

Miscellaneous utilities

This section contains miscellaneous functions that don't really belong in any other section.

Data types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 14)

Unknown directive type "c:type".

```
.. c:type:: uv_buf_t

    Buffer data type.

.. c:member:: char* uv_buf_t.base

    Pointer to the base of the buffer.

.. c:member:: size_t uv_buf_t.len

    Total bytes in the buffer.

.. note::
    On Windows this field is ULONG.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 29)

Unknown directive type "c:type".

```
.. c:type:: void* (*uv_malloc_func)(size_t size)

    Replacement function for :man:`malloc(3)`.
    See :c:func:`uv_replace_allocator`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 34)

Unknown directive type "c:type".

```
.. c:type:: void* (*uv_realloc_func)(void* ptr, size_t size)

    Replacement function for :man:`realloc(3)`.
    See :c:func:`uv_replace_allocator`.
```

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

Unknown directive type "c:type".

```
.. c:type:: void* (*uv_calloc_func)(size_t count, size_t size)

    Replacement function for :man:`calloc(3)`.
    See :c:func:`uv_replace_allocator`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 44)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_free_func)(void* ptr)

    Replacement function for :man:`free(3)`.
    See :c:func:`uv_replace_allocator`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 49)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_random_cb)(uv_random_t* req, int status, void* buf, size_t buflen)

    Callback passed to :c:func:`uv_random`. `status` is non-zero in case of
    error. The `buf` pointer is the same pointer that was passed to
    :c:func:`uv_random`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 55)

Unknown directive type "c:type".

```
.. c:type:: uv_file

    Cross platform representation of a file handle.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 59)

Unknown directive type "c:type".

```
.. c:type:: uv_os_sock_t

    Cross platform representation of a socket handle.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 63)

Unknown directive type "c:type".

```
.. c:type:: uv_os_fd_t
```

Abstract representation of a file descriptor. On Unix systems this is a `typedef` of `int` and on Windows a `HANDLE`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 68)

Unknown directive type "c:type".

```
.. c:type:: uv_pid_t
```

Cross platform representation of a `pid_t`.

```
.. versionadded:: 1.16.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 74)

Unknown directive type "c:type".

```
.. c:type:: uv_timeval_t
```

Data type for storing times.

```
::
```

```
typedef struct {
    long tv_sec;
    long tv_usec;
} uv_timeval_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 85)

Unknown directive type "c:type".

```
.. c:type:: uv_timeval64_t
```

Alternative data type for storing times.

```
::
```

```
typedef struct {
    int64_t tv_sec;
    int32_t tv_usec;
} uv_timeval64_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 96)

Unknown directive type "c:type".

```
.. c:type:: uv_rusage_t
```

Data type for resource usage results.

```
::
```

```
typedef struct {
    uv_timeval_t ru_utime; /* user CPU time used */
    uv_timeval_t ru_stime; /* system CPU time used */
    uint64_t ru_maxrss; /* maximum resident set size */
    uint64_t ru_ixrss; /* integral shared memory size (X) */
    uint64_t ru_idrss; /* integral unshared data size (X) */
    uint64_t ru_isrss; /* integral unshared stack size (X) */
    uint64_t ru_minflt; /* page reclaims (soft page faults) (X) */
    uint64_t ru_majflt; /* page faults (hard page faults) */
    uint64_t ru_nswap; /* swaps (X) */
    uint64_t ru_inblock; /* block input operations */
    uint64_t ru_oublock; /* block output operations */
    uint64_t ru_msgsnd; /* IPC messages sent (X) */
    uint64_t ru_msgrcv; /* IPC messages received (X) */
    uint64_t ru_nsignals; /* signals received (X) */
    uint64_t ru_nvcsw; /* voluntary context switches (X) */
    uint64_t ru_nivcsw; /* involuntary context switches (X) */
} uv_rusage_t;
```

Members marked with `(X)` are unsupported on Windows.
See :man:getrusage(2) for supported fields on Unix

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 124)

Unknown directive type "c:type".

```
.. c:type:: uv_cpu_info_t
```

Data type for CPU information.

```
::
```

```
typedef struct uv_cpu_info_s {
    char* model;
    int speed;
    struct uv_cpu_times_s {
        uint64_t user; /* milliseconds */
        uint64_t nice; /* milliseconds */
        uint64_t sys; /* milliseconds */
        uint64_t idle; /* milliseconds */
        uint64_t irq; /* milliseconds */
    } cpu_times;
} uv_cpu_info_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 142)

Unknown directive type "c:type".

```
.. c:type:: uv_interface_address_t

Data type for interface addresses.

::

typedef struct uv_interface_address_s {
    char* name;
    char phys_addr[6];
    int is_internal;
    union {
        struct sockaddr_in address4;
        struct sockaddr_in6 address6;
    } address;
    union {
        struct sockaddr_in netmask4;
        struct sockaddr_in6 netmask6;
    } netmask;
} uv_interface_address_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 162)

Unknown directive type "c:type".

```
.. c:type:: uv_passwd_t

Data type for password file information.

::

typedef struct uv_passwd_s {
    char* username;
    long uid;
    long gid;
    char* shell;
    char* homedir;
} uv_passwd_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 176)

Unknown directive type "c:type".

```
.. c:type:: uv_utsname_t

Data type for operating system name and version information.

::

typedef struct uv_utsname_s {
    char sysname[256];
    char release[256];
    char version[256];
    char machine[256];
} uv_utsname_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 189)

Unknown directive type "c:type".

```
.. c:type:: uv_env_item_t

Data type for environment variable storage.

::

typedef struct uv_env_item_s {
    char* name;
    char* value;
} uv_env_item_t;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 200)

Unknown directive type "c:type".

```
.. c:type:: uv_random_t

Random data request type.
```

API

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 207)

Unknown directive type "c:function".

```
.. c:function:: uv_handle_type uv_guess_handle(uv_file file)

Used to detect what type of stream should be used with a given file
descriptor. Usually this will be used during initialization to guess the
type of the stdio streams.

For :man:`isatty(3)` equivalent functionality use this function and test
for ``UV_TTY``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 216)

Unknown directive type "c:function".

```
.. c:function:: int uv_replace_allocator(uv_malloc_func malloc_func, uv_realloc_func realloc_func, uv_calloc_func calloc_func)

.. versionadded:: 1.6.0
```

Override the use of the standard library's `:man:malloc(3)`, `:man:calloc(3)`, `:man:realloc(3)`, `:man:free(3)`, memory allocation functions.

This function must be called before any other libuv function is called or after all resources have been freed and thus libuv doesn't reference any allocated memory chunk.

On success, it returns 0, if any of the function pointers is NULL it returns `UV_EINVAL`.

.. warning:: There is no protection against changing the allocator multiple times. If the user changes it they are responsible for making sure the allocator is changed while no memory was allocated with the previous allocator, or that they are compatible.

.. warning:: Allocator must be thread-safe.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 238)

Unknown directive type "c:function".

```
.. c:function:: void uv_library_shutdown(void);
```

.. versionadded:: 1.38.0

Release any global state that libuv is holding onto. Libuv will normally do so automatically when it is unloaded but it can be instructed to perform cleanup manually.

.. warning:: Only call `:c:func:`uv_library_shutdown()`` once.

.. warning:: Don't call `:c:func:`uv_library_shutdown()`` when there are still event loops or I/O requests active.

.. warning:: Don't call libuv functions after calling `:c:func:`uv_library_shutdown()``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 254)

Unknown directive type "c:function".

```
.. c:function:: uv_buf_t uv_buf_init(char* base, unsigned int len)
```

Constructor for `:c:type:`uv_buf_t``.

Due to platform differences the user cannot rely on the ordering of the ``base`` and ``len`` members of the `uv_buf_t` struct. The user is responsible for freeing ``base`` after the `uv_buf_t` is done. Return struct passed by value.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 262)

Unknown directive type "c:function".

```
.. c:function:: char** uv_setup_args(int argc, char** argv)
```

Store the program arguments. Required for getting / setting the process title or the executable path. Libuv may take ownership of the memory that ``argv`` points to. This function should be called exactly once, at program start-up.

Example:

```
::
```

```
    argv = uv_setup_args(argc, argv); /* May return a copy of argv. */
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 275)

Unknown directive type "c:function".

```
.. c:function:: int uv_get_process_title(char* buffer, size_t size)
```

Gets the title of the current process. You *must* call ``uv_setup_args`` before calling this function on Unix and AIX systems. If ``uv_setup_args`` has not been called on systems that require it, then ``UV_ENOBUFS`` is returned. If ``buffer`` is ``NULL`` or ``size`` is zero, ``UV_EINVAL`` is returned. If ``size`` cannot accommodate the process title and terminating ``nul`` character, the function returns ``UV_ENOBUFS``.

.. note::
On BSD systems, ``uv_setup_args`` is needed for getting the initial process title. The process title returned will be an empty string until either ``uv_setup_args`` or ``uv_set_process_title`` is called.

.. versionchanged:: 1.18.1 now thread-safe on all supported platforms.

.. versionchanged:: 1.39.0 now returns an error if ``uv_setup_args`` is needed but hasn't been called.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 294)

Unknown directive type "c:function".

```
.. c:function:: int uv_set_process_title(const char* title)
```

Sets the current process title. You *must* call ``uv_setup_args`` before calling this function on Unix and AIX systems. If ``uv_setup_args`` has not been called on systems that require it, then ``UV_ENOBUFS`` is returned. On platforms with a fixed size buffer for the process title the contents of ``title`` will be copied to the buffer and truncated if larger than the available space. Other platforms will return ``UV_ENOMEM`` if they cannot allocate enough space to duplicate the contents of ``title``.

.. versionchanged:: 1.18.1 now thread-safe on all supported platforms.

```
.. versionchanged:: 1.39.0 now returns an error if `uv_setup_args` is needed
but hasn't been called.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 309)

Unknown directive type "c:function".

```
.. c:function:: int uv_resident_set_memory(size_t* rss)

Gets the resident set size (RSS) for the current process.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 313)

Unknown directive type "c:function".

```
.. c:function:: int uv_uptime(double* uptime)

Gets the current system uptime.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 317)

Unknown directive type "c:function".

```
.. c:function:: int uv_getrusage(uv_rusage_t* rusage)

Gets the resource usage measures for the current process.

.. note::
    On Windows not all fields are set, the unsupported fields are filled with zeroes.
    See :c:type:`uv_rusage_t` for more details.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 325)

Unknown directive type "c:function".

```
.. c:function:: uv_pid_t uv_os_getpid(void)

Returns the current process ID.

.. versionadded:: 1.18.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 331)

Unknown directive type "c:function".

```
.. c:function:: uv_pid_t uv_os_getppid(void)

Returns the parent process ID.

.. versionadded:: 1.16.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 337)

Unknown directive type "c:function".

```
.. c:function:: int uv_cpu_info(uv_cpu_info_t** cpu_infos, int* count)

Gets information about the CPUs on the system. The `cpu_infos` array will
have `count` elements and needs to be freed with :c:func:`uv_free_cpu_info`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 342)

Unknown directive type "c:function".

```
.. c:function:: void uv_free_cpu_info(uv_cpu_info_t* cpu_infos, int count)

Frees the `cpu_infos` array previously allocated with :c:func:`uv_cpu_info`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 346)

Unknown directive type "c:function".

```
.. c:function:: int uv_interface_addresses(uv_interface_address_t** addresses, int* count)

Gets address information about the network interfaces on the system. An
array of `count` elements is allocated and returned in `addresses`. It must
be freed by the user, calling :c:func:`uv_free_interface_addresses`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 352)

Unknown directive type "c:function".

```
.. c:function:: void uv_free_interface_addresses(uv_interface_address_t* addresses, int count)

Free an array of :c:type:`uv_interface_address_t` which was returned by
:c:func:`uv_interface_addresses`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 357)

Unknown directive type "c:function".

```
.. c:function:: void uv_loadavg(double avg[3])

Gets the load average. See: `<https://en.wikipedia.org/wiki/Load\_\(computing\)>`

.. note::
    Returns [0,0,0] on Windows (i.e., it's not implemented).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 364)

Unknown directive type "c:function".

```
.. c:function:: int uv_ip4_addr(const char* ip, int port, struct sockaddr_in* addr)

Convert a string containing an IPv4 addresses to a binary structure.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 368)

Unknown directive type "c:function".

```
.. c:function:: int uv_ip6_addr(const char* ip, int port, struct sockaddr_in6* addr)

Convert a string containing an IPv6 addresses to a binary structure.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 372)

Unknown directive type "c:function".

```
.. c:function:: int uv_ip4_name(const struct sockaddr_in* src, char* dst, size_t size)

Convert a binary structure containing an IPv4 address to a string.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 376)

Unknown directive type "c:function".

```
.. c:function:: int uv_ip6_name(const struct sockaddr_in6* src, char* dst, size_t size)

Convert a binary structure containing an IPv6 address to a string.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 380)

Unknown directive type "c:function".

```
.. c:function:: int uv_ip_name(const struct sockaddr *src, char *dst, size_t size)

Convert a binary structure containing an IPv4 address or an IPv6 address to a string.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 384)

Unknown directive type "c:function".

```
.. c:function:: int uv_inet_ntop(int af, const void* src, char* dst, size_t size)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 385)

Unknown directive type "c:function".

```
.. c:function:: int uv_inet_pton(int af, const char* src, void* dst)

Cross-platform IPv6-capable implementation of :man:`inet_ntop(3)`
and :man:`inet_pton(3)`. On success they return 0. In case of error
the target `dst` pointer is unmodified.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 391)

Unknown directive type "c:macro".

```
.. c:macro:: UV_IF_NAMESIZE

Maximum IPv6 interface identifier name length. Defined as
`IFNAMSIZ` on Unix and `IF_NAMESIZE` on Linux and Windows.

.. versionadded:: 1.16.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 398)

Unknown directive type "c:function".

```
.. c:function:: int uv_if_indextoname(unsigned int ifindex, char* buffer, size_t* size)

IPv6-capable implementation of :man:`if_indextoname(3)`. When called,
`*size` indicates the length of the `buffer`, which is used to store the
result.
On success, zero is returned, `buffer` contains the interface name, and
`*size` represents the string length of the `buffer`, excluding the NUL
terminator byte from `*size`. On error, a negative result is
returned. If `buffer` is not large enough to hold the result,
`UV_ENOBUFS` is returned, and `*size` represents the necessary size in
bytes, including the NUL terminator byte into the `*size`.

On Unix, the returned interface name can be used directly as an
```

```
interface identifier in scoped IPv6 addresses, e.g.  
`fe80::abc:def1:2345%en0`.
```

On Windows, the returned interface cannot be used as an interface identifier, as Windows uses numerical interface identifiers, e.g. `fe80::abc:def1:2345%5`.

To get an interface identifier in a cross-platform compatible way, use `uv_if_indextoid()`.

Example:

```
::  
  
char ifname[UV_IF_NAMESIZE];  
size_t size = sizeof(ifname);  
uv_if_indextoname(sin6->sin6_scope_id, ifname, &size);  
  
.. versionadded:: 1.16.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 431)

Unknown directive type "c:function".

```
.. c:function:: int uv_if_indextoid(unsigned int ifindex, char* buffer, size_t* size)
```

Retrieves a network interface identifier suitable for use in an IPv6 scoped address. On Windows, returns the numeric `ifindex` as a string. On all other platforms, `uv_if_indextoname()` is called. The result is written to `buffer`, with `*size` indicating the length of `buffer`. If `buffer` is not large enough to hold the result, then `UV_ENOBUFS` is returned, and `*size` represents the size, including the NUL byte, required to hold the result.

See `uv_if_indextoname` for further details.

```
.. versionadded:: 1.16.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 445)

Unknown directive type "c:function".

```
.. c:function:: int uv_exepath(char* buffer, size_t* size)
```

Gets the executable path. You *must* call `uv_setup_args` before calling this function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 450)

Unknown directive type "c:function".

```
.. c:function:: int uv_cwd(char* buffer, size_t* size)
```

Gets the current working directory, and stores it in `buffer`. If the current working directory is too large to fit in `buffer`, this function returns `UV_ENOBUFS`, and sets `size` to the required length, including the null terminator.

```
.. versionchanged:: 1.1.0
```

On Unix the path no longer ends in a slash.

```
.. versionchanged:: 1.9.0 the returned length includes the terminating null  
   byte on `UV_ENOBUFS`, and the buffer is null terminated  
   on success.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 466)

Unknown directive type "c:function".

```
.. c:function:: int uv_chdir(const char* dir)
```

Changes the current working directory.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 470)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_homedir(char* buffer, size_t* size)
```

Gets the current user's home directory. On Windows, `uv_os_homedir()` first checks the `USERPROFILE` environment variable using `GetEnvironmentVariableW()`. If `USERPROFILE` is not set, `GetUserProfileDirectoryW()` is called. On all other operating systems, `uv_os_homedir()` first checks the `HOME` environment variable using `man:getenv(3)`. If `HOME` is not set, `man:getpwuid_r(3)` is called. The user's home directory is stored in `buffer`. When `uv_os_homedir()` is called, `size` indicates the maximum size of `buffer`. On success `size` is set to the string length of `buffer`. On `UV_ENOBUFS` failure `size` is set to the required length for `buffer`, including the null byte.

```
.. warning::  
   `uv_os_homedir()` is not thread safe.
```

```
.. versionadded:: 1.6.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 488)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_tmpdir(char* buffer, size_t* size)
```

```
Gets the temp directory. On Windows, `uv_os_tmpdir()` uses `GetTempPathW()`. On all other operating systems, `uv_os_tmpdir()` uses the first environment variable found in the ordered list `TMPDIR`, `TMP`, `TEMP`, and `TEMPDIR`. If none of these are found, the path `"/tmp"` is used, or, on Android, `"/data/local/tmp"` is used. The temp directory is stored in `buffer`. When `uv_os_tmpdir()` is called, `size` indicates the maximum size of `buffer`. On success `size` is set to the string length of `buffer` (which does not include the terminating null). On `UV_ENOBUFS` failure `size` is set to the required length for `buffer`, including the null byte.

.. warning::
    `uv_os_tmpdir()` is not thread safe.

.. versionadded:: 1.9.0
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 505)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_get_passwd(uv_passwd_t* pwd)

Gets a subset of the password file entry for the current effective uid (not the real uid). The populated data includes the username, euid, gid, shell, and home directory. On non-Windows systems, all data comes from :man:`getpwuid_r(3)`. On Windows, uid and gid are set to -1 and have no meaning, and shell is `NULL`. After successfully calling this function, the memory allocated to `pwd` needs to be freed with :c:func:`uv_os_free_passwd`.

.. versionadded:: 1.9.0
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 517)

Unknown directive type "c:function".

```
.. c:function:: void uv_os_free_passwd(uv_passwd_t* pwd)

Frees the `pwd` memory previously allocated with :c:func:`uv_os_get_passwd`.

.. versionadded:: 1.9.0
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 523)

Unknown directive type "c:function".

```
.. c:function:: uint64_t uv_get_free_memory(void)

Gets the amount of free memory available in the system, as reported by the kernel (in bytes).
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 527)

Unknown directive type "c:function".

```
.. c:function:: uint64_t uv_get_total_memory(void)

Gets the total amount of physical memory in the system (in bytes).
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 531)

Unknown directive type "c:function".

```
.. c:function:: uint64_t uv_get_constrained_memory(void)

Gets the amount of memory available to the process (in bytes) based on limits imposed by the OS. If there is no such constraint, or the constraint is unknown, `0` is returned. Note that it is not unusual for this value to be less than or greater than :c:func:`uv_get_total_memory`.

.. note::
    This function currently only returns a non-zero value on Linux, based on cgroups if it is present, and on z/OS based on RLIMIT_MEMPLIMIT.

.. versionadded:: 1.29.0
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 544)

Unknown directive type "c:function".

```
.. c:function:: uint64_t uv_hrtime(void)

Returns the current high-resolution real time. This is expressed in nanoseconds. It is relative to an arbitrary time in the past. It is not related to the time of day and therefore not subject to clock drift. The primary use is for measuring performance between intervals.

.. note::
    Not every platform can support nanosecond resolution; however, this value will always be in nanoseconds.
```

System Message: ERROR/3 (p:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 555)

Unknown directive type "c:function".

```
.. c:function:: void uv_print_all_handles(uv_loop_t* loop, FILE* stream)

Prints all handles associated with the given `loop` to the given `stream`.

Example:
```



```
::

uv_print_all_handles(uv_default_loop(), stderr);
/*
[~I] signal    0x1a25ea8
[~AI] async    0x1a25cf0
[R~~] idle     0x1a7a8c8
*/

The format is `[flags] handle-type handle-address`. For `flags`:

- `R` is printed for a handle that is referenced
- `A` is printed for a handle that is active
- `I` is printed for a handle that is internal

.. warning::
    This function is meant for ad hoc debugging, there is no API/ABI
    stability guarantees.

.. versionadded:: 1.8.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 582)

Unknown directive type "c:function".

```
.. c:function:: void uv_print_active_handles(uv_loop_t* loop, FILE* stream)

This is the same as :c:func:`uv_print_all_handles` except only active handles
are printed.

.. warning::
    This function is meant for ad hoc debugging, there is no API/ABI
    stability guarantees.

.. versionadded:: 1.8.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 593)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_environ(uv_env_item_t** envitems, int* count)

Retrieves all environment variables. This function will allocate memory
which must be freed by calling :c:func:`uv_os_free_environ`.

.. warning::
    This function is not thread safe.

.. versionadded:: 1.31.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 603)

Unknown directive type "c:function".

```
.. c:function:: void uv_os_free_environ(uv_env_item_t* envitems, int count);

Frees the memory allocated for the environment variables by
:c:func:`uv_os_environ`.

.. versionadded:: 1.31.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 610)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_getenv(const char* name, char* buffer, size_t* size)

Retrieves the environment variable specified by `name`, copies its value
into `buffer`, and sets `size` to the string length of the value. When
calling this function, `size` must be set to the amount of storage available
in `buffer`, including the null terminator. If the environment variable
exceeds the storage available in `buffer`, `UV_ENOBUFS` is returned, and
`size` is set to the amount of storage required to hold the value. If no
matching environment variable exists, `UV_ENOENT` is returned.

.. warning::
    This function is not thread safe.

.. versionadded:: 1.12.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 625)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_setenv(const char* name, const char* value)

Creates or updates the environment variable specified by `name` with
`value`.

.. warning::
    This function is not thread safe.

.. versionadded:: 1.12.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\node-master [deps] [uv] [docs] [src]misc.rst, line 635)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_unsetenv(const char* name)

Deletes the environment variable specified by `name`. If no such environment
```

variable exists, this function returns successfully.

```
.. warning::  
    This function is not thread safe.  
  
.. versionadded:: 1.12.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 645)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_gethostname(char* buffer, size_t* size)  
  
Returns the hostname as a null-terminated string in `buffer`, and sets  
`size` to the string length of the hostname. When calling this function,  
`size` must be set to the amount of storage available in `buffer`, including  
the null terminator. If the hostname exceeds the storage available in  
`buffer`, `UV_ENOBUFS` is returned, and `size` is set to the amount of  
storage required to hold the value.  
  
.. versionadded:: 1.12.0  
  
.. versionchanged:: 1.26.0 `UV_MAXHOSTNAMESIZE` is available and represents  
the maximum `buffer` size required to store a  
hostname and terminating `nul` character.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 660)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_getpriority(uv_pid_t pid, int* priority)  
  
Retrieves the scheduling priority of the process specified by `pid`. The  
returned value of `priority` is between -20 (high priority) and 19 (low  
priority).  
  
.. note::  
    On Windows, the returned priority will equal one of the `UV_PRIORITY`  
    constants.  
  
.. versionadded:: 1.23.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 672)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_setpriority(uv_pid_t pid, int priority)  
  
Sets the scheduling priority of the process specified by `pid`. The  
`priority` value range is between -20 (high priority) and 19 (low priority).  
The constants `UV_PRIORITY_LOW`, `UV_PRIORITY_BELOW_NORMAL`,  
`UV_PRIORITY_NORMAL`, `UV_PRIORITY_ABOVE_NORMAL`, `UV_PRIORITY_HIGH`, and  
`UV_PRIORITY_HIGHEST` are also provided for convenience.  
  
.. note::  
    On Windows, this function utilizes `SetPriorityClass()`. The `priority`  
    argument is mapped to a Windows priority class. When retrieving the  
    process priority, the result will equal one of the `UV_PRIORITY`  
    constants, and not necessarily the exact value of `priority`.  
  
.. note::  
    On Windows, setting `PRIORITY_HIGHEST` will only work for elevated user,  
    for others it will be silently reduced to `PRIORITY_HIGH`.  
  
.. note::  
    On IBM i PASE, the highest process priority is -10. The constant  
    `UV_PRIORITY_HIGHEST` is -10, `UV_PRIORITY_HIGH` is -7,  
    `UV_PRIORITY_ABOVE_NORMAL` is -4, `UV_PRIORITY_NORMAL` is 0,  
    `UV_PRIORITY_BELOW_NORMAL` is 15 and `UV_PRIORITY_LOW` is 39.  
  
.. note::  
    On IBM i PASE, you are not allowed to change your priority unless you  
    have the `*JOBCTL` special authority (even to lower it).  
  
.. versionadded:: 1.23.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 702)

Unknown directive type "c:function".

```
.. c:function:: int uv_os_utsname(uv_utsname_t* buffer)  
  
Retrieves system information in `buffer`. The populated data includes the  
operating system name, release, version, and machine. On non-Windows  
systems, `uv_os_utsname()` is a thin wrapper around :man:`uname(2)`. Returns  
zero on success, and a non-zero error value otherwise.  
  
.. versionadded:: 1.25.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 711)

Unknown directive type "c:function".

```
.. c:function:: int uv_gettimeofday(uv_timeval64_t* tv)  
  
Cross-platform implementation of :man:`gettimeofday(2)`. The timezone  
argument to `gettimeofday()` is not supported, as it is considered obsolete.  
  
.. versionadded:: 1.28.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 718)

Unknown directive type "c:function".

```
.. c:function:: int uv_random(uv_loop_t* loop, uv_random_t* req, void* buf, size_t buflen, unsigned int flags, uv_random_cb cb)

    Fill `buf` with exactly `buflen` cryptographically strong random bytes
    acquired from the system CSPRNG. `flags` is reserved for future extension
    and must currently be 0.

    Short reads are not possible. When less than `buflen` random bytes are
    available, a non-zero error value is returned or passed to the callback.

    The synchronous version may block indefinitely when not enough entropy
    is available. The asynchronous version may not ever finish when the system
    is low on entropy.

    Sources of entropy:

    - Windows: `RtlGenRandom` <https://docs.microsoft.com/en-us/windows/desktop/api/ntsecapi/nf-ntsecapi-rtlgenrandom>`.
    - Linux, Android: `:man:random(2)` if available, or `:man:urandom(4)`
      after reading from `/dev/random` once, or the `KERN_RANDOM`
      `:man:sysctl(2)`.
    - FreeBSD: `getrandom(2)` <https://www.freebsd.org/cgi/man.cgi?query=getrandom&section=2>`,
      or `/dev/urandom` after reading from `/dev/random` once.
    - NetBSD: `KERN_ARND` `sysctl(7)` <https://man.netbsd.org/sysctl.7>`.
    - macOS, OpenBSD: `getentropy(2)` <https://man.openbsd.org/getentropy.2>`
      if available, or `/dev/urandom` after reading from `/dev/random` once.
    - AIX: `/dev/random`.
    - IBM i: `/dev/urandom`.
    - Other UNIX: `/dev/urandom` after reading from `/dev/random` once.

    :returns: 0 on success, or an error code < 0 on failure. The contents of
    `buf` is undefined after an error.

    .. note::
        When using the synchronous version, both `loop` and `req` parameters
        are not used and can be set to `NULL`.

    .. versionadded:: 1.33.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]misc.rst, line 755)

Unknown directive type "c:function".

```
.. c:function:: void uv_sleep(unsigned int msec)

    Causes the calling thread to sleep for `msec` milliseconds.

    .. versionadded:: 1.34.0
```