

«interface»

Task

- handle: std::thread
- is_alive: std::atomic<bool> = false
- is_muted: std::atomic<bool> = true
- events_in: std::queue<Event>
- events_out: std::queue<Event>&
- mtx_events_in: std::mutex
- mtx_events_out: std::mutex&

+ start(events_out&, mutex_out&): bool

+ kill()

+ mute()

+ unmute()

+ sendEventToTask(e : Event)

sendEventFromTask(e : Event)

readLatestEvent(blocking : bool): Event*

init(): bool {abstract}

run(): void = 0 {abstract}

start() creates an std::thread which continuously invokes the run() method.

a muted task can still call sendEvent(), but these are not sent.