

EventPipe Class Reference

```
#include <libcom3.h>
```

Public Member Functions	
	EventPipe (event::EventType type)
	~EventPipe ()
	EventPipe (const EventPipe &eventPipe)
void	operator= (const EventPipe &eventPipe)
enum status::PipeStatus	recv (vfiipc::JObject &out, Milliseconds millis)
enum status::PipeStatus	recv (vfiipc::JObject &out, Milliseconds millis, int interruptFD)

Constructor & Destructor Documentation

◆ [EventPipe\(\)](#) [1/2]

EventPipe	(event::EventType	type)	
---------------------------	---	----------------------------------	------	---	--

◆ [~EventPipe\(\)](#)

~EventPipe	()	
----------------------------	---	---	--

◆ [EventPipe\(\)](#) [2/2]

EventPipe	(const EventPipe &	<i>eventPipe</i>)	
---------------------------	---	-----------------------------------	------------------	---	--

Member Function Documentation

◆ [operator=\(\)](#)

void operator=	(const EventPipe &	<i>eventPipe</i>)	
----------------	---	-----------------------------------	------------------	---	--

◆ [recv\(\)](#) [1/2]

enum status::PipeStatus recv	(vfiipc::JObject &	<i>out</i> ,
--	---	-----------------------------------	--------------

		Milliseconds	<i>millis</i>
)		

Read the next event If the message is interrupted (return code is ERR_USER_INTERRUPT or ERR_TIMEOUT) the [EventPipe](#) is still valid and can be used again.

Parameters

[out]	out	Response. Will be changed only if return is OK
[in]	millis	Milliseconds to wait for an event

Returns

OK in case of success, else failure code

◆ **recv() [2/2]**

enum status::PipeStatus recv	(vfiipc::JObject &	<i>out,</i>
		Milliseconds	<i>millis,</i>
		int	<i>interruptFD</i>
)		

Read the next event If the message is interrupted (return code is ERR_USER_INTERRUPT or ERR_TIMEOUT) the [EventPipe](#) is still valid and can be used again.

Parameters

[out]	out	Response. Will be changed only if return is OK
[in]	millis	Milliseconds to wait for an event
[in]	interruptFD	FD to listen for activity. Once activity is detected the function will return with error code ERR_USER_INTERRUPT

Returns

OK in case of success, else failure code

The documentation for this class was generated from the following file:

- com/src/[libcom3.h](#)