

iffparser

COLLABORATORS

	<i>TITLE :</i> iffparser		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	iffparser	1
1.1	IFFParser_oo Guide	1
1.2	Introduction	2
1.3	Author's Infos	2
1.4	Example Program	3
1.5	iffparser()	3
1.6	addr()	3
1.7	close()	4
1.8	closechunk()	4
1.9	createchunk(type, id, size)	4
1.10	exit(type, id)	5
1.11	first(type, id)	5
1.12	getheader(string, filename)	6
1.13	load(filename)	6
1.14	save(filename)	6
1.15	scan()	7
1.16	setscan(type, id)	7
1.17	size()	8
1.18	writchunk(datas, datalen)	8
1.19	succ()	8
1.20	EXCEPTIONS TABLE	9

Chapter 1

iffparser

1.1 IFFParser_oo Guide

** IFFParser_oo - Written By Fabio Rotondo **

** DOCUMENTATION GUIDE **

Introduction

Author's Infos

Example Program

COMMANDS

BRIEF DESCRIPTION

iffparser()

 Inits the iffparser object.

addr()

 Returns current data position in mem.

close()

 Closes currently IFF session.

closechunk()

 Closes last created chunk.

createchunk(type, id, size)

 Inits a new chunk to write.

exit(type, id)

 Defines when scan() must stop.

first(type, id)

 Prepares data examination.

getheader(string, filename)

 Returns header of a file.

```
load(filename)
    Inits an IFF Load session.

save(filename)
    Inits an IFF SAVE session.

scan()
    Begins scanning an IFF file.

setscan(type, id)
    Defines type of chunk to scan for.

succ()
    Gets next memory data location.

size()
    Returns current data size.

writechunk(datas, datalen)
    Writes some datas to a chunk.
```

EXCEPTIONS TABLE

1.2 Introduction

IFFParser is an Object to easily read/write IFF files.

It is not so powerful as IFFParse.library: it misses some features, but I have written it to match all my needs, and it does. I hope you will enjoy using. Look at the examples for working programs.

1.3 Author's Infos

My name is Fabio Rotondo. I am a free-lance Amiga programmer and I would like to get in touch with anyone who writes code for the Amiga. I write in AmigaE, BlitzII, C and a bunch of other languages.

Please, feel free to contact me for any suggestions/questions.

My address is:

Fabio Rotondo
C.so Vercelli 9
28100 Novara
ITALY
Tel. (ITA) - (0)321 - 459676
e-mail: fsoft@intercom.it

Check out my WWWPage with many of my AmigaE Modules/Sources!

Thanks!

```
-----
//                               | Fabio Rotondo           e-mail: fsoft@intercom.it
\X/ Amiga. Feel The Power. | URL:      http://www.intercom.it/~fsoft/index.html
+-----+
|GTBoxConv V4.20 | SafeLocks V1.00 | Locator.library V1.00 | MIDIBridge V1.00 |
|MIDIFilter V1.1 | PCATPro (65%)   | DOOPSI-GS (85%)       | Great Game (10%) |
+-----+
```

1.4 Example Program

See directory Docs/Examples/IFFParser_Examples for some working AmigaE programs. They are all PD sources! Feel free of examing and grab outwhatever you want!

1.5 iffparser()

NAME: iffparser()

DESCRIPTION: This command will initialize the iffparser object.

INPUT: NONE.

RESULTS: NONE.

EXCEPTIONS:

iffp

SEE ALSO:

1.6 addr()

NAME: addr()

DESCRIPTION: This command will return current memory data address.

INPUT: NONE.

RESULTS: a PTR TO LONG specifying memory data address (may be NIL)

EXCEPTIONS: NONE.

SEE ALSO:

size()

first()

succ()

1.7 close()

NAME: close()

DESCRIPTION: This command will close curent IFF session.

INPUT: NONE.

RESULTS: NONE.

EXCEPTIONS: NONE.

SEE ALSO:

load()

save()

1.8 closechunk()

NAME: closechunk()

DESCRIPTION: This command will close current chunk.

INPUT: NONE.

RESULTS: NONE.

EXCEPTIONS:

init

popc

SEE ALSO:

createchunk()

writechunk()

save()

1.9 createchunk(type, id, size)

NAME: createchunk(type, id, size=IFFSIZE_UNKNOWN)

DESCRIPTION: This command will create a new chunk where to write in.

INPUT: type - (LONG) type of the chunk. (ex. "ILBM")
 id - (LONG) id of the chunk. (ex. "FORM")

RESULTS: NONE.

EXCEPTIONS:

init

```
push
    SEE ALSO:
closechunk()

writechunk()

save()
```

1.10 exit(type, id)

NAME: exit(type, id)

DESCRIPTION: Use this command to determinate WHEN scan() should stop.

```
INPUT: type      - (LONG) type of the chunk. (ex. "ILBM")
      id         - (LONG) id of the chunk. (ex. "FORM")
```

RESULTS: NONE.

EXCEPTIONS:

```
init

exit
    SEE ALSO:
load()

scan()

setscan()
```

1.11 first(type, id)

NAME: first(type, id)

DESCRIPTION: Use this command to position IFFParser object to the FIRST memory data location of a specific kind.

```
INPUT: type      - (LONG) type of the chunk. (ex. "ILBM")
      id         - (LONG) id of the chunk. (ex. "FORM")
```

RESULTS: a PTR TO LONG memory location of the data (may be NIL)

EXCEPTIONS:

```
init
    SEE ALSO:
load()

scan()

setscan()

succ()
```

1.12 getheader(string, filename)

NAME: getheader(string:PTR TO CHAR, filename:PTR TO CHAR)

DESCRIPTION: Use this command to determinate the kind of a file.

INPUT: string - A `_VALID_ Estring` already initialized.
filename - Name of the file to examine.

RESULTS: string - your `_VALID_ Estring` will be filled with the file header (ex. "ILBM")

NOTE: This command returns a STRING not a LONG!

EXCEPTIONS: NONE.

SEE ALSO:

1.13 load(filename)

NAME: load(filename:PTR TO CHAR)

DESCRIPTION: Use this command to open an already existing IFF file and to begin a IFF load session.

INPUT: filename - Name of the file to open / load.

RESULTS: NONE.

EXCEPTIONS:

iffa

oold

oiff

SEE ALSO:

setscan()

scan()

close()

1.14 save(filename)

NAME: save(filename:PTR TO CHAR)

DESCRIPTION: Use this command to create a new IFF file and to begin a IFF save session.

INPUT: filename - Name of the file to open / load.

RESULTS: NONE.

EXCEPTIONS:

iffa

onew

oiff

SEE ALSO:

createchunk()

writetechunk()

closechunk()

close()

1.15 scan()

NAME: scan()

DESCRIPTION: Use this command to begin scanning an IFF file.

INPUT: NONE.

RESULTS: NONE.

EXCEPTIONS:

init

SEE ALSO:

setscan()

load()

1.16 setscan(type, id)

NAME: setscan(type, id)

DESCRIPTION: Use this command to set chunk that will be loaded by scan().

INPUT: type - (LONG) type of chunk. (ex. "ILBM")
id - (LONG) id of chunk. (ex. "FORM")

RESULTS: NONE.

EXCEPTIONS:

init

coll

NOTES: You can do more than a single setscan() before ↔
scan()ing the

IFF file. This is the biggie ;)
In this way you can load up in memory all you need in a single
file access.

SEE ALSO:

scan()

load()

1.17 size()

NAME: size()

DESCRIPTION: Use this command to get size of current memory data.

INPUT: NONE.

RESULTS: size - (LONG) size of the memory data. (May be NIL)

EXCEPTIONS: NONE.

SEE ALSO:

first()

succ()

1.18 writechunk(datas, datalen)

NAME: writechunk(data, datalen)

DESCRIPTION: Use this command to write some data into a chunk.

INPUT: data - (PTR TO LONG) memory location of your datas.
datalen - (LONG) length in bytes of your data.

RESULTS: NONE.

EXCEPTIONS:

init

writ

SEE ALSO:

save()

createchunk()

closechunk()

1.19 succ()

NAME: succ()

DESCRIPTION: Use this command to position to the next memory data.

INPUT: NONE.

RESULTS: a PTR TO LONG to the new location of memory data. (May be NIL)

EXCEPTIONS:

init

SEE ALSO:

load()

first()

1.20 EXCEPTIONS TABLE

EXCEPTIONS TABLE

EXCEPTION	- DESCRIPTION
"coll"	- Could not do a CollectionChunk() call.
"exit"	- Could not assign StopOnExit.
"iffa"	- Could not AllocIFF().
"iffp"	- Could not open iffparse.library.
"init"	- IFFParser object not initialized with load() or save().
"oiff"	- Could not OpenIFF().
"onew"	- Could not create filename with Open(fname, MODE_NEWFILE).
"oold"	- Could not Open() file for reading.
"popc"	- Could not Pop current chunk.
"push"	- Could not Push current chunk.
"writ"	- WriteChunkBytes() failed.