

## :c:type:`uv\_pipe\_t` --- Pipe handle

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

Unknown interpreted text role "c:type".

Pipe handles provide an abstraction over streaming files on Unix (including local domain sockets, pipes, and FIFOs) and named pipes on Windows.

:c:type:`uv\_pipe\_t` is a 'subclass' of :c:type:`uv\_stream\_t`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.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)pipe.rst, line 10); [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)pipe.rst, line 16)**

Unknown directive type "c:type".

```
.. c:type:: uv_pipe_t

    Pipe handle type.
```

## Public members

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 24)**

Unknown directive type "c:member".

```
.. c:member:: int uv_pipe_t.ipc

    Whether this pipe is suitable for handle passing between processes.
    Only a connected pipe that will be passing the handles should have this flag
    set, not the listening pipe that uv_accept is called on.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 30)**

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_stream_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)pipe.rst, line 36)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_init(uv_loop_t* loop, uv_pipe_t* handle, int ipc)

    Initialize a pipe handle. The `ipc` argument is a boolean to indicate if
    this pipe will be used for handle passing between processes (which may
    change the bytes on the wire). Only a connected pipe that will be
    passing the handles should have this flag set, not the listening pipe
    that uv_accept is called on.
```

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

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_open(uv_pipe_t* handle, uv_file file)

    Open an existing file descriptor or HANDLE as a pipe.

.. versionchanged:: 1.2.1 the file descriptor is set to non-blocking mode.

.. note::
    The passed file descriptor or HANDLE is not checked for its type, but
    it's required that it represents a valid pipe.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 54)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_bind(uv_pipe_t* handle, const char* name)

    Bind the pipe to a file path (Unix) or a name (Windows).

.. note::
    Paths on Unix get truncated to ``sizeof(sockaddr_un.sun_path)`` bytes, typically between
    92 and 108 bytes.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 62)**

Unknown directive type "c:function".

```
.. c:function:: void uv_pipe_connect(uv_connect_t* req, uv_pipe_t* handle, const char* name, uv_connect_t* req)

    Connect to the Unix domain socket or the named pipe.

.. note::
    Paths on Unix get truncated to ``sizeof(sockaddr_un.sun_path)`` bytes, typically between
    92 and 108 bytes.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 70)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_getsockname(const uv_pipe_t* handle, char* buffer, size_t* size)

    Get the name of the Unix domain socket or the named pipe.

    A preallocated buffer must be provided. The size parameter holds the length
    of the buffer and it's set to the number of bytes written to the buffer on
    output. If the buffer is not big enough ``UV_ENOBUFS`` will be returned and
    len will contain the required size.

.. versionchanged:: 1.3.0 the returned length no longer includes the terminating null byte,
    and the buffer is not null terminated.
```

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

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_getpeername(const uv_pipe_t* handle, char* buffer, size_t* size)

    Get the name of the Unix domain socket or the named pipe to which the handle
    is connected.

    A preallocated buffer must be provided. The size parameter holds the length
    of the buffer and it's set to the number of bytes written to the buffer on
    output. If the buffer is not big enough ``UV_ENOBUFS`` will be returned and
    len will contain the required size.

.. versionadded:: 1.3.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 94)**

Unknown directive type "c:function".

```
.. c:function:: void uv_pipe_pending_instances(uv_pipe_t* handle, int count)

    Set the number of pending pipe instance handles when the pipe server is
    waiting for connections.
```

```
.. note::
    This setting applies to Windows only.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 102)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_pending_count(uv_pipe_t* handle)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 103)**

Unknown directive type "c:function".

```
.. c:function:: uv_handle_type uv_pipe_pending_type(uv_pipe_t* handle)
```

Used to receive handles over IPC pipes.

First - call `:c:func:`uv_pipe_pending_count``, if it's > 0 then initialize a handle of the given ``type``, returned by `:c:func:`uv_pipe_pending_type`` and call ``uv_accept(pipe, handle)``.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 111)**

Unknown directive type "seealso".

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

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src)pipe.rst, line 113)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe_chmod(uv_pipe_t* handle, int flags)
```

Alters pipe permissions, allowing it to be accessed from processes run by different users. Makes the pipe writable or readable by all users. Mode can be ``UV_WRITABLE``, ``UV_READABLE`` or ``UV_WRITABLE | UV_READABLE``. This function is blocking.

```
.. 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)pipe.rst, line 122)**

Unknown directive type "c:function".

```
.. c:function:: int uv_pipe(uv_file fds[2], int read_flags, int write_flags)
```

Create a pair of connected pipe handles.

Data may be written to ``fds[1]`` and read from ``fds[0]``.

The resulting handles can be passed to ``uv_pipe_open``, used with ``uv_spawn``, or for any other purpose.

Valid values for ``flags`` are:

- `UV_NONBLOCK_PIPE`: Opens the specified socket handle for ``OVERLAPPED`` or ``FIONBIO`/`O_NONBLOCK`` I/O usage. This is recommended for handles that will be used by libuv, and not usually recommended otherwise.

Equivalent to `:man:`pipe(2)`` with the ``O_CLOEXEC`` flag set.

```
.. versionadded:: 1.41.0
```