Kernel Functions (Compact) 2013

The following table shows kernel functions with a description of the purpose of each and whether they can be called in kernel mode only.

Function	Kernel mode only	Description
AllocPhysMem	Yes	Allocates physically contiguous memory.
CacheRangeFlush	Yes	Flushes a specific range of the cache.
CacheSync	Yes	Flushes the cache.
CeCallUserProc	No	Loads the user interface (UI) proxy device driver.
CeGetCacheInfo	Yes	Obtains cache information.
CeGetRandomSeed	No	Obtains a random seed that can be used in an algorithm.
CeSetMemoryAttributes	Yes	Uses memory attributes supported on some hardware platforms that the kernel does not support by default.
CeSetPowerOnEvent	Yes	Signals events during suspend/resume.
CeVirtualSharedAlloc	Yes	Allocates read/write memory to the caller and read-only memory to other processes.
CreateAPIHandle	No	Creates a handle and associates the handle to the specified handle object.
CreateAPISet	No	Creates an API set from the list of functions passed as a parameter.
CreateStaticMapping	Yes	Creates a static virtual memory address that maps to a physical address.
CreateWatchDogTimer	No	Creates a watchdog timer.
DelayedBootWorkComplete	No	Informs the system that the delayed work is complete.
DeleteStaticMapping	No	Deletes a static virtual memory address that maps to a physical address.
DrainDelayedBootWork	No	Signals to begin the queued tasks.
DrWatsonClear	Yes	Clears Windows Embedded Compact error reporting dump storage by

		setting the Windows Embedded Compact error reporting dump area to zero.
DrWatsonFlush	Yes	Flushes Windows Embedded Compact error reporting dump data to the underlining storage.
DrWatsonGetSize	Yes	Obtains the size of the Windows Embedded Compact error reporting dump area.
DrWatsonReadData	Yes	Reads data from the Windows Embedded Compact error reporting dump area.
DrWatsonWriteData	Yes	Writes data to the Windows Embedded Compact error reporting dump area.
ForcePageout	Yes	Forces the OS to swap out all discardable pages from memory.
ForwardDeviceIoControl	No	Enables drivers to forward I/O controls to other drivers when the driver does not know anything about the I/O control so that no validation is performed. The driver that is called into still has the correct direct caller information to perform parameter validation.
FreePhysMem	Yes	Releases physical memory back to the system.
GetCallerVMProcessId	No	Obtains the process identifier of the caller that originated the call to the API.
GetDirectCallerProcessId	No	Obtains the direct caller's process identifier.
GetEPC	Yes	Obtains the interrupted program counter, which is the exception program counter (EPC).
GetEventData	No	Obtains data associated with an event.
GetOwnerProcess	No	Obtains the process handle of the current thread owner.
GetStdioPathW	Yes	Obtains the name of the device driver being used for a standard input, output, or error output operation.
GetSystemMemoryDivision	Yes	Obtains information about the object store and system memory. This function is obsolete.
GetThreadCallStack	No	Obtains the call stack of an arbitrary thread in the system.

HookInterrupt	Yes	Registers an interrupt service routine (ISR) with the kernel, specifying a hardware interrupt indicated by its interrupt request (IRQ) line value.
HookIPI		Installs an interrupt handler for inter processor interrupt.
InterruptDisable	Yes	Disables a hardware interrupt as specified by its interrupt identifier.
InterruptDone	Yes	Signals to the kernel that interrupt processing has been completed.
InterruptInitialize	Yes	Initializes a hardware interrupt with the kernel. This initialization allows the device driver to register an event and enable the interrupt.
InterruptMask	Yes	Masks hardware interrupts.
INTERRUPTS_ENABLE	Yes	Enables all interrupts based on a parameter and returns the current state.
INTERRUPTS_OFF	Yes	Disables all interrupts.
INTERRUPTS_ON	Yes	Enables all interrupts.
IsNamedEventSignaled	No	Checks whether a named event is signaled.
ISRHandler	Yes	Prototype function used by an OEM or an independent hardware vendor (IHV) to create and export an installable interrupt handler.
KCompareFileTime	Yes	Compares two file time values for equality.
KernelLibIoControl	No	Called by a driver to communicate with an interrupt handler.
KLibAllocShareMem	No	Allocates memory that can be shared between the ISR and IST.
KLibFreeShareMem	No	Frees memory that is allocated by the **KLibAllocShareMem** function.
KLibUnalignedAccessEnable	No	Enables or disables unaligned memory access on ARMv6 and later microprocessors.
LoadDriver	Yes	Maps the specified executable module into the address space of the calling process.
LockPages	Yes	Locks into memory the specified region of the virtual address space of the process, ensuring that subsequent access to the region does not incur a page fault.

OpenWatchDogTimer	Yes	Opens an existing watchdog timer.
PageOutModule	Yes	Swaps out all the pages of a process or DLL that are available for paging.
PFNAPIERRHANDLER	Not applicable	Prototype for the error handler for the SetAPIErrorHandler function.
PFN_UIENTRYPOINT	Not applicable	Prototype for the user interface (UI) proxy device driver entry point.
ProfileCaptureStatus	Yes	Queries the OEM profiler after it is running.
ProfilerHit	Yes	Implemented in the kernel and called by the profiler ISR to record a profiling sample.
ProfilerHitEx	Yes	Implemented in the kernel and called by OEM adaptation layer (OAL) code generating interrupts. **ProfilerHitEx** is called by the profiler ISR to record a profiling sample.
ProfileStart	No	Starts the Windows Embedded Compact instrumented kernel or Monte Carlo profiling.
ProfileStartEx	No	Starts the profiler.
ProfileStop	No	Stops the Windows Embedded Compact instrumented kernel or Monte Carlo profiling and displays a profile report on a debug terminal.
RefreshWatchDogTimer	Yes	Refreshes a watchdog timer.
RegisterAPISet	No	Registers an API set.
RegisterDelayedBootWork	No	Registers delayed work items.
RegisterDirectMethods	Yes	Registers a second internal method table.
SetAPIErrorHandler	No	Registers an error handler for a Program Static Library (PSL).
SetDbgZone	Yes	Sets or queries zones for either a process or a module, but not both.
SetEventData	No	Associates data with an event handle.
SetInterruptEvent	Yes	Used by a device driver to cause an artificial interrupt event.

SetJITDebuggerPath	Yes	Changes the default Just-in-Time (JIT) debugger dynamically. **SetJITDebuggerPath** can be called only by a privileged or trusted process.
SetStdioPathW	Yes	Sets the standard input, output, or error output destination path.
SetSystemMemoryDivision	Yes	Sets the specified number of pages for the object store. This function is obsolete.
SleepTillTick	No	Suspends the current thread until the next system tick.
StartWatchDogTimer	No	Starts a watchdog timer.
StopWatchDogTimer	No	Stops a watchdog timer.
UnhookInterrupt	Yes	Deregisters an ISR with a specific hardware interrupt.
UnlockPages	Yes	Unlocks a specified range of pages in the virtual address space of a process, enabling the system to swap the pages out if necessary.
VirtualCopy	Yes	Binds a specific physical memory range to a statically mapped virtual address.
VirtualSetAttributes	Yes	Changes the per-page attributes for a range of virtual memory, which is usually copied from a physical location not known to the kernel.
WaitForAPIReady	No	Indicates whether the specified API set has been registered.

Application Programming Interface (API)

APIs are assets of routines, protocols, and tools for building software applications.

Additional References

Kernel Functions https://web.archive.org/web/20160719112359/https://msdn.microsoft.com/en-us/library/ee482951.aspx

API https://web.archive.org/web/20160719112521/http://www.webopedia.com/TERM/A/API.html

Paging https://web.archive.org/web/20160719112636/http://pages.cs.wisc.edu/~remzi/OSTEP/vm-paging.pdf

Please contact the Course Coordinators if you are unable to access any of the Recommended Internet Sites.