:c:type:`uv_poll_t` --- Poll handle

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

Unknown interpreted text role "c:type".

Poll handles are used to watch file descriptors for readability, writability and disconnection similar to the purpose of :man:`poll(2)`.

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

Unknown interpreted text role "man".

The purpose of poll handles is to enable integrating external libraries that rely on the event loop to signal it about the socket status changes, like c-ares or libssh2. Using uv_poll_t for any other purpose is not recommended; :c:type:`uv_tcp_t`, :c:type:`uv_udp_t`, etc. provide an implementation that is faster and more scalable than what can be achieved with :c:type:`uv_poll_t`, especially on Windows.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst`, **line 10**); *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)poll.rst`, **line 10**); *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)poll.rst`, **line 10**); *backlink*

Unknown interpreted text role "c:type".

It is possible that poll handles occasionally signal that a file descriptor is readable or writable even when it isn't. The user should therefore always be prepared to handle EAGAIN or equivalent when it attempts to read from or write to the fd.

It is not okay to have multiple active poll handles for the same socket, this can cause libuv to busyloop or otherwise malfunction.

The user should not close a file descriptor while it is being polled by an active poll handle. This can cause the handle to report an error, but it might also start polling another socket. However the fd can be safely closed immediately after a call to :c:func:`uv_poll_stop` or :c:func:`uv_close`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst`, **line 25**); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst`, **line 25**); *backlink*

Unknown interpreted text role "c:func".

Note

On windows only sockets can be polled with poll handles. On Unix any file descriptor that would be accepted by :man:`poll(2)` can be used.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst`, **line 31**); *backlink*

Unknown interpreted text role "man".

Note

On AIX, watching for disconnection is not supported.

Data types

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

Unknown directive type "c:type".

```
.. c:type:: uv_poll_t

    Poll handle type.
```

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

Unknown directive type "c:type".

```
.. c:type:: void (*uv_poll_cb)(uv_poll_t* handle, int status, int events)

    Type definition for callback passed to :c:func:`uv_poll_start`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst, line 48)

Unknown directive type "c:type".

```
.. c:type:: uv_poll_event

    Poll event types

    ::

        enum uv_poll_event {
            UV_READABLE = 1,
            UV_WRITABLE = 2,
            UV_DISCONNECT = 4,
            UV_PRIORITIZED = 8
        };
```

Public members

N/A

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst, line 67)

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)poll.rst, line 73)

Unknown directive type "c:function".

```
.. c:function:: int uv_poll_init(uv_loop_t* loop, uv_poll_t* handle, int fd)

    Initialize the handle using a file descriptor.

    .. versionchanged:: 1.2.2 the file descriptor is set to non-blocking mode.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst, line 79)

Unknown directive type "c:function".

```
.. c:function:: int uv_poll_init_socket(uv_loop_t* loop, uv_poll_t* handle, uv_os_sock_t socket)

    Initialize the handle using a socket descriptor. On Unix this is identical
    to :c:func:`uv_poll_init`. On windows it takes a SOCKET handle.

    .. versionchanged:: 1.2.2 the socket is set to non-blocking mode.
```

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

Unknown directive type "c:function".

```
.. c:function:: int uv_poll_start(uv_poll_t* handle, int events, uv_poll_cb cb)

    Starts polling the file descriptor. `events` is a bitmask made up of
    `UV_READABLE`, `UV_WRITABLE`, `UV_PRIORITIZED` and `UV_DISCONNECT`. As soon
    as an event is detected the callback will be called with `status` set to 0,
    and the detected events set on the `events` field.

    The `UV_PRIORITIZED` event is used to watch for sysfs interrupts or TCP
    out-of-band messages.

    The `UV_DISCONNECT` event is optional in the sense that it may not be
    reported and the user is free to ignore it, but it can help optimize the
    shutdown path because an extra read or write call might be avoided.

    If an error happens while polling, `status` will be < 0 and corresponds
    with one of the `UV_E*` error codes (see :ref:`errors`). The user should
    not close the socket while the handle is active. If the user does that
    anyway, the callback *may* be called reporting an error status, but this is
    **not** guaranteed.

    .. note::
        Calling :c:func:`uv_poll_start` on a handle that is already active is
        fine. Doing so will update the events mask that is being watched for.

    .. note::
        Though `UV_DISCONNECT` can be set, it is unsupported on AIX and as such
        will not be set on the `events` field in the callback.

    .. note::
        If one of the events `UV_READABLE` or `UV_WRITABLE` are set, the
        callback will be called again, as long as the given fd/socket remains
        readable or writable accordingly. Particularly in each of the following
        scenarios:

        * The callback has been called because the socket became
          readable/writable and the callback did not conduct a read/write on
          this socket at all.
        * The callback committed a read on the socket, and has not read all the
          available data (when `UV_READABLE` is set).
        * The callback committed a write on the socket, but it remained
          writable afterwards (when `UV_WRITABLE` is set).
        * The socket has already became readable/writable before calling
          :c:func:`uv_poll_start` on a poll handle associated with this socket,
          and since then the state of the socket did not changed.

        In all of the above listed scenarios, the socket remains readable or
        writable and hence the callback will be called again (depending on the
        events set in the bitmask). This behaviour is known as level
        triggering.

    .. versionchanged:: 1.9.0 Added the `UV_DISCONNECT` event.
    .. versionchanged:: 1.14.0 Added the `UV_PRIORITIZED` event.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst, line 139)

Unknown directive type "c:function".

```
.. c:function:: int uv_poll_stop(uv_poll_t* poll)
```

```
Stop polling the file descriptor, the callback will no longer be called.

.. note::
    Calling :c:func:`uv_poll_stop` is effective immediately: any pending
    callback is also canceled, even if the socket state change notification
    was already pending.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)poll.rst, line 148)

Unknown directive type "seealso".

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