 中声明，如下所示：  typedef DWORD LCTYPE; |
| **LGRPID** | 语言组标识符。 有关列表，请参阅 [**EnumLanguageGroupLocales**](https://learn.microsoft.com/zh-cn/windows/desktop/api/winnls/nf-winnls-enumlanguagegrouplocalesa)。  此类型在 WinNls.h 中声明，如下所示：  typedef DWORD LGRPID; |
| **长** | 32 位带符号整数。 范围为 -2147483648 到 2147483647 十进制。  此类型在 WinNT.h 中声明，如下所示：  typedef long LONG; |
| **LONGLONG** | 64 位带符号整数。 范围为 -9223372036854775808 到 9223372036854775807 十进制。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #if !defined(\_M\_IX86)  typedef \_\_int64 LONGLONG;  #else  typedef double LONGLONG;  #endif | |
| **LONG\_PTR** | 指针精度的有符号长类型。 将指针强制转换为 long 以执行指针算术时使用。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if defined(\_WIN64)  typedef \_\_int64 LONG\_PTR;  #else  typedef long LONG\_PTR;  #endif | |
| **LONG32** | 32 位带符号整数。 范围为 -2147483648 到 2147483647 十进制。  此类型在 BaseTsd.h 中声明，如下所示：  typedef signed int LONG32; |
| **LONG64** | 64 位带符号整数。 范围为 -9223372036854775808 到 9223372036854775807 十进制。  此类型在 BaseTsd.h 中声明，如下所示：  typedef \_\_int64 LONG64; |
| **LPARAM** | 消息参数。  此类型在 WinDef.h 中声明，如下所示：  typedef LONG\_PTR LPARAM; |
| **LPBOOL** | 指向 [**BOOL**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#bool) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef BOOL far \*LPBOOL; |
| **LPBYTE** | 指向 [**BYTE**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#byte) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef BYTE far \*LPBYTE; |
| **LPCOLORREF** | 指向 [**COLORREF**](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/colorref) 值的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef DWORD \*LPCOLORREF; |
| **LPCSTR** | 指向以 null 结尾的常量字符串的指针，该字符串由 8 位 Windows (ANSI) 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef \_\_nullterminated CONST CHAR \*LPCSTR; |
| **LPCTSTR** | 如果定义了 **UNICODE**，则为 [**LPCWSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#lpcwstr)，否则为 [**LPCSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#lpcstr)。 有关详细信息，请参阅 [字符串的 Windows 数据类型](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/windows-data-types-for-strings)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef LPCWSTR LPCTSTR;  #else  typedef LPCSTR LPCTSTR;  #endif | |
| **LPCVOID** | 指向任何类型的常量的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef CONST void \*LPCVOID; |
| **LPCWSTR** | 指向 16 位 Unicode 字符的以 null 结尾的常量字符串的指针。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef CONST WCHAR \*LPCWSTR; |
| **LPDWORD** | 指向 [**DWORD**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dword) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef DWORD \*LPDWORD; |
| **LPHANDLE** | 指向 [**HANDLE**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#handle) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef HANDLE \*LPHANDLE; |
| **LPINT** | 指向 [**INT 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int)指针。  此类型在 WinDef.h 中声明，如下所示：  typedef int \*LPINT; |
| **LPLONG** | 指向 [**LONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef long \*LPLONG; |
| **LPSTR** | 指向以 null 结尾的字符串的指针，该字符串由 8 位 Windows (ANSI) 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef CHAR \*LPSTR; |
| **LPTSTR** | 如果定义了 **UNICODE**，则为 [**LPWSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#lpwstr);否则为 [**LPSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#lpstr)。 有关详细信息，请参阅 [字符串的 Windows 数据类型](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/windows-data-types-for-strings)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef LPWSTR LPTSTR;  #else  typedef LPSTR LPTSTR;  #endif | |
| **LPVOID** | 指向任意类型的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef void \*LPVOID; |
| **LPWORD** | 指向 [**WORD**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#word) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef WORD \*LPWORD; |
| **LPWSTR** | 指向以 null 结尾的 16 位 Unicode 字符字符串的指针。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef WCHAR \*LPWSTR; |
| **LRESULT** | 消息处理的签名结果。  此类型在 WinDef.h 中声明，如下所示：  typedef LONG\_PTR LRESULT; |
| **PBOOL** | 指向 [**BOOL**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#bool) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef BOOL \*PBOOL; |
| **PBOOLEAN** | 指向 [**BOOLEAN 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#boolean)指针。  此类型在 WinNT.h 中声明，如下所示：  typedef BOOLEAN \*PBOOLEAN; |
| **PBYTE** | 指向 [**BYTE**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#byte) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef BYTE \*PBYTE; |
| **PCHAR** | 指向 [**CHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#char) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef CHAR \*PCHAR; |
| **PCSTR** | 指向以 null 结尾的常量字符串的指针，该字符串由 8 位 Windows (ANSI) 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef CONST CHAR \*PCSTR; |
| **PCTSTR** | 如果定义了 **UNICODE**，则为 [**PCWSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#pcwstr)，否则为 [**PCSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#pcstr)。 有关详细信息，请参阅 [字符串的 Windows 数据类型](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/windows-data-types-for-strings)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef LPCWSTR PCTSTR;  #else  typedef LPCSTR PCTSTR;  #endif | |
| **PCWSTR** | 指向 16 位 Unicode 字符的以 null 结尾的常量字符串的指针。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef CONST WCHAR \*PCWSTR; |
| **PDWORD** | 指向 [**DWORD**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dword) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef DWORD \*PDWORD; |
| **PDWORDLONG** | 指向 [**DWORDLONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dwordlong) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef DWORDLONG \*PDWORDLONG; |
| **PDWORD\_PTR** | 指向 [**DWORD\_PTR的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dword_ptr)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef DWORD\_PTR \*PDWORD\_PTR; |
| **PDWORD32** | 指向 [**DWORD32 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dword32)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef DWORD32 \*PDWORD32; |
| **PDWORD64** | 指向 [**DWORD64 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#dword64)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef DWORD64 \*PDWORD64; |
| **PFLOAT** | 指向 [**FLOAT**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#float) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef FLOAT \*PFLOAT; |
| **PHALF\_PTR** | 指向 [**HALF\_PTR的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#half_ptr)指针。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #ifdef \_WIN64  typedef HALF\_PTR \*PHALF\_PTR;  #else  typedef HALF\_PTR \*PHALF\_PTR;  #endif | |
| **PHANDLE** | 指向 [**HANDLE**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#handle) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef HANDLE \*PHANDLE; |
| **PHKEY** | 指向 [**HKEY**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#hkey) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef HKEY \*PHKEY; |
| **品脱** | 指向 [**INT 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int)指针。  此类型在 WinDef.h 中声明，如下所示：  typedef int \*PINT; |
| **PINT\_PTR** | 指向 [**INT\_PTR的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int_ptr)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef INT\_PTR \*PINT\_PTR; |
| **PINT8** | 指向 [**INT8**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int8) 的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef INT8 \*PINT8; |
| **PINT16** | 指向 [**INT16**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int16) 的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef INT16 \*PINT16; |
| **PINT32** | 指向 [**INT32**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int32) 的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef INT32 \*PINT32; |
| **PINT64** | 指向 [**INT64 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int64)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef INT64 \*PINT64; |
| **PLCID** | 指向 [**LCID**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#lcid) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef PDWORD PLCID; |
| **PLONG** | 指向 [**LONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef LONG \*PLONG; |
| **PLONGLONG** | 指向 [**LONGLONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#longlong) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef LONGLONG \*PLONGLONG; |
| **PLONG\_PTR** | 指向 [**LONG\_PTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long_ptr)的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef LONG\_PTR \*PLONG\_PTR; |
| **PLONG32** | 指向 [**LONG32**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long32) 的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef LONG32 \*PLONG32; |
| **PLONG64** | 指向 [**LONG64 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long64)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef LONG64 \*PLONG64; |
| **POINTER\_32** | 32 位指针。 在 32 位系统上，这是一个本机指针。 在 64 位系统上，这是截断的 64 位指针。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if defined(\_WIN64)  #define POINTER\_32 \_\_ptr32  #else  #define POINTER\_32  #endif | |
| **POINTER\_64** | 64 位指针。 在 64 位系统上，这是一个本机指针。 在 32 位系统上，这是一个符号扩展的 32 位指针。  请注意，假定高指针位的状态是不安全的。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if (\_MSC\_VER >= 1300)  #define POINTER\_64 \_\_ptr64  #else  #define POINTER\_64  #endif | |
| **POINTER\_SIGNED** | 有符号的指针。  此类型在 BaseTsd.h 中声明，如下所示：  #define POINTER\_SIGNED \_\_sptr |
| **POINTER\_UNSIGNED** | 无符号指针。  此类型在 BaseTsd.h 中声明，如下所示：  #define POINTER\_UNSIGNED \_\_uptr |
| **PSHORT** | 指向 [**SHORT**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#short) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef SHORT \*PSHORT; |
| **PSIZE\_T** | 指向 [**SIZE\_T的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#size_t)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef SIZE\_T \*PSIZE\_T; |
| **PSSIZE\_T** | 指向 [**SSIZE\_T的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ssize_t)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef SSIZE\_T \*PSSIZE\_T; |
| **PSTR** | 指向以 null 结尾的 8 位 Windows 字符串的指针， (ANSI) 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef CHAR \*PSTR; |
| **PTBYTE** | 指向 [**TBYTE 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#tbyte)指针。  此类型在 WinNT.h 中声明，如下所示：  typedef TBYTE \*PTBYTE; |
| **PTCHAR** | 指向 [**TCHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#tchar) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef TCHAR \*PTCHAR; |
| **PTSTR** | 如果定义了 **UNICODE**，则为 [**PWSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#pwstr)，否则为 [**PSTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#pstr)。 有关详细信息，请参阅 [字符串的 Windows 数据类型](https://learn.microsoft.com/zh-cn/windows/desktop/Intl/windows-data-types-for-strings)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef LPWSTR PTSTR;  #else typedef LPSTR PTSTR;  #endif | |
| **PUCHAR** | 指向 [**UCHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uchar) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef UCHAR \*PUCHAR; |
| **PUHALF\_PTR** | 指向 [**UHALF\_PTR的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uhalf_ptr)指针。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #ifdef \_WIN64  typedef UHALF\_PTR \*PUHALF\_PTR;  #else  typedef UHALF\_PTR \*PUHALF\_PTR;  #endif | |
| **PUINT** | 指向 [**UINT**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef UINT \*PUINT; |
| **PUINT\_PTR** | 指向 [**UINT\_PTR的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint_ptr)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef UINT\_PTR \*PUINT\_PTR; |
| **PUINT8** | 指向 [**UINT8 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint8)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef UINT8 \*PUINT8; |
| **PUINT16** | 指向 [**UINT16 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint16)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef UINT16 \*PUINT16; |
| **PUINT32** | 指向 [**UINT32 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint32)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef UINT32 \*PUINT32; |
| **PUINT64** | 指向 [**UINT64 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#uint64)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef UINT64 \*PUINT64; |
| **PULONG** | 指向 [**ULONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ulong) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef ULONG \*PULONG; |
| **PULONGLONG** | 指向 [**ULONGLONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ulonglong) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef ULONGLONG \*PULONGLONG; |
| **PULONG\_PTR** | 指向 [**ULONG\_PTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ulong_ptr)的指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef ULONG\_PTR \*PULONG\_PTR; |
| **PULONG32** | 指向 [**ULONG32 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ulong32)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef ULONG32 \*PULONG32; |
| **PULONG64** | 指向 [**ULONG64 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ulong64)指针。  此类型在 BaseTsd.h 中声明，如下所示：  typedef ULONG64 \*PULONG64; |
| **PUSHORT** | 指向 [**USHORT 的**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#ushort)指针。  此类型在 WinDef.h 中声明，如下所示：  typedef USHORT \*PUSHORT; |
| **PVOID** | 指向任意类型的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef void \*PVOID; |
| **PWCHAR** | 指向 [**WCHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#wchar) 的指针。  此类型在 WinNT.h 中声明，如下所示：  typedef WCHAR \*PWCHAR; |
| **PWORD** | 指向 [**WORD**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#word) 的指针。  此类型在 WinDef.h 中声明，如下所示：  typedef WORD \*PWORD; |
| **PWSTR** | 指向以 null 结尾的 16 位 Unicode 字符字符串的指针。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef WCHAR \*PWSTR; |
| **QWORD** | 64 位无符号整数。  此类型声明如下：  typedef unsigned \_\_int64 QWORD; |
| **SC\_HANDLE** | 服务控制管理器数据库的句柄。 有关详细信息，请参阅 [SCM 句柄](https://learn.microsoft.com/zh-cn/windows/desktop/Services/scm-handles)。  此类型在 WinSvc.h 中声明，如下所示：  typedef HANDLE SC\_HANDLE; |
| **SC\_LOCK** | 服务控制管理器数据库的锁。 有关详细信息，请参阅 [SCM 句柄](https://learn.microsoft.com/zh-cn/windows/desktop/Services/scm-handles)。  此类型在 WinSvc.h 中声明，如下所示：  typedef LPVOID SC\_LOCK; |
| **SERVICE\_STATUS\_HANDLE** | 服务状态值的句柄。 有关详细信息，请参阅 [SCM 句柄](https://learn.microsoft.com/zh-cn/windows/desktop/Services/scm-handles)。  此类型在 WinSvc.h 中声明，如下所示：  typedef HANDLE SERVICE\_STATUS\_HANDLE; |
| **短** | 16 位整数。 范围是 -32768 到 32767 十进制。  此类型在 WinNT.h 中声明，如下所示：  typedef short SHORT; |
| **SIZE\_T** | 指针可以指向的最大字节数。 对于必须跨越指针全范围的计数，请使用 。  此类型在 BaseTsd.h 中声明，如下所示：  typedef ULONG\_PTR SIZE\_T; |
| **SSIZE\_T** | [**SIZE\_T**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#size_t)的签名版本。  此类型在 BaseTsd.h 中声明，如下所示：  typedef LONG\_PTR SSIZE\_T; |
| **TBYTE** | 如果定义了 **UNICODE**，则为 [**WCHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#wchar)，否则为 [**CHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#char)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef WCHAR TBYTE;  #else  typedef unsigned char TBYTE;  #endif | |
| **TCHAR** | 如果定义了 **UNICODE**，则为 [**WCHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#wchar)，否则为 [**CHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#char)。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #ifdef UNICODE  typedef WCHAR TCHAR;  #else  typedef char TCHAR;  #endif | |
| **UCHAR** | 未签名 [**的 CHAR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#char)。  此类型在 WinDef.h 中声明，如下所示：  typedef unsigned char UCHAR; |
| **UHALF\_PTR** | 无符号 [**HALF\_PTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#half_ptr)。 在包含指针和两个小字段的结构中使用。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #ifdef \_WIN64  typedef unsigned int UHALF\_PTR;  #else  typedef unsigned short UHALF\_PTR;  #endif | |
| **UINT** | 未签名 [**的 INT**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int)。 范围为 0 到 4294967295 decimal。  此类型在 WinDef.h 中声明，如下所示：  typedef unsigned int UINT; |
| **UINT\_PTR** | 无符号 [**INT\_PTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int_ptr)。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if defined(\_WIN64)  typedef unsigned \_\_int64 UINT\_PTR;  #else  typedef unsigned int UINT\_PTR;  #endif | |
| **UINT8** | 未签名 [**的 INT8**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int8)。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned char UINT8; |
| **UINT16** | 未签名 [**的 INT16**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int16)。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned short UINT16; |
| **UINT32** | 未签名 [**的 INT32**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int32)。 范围为 0 到 4294967295 decimal。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned int UINT32; |
| **UINT64** | 未签名 [**的 INT64**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#int64)。 范围是 0 到 18446744073709551615 十进制。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned \_\_int64 UINT64; |
| **ULONG** | 无符号 [**LONG**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long)。 范围为 0 到 4294967295 decimal。  此类型在 WinDef.h 中声明，如下所示：  typedef unsigned long ULONG; |
| **ULONGLONG** | 64 位无符号整数。 范围是 0 到 18446744073709551615 十进制。  此类型在 WinNT.h 中声明，如下所示：   | **C++** | | --- | | #if !defined(\_M\_IX86)  typedef unsigned \_\_int64 ULONGLONG;  #else  typedef double ULONGLONG;  #endif | |
| **ULONG\_PTR** | 无符号 [**LONG\_PTR**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long_ptr)。  此类型在 BaseTsd.h 中声明，如下所示：   | **C++** | | --- | | #if defined(\_WIN64)  typedef unsigned \_\_int64 ULONG\_PTR;  #else  typedef unsigned long ULONG\_PTR;  #endif | |
| **ULONG32** | 无符号 [**LONG32**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long32)。 范围为 0 到 4294967295 decimal。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned int ULONG32; |
| **ULONG64** | 无符号 [**LONG64**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#long64)。 范围是 0 到 18446744073709551615 十进制。  此类型在 BaseTsd.h 中声明，如下所示：  typedef unsigned \_\_int64 ULONG64; |
| **UNICODE\_STRING** | 一个 Unicode 字符串。  此类型在 Winternl.h 中声明，如下所示：   | **C++** | | --- | | typedef struct \_UNICODE\_STRING {  USHORT Length;  USHORT MaximumLength;  PWSTR Buffer;  } UNICODE\_STRING;  typedef UNICODE\_STRING \*PUNICODE\_STRING;  typedef const UNICODE\_STRING \*PCUNICODE\_STRING; | |
| **USHORT** | 无符号 [**SHORT**](read://https_learn.microsoft.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fzh-cn%2Fwindows%2Fwin32%2Fwinprog%2Fwindows-data-types#short)。 范围是 0 到 65535 十进制。  此类型在 WinDef.h 中声明，如下所示：  typedef unsigned short USHORT; |
| **Usn** | 更新序列号 (USN) 。  此类型在 WinNT.h 中声明，如下所示：  typedef LONGLONG USN; |
| **无效** | 任何类型。  此类型在 WinNT.h 中声明，如下所示：  #define VOID void |
| **WCHAR** | 16 位 Unicode 字符。 有关详细信息，请参阅 [字体使用的字符集](https://learn.microsoft.com/zh-cn/windows/desktop/gdi/character-sets-used-by-fonts)。  此类型在 WinNT.h 中声明，如下所示：  typedef wchar\_t WCHAR; |
| **WINAPI** | 系统函数的调用约定。  此类型在 WinDef.h 中声明，如下所示：  #define WINAPI \_\_stdcall  **CALLBACK**、 **WINAPI** 和 **APIENTRY** 都用于定义具有\_\_stdcall调用约定的函数。 Windows API 中的大多数函数都是使用 **WINAPI** 声明的。 你可能希望对实现的回调函数使用 **CALLBACK** 来帮助将函数标识为回调函数。 |
| **词** | 16 位无符号整数。 范围是 0 到 65535 十进制。  此类型在 WinDef.h 中声明，如下所示：  typedef unsigned short WORD; |
| **WPARAM** | 消息参数。  此类型在 WinDef.h 中声明，如下所示：  typedef UINT\_PTR WPARAM; |

显示详细信息

# 其中，常用的数据类型

# windows编程常用数据类型

## Windows数据类型命名规律

        基本数据类型包括：BYTE、CHAR、WORD、SHORT、INT等。

        指针类型的命令方式一般是在其指向的数据类型前加“LP”或“P”，比如指向DWORD的指针类型为“LPDWORD”和“PDWORD”

　　各种句柄类型的命名方式一般都是在对象名前加“H”，比如位图（BITMAP）对应的句柄类型为“HBITMAP”。

　　无符号类型一般是以“U”开头，比如“INT”是符号类型，“UINT”是无符号类型。

　　根据这些命名规律以及自己的经验看到一些没见过的数据类型也就能知道它的代表的意思。

## Windows数据类型与标准C数据类型的关系

        C数据类型经过类型重定义得到的。如DWORD实质上就是 unsigned long 数据类型，32位无符号整型。而经常要用到的HANDLE类型实质上是无类型指针void,HANDLE定义为：

typedef PVOID HANDLE;

        HANDLE实际上就是一个PVOID，那PVOID又是什么呢？

typedef void\* PVOID;

        PVOID就是指向void的指针。

## Windows数据类型:

WORD：16位无符号整形数据

DWORD：32位无符号整型数据（DWORD32）

DWORD64：64位无符号整型数据

INT：32位有符号整型数据类型

INT\_PTR：指向INT数据类型的指针类型

INT32：32位符号整型

int64：64位符号整型

UINT：无符号INT

PVOID:普通指针

LONG：32位符号整型（LONG32）

ULONG：无符号LONG

LONGLONG：64位符号整型（LONG64）

SHORT：无符号短整型（16位）

LPARAM：消息的L参数

WPARAM：消息的W参数

HANDLE：对象的句柄，最基本的句柄类型

HICON：图标的句柄

HINSTANCE：程序实例的句柄

HKEY：注册表键的句柄

HMODULE：模块的句柄

HWND：窗口的句柄

LPSTR：字符指针，也就是字符串变量

LPCSTR：字符串常量

LPCTSTR：根据环境配置，如果定义了UNICODE宏，则是LPCWSTR类型，否则则为LPCSTR类型

LPCWSTR：UNICODE字符串常量

LPDWORD：指向DWORD类型数据的指针

CHAR：8比特字节

TCHAR：如果定义了UNICODE，则为WCHAR，否则为CHAR

UCHAR：无符号CHAR

WCHAR：16位Unicode字符

BOOL：布尔型变量

BYTE：字节类型（8位）

CONST：常量

FLOAT：浮点数据类型

SIZE\_T：表示内存大小，以字节为单位，其最大值是CPU最大寻址范围

VOID：无类型，相当于标准C语言中的void

## 注意：在windows程序里面也是可以正常使用c语言的数据类型的，成员然后问题，只是c语言的事件类型太少，微软需要创建很多事件类型，widows的数据类型都是全部大写的。

# 我们可以在控制台程序里面使用windows的数据类型，

|  |
| --- |
|  |

## 1.我们创建一个控制台项目，然后添加下面的代码

|  |
| --- |
|  |

### 编译运行程序，效果如下

|  |
| --- |
|  |

## 2.在这里调用一下CopyMemory函数来为字符串赋值

|  |
| --- |
|  |

### 效果如下

|  |
| --- |
|  |

# windows的数据类型是在c语言数据类型的基础上进行重新定义的，功能更加强大，实现很多c语言数据类型不能实现的功能