PINC

Generated by Doxygen 1.8.10

Tue Oct 27 2015 16:23:59

Contents

1 File Index **1**

1.1 File List 1

2 File Documentation **3**

2.1 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/aux.c File Reference 3

2.1.1 Detailed Description 3

2.1.2 Function Documentation 3

2.1.2.1 msg(msg_kind kind, const char *restrict format,...) 3

2.2 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/input.c File Reference 4

2.2.1 Detailed Description 4

2.2.2 Function Documentation 5

2.2.2.1 free_strarr(char **strarr) 5

2.2.2.2 list_getnelements(const char *list) 5

2.2.2.3 list_to_strarr(const char *list) 5

2.2.2.4 listparser_getallocdouble(const dictionary *d, const char *key, int *count) 5

2.2.2.5 listparser_getallocint(const dictionary *d, const char *key, int *count) 6

2.2.2.6 listparser_getdouble(const dictionary *d, const char *key, double *result) 6

2.2.2.7 listparser_getint(const dictionary *d, const char *key, int *result) 6

2.2.2.8 listparser_getnelements(const dictionary *d, const char *key) 7

2.2.2.9 parse_input(int argc, char *argv[]) 7

2.3 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/main.c File Reference 7

2.3.1 Detailed Description 8

2.4 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/pinc.h File Reference 8

2.4.1 Detailed Description 8

2.4.2 Function Documentation 9

2.4.2.1 msg(msg_kind kind, const char *restrict format,...) 9

2.4.2.2 parse_input(int argc, char *argv[]) 9

Index **11**

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/aux.c
PINC auxiliary functions 3
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/input.c
PINC main routine 4
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/main.c
PINC main routine 7
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/pinc.h
PINC main routine 8

Chapter 2

File Documentation

2.1 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/aux.c File Reference

PINC auxiliary functions.

```
#include <stdlib.h>
#include <stdio.h>
#include <stdarg.h>
#include "pinc.h"
#include <string.h>
```

Functions

- void msg (msg_kind kind, const char *restrict format,...)
  The PINC equivalent of printf().

2.1.1 Detailed Description

PINC auxiliary functions.

Author

Sigvald Marholm sigvaldm@fys.uio.no

Copyright

University of Oslo, Norway

Date

11.10.15

Small auxiliary functions used throughout the PINC program.

2.1.2 Function Documentation

2.1.2.1 void msg (msg_kind *kind*, const char *restrict *format*, ...)

The PINC equivalent of printf().

kind STATUS, WARNING or ERROR depending on what to output.

Parameters

format	Error message with specification of how to interpret data.
...	Data to be interpreted in message.

Returns

void

This is the way to output information from PINC. Similar syntax to printf(). In the case of an ERROR, the program is terminated. Appends end-of-line automatically at the end.

2.2 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/input.c File Reference

PINC main routine.

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "iniparser.h"
#include "pinc.h"
```

Functions

- static char ** list_to_strarr (const char *list)

  Splits a comma-separated list-string to an array of strings.
- static void free_strarr (char **strarr)

  Frees dynamically allocated NULL-terminated array of strings.
- static int list_getnelements (const char *list)

  Gets number of elements in comma-separated list-string.
- static int listparser_getnelements (const dictionary *d, const char *key)

  Gets number of elements in comma-separated entry in ini file.
- static void listparser_getdouble (const dictionary *d, const char *key, double *result)

  Get the array of doubles associated to a key in ini file.
- static double * listparser_getallocdouble (const dictionary *d, const char *key, int *count)

  Get the array of doubles associated to a key in ini file.
- static void listparser_getint (const dictionary *d, const char *key, int *result)

  Get the array of integers associated to a key in ini file.
- static int * listparser_getallocint (const dictionary *d, const char *key, int *count)

  Get the array of integers associated to a key in ini file.
- static void **fprintarr** (FILE *stream, const char *restrict format, void *arr, int count)
- void parse_input (int argc, char *argv[])

  Parse PINC's input argument and input file.

2.2.1 Detailed Description

PINC main routine.

Author

Sigvald Marholm sigvaldm@fys.uio.no

Copyright

University of Oslo, Norway

Date

13.10.15

Functions for parsing input to PINC. Replaces old DiP3D input.c file by Wojciech Jacek Miloch.

2.2.2 Function Documentation

2.2.2.1 static void free_strarr (char ** *strarr*) [static]

Frees dynamically allocated NULL-terminated array of strings.

Parameters

strarr	Pointer to array of strings

Returns

void

2.2.2.2 static int list_getnelements (const char * *list*) [static]

Gets number of elements in comma-separated list-string.

Parameters

list	Comma-separated list

Returns

Number of elements in list. 0 if string is empty.

2.2.2.3 static char ** list_to_strarr (const char * *list*) [static]

Splits a comma-separated list-string to an array of strings.

Parameters

list	Comma-sepaprated list

Returns

A NULL-terminated array of NULL-terminated strings

Example: when str="abc ,def, ghi", list2strarr(str); will return an array arr such that :

arr[0]="abc" arr[1]="def" arr[2]="ghi" arr[3]=NULL

Note that whitespaces are trimmed away. Remember to free string array. This can be done with free_strarr().

2.2.2.4 static double * listparser_getallocdouble (const dictionary * *d*, const char * *key*, int * *count*) [static]

Get the array of doubles associated to a key in ini file.

Parameters

	d	Dictionary to search
	key	Key string to look for in iniparser dictionary
out	*count*	Number of elements in returned array

Returns

Array of doubles

This function can be seen as an extension to iniparser in order to handle comma-separated entries. It is syntactically similar to the functions in iniparser.

listparser_getdouble() is similar to this but does not allocate memory.

2.2.2.5 static int * listparser_getallocint (const dictionary * *d*, const char * *key*, int * *count*) [static]

Get the array of integers associated to a key in ini file.

Parameters

	d	Dictionary to search
	key	Key string to look for in iniparser dictionary
out	*count*	Number of elements in returned array

Returns

Array of integers

This function can be seen as an extension to iniparser in order to handle comma-separated entries. It is syntactically similar to the functions in iniparser.

listparser_getint() is similar to this but does not allocate memory.

2.2.2.6 static void listparser_getdouble (const dictionary * *d*, const char * *key*, double * *result*) [static]

Get the array of doubles associated to a key in ini file.

Parameters

	d	Dictionary to search
	key	Key string to look for in iniparser dictionary
out	*result*	Pointer to pre-allocated array to store results in

Returns

void

This function can be seen as an extension to iniparser in order to handle comma-separated entries. It is syntactically similar to the functions in iniparser.

Array of results must be allocated before being passed to this function. The number of elements to allocate can be obtained by listparser_getnelements().

listparser_getallocdouble() is similar to this but allocates memory.

2.2.2.7 static void listparser_getint (const dictionary * *d*, const char * *key*, int * *result*) [static]

Get the array of integers associated to a key in ini file.

Parameters

d	Dictionary to search
key	Key string to look for in iniparser dictionary
result	Pointer to pre-allocated array to store results in

Returns

void

This function can be seen as an extension to iniparser in order to handle comma-separated entries. It is syntactically similar to the functions in iniparser.

Array of results must be allocated before being passed to this function. The number of elements to allocate can be obtained by listparser_getnelements().

2.2.2.8 static int listparser_getnelements (const dictionary * *d*, const char * *key*) [static]

Gets number of elements in comma-separated entry in ini file.

Parameters

d	Dictionary to search
key	Key string to look for in iniparser dictionary

Returns

Number of elements in entry. 0 if entry does not exist.

This function can be seen as an extension to iniparser in order to handle comma-separated entries. It is syntactically similar to the functions in iniparser.

2.2.2.9 void parse_input (int *argc*, char * *argv[]*)

Parse PINC's input argument and input file.

Parameters

argc	Argument count
argv	Argument vector

Returns

void

This function performs a sanity check of argc and argv and reads the specified input file. It performs the necessary sanity checks of its data and stores the values. It also computes derived values. In case of failure it prints an error to stderr and terminates PINC.

2.3 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/main.c File Reference

PINC main routine.

```
#include <stdlib.h>
#include <stdio.h>
#include "pinc.h"
```

Functions

- int **main** (int argc, char ∗argv[ ])

2.3.1 Detailed Description

PINC main routine.

Author

Sigvald Marholm sigvaldm@fys.uio.no

Copyright

University of Oslo, Norway

Date

08.10.15

Main routine for PINC (Particle-IN-Cell). Replaces old DiP3D main.c file by Wojciech Jacek Miloch.

2.4 /home/gullik/Documents/PiC/mn-fysrp-pic/DiP3D/src/pinc.h File Reference

PINC main routine.

Enumerations

- enum **msg_kind** { **STATUS**, **WARNING**, **ERROR** }

Functions

- void parse_input (int argc, char ∗argv[ ])
  Parse PINC's input argument and input file.
- void msg (msg_kind kind, const char ∗restrict format,...)
  The PINC equivalent of printf().

2.4.1 Detailed Description

PINC main routine.

Author

Sigvald Marholm sigvaldm@fys.uio.no

Copyright

University of Oslo, Norway

Date

11.10.15

PINC main header file. Holding the central function declarations comprising the PINC program.

2.4.2 Function Documentation

2.4.2.1 void msg (msg_kind *kind,* const char ∗restrict *format, ...*)

The PINC equivalent of printf().

kind STATUS, WARNING or ERROR depending on what to output.

Parameters

format	Error message with specification of how to interpret data.
...	Data to be interpreted in message.

Returns

void

This is the way to output information from PINC. Similar syntax to printf(). In the case of an ERROR, the program is terminated. Appends end-of-line automatically at the end.

2.4.2.2 void parse_input (int *argc,* char ∗ *argv[ ]*)

Parse PINC's input argument and input file.

Parameters

argc	Argument count
argv	Argument vector

Returns

void

This function performs a sanity check of argc and argv and reads the specified input file. It performs the necessary sanity checks of its data and stores the values. It also computes derived values. In case of failure it prints an error to stderr and terminates PINC.

Index

/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3↩
D/src/aux.c, 3
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3↩
D/src/input.c, 4
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3↩
D/src/main.c, 7
/home/gullik/Documents/PiC/mn-fysrp-pic/DiP3↩
D/src/pinc.h, 8

aux.c
msg, 3

free_strarr
input.c, 5

input.c
free_strarr, 5
list_getnelements, 5
list_to_strarr, 5
listparser_getallocdouble, 5
listparser_getallocint, 6
listparser_getdouble, 6
listparser_getint, 6
listparser_getnelements, 7
parse_input, 7

list_getnelements
input.c, 5
list_to_strarr
input.c, 5
listparser_getallocdouble
input.c, 5
listparser_getallocint
input.c, 6
listparser_getdouble
input.c, 6
listparser_getint
input.c, 6
listparser_getnelements
input.c, 7

msg
aux.c, 3
pinc.h, 9

parse_input
input.c, 7
pinc.h, 9
pinc.h
msg, 9
parse_input, 9