PsSetCreateProcessNotifyRoutine function (ntddk.h) - Windows drivers

By EliotSeattle

Archived: 2026-04-05 16:01:38 UTC

The **PsSetCreateProcessNotifyRoutine** routine adds a driver-supplied callback routine to, or removes it from, a list of routines to be called whenever a process is created or deleted.

Syntax

```
NTSTATUS PsSetCreateProcessNotifyRoutine(
  [in] PCREATE_PROCESS_NOTIFY_ROUTINE NotifyRoutine,
  [in] BOOLEAN                        Remove
);
```

Parameters

`[in] NotifyRoutine`

Specifies the entry point of a caller-supplied process-creation callback routine. See PCREATE_PROCESS_NOTIFY_ROUTINE.

`[in] Remove`

Indicates whether the routine specified by *NotifyRoutine* should be added to or removed from the system's list of notification routines. If **FALSE**, the specified routine is added to the list. If **TRUE**, the specified routine is removed from the list.

Return value

PsSetCreateProcessNotifyRoutine can return one of the following:

Return code	Description
STATUS_SUCCESS	The given *NotifyRoutine* is now registered with the system.
STATUS_INVALID_PARAMETER	The given *NotifyRoutine* has already been registered, so this call is a redundant call, or the system has reached its limit for registering process-creation callbacks.

Highest-level drivers can call **PsSetCreateProcessNotifyRoutine** to set up their process-creation notify routines implemented as PCREATE_PROCESS_NOTIFY_ROUTINE.

An IFS or highest-level system-profiling driver might register a process-creation callback to track the system-wide creation and deletion of processes against the driver's internal state. For Windows Vista and later versions of Windows, the system can register up to 64 process-creation callback routines.

A driver must remove any callbacks that it registers before it unloads. You can remove the callback by calling **PsSetCreateProcessNotify** with *Remove* = **TRUE**. A driver must not make this call from its implementation of the *PCREATE_PROCESS_NOTIFY_ROUTINE* callback routine.

After a driver-supplied routine is registered, it is called with *Create* set to **TRUE** just after the initial thread is created within the newly created process designated by the input *ProcessId* handle. The input *ParentId* handle identifies the parent process of the newly-created process (this is the parent used for priority, affinity, quota, token, and handle inheritance, among others).

Requirements

See also

PCREATE_PROCESS_NOTIFY_ROUTINE

PsGetCurrentProcessId

PsSetCreateProcessNotifyRoutineEx

PsSetCreateThreadNotifyRoutine

PsSetLoadImageNotifyRoutine
