

# Copy Service



[Main Page](#)[Classes ▾](#)[Files ▾](#)

## Copy Service Documentation

Generated by [doxygen](#) 1.8.18

## Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

 <b>IPCmessageToClient</b>	Struct contains the id of the current job, the status and a list of other jobs
 <b>IPCmessageToDaemon</b>	Struct contains the source, destination, the task and the type of job

## IPCmessageToClient Struct Reference

Struct contains the id of the current job, the status and a list of other jobs. [More...](#)

```
#include <IPC.h>
```

### Public Attributes

int [jobList](#) [1024]

int [jobId](#)

float [status](#)

### Detailed Description

Struct contains the id of the current job, the status and a list of other jobs.

#### Parameters

**jobList** Integer array with len of 1024.

**jobId** Integer, tells us the id / thread id.

**status** Float, the status of the current job.

## IPCmessageToDaemon Struct Reference

Struct contains the source, destination, the task and the type of job. [More...](#)

```
#include <IPC.h>
```

### Public Attributes

char	<a href="#">source</a>	[1024]
char	<a href="#">destination</a>	[1024]
char	<a href="#">task</a>	[1024]
int	<a href="#">jobID</a>	

### Detailed Description

Struct contains the source, destination, the task and the type of job.

#### Parameters

- source** Char array with len of 1024, contains the source from where it should copy.
- destination** Char array with len of 1024, contains the destination to where it should copy.
- task** Char array with len of 1024, tells the daemon what job it should accomplish: -"copy" = tells the daemon to start a copy tipe job -"status" = gets the status of a current job -"stop" = stops a job -"suspent" = pauses / suspends a job -"resume" = resumes a job
- .

#### Warning

A wrong typed task will result in the daemon doing nothing.

Parameters

**jobID** Integer, tells us the thread id.

Member Data Documentation

◆ destination

char IPCMessageToDaemon::destination[1024]

◆ jobID

int IPCMessageToDaemon::jobID

◆ source

char IPCMessageToDaemon::source[1024]

◆ task

char IPCMessageToDaemon::task[1024]

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

- [IPC.h](#)

# Copy Service









[Main Page](#)[Classes ▾](#)[Files ▾](#)

Here is a list of all class members with links to the classes they belong to:

- destination : [IPCmessageToDaemon](#)
- jobId : [IPCmessageToClient](#)
- jobId : [IPCmessageToDaemon](#)
- jobList : [IPCmessageToClient](#)
- source : [IPCmessageToDaemon](#)
- status : [IPCmessageToClient](#)
- task : [IPCmessageToDaemon](#)

## File List

Here is a list of all files with brief descriptions:

 <a href="#">client.c</a>	
 <a href="#">client.h</a>	Makes connection between user and back-end (Daemon)
 <a href="#">daemon.c</a>	
 <a href="#">daemon.h</a>	
 <a href="#">IPC.c</a>	
 <a href="#">IPC.h</a>	Inter Process Communication Library
 <a href="#">threadService.c</a>	
 <a href="#">threadService.h</a>	Thread Service Library

## client.h File Reference

Makes connection between user and back-end (Daemon). [More...](#)

[Go to the source code of this file.](#)

### Functions

int [main](#) ()

### Detailed Description

Makes connection between user and back-end (Daemon).

Receives messages from user (input)

Sends to daemon a message which contains the requirement (writing a copy job, etc).

### Function Documentation

#### ◆ [main\(\)](#)

[main](#) ( )

#### Parameters

None

#### Returns

0



# daemon.h File Reference

Daemon runs as a background process. [More...](#)

[Go to the source code of this file.](#)

## Functions

int [main](#) ()

## Detailed Description

Daemon runs as a background process.

Receives messages from client

Starts SEPARATED threads for each copy job.

## Function Documentation

### ◆ main()

int main ( )

#### Parameters

None

#### Returns

0

## IPC.h File Reference

Inter Process Communication Library. [More...](#)

```
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
```

[Go to the source code of this file.](#)

## Classes

struct	<a href="#">IPCmessageToDaemon</a>	Struct contains the source, destination, the task and the type of job. <a href="#">More...</a>
struct	<a href="#">IPCmessageToClient</a>	Struct contains the id of the current job, the status and a list of other jobs. <a href="#">More...</a>

## Typedefs

typedef struct	<a href="#">IPCmessageToDaemon</a>	<a href="#">IPCmessageToDaemon</a>
typedef struct	<a href="#">IPCmessageToClient</a>	<a href="#">IPCmessageToClient</a>

## Functions

void	<a href="#">IPC_ClientSendMessage</a>	( <a href="#">IPCmessageToDaemon</a> msg)
<a href="#">IPCmessageToDaemon</a>	<a href="#">IPC_DaemonReceiveMessage</a>	()
		Receive the <a href="#">IPCmessageToDaemon</a> struct. <a href="#">More...</a>

## Functions

	void	<b>IPC_ClientSendMessage</b> ( <b>IPCmessageToDaemon</b> msg)
<b>IPCmessageToDaemon</b>		<b>IPC_DaemonReceiveMessage</b> () Receive the <b>IPCmessageToDaemon</b> struct. <a href="#">More...</a>
	void	<b>IPC_DaemonSendMessage</b> ( <b>IPCmessageToClient</b> msg)
		Send the <b>IPCmessageToClient</b> struct to the Client. <a href="#">More...</a>
<b>IPCmessageToClient</b>		<b>IPC_ClientReceiveMessage</b> () Receive the <b>IPCmessageToClient</b> struct. <a href="#">More...</a>

## Detailed Description

Inter Process Communication Library.

Inter-Process communication library that allows the daemon and the client to send and receive messages in a defined way.

## Typedef Documentation

### ◆ IPCmessageToClient

typedef struct **IPCmessageToClient** **IPCmessageToClient**

### ◆ IPCmessageToDaemon

typedef struct **IPCmessageToDaemon** **IPCmessageToDaemon**

◆ IPC\_ClientReceiveMessage()

**IPCmessageToClient** IPC\_ClientReceiveMessage ( )

Receive the **IPCmessageToClient** struct.

**Parameters**

**None**

**Returns**

The message from Client

**Warning**

Use only in Clients

◆ IPC\_ClientSendMessage()

void IPC\_ClientSendMessage ( **IPCmessageToDaemon** msg )

◆ IPC\_DaemonReceiveMessage()

IPC\_DaemonReceiveMessage ( )

Receive the **IPCmessageToDaemon** struct.

**Parameters**

**None**

## ◆ IPC\_DaemonReceiveMessage()

IPC\_DaemonReceiveMessage ( )

Receive the **IPCmessageToDaemon** struct.

### Parameters

**None**

### Returns

The message from Deamon

### Warning

Use only in Daemon

## ◆ IPC\_DaemonSendMessage()

void IPC\_DaemonSendMessage ( **IPCmessageToClient** msg )

Send the **IPCmessageToClient** struct to the Client.

### Parameters

**msg** The message to be sent to the Client

### Returns

void

### Warning

Use only in Daemon

## threadService.h File Reference

Thread Service Library. [More...](#)

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <errno.h>
#include <sys/stat.h>
#include <string.h>
#include <signal.h>
#include "IPC.h"
```

[Go to the source code of this file.](#)

## Functions

void **stopThread** (int index)  
Stops a thread found at position index in threads array. [More...](#)

int **copyThread (IPCmessageToDaemon \*client\_message)**  
Starts a new copy job on a new thread;. [More...](#)

void **pauseThread** (int jobID)  
Pauses a thread with the idex / jobID. [More...](#)

float **statusThread** (int jobID)  
Gets the progress of a job. [More...](#)

void **resumeThread** (int jobID)  
Resumes a job that has been paused. [More...](#)

## Variables

pthread_t	<b>threads</b> [1024] Array of threads, used to store thee jobs on independent threads. <a href="#">More...</a>
int	<b>maxThreads</b> The maximum number of threads that can run in the same time. Configurable. <a href="#">More...</a>
int	<b>pause_status</b> [1024] Array of ints that store the status of the thread (running/paused/stopped). <a href="#">More...</a>
float	<b>progress</b> [1024] Array of floats that store the progress of the thread (from 0=no progress to 1=done) <a href="#">More...</a>
pthread_mutex_t	<b>mtx</b> General mutex, for all type of jobs. <a href="#">More...</a>
pthread_mutex_t	<b>copyThreadMutex</b> Mutex for the copy jobs. <a href="#">More...</a>
pthread_mutex_t	<b>pauseMutex</b> Mutex got the pause jobs. <a href="#">More...</a>
pthread_mutex_t	<b>progressMutex</b> Mutex for the jobs progress. <a href="#">More...</a>

## Detailed Description

Thread Service Library.

Thread Service is a Library that allows the start, pause, resume and cancel of a copy job.

### ◆ copyThread()

```
int copyThread ( IPCmessageToDaemon * client_message )
```

Starts a new copy job on a new thread;.

#### Parameters

**client\_message** Informations got form the client like from where to where to copy.

#### Returns

Integer, is the index of the new thread. Needs to be stored in order to know what thread it is.

### ◆ pauseThread()

```
void pauseThread ( int jobID )
```

Pauses a thread with the idex / jobID.

#### Parameters

**jobID** Integer, the job that needs to be paused

#### Returns

void



### ◆ resumeThread()

```
void resumeThread ( int jobId )
```

Resumes a job that has been paused.

#### Parameters

**jobID** Integer, the job that needs to be resumed

#### Returns

void

### ◆ statusThread()

```
float statusThread ( int jobId )
```

Gets the progress of a job.

#### Parameters

**jobID** Integer, the job that i want to know it's progress

#### Returns

Float, the progress of the job

### ◆ stopThread()

```
void stopThread ( int index )
```

Stops a thread found at position index in threads array.

## ◆ stopThread()

```
void stopThread ( int index )
```

Stops a thread found at position index in threads array.

### Parameters

**index** Index of the thread that needs to be stopped

### Returns

void

## Variable Documentation

## ◆ copyThreadMutex

```
pthread_mutex_t copyThreadMutex
```

Mutex for the copy jobs.

## ◆ maxThreads

```
int maxThreads
```

The maximum number of threads that can run in the same time. Configurable.

#### ◆ mtx

```
pthread_mutex_t mtx
```

General mutex, for all type of jobs.

#### ◆ pause\_status

```
int pause_status[1024]
```

Array of ints that store the status of the thread (running/paused/stopped).

#### ◆ pauseMutex

```
pthread_mutex_t pauseMutex
```

Mutex got the pause jobs.

#### ◆ progress

```
float progress[1024]
```

Array of floats that store the progress of the thread (from 0=no progress to 1=done)

### ◆ progressMutex

`pthread_mutex_t progressMutex`

Mutex for the jobs progress.

### ◆ threads

`pthread_t threads[1024]`

Array of threads, used to store thee jobs on independent threads.

# Copy Service

[Main Page](#)[Classes ▾](#)[Files ▾](#)

Here is a list of all file members with links to the files they belong to:

- c -

- config() : [threadService.c](#)
- copy() : [threadService.c](#)
- copyThread() : [threadService.h](#) , [threadService.c](#)
- copyThreadMutex : [threadService.h](#)

- i -

- IPC\_ClientReceiveMessage() : [IPC.h](#)
- IPC\_ClientReciveMessage() : [IPC.c](#)
- IPC\_ClientSendMessage() : [IPC.h](#) , [IPC.c](#)
- IPC\_DaemonReceiveMessage() : [IPC.h](#)
- IPC\_DaemonReciveMessage() : [IPC.c](#)
- IPC\_DaemonSendMessage() : [IPC.c](#) , [IPC.h](#)
- IPCmessageToClient : [IPC.h](#)
- IPCmessageToDaemon : [IPC.h](#)

- m -

- main() : [client.c](#) , [client.h](#) , [daemon.h](#) , [daemon.c](#)
- maxThreads : [threadService.h](#)
- mtx : [threadService.h](#)

- p -

- pause\_status : [threadService.h](#)
- pauseMutex : [threadService.h](#)
- pauseThread() : [threadService.h](#) , [threadService.c](#)
- progress : [threadService.h](#)

- IPC\_ClientReceiveMessage() : **IPC.h**
- IPC\_ClientRecvMessage() : **IPC.c**
- IPC\_ClientSendMessage() : **IPC.h** , **IPC.c**
- IPC\_DaemonReceiveMessage() : **IPC.h**
- IPC\_DaemonRecvMessage() : **IPC.c**
- IPC\_DaemonSendMessage() : **IPC.c** , **IPC.h**
- IPCmessageToClient : **IPC.h**
- IPCmessageToDaemon : **IPC.h**

#### - m -

- main() : **client.c** , **client.h** , **daemon.h** , **daemon.c**
- maxThreads : **threadService.h**
- mtx : **threadService.h**

#### - p -

- pause\_status : **threadService.h**
- pauseMutex : **threadService.h**
- pauseThread() : **threadService.h** , **threadService.c**
- progress : **threadService.h**
- progressMutex : **threadService.h**

#### - r -

- resumeThread() : **threadService.c** , **threadService.h**

#### - s -

- statusThread() : **threadService.c** , **threadService.h**
- stopThread() : **threadService.c** , **threadService.h**

#### - t -

- threads : **threadService.h**

# Copy Service

[Main Page](#)[Classes ▾](#)[Files ▾](#)[File List](#)[File Members ▸](#)[All ▸](#)[Functions](#)[Variables](#)[Typedefs](#)

- `config()` : [threadService.c](#) , [threadService.h](#)
- `copy()` : [threadService.c](#) , [threadService.h](#)
- `copyThread()` : [threadService.c](#) , [threadService.h](#)
- `IPC_ClientReceiveMessage()` : [IPC.h](#)
- `IPC_ClientReciveMessage()` : [IPC.c](#)
- `IPC_ClientSendMessage()` : [IPC.c](#) , [IPC.h](#)
- `IPC_DaemonReceiveMessage()` : [IPC.h](#)
- `IPC_DaemonReciveMessage()` : [IPC.c](#)
- `IPC_DaemonSendMessage()` : [IPC.c](#) , [IPC.h](#)
- `main()` : [client.c](#) , [client.h](#) , [daemon.c](#) , [daemon.h](#)
- `pauseThread()` : [threadService.c](#) , [threadService.h](#)
- `resumeThread()` : [threadService.c](#) , [threadService.h](#)
- `statusThread()` : [threadService.c](#) , [threadService.h](#)
- `stopThread()` : [threadService.c](#) , [threadService.h](#)

# Copy Service

[Main Page](#)[Classes ▾](#)[Files ▾](#)[File List](#)[File Members ▸](#)[All ▸](#)[Functions](#)[Variables](#)[Typedefs](#)

- copyThreadMutex : [threadService.h](#)
- maxThreads : [threadService.h](#)
- mtx : [threadService.h](#)
- pause\_status : [threadService.h](#)
- pauseMutex : [threadService.h](#)
- progress : [threadService.h](#)
- progressMutex : [threadService.h](#)
- threads : [threadService.h](#)



# Copy Service

Main Page

Classes ▾

Files ▾

Search

File List

File Members ▶

All ▶

Functions

Variables

**Typedefs**

- IPCmessageToClient
- IPCmessageToDaem

Generated by **doxygen** 1.8.18