:c:type:`uv_stream_t` --- Stream handle

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 4); ***backlink***

Unknown interpreted text role "c:type".

Stream handles provide an abstraction of a duplex communication channel. :c:type:`uv_stream_t` is an abstract type, libuv provides 3 stream implementations in the form of :c:type:`uv_tcp_t`, :c:type:`uv_pipe_t` and :c:type:`uv_tty_t`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 7); ***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) stream.rst`, line 7); ***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) stream.rst`, line 7); ***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) stream.rst`, line 7); ***backlink***

Unknown interpreted text role "c:type".

Data types

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 15)

Unknown directive type "c:type".

```
.. c:type:: uv_stream_t

    Stream handle type.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 19)

Unknown directive type "c:type".

```
.. c:type:: uv_connect_t

    Connect request type.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 23)

Unknown directive type "c:type".

```
.. c:type:: uv_shutdown_t

    Shutdown request type.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 27)

Unknown directive type "c:type".

```
.. c:type:: uv_write_t

    Write request type. Careful attention must be paid when reusing objects of
    this type. When a stream is in non-blocking mode, write requests sent
    with ``uv_write`` will be queued. Reusing objects at this point is undefined
    behaviour. It is safe to reuse the ``uv_write_t`` object only after the
    callback passed to ``uv_write`` is fired.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 35)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_read_cb)(uv_stream_t* stream, ssize_t nread, const uv_buf_t* buf)

    Callback called when data was read on a stream.

    `nread` is > 0 if there is data available or < 0 on error. When we've
    reached EOF, `nread` will be set to ``UV_EOF``. When `nread` < 0,
    the `buf` parameter might not point to a valid buffer; in that case
    `buf.len` and `buf.base` are both set to 0.

    .. note::
        `nread` might be 0, which does *not* indicate an error or EOF. This
        is equivalent to ``EAGAIN`` or ``EWOULDBLOCK`` under ``read(2)``.

    The callee is responsible for stopping/closing the stream when an error happens
    by calling :c:func:`uv_read_stop` or :c:func:`uv_close`. Trying to read
    from the stream again is undefined.

    The callee is responsible for freeing the buffer, libuv does not reuse it.
    The buffer may be a null buffer (where `buf->base` == NULL and `buf->len` == 0)
    on error.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 56)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_write_cb)(uv_write_t* req, int status)

    Callback called after data was written on a stream. `status` will be 0 in
    case of success, < 0 otherwise.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 61)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_connect_cb)(uv_connect_t* req, int status)

    Callback called after a connection started by :c:func:`uv_connect` is done.
    `status` will be 0 in case of success, < 0 otherwise.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 66)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_shutdown_cb)(uv_shutdown_t* req, int status)

    Callback called after a shutdown request has been completed. `status` will
    be 0 in case of success, < 0 otherwise.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 71)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_connection_cb)(uv_stream_t* server, int status)

    Callback called when a stream server has received an incoming connection.
    The user can accept the connection by calling :c:func:`uv_accept`.
    `status` will be 0 in case of success, < 0 otherwise.
```

Public members

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 81)

Unknown directive type "c:member".

```
.. c:member:: size_t uv_stream_t.write_queue_size

    Contains the amount of queued bytes waiting to be sent. Readonly.
```

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

Unknown directive type "c:member".

```
.. c:member:: uv_stream_t* uv_connect_t.handle

    Pointer to the stream where this connection request is running.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 89)

Unknown directive type "c:member".

```
.. c:member:: uv_stream_t* uv_shutdown_t.handle

    Pointer to the stream where this shutdown request is running.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)stream.rst, line 93)

Unknown directive type "c:member".

```
.. c:member:: uv_stream_t* uv_write_t.handle

    Pointer to the stream where this write request is running.
```

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

Unknown directive type "c:member".

```
.. c:member:: uv_stream_t* uv_write_t.send_handle

    Pointer to the stream being sent using this write request.
```

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

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)stream.rst, line 107)

Unknown directive type "c:function".

```
.. c:function:: int uv_shutdown(uv_shutdown_t* req, uv_stream_t* handle, uv_shutdown_cb cb)

    Shutdown the outgoing (write) side of a duplex stream. It waits for pending
    write requests to complete. The `handle` should refer to a initialized stream.
    `req` should be an uninitialized shutdown request struct. The `cb` is called
    after shutdown is complete.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst, line 114)

Unknown directive type "c:function".

```
.. c:function:: int uv_listen(uv_stream_t* stream, int backlog, uv_connection_cb cb)

    Start listening for incoming connections. `backlog` indicates the number of
    connections the kernel might queue, same as :man:`listen(2)`. When a new
    incoming connection is received the :c:type:`uv_connection_cb` callback is
    called.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst, line 121)

Unknown directive type "c:function".

```
.. c:function:: int uv_accept(uv_stream_t* server, uv_stream_t* client)

    This call is used in conjunction with :c:func:`uv_listen` to accept incoming
    connections. Call this function after receiving a :c:type:`uv_connection_cb`
    to accept the connection. Before calling this function the client handle must
    be initialized. < 0 return value indicates an error.

    When the :c:type:`uv_connection_cb` callback is called it is guaranteed that
    this function will complete successfully the first time. If you attempt to use
    it more than once, it may fail. It is suggested to only call this function once
    per :c:type:`uv_connection_cb` call.

    .. note::
        `server` and `client` must be handles running on the same loop.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst, line 136)

Unknown directive type "c:function".

```
.. c:function:: int uv_read_start(uv_stream_t* stream, uv_alloc_cb alloc_cb, uv_read_cb read_cb)

    Read data from an incoming stream. The :c:type:`uv_read_cb` callback will
    be made several times until there is no more data to read or
    :c:func:`uv_read_stop` is called.

    .. versionchanged:: 1.38.0 :c:func:`uv_read_start()` now consistently
      returns `UV_EALREADY` when called twice, and `UV_EINVAL` when the
      stream is closing. With older libuv versions, it returns `UV_EALREADY`
      on Windows but not UNIX, and `UV_EINVAL` on UNIX but not Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst, line 147)

Unknown directive type "c:function".

```
.. c:function:: int uv_read_stop(uv_stream_t*)

    Stop reading data from the stream. The :c:type:`uv_read_cb` callback will
    no longer be called.

    This function is idempotent and may be safely called on a stopped stream.

    This function will always succeed; hence, checking its return value is
    unnecessary. A non-zero return indicates that finishing releasing resources
    may be pending on the next input event on that TTY on Windows, and does not
    indicate failure.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst, line 159)

Unknown directive type "c:function".

```
.. c:function:: int uv_write(uv_write_t* req, uv_stream_t* handle, const uv_buf_t bufs[], unsigned int nbufs, uv_write_cb cb)

    Write data to stream. Buffers are written in order. Example:

    ::

        void cb(uv_write_t* req, int status) {
            /* Logic which handles the write result */
        }

        uv_buf_t a[] = {
            { .base = "1", .len = 1 },
            { .base = "2", .len = 1 }
        };

        uv_buf_t b[] = {
            { .base = "3", .len = 1 },
            { .base = "4", .len = 1 }
        };

        uv_write_t req1;
        uv_write_t req2;

        /* writes "1234" */
        uv_write(&req1, stream, a, 2, cb);
        uv_write(&req2, stream, b, 2, cb);

    .. note::
        The memory pointed to by the buffers must remain valid until the callback gets called.
        This also holds for :c:func:`uv_write2`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 190)

Unknown directive type "c:function".

```
.. c:function:: int uv_write2(uv_write_t* req, uv_stream_t* handle, const uv_buf_t bufs[], unsigned int nbufs, uv_stream_t* s

    Extended write function for sending handles over a pipe. The pipe must be
    initialized with `ipc` == 1.

    .. note::
        `send_handle` must be a TCP, pipe and UDP handle on Unix, or a TCP
        handle on Windows, which is a server or a connection (listening or
        connected state). Bound sockets or pipes will be assumed to be servers.
```

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

Unknown directive type "c:function".

```
.. c:function:: int uv_try_write(uv_stream_t* handle, const uv_buf_t bufs[], unsigned int nbufs)

    Same as :c:func:`uv_write`, but won't queue a write request if it can't be
    completed immediately.

    Will return either:

    * > 0: number of bytes written (can be less than the supplied buffer size).
    * < 0: negative error code (``UV_EAGAIN`` is returned if no data can be sent
      immediately).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 211)

Unknown directive type "c:function".

```
.. c:function:: int uv_try_write2(uv_stream_t* handle, const uv_buf_t bufs[], unsigned int nbufs, uv_stream_t* send_handle)

    Same as :c:func:`uv_try_write` and extended write function for sending
    handles over a pipe like c:func:`uv_write2`.

    Try to send a handle is not supported on Windows,
    where it returns ``UV_EAGAIN``.

    .. versionadded:: 1.42.0
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 221)

Unknown directive type "c:function".

```
.. c:function:: int uv_is_readable(const uv_stream_t* handle)

    Returns 1 if the stream is readable, 0 otherwise.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 225)

Unknown directive type "c:function".

```
.. c:function:: int uv_is_writable(const uv_stream_t* handle)

    Returns 1 if the stream is writable, 0 otherwise.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 229)

Unknown directive type "c:function".

```
.. c:function:: int uv_stream_set_blocking(uv_stream_t* handle, int blocking)

    Enable or disable blocking mode for a stream.

    When blocking mode is enabled all writes complete synchronously. The
    interface remains unchanged otherwise, e.g. completion or failure of the
    operation will still be reported through a callback which is made
    asynchronously.

    .. warning::
        Relying too much on this API is not recommended. It is likely to change
        significantly in the future.

        Currently only works on Windows for :c:type:`uv_pipe_t` handles.
        On UNIX platforms, all :c:type:`uv_stream_t` handles are supported.

        Also libuv currently makes no ordering guarantee when the blocking mode
        is changed after write requests have already been submitted. Therefore it is
        recommended to set the blocking mode immediately after opening or creating
        the stream.

    .. versionchanged:: 1.4.0 UNIX implementation added.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 252)

Unknown directive type "c:function".

```
.. c:function:: size_t uv_stream_get_write_queue_size(const uv_stream_t* stream)

    Returns `stream->write_queue_size`.

    .. versionadded:: 1.19.0
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) stream.rst`, line 258)

Unknown directive type "seealso".

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