

SParse

Generated by Doxygen 1.8.5

Fri Sep 14 2018 17:01:47

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Error Struct Reference	5
3.2	ErrorMgr Class Reference	5
3.2.1	Member Function Documentation	6
3.2.1.1	ErrorMgr_copy	6
3.2.1.2	ErrorMgr_exit	6
3.2.1.3	ErrorMgr_getRef	6
3.3	FileDesc Struct Reference	7
3.4	FileMgr Class Reference	8
3.4.1	Member Function Documentation	9
3.4.1.1	FileMgr_addDirectory	9
3.4.1.2	FileMgr_addFile	9
3.4.1.3	FileMgr_copy	9
3.4.1.4	FileMgr_filterFiles	9
3.4.1.5	FileMgr_getRef	9
3.4.1.6	FileMgr_load	9
3.5	FileReader Class Reference	10
3.6	Grammar1 Struct Reference	10
3.7	Grammar2 Struct Reference	11
3.8	List Class Reference	12
3.8.1	Member Function Documentation	13
3.8.1.1	List_copy	13
3.8.1.2	List_forEach	13
3.8.1.3	List_getSize	13
3.8.1.4	List_insertHead	13
3.8.1.5	List_insertTail	13

3.8.1.6	List_merge	13
3.8.1.7	List_new	13
3.9	Map Class Reference	14
3.9.1	Member Function Documentation	15
3.9.1.1	Map_copy	15
3.9.1.2	Map_insert	15
3.10	MapEntry Struct Reference	15
3.11	Node Struct Reference	16
3.12	Object Struct Reference	16
3.12.1	Member Function Documentation	16
3.12.1.1	Object_copy	16
3.12.1.2	Object_getRef	17
3.12.1.3	Object_new	17
3.13	ObjectInfo Struct Reference	17
3.14	ObjectMgr Class Reference	18
3.14.1	Member Function Documentation	19
3.14.1.1	ObjectMgr_allocate	19
3.14.1.2	ObjectMgr_copy	20
3.14.1.3	ObjectMgr_deallocate	20
3.14.1.4	ObjectMgr_getRef	20
3.14.2	Member Data Documentation	20
3.14.2.1	maxNbObjectAllocated	20
3.15	OptionDefault Struct Reference	20
3.16	OptionMgr Class Reference	21
3.16.1	Member Function Documentation	22
3.16.1.1	OptionMgr_getRef	22
3.16.1.2	OptionMgr_readFromCmdLine	22
3.17	Point Struct Reference	22
3.18	SdbMgr Class Reference	22
3.19	SdbRequest Struct Reference	23
3.20	SParse Class Reference	24
3.20.1	Member Function Documentation	24
3.20.1.1	SParse_delete	24
3.20.1.2	SParse_new	24
3.20.1.3	SParse_parse	25
3.21	String Class Reference	25
3.21.1	Member Function Documentation	26
3.21.1.1	String_copy	26
3.21.1.2	String_getRef	26
3.21.1.3	String_new	26

3.22	TestClass Struct Reference	27
3.23	TestFileMgr Struct Reference	27
3.24	TestItem Struct Reference	28
3.25	TestObjectMgr Struct Reference	29
3.26	TestSdbMgr Struct Reference	30
3.27	TimeMgr Class Reference	30
3.28	yy_buffer_state Struct Reference	31
3.28.1	Member Data Documentation	31
3.28.1.1	yy_bs_column	31
3.28.1.2	yy_bs_lineno	32
3.29	yy_trans_info Struct Reference	32
3.30	yyalloc Union Reference	32
3.31	yyguts_t Struct Reference	33
3.31.1	Member Data Documentation	33
3.31.1.1	yy_buffer_stack	33
3.31.1.2	yy_buffer_stack_max	34
3.31.1.3	yy_buffer_stack_top	34
4	File Documentation	35
4.1	C:/temp/SParse/src/AppliLib/ErrorMgr/ErrorMgr.c File Reference	35
4.1.1	Detailed Description	36
4.2	C:/temp/SParse/src/AppliLib/FileMgr/FileDesc.c File Reference	36
4.2.1	Detailed Description	36
4.3	C:/temp/SParse/src/AppliLib/FileMgr/FileMgr.c File Reference	36
4.3.1	Detailed Description	37
4.4	C:/temp/SParse/src/AppliLib/OptionMgr/OptionMgr.c File Reference	37
4.4.1	Detailed Description	38
4.4.2	Variable Documentation	38
4.4.2.1	optionDefault	38
4.5	C:/temp/SParse/src/AppliLib/SdbMgr/SdbMgr.c File Reference	39
4.5.1	Detailed Description	40
4.6	C:/temp/SParse/src/AppliLib/TimeMgr/TimeMgr.c File Reference	40
4.6.1	Detailed Description	40
4.7	C:/temp/SParse/src/CommonLib/Debug/Debug.c File Reference	40
4.7.1	Detailed Description	41
4.8	C:/temp/SParse/src/CommonLib/List/List.c File Reference	41
4.8.1	Detailed Description	42
4.9	C:/temp/SParse/src/CommonLib/Map/Map.c File Reference	42
4.9.1	Detailed Description	43
4.10	C:/temp/SParse/src/CommonLib/Memory/Memory.c File Reference	43

4.10.1 Detailed Description	44
4.11 C:/temp/SParse/src/CommonLib/Object/Object.c File Reference	44
4.11.1 Detailed Description	45
4.12 C:/temp/SParse/src/CommonLib/ObjectMgr/ObjectMgr.c File Reference	45
4.12.1 Detailed Description	46
4.13 C:/temp/SParse/src/CommonLib/String/String2.c File Reference	46
4.13.1 Detailed Description	47
4.14 C:/temp/SParse/src/ParseLib/FileReader/FileReader.c File Reference	47
4.14.1 Detailed Description	48
4.15 C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.c File Reference	48
4.15.1 Detailed Description	49
4.16 C:/temp/SParse/src/ParseLib/SParse/SParse.c File Reference	49
4.16.1 Detailed Description	50
 Index	 51

Chapter 1

Class Index

1.1 Class List

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

Error	5
ErrorMgr	5
FileDesc	7
FileMgr	8
FileReader	10
Grammar1	10
Grammar2	11
List	12
Map	14
MapEntry	15
Node	16
Object	16
ObjectInfo	17
ObjectMgr	18
OptionDefault	20
OptionMgr	21
Point	22
SdbMgr	22
SdbRequest	23
SParse	24
String	25
TestClass	27
TestFileMgr	27
TestItem	28
TestObjectMgr	29
TestSdbMgr	30
TimeMgr	30
yy_buffer_state	31
yy_trans_info	32
yyalloc	32
yyguts_t	33

Chapter 2

File Index

2.1 File List

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

C:/temp/SParse/src/AppliLib/ErrorMgr/ Error.h	??
C:/temp/SParse/src/AppliLib/ErrorMgr/ ErrorMgr.c	35
C:/temp/SParse/src/AppliLib/ErrorMgr/ ErrorMgr.h	??
C:/temp/SParse/src/AppliLib/FileMgr/ FileDesc.c The FileDesc class describe a File in the FileMgr	36
C:/temp/SParse/src/AppliLib/FileMgr/ FileDesc.h	??
C:/temp/SParse/src/AppliLib/FileMgr/ FileMgr.c The FileMgr class manages a list of files contained in a group of locations	36
C:/temp/SParse/src/AppliLib/FileMgr/ FileMgr.h	??
C:/temp/SParse/src/AppliLib/OptionMgr/ OptionMgr.c The OptionMgr class manages the application configuration	37
C:/temp/SParse/src/AppliLib/OptionMgr/ OptionMgr.h	??
C:/temp/SParse/src/AppliLib/SdbMgr/ SdbMgr.c TBD	39
C:/temp/SParse/src/AppliLib/SdbMgr/ SdbMgr.h	??
C:/temp/SParse/src/AppliLib/SdbMgr/ SdbRequest.h	??
C:/temp/SParse/src/AppliLib/TimeMgr/ TimeMgr.c This file implements the class TimeMgr	40
C:/temp/SParse/src/AppliLib/TimeMgr/ TimeMgr.h	??
C:/temp/SParse/src/AppliLib/TimeMgr/ Timer.h	??
C:/temp/SParse/src/CommonLib/Debug/ Debug.c This file contains debugging functions	40
C:/temp/SParse/src/CommonLib/Debug/ Debug.h	??
C:/temp/SParse/src/CommonLib/Error/ Error.h	??
C:/temp/SParse/src/CommonLib/List/ List.c This file contains the implementation of the class List	41
C:/temp/SParse/src/CommonLib/List/ List.h	??
C:/temp/SParse/src/CommonLib/Map/ Map.c A Map class. This class provides a container indexed by a string	42
C:/temp/SParse/src/CommonLib/Map/ Map.h	??
C:/temp/SParse/src/CommonLib/Map/ MapEntry.h	??
C:/temp/SParse/src/CommonLib/Memory/ Memory.c This file provides the implementation of the memory functions	43
C:/temp/SParse/src/CommonLib/Memory/ Memory.h	??
C:/temp/SParse/src/CommonLib/Object/ Object.c This file contains the implementation for the class Object	44
C:/temp/SParse/src/CommonLib/Object/ Object.h	??

C:/temp/SParse/src/CommonLib/ObjectMgr/ ObjectMgr.c	
An object management class	45
C:/temp/SParse/src/CommonLib/ObjectMgr/ ObjectMgr.h	??
C:/temp/SParse/src/CommonLib/String/ String2.c	
The String class provide a dynamic array of char terminated by 0	46
C:/temp/SParse/src/CommonLib/String/ String2.h	??
C:/temp/SParse/src/CommonLib/Types/ Types.h	??
C:/temp/SParse/src/ParseLib/C89Grammar/ C99_Grammar.parse.h	??
C:/temp/SParse/src/ParseLib/C89Grammar/ Node.h	??
C:/temp/SParse/src/ParseLib/FileReader/ FileReader.c	
This file contains the implementation for the class FileReader	47
C:/temp/SParse/src/ParseLib/FileReader/ FileReader.h	??
C:/temp/SParse/src/ParseLib/Grammar1/ Grammar1.c	
TBD	48
C:/temp/SParse/src/ParseLib/Grammar1/ Grammar1.h	??
C:/temp/SParse/src/ParseLib/Grammar1/ Grammar1.parse.h	??
C:/temp/SParse/src/ParseLib/Grammar2/ Grammar2.h	??
C:/temp/SParse/src/ParseLib/Grammar2/ Grammar2.parse.h	??
C:/temp/SParse/src/ParseLib/SParse/ SParse.c	
This file contains the implementation for the class SParse	49
C:/temp/SParse/src/ParseLib/SParse/ SParse.h	??

Chapter 3

Class Documentation

3.1 Error Struct Reference

Public Attributes

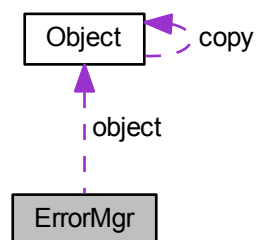
- ErrorSeverity **severity**
- char * **message**
- unsigned int **param** [5]

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

- C:/temp/SParse/src/AppliLib/ErrorMgr/Error.h

3.2 ErrorMgr Class Reference

Collaboration diagram for ErrorMgr:



Public Member Functions

- PUBLIC void [ErrorMgr_delete](#) ([ErrorMgr](#) *this)
Delete an instance of the class [ErrorMgr](#).
- PUBLIC [ErrorMgr](#) * [ErrorMgr_copy](#) ([ErrorMgr](#) *this)
Copy an instance of the class [ErrorMgr](#).

- PUBLIC [ErrorMgr](#) * [ErrorMgr_getRef](#) ()

TBD.

- PUBLIC void [ErrorMgr_exit](#) ([ErrorMgr](#) *this, const char *fmt,...)

Exit the processing of the main function and display an error message.

Public Attributes

- [Object](#) **object**

3.2.1 Member Function Documentation

3.2.1.1 PUBLIC [ErrorMgr](#) * [ErrorMgr_copy](#) ([ErrorMgr](#) * *this*)

Copy an instance of the class [ErrorMgr](#).

Returns

Copy of the instance given as a parameter.

3.2.1.2 PUBLIC void [ErrorMgr_exit](#) ([ErrorMgr](#) * *this*, const char * *fmt*, ...)

Exit the processing of the main function and display an error message.

Parameters

<i>in</i>	<i>fmt</i>	Printf style display string.
-----------	------------	------------------------------

3.2.1.3 PUBLIC [ErrorMgr](#) * [ErrorMgr_getRef](#) ()

TBD.

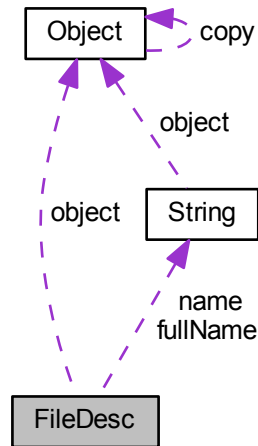
TBD

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

- C:/temp/SParse/src/AppliLib/ErrorMgr/[ErrorMgr.c](#)

3.3 FileDesc Struct Reference

Collaboration diagram for FileDesc:



Public Member Functions

- PUBLIC [FileDesc](#) * [FileDesc_new](#) ()
TBD.
- PUBLIC void [FileDesc_delete](#) ([FileDesc](#) *this)
TBD.
- PUBLIC [FileDesc](#) * [FileDesc_copy](#) ([FileDesc](#) *this)
TBD.
- PUBLIC void [FileDesc_setFullName](#) ([FileDesc](#) *this, [String](#) *fullName)
TBD.
- PUBLIC [String](#) * [FileDesc_getFullName](#) ([FileDesc](#) *this)
TBD.
- PUBLIC [String](#) * [FileDesc_getName](#) ([FileDesc](#) *this)
TBD.

Public Attributes

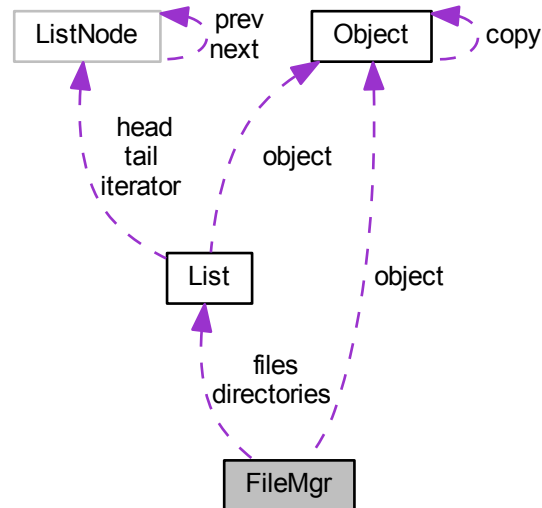
- [Object](#) **object**
- [String](#) * **name**
- [String](#) * **fullName**

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

- C:/temp/SParse/src/AppliLib/FileMgr/[FileDesc.c](#)

3.4 FileMgr Class Reference

Collaboration diagram for FileMgr:



Public Member Functions

- PUBLIC void `FileMgr_delete` (`FileMgr *this`)
Delete an instance of the class `FileMgr`.
- PUBLIC `FileMgr *` `FileMgr_copy` (`FileMgr *this`)
Copy an instance of the class `FileMgr`.
- PUBLIC `FileMgr *` `FileMgr_getRef` ()
Get a reference to the singleton instance of `FileMgr`.
- PUBLIC unsigned int `FileMgr_addDirectory` (`FileMgr *this`, const char *directoryName)
Add all files in the given directory to the list of managed files.
- PUBLIC unsigned int `FileMgr_addFile` (`FileMgr *this`, const char *fileName)
Add a files to the list of managed files.
- PUBLIC `String *` `FileMgr_load` (`FileMgr *this`, const char *fileName)
Load a managed file into a `String`.
- PUBLIC `List *` `FileMgr_filterFiles` (`FileMgr *this`, const char *pattern)
TBD.

Public Attributes

- `Object` **object**
- `List *` **files**
- `List *` **directories**
- char **rootLocation** [FILEMGR_MAX_PATH]

3.4.1 Member Function Documentation

3.4.1.1 PUBLIC unsigned int FileMgr_addDirectory (FileMgr * *this*, const char * *directoryName*)

Add all files in the given directory to the list of managed files.

Returns

Status.

3.4.1.2 PUBLIC unsigned int FileMgr_addFile (FileMgr * *this*, const char * *fileName*)

Add a files to the list of managed files.

Returns

Status.

3.4.1.3 PUBLIC FileMgr * FileMgr_copy (FileMgr * *this*)

Copy an instance of the class [FileMgr](#).

Returns

New instance

3.4.1.4 PUBLIC List * FileMgr_filterFiles (FileMgr * *this*, const char * *pattern*)

TBD.

Returns

TBD

3.4.1.5 PUBLIC FileMgr * FileMgr_getRef ()

Get a reference to the singleton instance of [FileMgr](#).

Returns

Reference to the singleton.

3.4.1.6 PUBLIC String * FileMgr_load (FileMgr * *this*, const char * *fileName*)

Load a managed file into a [String](#).

Returns

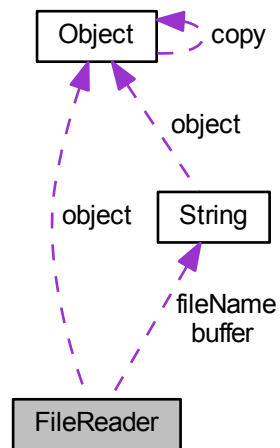
Content of file.

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

- C:/temp/SParse/src/AppliLib/FileMgr/[FileMgr.c](#)

3.5 FileReader Class Reference

Collaboration diagram for FileReader:



Public Attributes

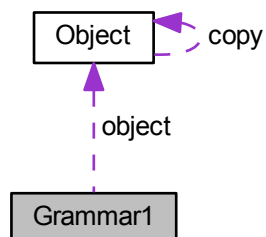
- `Object` `object`
- `String` * `buffer`
- `String` * `fileName`

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

- `C:/temp/SParse/src/ParseLib/FileReader/FileReader.c`

3.6 Grammar1 Struct Reference

Collaboration diagram for Grammar1:



Public Attributes

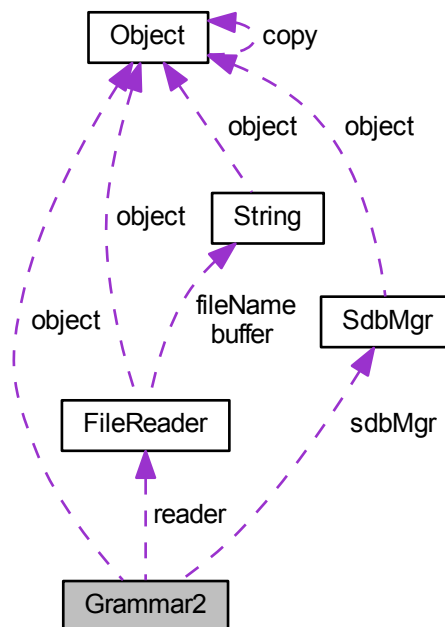
- [Object](#) **object**

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

- C:/temp/SParse/src/ParseLib/Grammar1/[Grammar1.c](#)

3.7 Grammar2 Struct Reference

Collaboration diagram for Grammar2:



Public Attributes

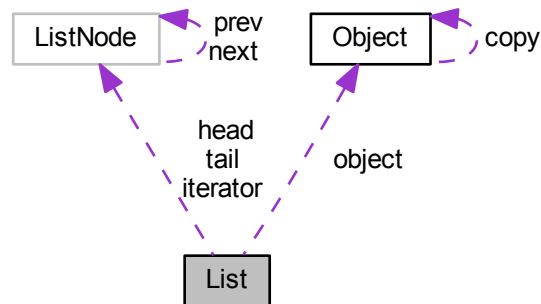
- [Object](#) **object**
- void * **scanner**
- [SdbMgr](#) * **sdbMgr**
- [FileReader](#) * **reader**

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

- C:/temp/SParse/src/ParseLib/Grammar2/Grammar2.c

3.8 List Class Reference

Collaboration diagram for List:



Public Member Functions

- PUBLIC `List * List_new ()`
Create a new instance of the class `List`.
- PUBLIC void `List_delete (List *this)`
Delete an instance of the class `List`.
- PUBLIC `List * List_copy (List *this)`
Copy an instance of the class `List`.
- PUBLIC void `List_insertHead (List *this, void *item)`
Insert an item at the head of a list instance.
- PUBLIC void `List_insertTail (List *this, void *item)`
Insert an item at the tail of a `List` instance.
- PUBLIC void `List_merge (List *this, List *l1)`
Merge a list into a `List` instance.
- PUBLIC void `List_forEach (List *this, void(*method)(void *o, void *p), void *object)`
Execute a given function for each item in an instance of `List`.
- PUBLIC unsigned int `List_getSize (List *this)`
Get the number of item in `List` instance.

Public Attributes

- `Object object`
- `ListNode * head`
- `ListNode * tail`
- `ListNode * iterator`
- unsigned int `nbNodes`

3.8.1 Member Function Documentation

3.8.1.1 PUBLIC List * List_copy (List * *this*)

Copy an instance of the class [List](#).

Returns

Copy of the given instance.

3.8.1.2 PUBLIC void List_forEach (List * *this*, void(*)(void *o, void *p) *method*, void * *object*)

Execute a given function for each item in an instance of [List](#)..

Parameters

<i>in</i>	<i>f</i>	Pointer to function.
-----------	----------	----------------------

3.8.1.3 PUBLIC unsigned int List_getSize (List * *this*)

Get the number of item in [List](#) instance.

Returns

Number of items.

3.8.1.4 PUBLIC void List_insertHead (List * *this*, void * *item*)

Insert an item at the head of a list instance.

Parameters

<i>in</i>	<i>item</i>	Reference to item.
-----------	-------------	--------------------

3.8.1.5 PUBLIC void List_insertTail (List * *this*, void * *item*)

Insert an item at the tail of a [List](#) instance.

Parameters

<i>in</i>	<i>item</i>	Reference to item.
-----------	-------------	--------------------

3.8.1.6 PUBLIC void List_merge (List * *this*, List * *l1*)

Merge a list into a [List](#) instance.

Parameters

<i>in</i>	<i>l1</i>	Reference to list to merge.
-----------	-----------	-----------------------------

3.8.1.7 PUBLIC List * List_new ()

Create a new instance of the class [List](#).

Returns

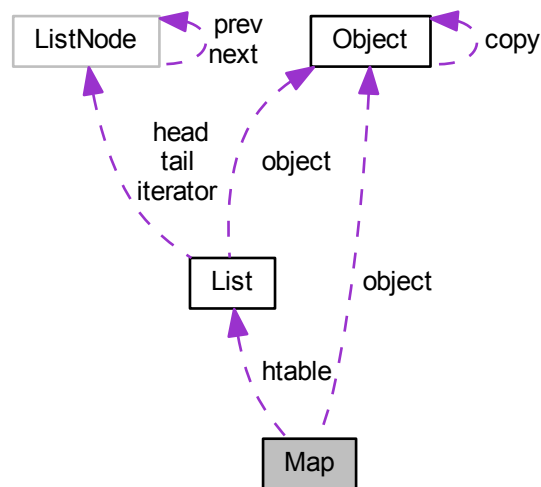
New instance.

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

- C:/temp/SParse/src/CommonLib/List/[List.c](#)

3.9 Map Class Reference

Collaboration diagram for Map:

**Public Member Functions**

- PUBLIC [Map](#) * [Map_new](#) ()
Create a new instance of the class [Map](#).
- PUBLIC void [Map_delete](#) ([Map](#) *this)
TBD.
- PUBLIC [Map](#) * [Map_copy](#) ([Map](#) *this)
Copy an instance of the class [Map](#).
- PUBLIC unsigned int [Map_insert](#) ([Map](#) *this, [String](#) *s, void *p)
TBD.
- PUBLIC unsigned int [Map_find](#) ([Map](#) *this, [String](#) *s, void **p)
TBD.

Public Attributes

- [Object](#) **object**
- [List](#) * **htable** [HTABLE_SIZE]

3.9.1 Member Function Documentation

3.9.1.1 PUBLIC Map * Map_copy (Map * *this*)

Copy an instance of the class [Map](#).

Returns

Copy of instance.

3.9.1.2 PUBLIC unsigned int Map_insert (Map * *this*, String * *s*, void * *p*)

TBD.

Returns

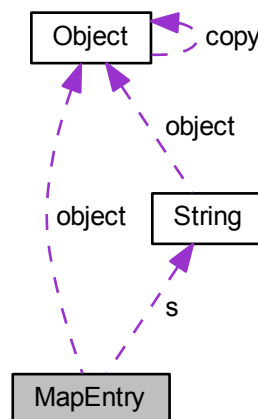
TBD

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

- C:/temp/SParse/src/CommonLib/Map/[Map.c](#)

3.10 MapEntry Struct Reference

Collaboration diagram for MapEntry:



Public Attributes

- [Object](#) **object**
- [String](#) * **s**
- void * **item**

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

- C:/temp/SParse/src/CommonLib/Map/[MapEntry.c](#)

3.11 Node Struct Reference

Public Attributes

- NodeType **type**
- NodeScope **scope**
- char * **name**

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

- C:/temp/SParse/src/ParseLib/C89Grammar/Node.c

3.12 Object Struct Reference

Collaboration diagram for Object:



Public Member Functions

- PUBLIC [Object](#) * [Object_new](#) (unsigned int size, void(*f_delete)([Object](#) *), [Object](#) *(*f_copy)([Object](#) *))
Create an instance of the class [Object](#).
- PUBLIC void [Object_delete](#) ([Object](#) *this)
Delete an instance of the class [Object](#).
- PUBLIC [Object](#) * [Object_copy](#) ([Object](#) *this)
Copy an instance of the class [Object](#).
- PUBLIC [Object](#) * [Object_getRef](#) ([Object](#) *this)
Get a reference to an instance of the class [Object](#).

Public Attributes

- unsigned int **id**
- void(* **delete**)([Object](#) *this)
- [Object](#) *(* **copy**)([Object](#) *this)
- unsigned int **refCount**
- unsigned int **size**

3.12.1 Member Function Documentation

3.12.1.1 PUBLIC [Object](#) * [Object_copy](#) ([Object](#) * *this*)

Copy an instance of the class [Object](#).

Returns

New instance

3.12.1.2 PUBLIC Object * Object_getRef (Object * *this*)

Get a reference to an instance of the class [Object](#).

Returns

Reference to instance

3.12.1.3 PUBLIC Object * Object_new (unsigned int *size*, void (*)(Object *) *f_delete*, Object (*)(Object *) *f_copy*)

Create an instance of the class [Object](#).

Parameters

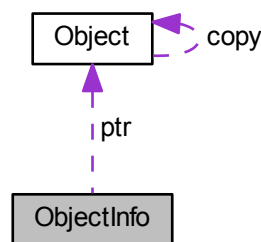
in	<i>f_delete</i>	Destructor function.
in	<i>f_copy</i>	Copy_operator function.

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

- C:/temp/SParse/src/CommonLib/Object/Object.h
- C:/temp/SParse/src/CommonLib/Object/[Object.c](#)

3.13 ObjectInfo Struct Reference

Collaboration diagram for ObjectInfo:



Public Attributes

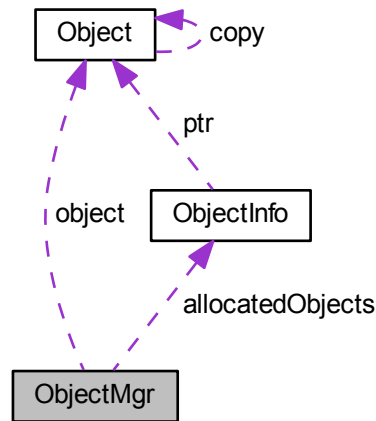
- [Object](#) * **ptr**
- unsigned int **prevId**
- unsigned int **nextId**

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

- C:/temp/SParse/src/CommonLib/ObjectMgr/[ObjectMgr.c](#)
- C:/temp/SParse/src/CommonLib/ObjectMgr/tests/main.c

3.14 ObjectMgr Class Reference

Collaboration diagram for ObjectMgr:



Public Member Functions

- PUBLIC void [ObjectMgr_delete](#) ([ObjectMgr](#) *this)
Delete an instance of the class [ObjectMgr](#).
- PUBLIC [ObjectMgr](#) * [ObjectMgr_copy](#) ([ObjectMgr](#) *this)
Copy an instance of the class [ObjectMgr](#).
- PUBLIC [ObjectMgr](#) * [ObjectMgr_getRef](#) ()
Get a reference to the singleton instance of [ObjectMgr](#).
- PUBLIC void [ObjectMgr_report](#) ([ObjectMgr](#) *this)
Reports the usage statistics for an instance of [ObjectMgr](#).
- PUBLIC [Object](#) * [ObjectMgr_allocate](#) ([ObjectMgr](#) *this, unsigned int size)
Allocate a new object memory footprint of a given size.
- PUBLIC void [ObjectMgr_deallocate](#) ([ObjectMgr](#) *this, [Object](#) *object)
De Allocate a given object.

Public Attributes

- [Object](#) **object**
- unsigned int **maxNbObjectAllocated**
- unsigned int **allocRequestId**
- unsigned int **freeRequestId**
- unsigned int **nbAllocatedObjects**
- [ObjectInfo](#) **allocatedObjects** [MAX_NB_OBJECTS]
- unsigned int **freeSpace**
- unsigned int **usedSpace**

3.14.1 Member Function Documentation

3.14.1.1 PUBLIC Object * ObjectMgr_allocate (ObjectMgr * *this*, unsigned int *size*)

Allocate a new object memory footprint of a given size.

Parameters

<i>in</i>	<i>size</i>	size in bytes of the memory footprint.
-----------	-------------	--

Returns

Reference to a instance of [Object](#).

3.14.1.2 PUBLIC ObjectMgr * ObjectMgr_copy (ObjectMgr * this)

Copy an instance of the class [ObjectMgr](#).

Returns

New instance

3.14.1.3 PUBLIC void ObjectMgr_deallocate (ObjectMgr * this, Object * object)

De Allocate a given object.

Parameters

<i>in</i>	<i>object</i>	Reference to instance of Object .
-----------	---------------	---

3.14.1.4 PUBLIC ObjectMgr * ObjectMgr_getRef ()

Get a reference to the singleton instance of [ObjectMgr](#).

Returns

Reference to the singleton.

3.14.2 Member Data Documentation**3.14.2.1 unsigned int ObjectMgr::maxNbObjectAllocated**

This is member B

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

- C:/temp/SParse/src/CommonLib/ObjectMgr/[ObjectMgr.c](#)

3.15 OptionDefault Struct Reference**Public Attributes**

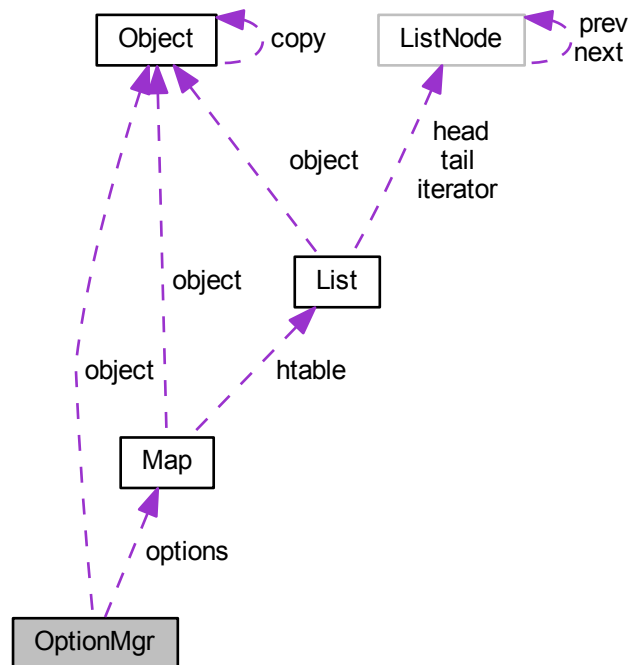
- char * **name**
- char * **flag**
- char * **value**

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

- C:/temp/SParse/src/AppliLib/OptionMgr/[OptionMgr.c](#)

3.16 OptionMgr Class Reference

Collaboration diagram for OptionMgr:



Public Member Functions

- PUBLIC void `OptionMgr_delete` (`OptionMgr *this`)
TBD.
- PUBLIC `OptionMgr *` `OptionMgr_copy` (`OptionMgr *this`)
TBD.
- PUBLIC `OptionMgr *` `OptionMgr_getRef` ()
TBD.
- PUBLIC `String *` `OptionMgr_getOption` (`OptionMgr *this`, `const char *name`)
TBD.
- PUBLIC void `OptionMgr_setOption` (`OptionMgr *this`, `const char *optionName`, `String *value`)
TBD.
- PUBLIC unsigned int `OptionMgr_readFromFile` (`OptionMgr *this`)
TBD.
- PUBLIC unsigned int `OptionMgr_readFromCmdLine` (`OptionMgr *this`, `const int argc`, `const char **argv`)
TBD.

Public Attributes

- `Object` `object`
- `Map *` `options`

3.16.1 Member Function Documentation

3.16.1.1 PUBLIC OptionMgr * OptionMgr_getRef ()

TBD.

TBD

3.16.1.2 PUBLIC unsigned int OptionMgr_readFromCmdLine (OptionMgr * *this*, const int *argc*, const char ** *argv*)

TBD.

Parameters

in	<i>argc</i>	Number of commandline arguments.
in	<i>argv</i>	List os commandline arguments.

Returns

Status of operation.

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

- C:/temp/SParse/src/AppliLib/OptionMgr/[OptionMgr.c](#)

3.17 Point Struct Reference

Public Attributes

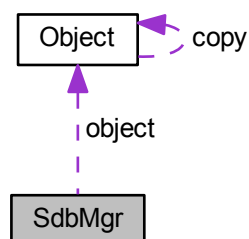
- float **x**
- float **y**

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

- C:/temp/SParse/src/ParseLib/C89Grammar/test.c

3.18 SdbMgr Class Reference

Collaboration diagram for SdbMgr:



Public Attributes

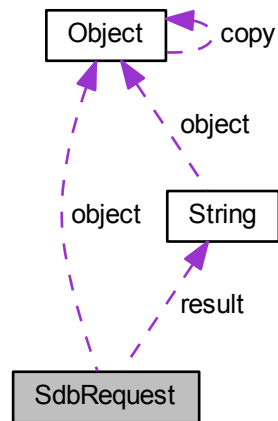
- [Object](#) **object**
- sqlite3 * **db**

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

- C:/temp/SParse/src/AppliLib/SdbMgr/[SdbMgr.c](#)

3.19 SdbRequest Struct Reference

Collaboration diagram for SdbRequest:



Public Attributes

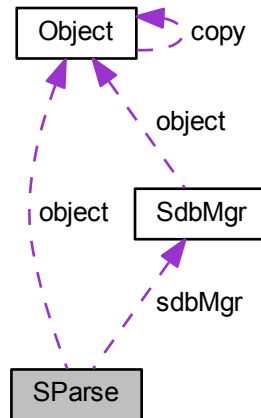
- [Object](#) **object**
- char * **buffer**
- unsigned int **size**
- const char * **fmt**
- [String](#) ** **result**

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

- C:/temp/SParse/src/AppliLib/SdbMgr/[SdbRequest.c](#)

3.20 SParse Class Reference

Collaboration diagram for SParse:



Public Member Functions

- PUBLIC `SParse * SParse_new` (`String *sdbName`)
Create a new `SParse` object.
- PUBLIC void `SParse_delete` (`SParse *this`)
Delete a `SParse` object.
- PUBLIC unsigned int `SParse_parse` (`SParse *this`, `const char *extension`)
Parse all files with a given extension.

Public Attributes

- `Object` **object**
- char * **extension**
- `SdbMgr` * **sdbMgr**

3.20.1 Member Function Documentation

3.20.1.1 PUBLIC void SParse_delete (SParse * this)

Delete a `SParse` object.

Parameters

<code>Object</code>	to delete.
---------------------	------------

3.20.1.2 PUBLIC SParse * SParse_new (String * sdbName)

Create a new `SParse` object.

Returns

New [SParse](#) object.

3.20.1.3 PUBLIC unsigned int SParse_parse (SParse * *this*, const char * *extension*)

Parse all files with a given extension.

Parameters

<i>in</i>	<i>extension</i>	Extension of the files to parse.
-----------	------------------	----------------------------------

Returns

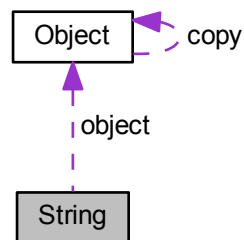
Status of the operation.

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

- C:/temp/SParse/src/ParseLib/SParse/[SParse.c](#)

3.21 String Class Reference

Collaboration diagram for String:

**Public Member Functions**

- PUBLIC [String](#) * [String_new](#) (const char *initString)
Create a new [String](#) object.
- PUBLIC void [String_delete](#) ([String](#) *this)
Delete an instance of class [String](#).
- PUBLIC [String](#) * [String_copy](#) ([String](#) *this)
Copy an instance of class [String](#).
- PUBLIC [String](#) * [String_getRef](#) ([String](#) *this)
Copy an instance of class [String](#).
- PUBLIC unsigned int [String_isEqual](#) ([String](#) *this, [String](#) *compared)
TBD.
- PUBLIC [String](#) * [String_subString](#) ([String](#) *this, unsigned int idx, unsigned int length)
TBD.
- PUBLIC int [String_toInt](#) ([String](#) *this)
TBD.

Public Attributes

- [Object](#) **object**
- char * **buffer**
- unsigned int **length**
- unsigned char * **buffer**

3.21.1 Member Function Documentation

3.21.1.1 PUBLIC String * String_copy (String * *this*)

Copy an instance of class [String](#).

Returns

Copy of instance.

3.21.1.2 PUBLIC String * String_getRef (String * *this*)

Copy an instance of class [String](#).

Returns

Copy of instance.

3.21.1.3 PUBLIC String * String_new (const char * *initString*)

Create a new [String](#) object.

Returns

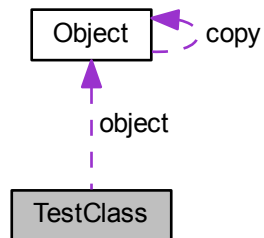
Created [String](#) object.

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

- C:/temp/SParse/src/CommonLib/String/[String2.c](#)
- C:/temp/SParse/src/ParseLib/C89Grammar/main.c
- C:/temp/SParse/src/ParseLib/Grammar1/main.c
- C:/temp/SParse/src/ParseLib/Grammar2/main.c

3.22 TestClass Struct Reference

Collaboration diagram for TestClass:



Public Attributes

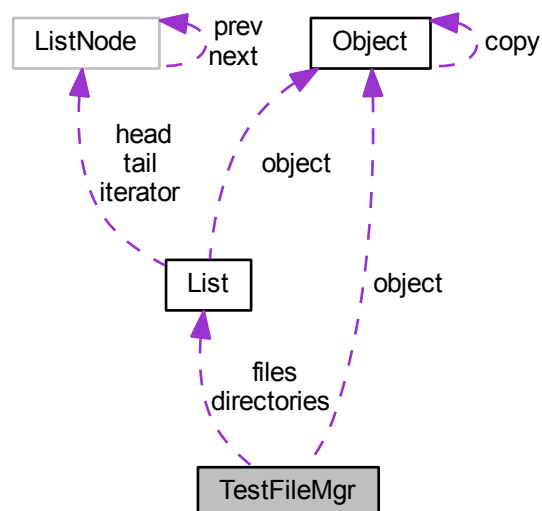
- [Object](#) **object**
- int **testValue**

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

- C:/temp/SParse/src/CommonLib/Object/tests/test1.c

3.23 TestFileMgr Struct Reference

Collaboration diagram for TestFileMgr:



Public Attributes

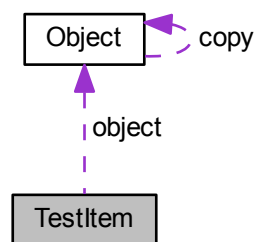
- [Object](#) **object**
- [List](#) * **files**
- [List](#) * **directories**
- char **rootLocation** [FILEMGR_MAX_PATH]

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

- C:/temp/SParse/src/AppliLib/FileMgr/tests/main.c

3.24 TestItem Struct Reference

Collaboration diagram for TestItem:



Public Attributes

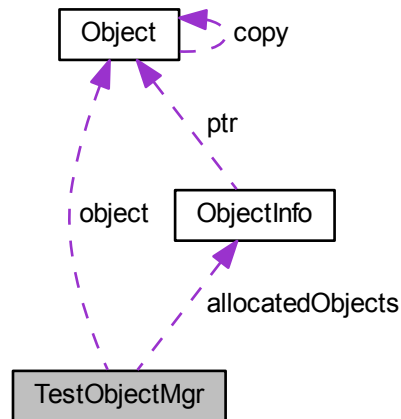
- [Object](#) **object**
- int **x**
- int **y**

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

- C:/temp/SParse/src/CommonLib/List/tests/main.c

3.25 TestObjectMgr Struct Reference

Collaboration diagram for TestObjectMgr:



Public Attributes

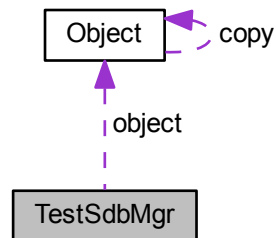
- [Object](#) **object**
- unsigned int **nbBytesAllocated**
- unsigned int **maxNbBytesAllocated**
- unsigned int **allocRequestId**
- unsigned int **freeRequestId**
- unsigned int **nbAllocatedObjects**
- [ObjectInfo](#) **allocatedObjects** [MAX_NB_OBJECTS]
- unsigned int **freeSpace**
- unsigned int **usedSpace**

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

- C:/temp/SParse/src/CommonLib/ObjectMgr/tests/main.c

3.26 TestSdbMgr Struct Reference

Collaboration diagram for TestSdbMgr:



Public Attributes

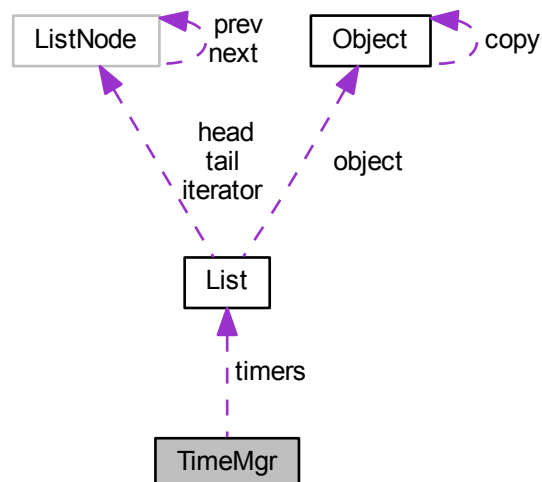
- [Object](#) **object**

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

- C:/temp/SParse/src/AppliLib/SdbMgr/tests/main.c

3.27 TimeMgr Class Reference

Collaboration diagram for TimeMgr:



Public Member Functions

- PUBLIC void [TimeMgr_delete](#) ([TimeMgr](#) *this)
TBD.
- PUBLIC [TimeMgr](#) * [TimeMgr_copy](#) ([TimeMgr](#) *this)
TBD.
- PUBLIC [TimeMgr](#) * [TimeMgr_getRef](#) ()
TBD.
- PUBLIC void [TimeMgr_createTimer](#) ([TimeMgr](#) *this, [String](#) *s)
TBD.
- PUBLIC void [TimeMgr_timeStamp](#) ([TimeMgr](#) *this, [String](#) *s)
TBD.
- PUBLIC void [TimeMgr_report](#) ([TimeMgr](#) *this)
TBD.

Public Attributes

- [List](#) * **timers**

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

- C:/temp/SParse/src/AppliLib/TimeMgr/[TimeMgr.c](#)

3.28 yy_buffer_state Struct Reference

Public Attributes

- FILE * **yy_input_file**
- char * **yy_ch_buf**
- char * **yy_buf_pos**
- [yy_size_t](#) **yy_buf_size**
- int **yy_n_chars**
- int **yy_is_our_buffer**
- int **yy_is_interactive**
- int **yy_at_bol**
- int [yy_bs_lineno](#)
- int [yy_bs_column](#)
- int **yy_fill_buffer**
- int **yy_buffer_status**

3.28.1 Member Data Documentation

3.28.1.1 int yy_buffer_state::yy_bs_column

The column count.

3.28.1.2 int yy_buffer_state::yy_bs_lineno

The line count.

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

- C:/temp/SParse/src/ParseLib/C89Grammar/lex.C99_Grammar.c
- C:/temp/SParse/src/ParseLib/C89Grammar/lex.C99Grammar.c
- C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.lex.c
- C:/temp/SParse/src/ParseLib/Grammar2/Grammar2.lex.c

3.29 yy_trans_info Struct Reference

Public Attributes

- flex_int32_t **yy_verify**
- flex_int32_t **yy_nxt**

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

- C:/temp/SParse/src/ParseLib/C89Grammar/lex.C99_Grammar.c
- C:/temp/SParse/src/ParseLib/C89Grammar/lex.C99Grammar.c
- C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.lex.c
- C:/temp/SParse/src/ParseLib/Grammar2/Grammar2.lex.c

3.30 yyalloc Union Reference

Public Attributes

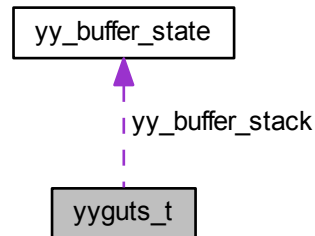
- yytype_int16 **yyss_alloc**
- YYSTYPE **yyvs_alloc**

The documentation for this union was generated from the following files:

- C:/temp/SParse/src/ParseLib/C89Grammar/C99_Grammar.parse.c
- C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.parse.c
- C:/temp/SParse/src/ParseLib/Grammar2/Grammar2.parse.c
- C:/temp/SParse/src/ParseLib/Grammar2.parse.c

3.31 yyguts_t Struct Reference

Collaboration diagram for yyguts_t:



Public Attributes

- YY_EXTRA_TYPE `yyextra_r`
- FILE * `yyin_r`
- FILE * `yyout_r`
- size_t `yy_buffer_stack_top`
- size_t `yy_buffer_stack_max`
- YY_BUFFER_STATE * `yy_buffer_stack`
- char `yy_hold_char`
- int `yy_n_chars`
- int `yy leng_r`
- char * `yy_c_buf_p`
- int `yy_init`
- int `yy_start`
- int `yy_did_buffer_switch_on_eof`
- int `yy_start_stack_ptr`
- int `yy_start_stack_depth`
- int * `yy_start_stack`
- yy_state_type `yy_last_accepting_state`
- char * `yy_last_accepting_cpos`
- int `yylineno_r`
- int `yy_flex_debug_r`
- char * `yytext_r`
- int `yy_more_flag`
- int `yy_more_len`
- YYSTYPE * `yyval_r`

3.31.1 Member Data Documentation

3.31.1.1 YY_BUFFER_STATE* yyguts_t::yy_buffer_stack

Stack as an array.

3.31.1.2 `size_t yyguts_t::yy_buffer_stack_max`

capacity of stack.

3.31.1.3 `size_t yyguts_t::yy_buffer_stack_top`

index of top of stack.

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

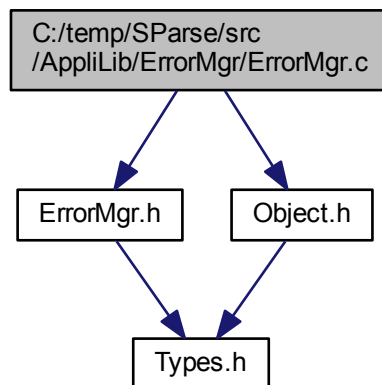
- `C:/temp/SParse/src/ParseLib/Grammar2/Grammar2.lex.c`

Chapter 4

File Documentation

4.1 C:/temp/SParse/src/AppliLib/ErrorMgr/ErrorMgr.c File Reference

```
#include "ErrorMgr.h"  
#include "Object.h"  
Include dependency graph for ErrorMgr.c:
```



Classes

- class `ErrorMgr`

Functions

- PRIVATE `ErrorMgr * ErrorMgr_new ()`

Variables

- PRIVATE `ErrorMgr * errorMgr = 0`

4.1.1 Detailed Description

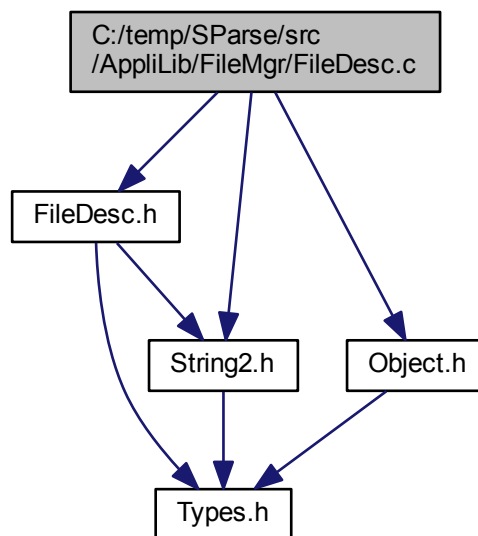
The class `ErrorMgr` is TBD

4.2 C:/temp/SParse/src/AppliLib/FileMgr/FileDesc.c File Reference

The `FileDesc` class describe a File in the FileMgr.

```
#include "FileDesc.h"
#include "String2.h"
#include "Object.h"
```

Include dependency graph for FileDesc.c:



Classes

- struct `FileDesc`

Functions

- PRIVATE `String * FileDesc_getBasename (FileDesc *this)`

4.2.1 Detailed Description

The `FileDesc` class describe a File in the FileMgr. The class `FileDesc` is TBD

4.3 C:/temp/SParse/src/AppliLib/FileMgr/FileMgr.c File Reference

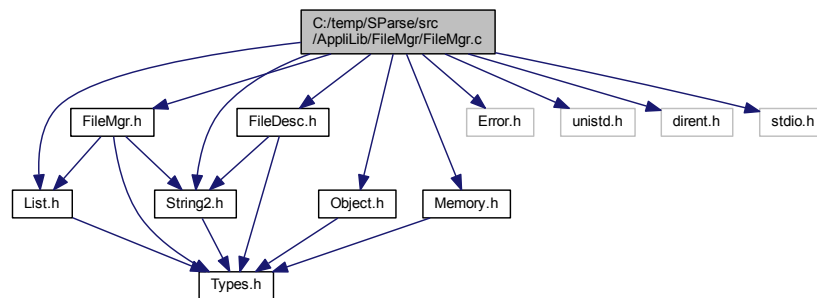
The `FileMgr` class manages a list of files contained in a group of locations.

```

#include "FileMgr.h"
#include "String2.h"
#include "Object.h"
#include "List.h"
#include "FileDesc.h"
#include "Memory.h"
#include "Error.h"
#include <unistd.h>
#include <dirent.h>
#include <stdio.h>

```

Include dependency graph for FileMgr.c:



Classes

- class [FileMgr](#)

Macros

- `#define FILEMGR_MAX_PATH (1024)`

Functions

- PRIVATE void **FileMgr_listFiles** ([FileMgr](#) *this, [String](#) *directory)
- PRIVATE [FileDesc](#) * **FileMgr_isManaged** ([FileMgr](#) *this, [String](#) *fullName)
- PRIVATE unsigned int **FileMgr_existFS** ([FileMgr](#) *this, [String](#) *fullName)

Variables

- PRIVATE [FileMgr](#) * **fileMgr** = 0

4.3.1 Detailed Description

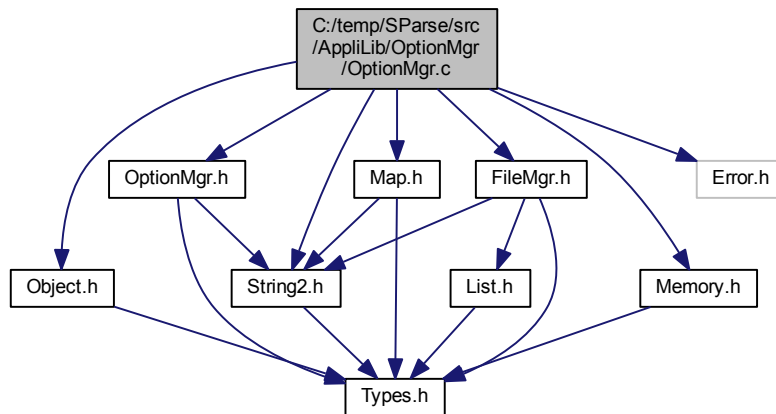
The [FileMgr](#) class manages a list of files contained in a group of locations. The class [FileMgr](#) is TBD

4.4 C:/temp/SParse/src/AppliLib/OptionMgr/OptionMgr.c File Reference

The [OptionMgr](#) class manages the application configuration.

```
#include "OptionMgr.h"
#include "Object.h"
#include "String2.h"
#include "Map.h"
#include "FileMgr.h"
#include "Memory.h"
#include "Error.h"
```

Include dependency graph for OptionMgr.c:



Classes

- class [OptionMgr](#)
- struct [OptionDefault](#)

Functions

- PRIVATE unsigned int **OptionMgr_parseFile** ([OptionMgr](#) *this, [String](#) *fileContent)

Variables

- PRIVATE [OptionMgr](#) * **optionMgr** = 0
- PRIVATE struct [OptionDefault](#) **optionDefault** []

4.4.1 Detailed Description

The [OptionMgr](#) class manages the application configuration. The class [OptionMgr](#) is TBD

4.4.2 Variable Documentation

4.4.2.1 PRIVATE struct OptionDefault optionDefault[]

Initial value:

```

=
{
  {"DB Name", "-o", "test.db"},
  {"Input directory", "-d", "."},
  {"Config file name", "-c", "sparse.txt"}
}

```

4.5 C:/temp/SParse/src/AppliLib/SdbMgr/SdbMgr.c File Reference

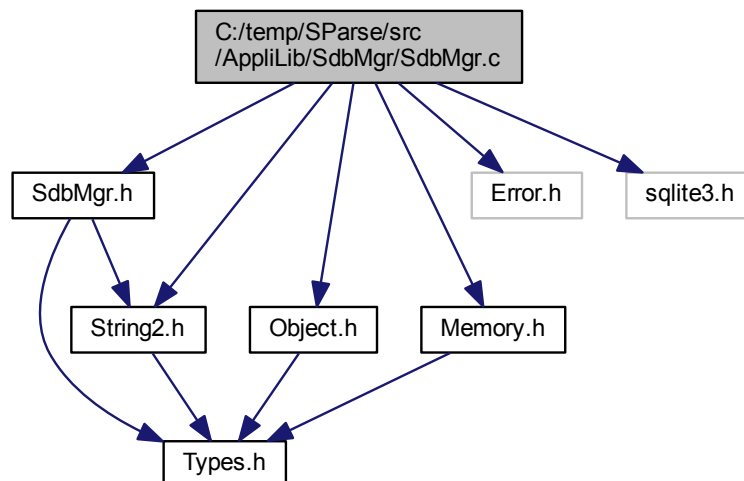
TBD.

```

#include "SdbMgr.h"
#include "Object.h"
#include "String2.h"
#include "Memory.h"
#include "Error.h"
#include <sqlite3.h>

```

Include dependency graph for SdbMgr.c:



Classes

- class [SdbMgr](#)

Functions

- PRIVATE unsigned int **SdbMgr_open** ([SdbMgr](#) *this, [String](#) *sdbName)
- PRIVATE void **SdbMgr_close** ([SdbMgr](#) *this)
- PUBLIC void **SdbMgr_delete** ([SdbMgr](#) *this)
- PUBLIC [SdbMgr](#) * **SdbMgr_copy** ([SdbMgr](#) *this)
- PUBLIC [SdbMgr](#) * **SdbMgr_getRef** ()
- PUBLIC unsigned int **SdbMgr_execute** ([SdbMgr](#) *this, const char *statement, [String](#) ***result)

Variables

- PRIVATE `SdbMgr` * `sdbMgr` = 0

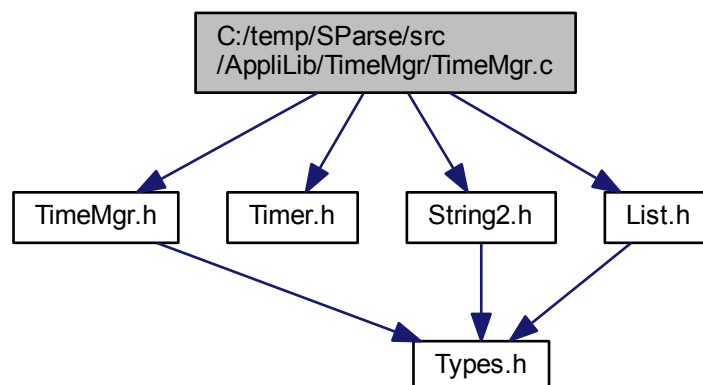
4.5.1 Detailed Description

TBD. TBD

4.6 C:/temp/SParse/src/AppliLib/TimeMgr/TimeMgr.c File Reference

This file implements the class `TimeMgr`.

```
#include "TimeMgr.h"
#include "Timer.h"
#include "String2.h"
#include "List.h"
Include dependency graph for TimeMgr.c:
```



Classes

- class `TimeMgr`

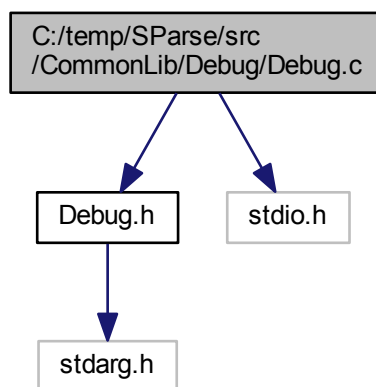
4.6.1 Detailed Description

This file implements the class `TimeMgr`. The class `TimeMgr` is TBD

4.7 C:/temp/SParse/src/CommonLib/Debug/Debug.c File Reference

This file contains debugging functions.

```
#include "Debug.h"
#include <stdio.h>
Include dependency graph for Debug.c:
```



Functions

- void **dbg_printf** (const char *fmt,...)

4.7.1 Detailed Description

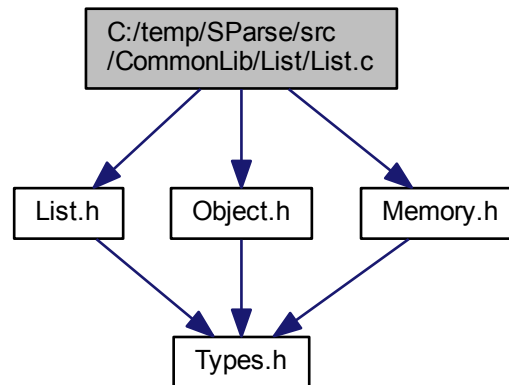
This file contains debugging functions. The debugging function are TBD

4.8 C:/temp/SParse/src/CommonLib/List/List.c File Reference

This file contains the implementation of the class [List](#).

```
#include "List.h"
#include "Object.h"
#include "Memory.h"
```

Include dependency graph for List.c:



Classes

- class [List](#)

Typedefs

- typedef struct ListNode **ListNode**

Functions

- PUBLIC void * **List_getNext** ([List](#) *this)

4.8.1 Detailed Description

This file contains the implementation of the class [List](#). The class [List](#) implement the [List](#) operations:

- init
- add

4.9 C:/temp/SParse/src/CommonLib/Map/Map.c File Reference

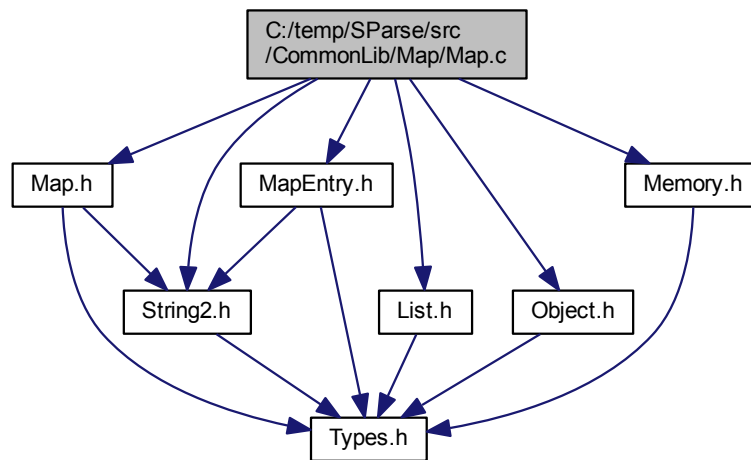
A [Map](#) class. This class provides a container indexed by a string.

```

#include "Map.h"
#include "MapEntry.h"
#include "List.h"
#include "Object.h"
#include "String2.h"
#include "Memory.h"

```


Include dependency graph for Map.c:



Classes

- class [Map](#)

Macros

- `#define HTABLE_SIZE (50)`

Functions

- `PRIVATE MapEntry * Map_findEntry (Map *this, String *s)`

4.9.1 Detailed Description

A [Map](#) class. This class provides a container indexed by a string. A support class for theMap class.

4.10 C:/temp/SParse/src/CommonLib/Memory/Memory.c File Reference

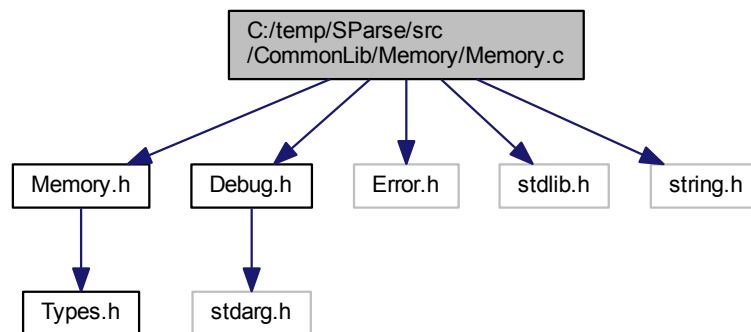
This file provides the implementation of the memory functions.

```

#include "Memory.h"
#include "Debug.h"
#include "Error.h"
#include <stdlib.h>
#include <string.h>

```

Include dependency graph for Memory.c:



Macros

- `#define DEBUG (0)`

Functions

- PUBLIC void * **Memory_alloc** (unsigned int nbBytes)
- PUBLIC void **Memory_free** (void *pointer, unsigned int nbBytes)
- PUBLIC void * **Memory_realloc** (void *pointer, unsigned int prevSizeBytes, unsigned int newSizeBytes)
- PUBLIC void **Memory_set** (void *pointer, unsigned char val, unsigned int nbBytes)
- PUBLIC void **Memory_copy** (void *pointer, void *src, unsigned int nbBytes)
- PUBLIC unsigned int **Memory_ncmp** (void *pointer, void *compared, unsigned int nbBytes)
- PUBLIC unsigned int **Memory_cmp** (void *pointer, void *compared)
- PUBLIC unsigned int **Memory_len** (void *pointer)
- PUBLIC void **Memory_report** ()

Variables

- PRIVATE unsigned int **Memory_allocRequestId** = 0
- PRIVATE unsigned int **Memory_freeRequestId** = 0
- PRIVATE unsigned int **Memory_nbBytesAllocated** = 0

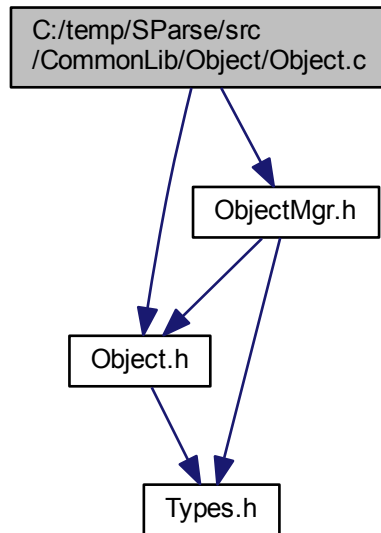
4.10.1 Detailed Description

This file provides the implementation of the memory functions. TBD

4.11 C:/temp/SParse/src/CommonLib/Object/Object.c File Reference

This file contains the implementation for the class [Object](#).

```
#include "Object.h"
#include "ObjectMgr.h"
Include dependency graph for Object.c:
```



Variables

- PRIVATE `ObjectMgr` * `Object_objMgrPtr` = 0

4.11.1 Detailed Description

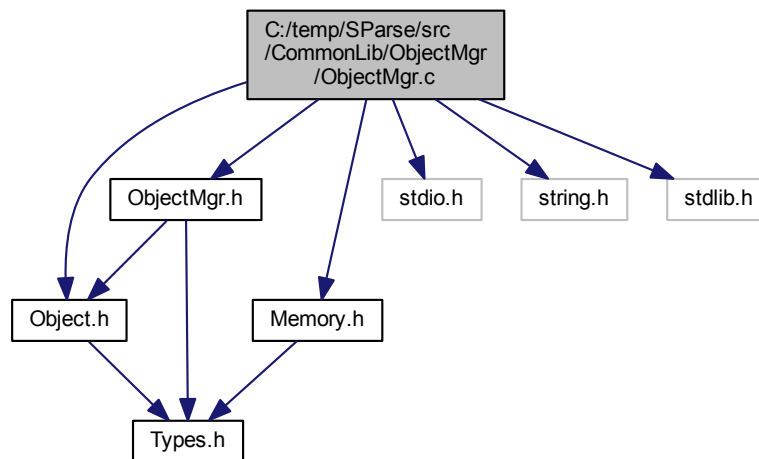
This file contains the implementation for the class `Object`. The class `Object` is TBD

4.12 C:/temp/SParse/src/CommonLib/ObjectMgr/ObjectMgr.c File Reference

An object management class.

```
#include "ObjectMgr.h"
#include "Object.h"
#include "Memory.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ObjectMgr.c:



Classes

- struct [ObjectInfo](#)
- class [ObjectMgr](#)

Macros

- #define **MAX_NB_OBJECTS** (40000)
- #define **END_OF_QUEUE** (0xFFFFFFFF)

Typedefs

- typedef struct [ObjectInfo](#) **ObjectInfo**

Variables

- PRIVATE [ObjectMgr](#) * **objectMgr** = 0

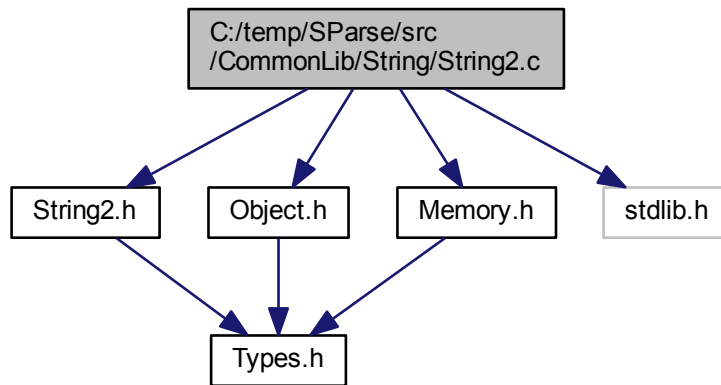
4.12.1 Detailed Description

An object management class. This class provides an object allocation and de-allocation service. Only one instance of this class can be created.

4.13 C:/temp/SParse/src/CommonLib/String/String2.c File Reference

The [String](#) class provide a dynamic array of char terminated by 0.

```
#include "String2.h"
#include "Object.h"
#include "Memory.h"
#include <stdlib.h>
Include dependency graph for String2.c:
```



Classes

- class [String](#)

Functions

- PUBLIC unsigned int **String_getLength** ([String](#) *this)
- PUBLIC char * **String_getBuffer** ([String](#) *this)
- PUBLIC void **String_setBuffer** ([String](#) *this, char *buffer)
- PUBLIC unsigned int **String_isContained** ([String](#) *this, [String](#) *s2)
- PUBLIC unsigned int **String_prepend** ([String](#) *this, const char *prefix)
- PUBLIC unsigned int **String_matchWildcard** ([String](#) *this, const char *wildcard)

4.13.1 Detailed Description

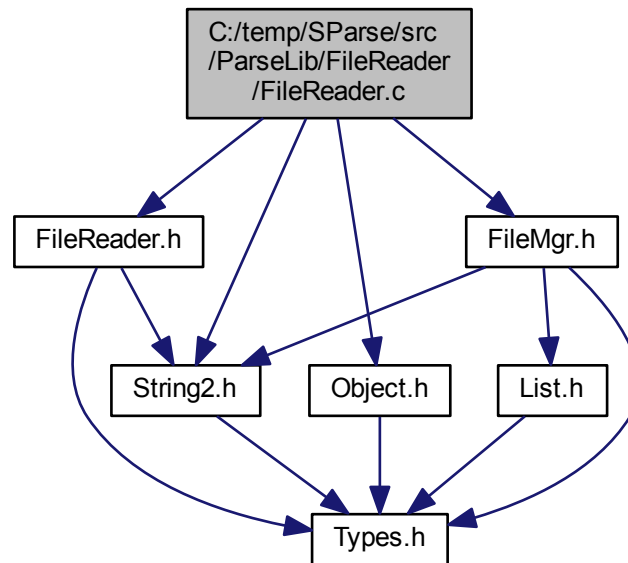
The [String](#) class provide a dynamic array of char terminated by 0. The class [String](#) is TBD

4.14 C:/temp/SParse/src/ParseLib/FileReader/FileReader.c File Reference

This file contains the implementation for the class [FileReader](#).

```
#include "FileReader.h"
#include "Object.h"
#include "String2.h"
#include "FileMgr.h"
```

Include dependency graph for FileReader.c:



Classes

- class `FileReader`

Functions

- PUBLIC `FileReader * FileReader_new (String *fileName)`
- PUBLIC void `FileReader_delete (FileReader *this)`
- PUBLIC `FileReader * FileReader_copy (FileReader *this)`
- PUBLIC char * `FileReader_getBuffer (FileReader *this)`
- PUBLIC `String * FileReader_getName (FileReader *this)`

4.14.1 Detailed Description

This file contains the implementation for the class `FileReader`. The class `FileReader` is TBD

4.15 C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.c File Reference

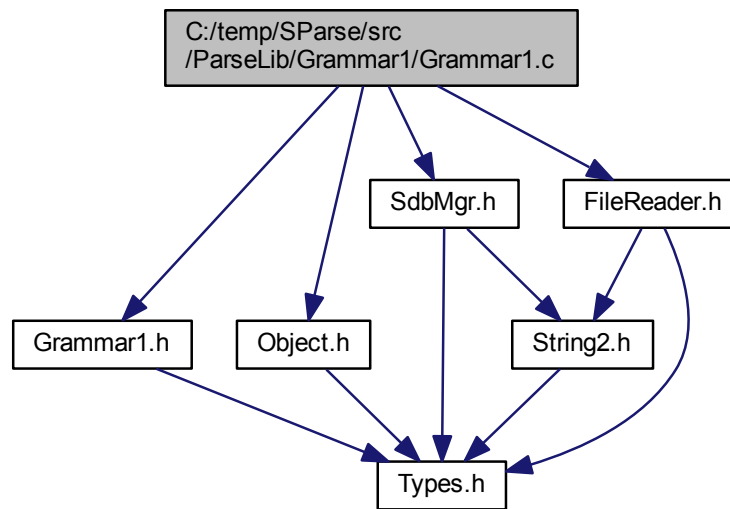
TBD.

```

#include "Grammar1.h"
#include "Object.h"
#include "SdbMgr.h"
#include "FileReader.h"

```

Include dependency graph for Grammar1.c:



Classes

- struct [Grammar1](#)

Functions

- PUBLIC [Grammar1](#) * **Grammar1_new** ([FileReader](#) *f, [SdbMgr](#) *sdbMgr)
- PUBLIC void **Grammar1_delete** ([Grammar1](#) *this)
- PUBLIC [Grammar1](#) * **Grammar_copy** ([Grammar1](#) *this)
- void **Grammar1_process** ([Grammar1](#) *this)
- PUBLIC void **Grammar1_changeBuffer** ([Grammar1](#) *this)

4.15.1 Detailed Description

TBD. TBD

4.16 C:/temp/SParse/src/ParseLib/SParse/SParse.c File Reference

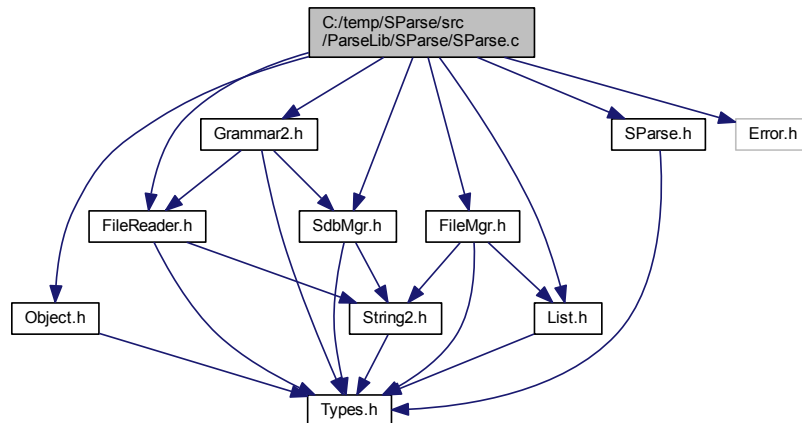
This file contains the implementation for the class [SParse](#).

```

#include "Object.h"
#include "FileReader.h"
#include "SdbMgr.h"
#include "Error.h"
#include "Grammar2.h"
#include "FileMgr.h"
#include "List.h"
#include "SParse.h"

```

Include dependency graph for SParse.c:



Classes

- class [SParse](#)

Typedefs

- typedef struct SParseDefault **SParseDefault**

Functions

- PRIVATE unsigned int **SParse_parseFile** ([SParse](#) *this, [String](#) *file)
- PUBLIC [SParse](#) * **SParse_copy** ([SParse](#) *this)

4.16.1 Detailed Description

This file contains the implementation for the class [SParse](#). The class [SParse](#) parses all files with extension .X and stores the result of the parsing in the SQLite DB name.

Index

C:/temp/SParse/src/AppliLib/ErrorMgr/ErrorMgr.c, 35
C:/temp/SParse/src/AppliLib/FileMgr/FileDesc.c, 36
C:/temp/SParse/src/AppliLib/FileMgr/FileMgr.c, 36
C:/temp/SParse/src/AppliLib/OptionMgr/OptionMgr.c, 37
C:/temp/SParse/src/AppliLib/SdbMgr/SdbMgr.c, 39
C:/temp/SParse/src/AppliLib/TimeMgr/TimeMgr.c, 40
C:/temp/SParse/src/CommonLib/Debug/Debug.c, 40
C:/temp/SParse/src/CommonLib/List/List.c, 41
C:/temp/SParse/src/CommonLib/Map/Map.c, 42
C:/temp/SParse/src/CommonLib/Memory/Memory.c, 43
C:/temp/SParse/src/CommonLib/Object/Object.c, 44
C:/temp/SParse/src/CommonLib/ObjectMgr/ObjectMgr.-
c, 45
C:/temp/SParse/src/CommonLib/String/String2.c, 46
C:/temp/SParse/src/ParseLib/Reader/Reader.c,
47
C:/temp/SParse/src/ParseLib/Grammar1/Grammar1.c,
48
C:/temp/SParse/src/ParseLib/SParse/SParse.c, 49

Error, 5
ErrorMgr, 5
 ErrorMgr_copy, 6
 ErrorMgr_exit, 6
 ErrorMgr_getRef, 6
ErrorMgr_copy
 ErrorMgr, 6
ErrorMgr_exit
 ErrorMgr, 6
ErrorMgr_getRef
 ErrorMgr, 6

FileDesc, 7
FileMgr, 8
 FileMgr_addDirectory, 9
 FileMgr_addFile, 9
 FileMgr_copy, 9
 FileMgr_filterFiles, 9
 FileMgr_getRef, 9
 FileMgr_load, 9
FileMgr_addDirectory
 FileMgr, 9
FileMgr_addFile
 FileMgr, 9
FileMgr_copy
 FileMgr, 9
FileMgr_filterFiles
 FileMgr, 9
FileMgr_getRef
 FileMgr, 9

FileMgr_load
 FileMgr, 9
FileReader, 10
Grammar1, 10
Grammar2, 11
List, 12
 List_copy, 13
 List_forEach, 13
 List_getSize, 13
 List_insertHead, 13
 List_insertTail, 13
 List_merge, 13
 List_new, 13
List_copy
 List, 13
List_forEach
 List, 13
List_getSize
 List, 13
List_insertHead
 List, 13
List_insertTail
 List, 13
List_merge
 List, 13
List_new
 List, 13
Map, 14
 Map_copy, 15
 Map_insert, 15
Map_copy
 Map, 15
Map_insert
 Map, 15
MapEntry, 15
maxNbObjectAllocated
 ObjectMgr, 20
Node, 16
Object, 16
 Object_copy, 16
 Object_getRef, 17
 Object_new, 17
Object_copy
 Object, 16
Object_getRef
 Object, 17

- Object_new
 - Object, [17](#)
- ObjectInfo, [17](#)
- ObjectMgr, [18](#)
 - maxNbObjectAllocated, [20](#)
 - ObjectMgr_allocate, [19](#)
 - ObjectMgr_copy, [20](#)
 - ObjectMgr_deallocate, [20](#)
 - ObjectMgr_getRef, [20](#)
- ObjectMgr_allocate
 - ObjectMgr, [19](#)
- ObjectMgr_copy
 - ObjectMgr, [20](#)
- ObjectMgr_deallocate
 - ObjectMgr, [20](#)
- ObjectMgr_getRef
 - ObjectMgr, [20](#)
- OptionDefault, [20](#)
- optionDefault
 - OptionMgr.c, [38](#)
- OptionMgr, [21](#)
 - OptionMgr_getRef, [22](#)
 - OptionMgr_readFromCmdLine, [22](#)
- OptionMgr.c
 - optionDefault, [38](#)
- OptionMgr_getRef
 - OptionMgr, [22](#)
- OptionMgr_readFromCmdLine
 - OptionMgr, [22](#)
- Point, [22](#)
- SParse, [24](#)
 - SParse_delete, [24](#)
 - SParse_new, [24](#)
 - SParse_parse, [25](#)
- SParse_delete
 - SParse, [24](#)
- SParse_new
 - SParse, [24](#)
- SParse_parse
 - SParse, [25](#)
- SdbMgr, [22](#)
- SdbRequest, [23](#)
- String, [25](#)
 - String_copy, [26](#)
 - String_getRef, [26](#)
 - String_new, [26](#)
- String_copy
 - String, [26](#)
- String_getRef
 - String, [26](#)
- String_new
 - String, [26](#)
- TestClass, [27](#)
- TestFileMgr, [27](#)
- TestItem, [28](#)
- TestObjectMgr, [29](#)
- TestSdbMgr, [30](#)
- TimeMgr, [30](#)
- yy_bs_column
 - yy_buffer_state, [31](#)
- yy_bs_lineno
 - yy_buffer_state, [31](#)
- yy_buffer_stack
 - yyguts_t, [33](#)
- yy_buffer_stack_max
 - yyguts_t, [33](#)
- yy_buffer_stack_top
 - yyguts_t, [34](#)
- yy_buffer_state, [31](#)
 - yy_bs_column, [31](#)
 - yy_bs_lineno, [31](#)
- yy_trans_info, [32](#)
- yyalloc, [32](#)
- yyguts_t, [33](#)
 - yy_buffer_stack, [33](#)
 - yy_buffer_stack_max, [33](#)
 - yy_buffer_stack_top, [34](#)