

Forum:[Hack Forums](#)

Thread: [Introduction To The Windows API](#)

ByteDoktor - March 15, 2025, 2:03 pm

Introduction

The Windows API helps programs talk to the Windows system. It lets apps show things on the screen, change files, or ask for info from the registry. And if you want to know more about how it works, Microsoft has all the details [here](#)

Windows Data Types

The Windows API uses many different kinds of data types that might not be as familiar as basic ones like "int" or "float." Here's an overview, and you can read more about them all on this page: [link].

Here are some examples:

DWORD - A 32-bit unsigned integer, on both 32-bit and 64-bit systems, used to represent values from 0 up to $(2^{32} - 1)$.

Code `DWORD dwVariable = 42;`

SIZE_T: Used to represent the size of an object. It's a 32-bit unsigned integer on 32-bit systems representing values from 0 up to $(2^{32} - 1)$. On the other hand, it's a 64-bit unsigned integer on 64-bit systems representing values from 0 up to $(2^{64} - 1)$.

Code `SIZE_T sVariable = sizeof(int);`

VOID: This means "no data type"—it's used when we don't want to imply that we're dealing with a particular kind of data.

Code `void* pVariable = NULL; // This is the same as PVOID`

HANDLE: A value that specifies a particular object that the operating system is managing (e.g. file, process, thread).

Code `HANDLE hFile = CreateFile(...);`

HMODULE: This is how you refer to a module (like a DLL) by its memory address.

Code `HMODULE hModule = GetModuleHandle(...);`

LPCSTR/PCSTR, LPSTR/PSTR, LPCWSTR/PCWSTR, and PWSTR/LPWSTR: These are ways of referring to strings in different forms. "L" means it's a long type (from 16-bit times), "C" means the string is read-only, and "P" means you can read from and write to this string.

Code `LPCSTR lpcString = "Hello, world!";`

`PCSTR pcString = "Hello, world!";`

Code `LPSTR lpString = "Hello, world!";`

`PSTR pString = "Hello, world!";`

Code `LPCWSTR lpwcString = L"Hello, world!";`

`PCWSTR pcwString = L"Hello, world!";`

Code `LPWSTR lpwString = L"Hello, world!";`

`PWSTR pwString = L"Hello, world!";`

wchar_t: This represents wide characters, or "wchar."

Then there are types like PVOID, which is a pointer to any data type. And ULONG_PTR is used for math with pointers—it's an unsigned integer that's the same size as a pointer on your architecture.

These make it easier to write programs that can do more complex tasks and interact properly with Windows.

Data Types Pointers

The Windows API lets you define what kind of data a variable can hold. If it's a pointer (a reference to another piece of memory), you'll see a "P" in front of its name, like POINTERSIZE_T. Otherwise, there's no "P". This is handy when you're working with functions that expect a certain type of data or a pointer to that data.

Here are some examples:

HANDLE: If you have an HANDLE and want a pointer to it, you add the "P" to get PHANDLE.

SIZE_T: You'd say SIZE_T for the value itself, but PSIZE_T if you're passing around its memory address.

DWORD: A regular DWORD is just a number, so it's PDWORD when you're dealing with its pointer.

These will be important when we use Windows APIs that require us to pass data types or pointers as function parameters.

If you'd like to read more, DM me and i will send the whole PDF.

Phantom_NiNja - March 16, 2025, 2:27 am

(Mar 15, 2025 02:03 PM)HyperHound Wrote: [\[CITE\]](#) Introduction

The Windows API helps programs talk to the Windows system. It lets apps show things on the screen, change files, or ask for info from the registry. And if you want to know more about how it works, Microsoft has all the details [here](#)

Windows Data Types

The Windows API uses many different kinds of data types that might not be as familiar as basic ones like "int" or "float." Here's an overview, and you can read more about them all on this page: [\[link\]](#).

Here are some examples:

DWORD: A number without a sign (positive only) that takes up 32 bits of space in memory.

Code DWORD dwVariable = 42;

SIZE_T: Used for counting how many bytes something is. It's always the same size as a pointer on your system—32 bits if you're using a 32-bit system, and 64 bits on a 64-bit system.

Code SIZE_T sVariable = sizeof(int);

VOID: This means "no data type"—it's used when we don't want to imply that we're dealing with a particular kind of data.

Code `void* pVariable = NULL; // This is the same as PVOID`

HANDLE: A special kind of number that the operating system uses to manage things like files or processes.

Code `HANDLE hFile = CreateFile(...);`

HMODULE: This is how you refer to a module (like a DLL) by its memory address.

Code `HMODULE hModule = GetModuleHandle(...);`

LPCSTR/PCSTR, LPSTR/PSTR, LPCWSTR/PCWSTR, and PWSTR/LPWSTR: These are ways of referring to strings in different forms. "L" means it's a long type (from 16-bit times), "C" means the string is read-only, and "P" means you can read from and write to this string.

Code `LPCSTR lpString = "Hello, world!";`

`PCSTR pcString = "Hello, world!";`

Code `LPSTR lpString = "Hello, world!";`

`PSTR pString = "Hello, world!";`

Code `LPCWSTR lpwString = L"Hello, world!";`

`PCWSTR pcwString = L"Hello, world!";`

Code `LPWSTR lpwString = L"Hello, world!";`

`PWSTR pwString = L"Hello, world!";`

`wchar_t`: This represents wide characters, or "wchar."

Then there are types like PVOID, which is a pointer to any data type. And ULONG_PTR is used for math with pointers—it's an unsigned integer that's the same size as a pointer on your architecture.

These make it easier to write programs that can do more complex tasks and interact properly with Windows.

Data Types Pointers

The Windows API lets you define what kind of data a variable can hold. If it's a pointer (a reference to another piece of memory), you'll see a "P" in front of its name, like POINTERSIZE_T. Otherwise, there's no "P". This is handy when you're working with functions that expect a certain type of data or a pointer to that data.

Here are some examples:

HANDLE: If you have an HANDLE and want a pointer to it, you add the "P" to get PHANDLE.

SIZE_T: You'd say SIZE_T for the value itself, but PSIZE_T if you're passing around its memory address.

DWORD: A regular DWORD is just a number, so it's PDWORD when you're dealing with its pointer.

These will be important when we use Windows APIs that require us to pass data types or pointers as function parameters.

If you'd like to read more, DM me and i will send the whole PDF.

I thought windows api is mostly just 8bit data and 16 bit data. I know the OS is 32 ot 64bits but I thought the whole windows api is still 8 bits and 16 bits.

Are you talking about the windows internals book?

DarkVision - *March 20, 2025, 12:04 am*

Good initiative. The Windows API is definitely the tool of the trade. ?

MrBot010 - *April 1, 2025, 3:29 am*

Good initiative.....