SYSTEM_INFO structure

Contains information about the current computer system. This includes the architecture and type of the processor, the number of processors in the system, the page size, and other such information.

Syntax

C++

```cpp
typedef struct _SYSTEM_INFO {
  union {
    DWORD  dwOemId;
    struct {
      WORD wProcessorArchitecture;
      WORD wReserved;
    };
  };
  DWORD     dwPageSize;
  LPVOID    lpMinimumApplicationAddress;
  LPVOID    lpMaximumApplicationAddress;
  DWORD_PTR dwActiveProcessorMask;
  DWORD     dwNumberOfProcessors;
  DWORD     dwProcessorType;
  DWORD     dwAllocationGranularity;
  WORD      wProcessorLevel;
  WORD      wProcessorRevision;
} SYSTEM_INFO;
```

Members

dwOemId

An obsolete member that is retained for compatibility. Applications should use the **wProcessorArchitecture** branch of the union.

wProcessorArchitecture

The processor architecture of the installed operating system. This member can be one of the following values.

Value	Meaning

PROCESSOR_ARCHITECTURE_AMD64 9	x64 (AMD or Intel)
PROCESSOR_ARCHITECTURE_ARM 5	ARM
PROCESSOR_ARCHITECTURE_IA64 6	Intel Itanium-based
PROCESSOR_ARCHITECTURE_INTEL 0	x86
PROCESSOR_ARCHITECTURE_UNKNOWN 0xffff	Unknown architecture.

wReserved

This member is reserved for future use.

dwPageSize

The page size and the granularity of page protection and commitment. This is the page size used by the VirtualAlloc function.

lpMinimumApplicationAddress

A pointer to the lowest memory address accessible to applications and dynamic-link libraries (DLLs).

lpMaximumApplicationAddress

A pointer to the highest memory address accessible to applications and DLLs.

dwActiveProcessorMask

A mask representing the set of processors configured into the system. Bit 0 is processor 0; bit 31 is processor 31.

dwNumberOfProcessors

The number of logical processors in the current group. To retrieve this value, use the

GetLogicalProcessorInformation function.

> **Note** For information about the physical processors shared by logical processors, call GetLogicalProcessorInformationEx with the *RelationshipType* parameter set to RelationProcessorPackage (3).

dwProcessorType

An obsolete member that is retained for compatibility. Use the **wProcessorArchitecture**, **wProcessorLevel**, and **wProcessorRevision** members to determine the type of processor.

PROCESSOR_INTEL_386 (386)
PROCESSOR_INTEL_486 (486)
PROCESSOR_INTEL_PENTIUM (586)
PROCESSOR_INTEL_IA64 (2200)
PROCESSOR_AMD_X8664 (8664)
PROCESSOR_ARM (Reserved)

dwAllocationGranularity

The granularity for the starting address at which virtual memory can be allocated. For more information, see VirtualAlloc.

wProcessorLevel

The architecture-dependent processor level. It should be used only for display purposes. To determine the feature set of a processor, use the IsProcessorFeaturePresent function.

If **wProcessorArchitecture** is PROCESSOR_ARCHITECTURE_INTEL, **wProcessorLevel** is defined by the CPU vendor.

If **wProcessorArchitecture** is PROCESSOR_ARCHITECTURE_IA64, **wProcessorLevel** is set to 1.

wProcessorRevision

The architecture-dependent processor revision. The following table shows how the revision value is assembled for each type of processor architecture.

Processor	Value
Intel Pentium, Cyrix, or NextGen 586	The high byte is the model and the low byte is the stepping. For example, if the value is *xxyy*, the model number and stepping can be displayed as follows: Model *xx*, Stepping *yy*
Intel 80386 or 80486	A value of the form *xxyz*. If *xx* is equal to 0xFF, *y* - 0xA is the model number, and *z* is the stepping identifier.

	If *xx* is not equal to 0xFF, *xx* + 'A' is the stepping letter and *yz* is the minor stepping.
ARM	Reserved.

Examples

For an example, see Getting Hardware Information.

Requirements

Minimum supported client	Windows 2000 Professional [desktop apps | Windows Store apps]
Minimum supported server	Windows 2000 Server [desktop apps | Windows Store apps]
Header	Winbase.h (include Windows.h)

See also

GetNativeSystemInfo
GetSystemInfo
MapViewOfFile
MapViewOfFileEx