

:c:type:`uv_signal_t` --- Signal handle

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 4); [backlink](#)

Unknown interpreted text role "c:type".

Signal handles implement Unix style signal handling on a per-event loop bases.

Windows notes

Reception of some signals is emulated:

- SIGINT is normally delivered when the user presses CTRL+C. However, like on Unix, it is not generated when terminal raw mode is enabled.
- SIGBREAK is delivered when the user pressed CTRL + BREAK.
- SIGHUP is generated when the user closes the console window. On SIGHUP the program is given approximately 10 seconds to perform cleanup. After that Windows will unconditionally terminate it.
- SIGWINCH is raised whenever libuv detects that the console has been resized. When a libuv app is running under a console emulator, or when a 32-bit libuv app is running on 64-bit system, SIGWINCH will be emulated. In such cases SIGWINCH signals may not always be delivered in a timely manner. For a writable `:c:type:`uv_tty_t`` handle libuv will only detect size changes when the cursor is moved. When a readable `:c:type:`uv_tty_t`` handle is used, resizing of the console buffer will be detected only if the handle is in raw mode and is being read.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 23); [backlink](#)

Unknown interpreted text role "c:type".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 23); [backlink](#)

Unknown interpreted text role "c:type".

- Watchers for other signals can be successfully created, but these signals are never received. These signals are: *SIGILL*, *SIGABRT*, *SIGFPE*, *SIGSEGV*, *SIGTERM* and *SIGKILL*.
- Calls to `raise()` or `abort()` to programmatically raise a signal are not detected by libuv; these will not trigger a signal watcher.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 39)

Unknown directive type "versionchanged".

```
.. versionchanged:: 1.15.0 SIGWINCH support on Windows was improved.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 40)

Unknown directive type "versionchanged".

```
.. versionchanged:: 1.31.0 32-bit libuv SIGWINCH support on 64-bit Windows was
   rolled back to old implementation.
```

Unix notes

- SIGKILL and SIGSTOP are impossible to catch.
- Handling SIGBUS, SIGFPE, SIGILL or SIGSEGV via libuv results into undefined behavior.
- SIGABRT will not be caught by libuv if generated by `abort()`, e.g. through `assert()`.
- On Linux SIGRT0 and SIGRT1 (signals 32 and 33) are used by the NPTL pthreads library to manage threads. Installing watchers for those signals will lead to unpredictable behavior and is strongly discouraged. Future versions of libuv may simply

reject them.

Data types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 60)

Unknown directive type "c.type".

```
.. c:type:: uv_signal_t
    Signal handle type.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 64)

Unknown directive type "c.type".

```
.. c:type:: void (*uv_signal_cb)(uv_signal_t* handle, int signum)
    Type definition for callback passed to :c:func:`uv_signal_start`.
```

Public members

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 72)

Unknown directive type "c.member".

```
.. c:member:: int uv_signal_t.signum
    Signal being monitored by this handle. Readonly.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 76)

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_handle_t` members also apply.
```

API

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 82)

Unknown directive type "c.function".

```
.. c:function:: int uv_signal_init(uv_loop_t* loop, uv_signal_t* signal)
    Initialize the handle.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 86)

Unknown directive type "c.function".

```
.. c:function:: int uv_signal_start(uv_signal_t* signal, uv_signal_cb cb, int signum)
    Start the handle with the given callback, watching for the given signal.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 90)

Unknown directive type "c:function".

```
.. c:function:: int uv_signal_start_oneShot(uv_signal_t* signal, uv_signal_cb cb, int signum)
```

```
.. versionadded:: 1.12.0
```

Same functionality as :c:func:`uv_signal_start` but the signal handler is reset the moment the signal is received.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 97)

Unknown directive type "c:function".

```
.. c:function:: int uv_signal_stop(uv_signal_t* signal)
```

Stop the handle, the callback will no longer be called.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] signal.rst, line 101)

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_handle_t` API functions also apply.
```