

MessagePortMain

`MessagePortMain` is the main-process-side equivalent of the DOM [MessagePort](#) object. It behaves similarly to the DOM version, with the exception that it uses the Node.js `EventEmitter` event system, instead of the DOM `EventTarget` system. This means you should use `port.on('message', ...)` to listen for events, instead of `port.onmessage = ...` or `port.addEventListener('message', ...)`

See the [Channel Messaging API](#) documentation for more information on using channel messaging.

`MessagePortMain` is an `[EventEmitter][event-emitter]`.

Class: MessagePortMain

Port interface for channel messaging in the main process.

Process: [Main](#)

This class is not exported from the `'electron'` module. It is only available as a return value of other methods in the Electron API.

Instance Methods

`port.postMessage(message, [transfer])`

- `message` any
- `transfer` `MessagePortMain[]` (optional)

Sends a message from the port, and optionally, transfers ownership of objects to other browsing contexts.

`port.start()`

Starts the sending of messages queued on the port. Messages will be queued until this method is called.

`port.close()`

Disconnects the port, so it is no longer active.

Instance Events

Event: 'message'

Returns:

- `messageEvent` Object
 - `data` any
 - `ports` `MessagePortMain[]`

Emitted when a `MessagePortMain` object receives a message.

Event: 'close'

Emitted when the remote end of a `MessagePortMain` object becomes disconnected.