

# SParse

Generated by Doxygen 1.8.5

Fri Sep 13 2019 10:16:00



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Class Struct Reference	5
3.2	FileDesc Class Reference	6
3.3	FileMgr Class Reference	7
3.3.1	Member Function Documentation	8
3.3.1.1	FileMgr_addDirectory	8
3.3.1.2	FileMgr_addFile	8
3.3.1.3	FileMgr_copy	8
3.3.1.4	FileMgr_filterFiles	8
3.3.1.5	FileMgr_getRef	8
3.3.1.6	FileMgr_getRootLocation	9
3.3.1.7	FileMgr_load	9
3.3.1.8	FileMgr_setRootLocation	9
3.4	FileReader Class Reference	9
3.4.1	Member Function Documentation	10
3.4.1.1	FileReader_addFile	10
3.4.1.2	FileReader_copy	10
3.4.1.3	FileReader_getBuffer	10
3.4.1.4	FileReader_getName	11
3.4.1.5	FileReader_new	11
3.5	Grammar2 Class Reference	11
3.5.1	Member Function Documentation	12
3.5.1.1	Grammar2_copy	12
3.5.1.2	Grammar2_new	12
3.6	GrammarContext Struct Reference	12
3.7	IncludeInfo Struct Reference	13

3.8	List Class Reference	14
3.8.1	Member Function Documentation	15
3.8.1.1	List_copy	15
3.8.1.2	List_forEach	15
3.8.1.3	List_getHead	15
3.8.1.4	List_getSize	15
3.8.1.5	List_insertHead	15
3.8.1.6	List_insertTail	15
3.8.1.7	List_isEqual	16
3.8.1.8	List_merge	16
3.8.1.9	List_new	16
3.8.1.10	List_removeHead	16
3.9	Map Class Reference	17
3.9.1	Member Function Documentation	17
3.9.1.1	Map_copy	17
3.9.1.2	Map_getAll	18
3.9.1.3	Map_insert	18
3.10	MapEntry Struct Reference	18
3.11	Object Struct Reference	19
3.11.1	Member Function Documentation	20
3.11.1.1	Object_copy	20
3.11.1.2	Object_getRef	20
3.11.1.3	Object_isEqual	20
3.11.1.4	Object_new	20
3.11.1.5	Object_print	20
3.12	ObjectInfo Struct Reference	21
3.13	ObjectMgr Class Reference	22
3.13.1	Member Function Documentation	23
3.13.1.1	ObjectMgr_allocate	23
3.13.1.2	ObjectMgr_copy	23
3.13.1.3	ObjectMgr_deallocate	23
3.13.1.4	ObjectMgr_getRef	23
3.13.2	Member Data Documentation	23
3.13.2.1	maxNbObjectAllocated	23
3.14	OptionDefault Struct Reference	23
3.15	OptionMgr Class Reference	24
3.15.1	Member Function Documentation	25
3.15.1.1	OptionMgr_getRef	25
3.15.1.2	OptionMgr_readFromCmdLine	25
3.16	SdbMgr Class Reference	25

3.16.1	Member Function Documentation	26
3.16.1.1	SdbMgr_copy	26
3.16.1.2	SdbMgr_execute	26
3.16.1.3	SdbMgr_getRef	26
3.17	SdbRequest Class Reference	27
3.17.1	Member Function Documentation	27
3.17.1.1	SdbRequest_delete	27
3.17.1.2	SdbRequest_execute	28
3.17.1.3	SdbRequest_new	28
3.18	SParse Class Reference	28
3.18.1	Member Function Documentation	29
3.18.1.1	SParse_delete	29
3.18.1.2	SParse_new	29
3.18.1.3	SParse_parse	29
3.19	String Class Reference	30
3.19.1	Member Function Documentation	30
3.19.1.1	String_copy	30
3.19.1.2	String_getRef	31
3.19.1.3	String_new	31
3.20	TestClass Struct Reference	31
3.21	TestFileMgr Struct Reference	32
3.22	TestItem Struct Reference	33
3.23	TestObjectMgr Struct Reference	34
3.24	TestSdbMgr Struct Reference	35
3.25	TestTimeMgr Struct Reference	36
3.26	TimeMgr Class Reference	37
3.27	Timer Class Reference	38
3.27.1	Member Function Documentation	39
3.27.1.1	Timer_copy	39
3.27.1.2	Timer_new	39
3.28	yy_buffer_state Struct Reference	39
3.28.1	Member Data Documentation	39
3.28.1.1	yy_bs_column	39
3.28.1.2	yy_bs_lineno	39
3.29	yy_trans_info Struct Reference	40
3.30	yyalloc Union Reference	40
3.31	yyguts_t Struct Reference	41
3.31.1	Member Data Documentation	42
3.31.1.1	yy_buffer_stack	42
3.31.1.2	yy_buffer_stack_max	42

3.31.1.3	yy_buffer_stack_top . . . . .	42
3.32	YYSTYPE Union Reference . . . . .	42
<b>4</b>	<b>File Documentation</b>	<b>45</b>
4.1	Debug.c File Reference . . . . .	45
4.1.1	Detailed Description . . . . .	45
4.2	FileDesc.c File Reference . . . . .	45
4.2.1	Detailed Description . . . . .	46
4.2.2	Variable Documentation . . . . .	46
4.2.2.1	fileDescClass . . . . .	46
4.3	FileMgr.c File Reference . . . . .	47
4.3.1	Detailed Description . . . . .	48
4.3.2	Variable Documentation . . . . .	48
4.3.2.1	fileMgrClass . . . . .	48
4.4	FileReader.c File Reference . . . . .	48
4.4.1	Detailed Description . . . . .	49
4.4.2	Variable Documentation . . . . .	49
4.4.2.1	fileReaderClass . . . . .	49
4.5	Grammar2.c File Reference . . . . .	49
4.5.1	Detailed Description . . . . .	51
4.5.2	Variable Documentation . . . . .	51
4.5.2.1	grammar2Class . . . . .	51
4.6	List.c File Reference . . . . .	51
4.6.1	Detailed Description . . . . .	52
4.6.2	Variable Documentation . . . . .	53
4.6.2.1	listClass . . . . .	53
4.7	Map.c File Reference . . . . .	53
4.7.1	Detailed Description . . . . .	54
4.7.2	Variable Documentation . . . . .	54
4.7.2.1	mapClass . . . . .	54
4.8	Memory.c File Reference . . . . .	54
4.8.1	Detailed Description . . . . .	55
4.9	Object.c File Reference . . . . .	55
4.9.1	Detailed Description . . . . .	56
4.10	ObjectMgr.c File Reference . . . . .	56
4.10.1	Detailed Description . . . . .	57
4.11	OptionMgr.c File Reference . . . . .	57
4.11.1	Detailed Description . . . . .	58
4.11.2	Variable Documentation . . . . .	58
4.11.2.1	optionDefault . . . . .	58

4.11.2.2 optionMgrClass . . . . .	59
4.12 SdbMgr.c File Reference . . . . .	59
4.12.1 Detailed Description . . . . .	60
4.12.2 Variable Documentation . . . . .	60
4.12.2.1 sdbMgrClass . . . . .	60
4.13 SParse.c File Reference . . . . .	60
4.13.1 Detailed Description . . . . .	61
4.13.2 Variable Documentation . . . . .	61
4.13.2.1 sparseClass . . . . .	61
4.14 String2.c File Reference . . . . .	61
4.14.1 Detailed Description . . . . .	62
4.14.2 Variable Documentation . . . . .	62
4.14.2.1 stringClass . . . . .	62
4.15 TimeMgr.c File Reference . . . . .	63
4.15.1 Detailed Description . . . . .	64
4.15.2 Variable Documentation . . . . .	64
4.15.2.1 timeMgrClass . . . . .	64
<b>Index</b>	<b>65</b>





# Chapter 1

## Class Index

### 1.1 Class List

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

Class	5
FileDesc	6
FileMgr	7
FileReader	9
Grammar2	11
GrammarContext	12
IncludeInfo	13
List	14
Map	17
MapEntry	18
Object	19
ObjectInfo	21
ObjectMgr	22
OptionDefault	23
OptionMgr	24
SdbMgr	25
SdbRequest	27
SParse	28
String	30
TestClass	31
TestFileMgr	32
TestItem	33
TestObjectMgr	34
TestSdbMgr	35
TestTimeMgr	36
TimeMgr	37
Timer	38
yy_buffer_state	39
yy_trans_info	40
yyalloc	40
yyguts_t	41
YYSTYPE	42



## Chapter 2

# File Index

### 2.1 File List

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

<b>Class.h</b>	??
<a href="#">Debug.c</a>	
This file contains debugging functions	45
<b>Debug.h</b>	??
<b>Error.h</b>	??
<a href="#">FileDesc.c</a>	
The <a href="#">FileDesc</a> class describe a File in the FileMgr	45
<b>FileDesc.h</b>	??
<a href="#">FileMgr.c</a>	
The <a href="#">FileMgr</a> class manages a list of files contained in a group of locations	47
<b>FileMgr.h</b>	??
<a href="#">FileReader.c</a>	
This file contains the implementation for the class <a href="#">FileReader</a>	48
<b>FileReader.h</b>	??
<a href="#">Grammar2.c</a>	
A test grammar class. The <a href="#">Grammar2</a> class implements a test grammar parsing comments code blocks and allowing includes files	49
<b>Grammar2.h</b>	??
<b>Grammar2.parse.h</b>	??
<a href="#">List.c</a>	
This file contains the implementation of the class <a href="#">List</a>	51
<b>List.h</b>	??
<a href="#">Map.c</a>	
A <a href="#">Map</a> class. This class provides a container indexed by a string	53
<b>Map.h</b>	??
<b>MapEntry.h</b>	??
<a href="#">Memory.c</a>	
This file provides the implementation of the memory functions	54
<b>Memory.h</b>	??
<a href="#">Object.c</a>	
This file contains the implementation for the class <a href="#">Object</a>	55
<b>Object.h</b>	??
<a href="#">ObjectMgr.c</a>	
An object management class	56
<b>ObjectMgr.h</b>	??
<a href="#">OptionMgr.c</a>	
The <a href="#">OptionMgr</a> class manages the application configuration	57
<b>OptionMgr.h</b>	??

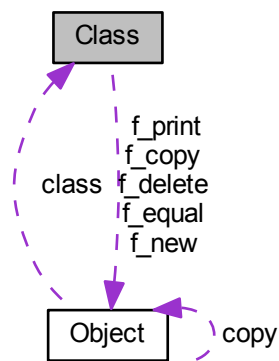
<a href="#">SdbMgr.c</a>	
TBD . . . . .	59
<b>SdbMgr.h</b> . . . . .	??
<b>SdbRequest.h</b> . . . . .	??
<a href="#">SParse.c</a>	
This file contains the implementation for the class <a href="#">SParse</a> . . . . .	60
<b>SParse.h</b> . . . . .	??
<a href="#">String2.c</a>	
The <a href="#">String</a> class provide a dynamic array of char terminated by 0 . . . . .	61
<b>String2.h</b> . . . . .	??
<a href="#">TimeMgr.c</a>	
This file implements the class <a href="#">TimeMgr</a> . . . . .	63
<b>TimeMgr.h</b> . . . . .	??
<b>Timer.h</b> . . . . .	??
<b>Types.h</b> . . . . .	??

## Chapter 3

# Class Documentation

### 3.1 Class Struct Reference

Collaboration diagram for Class:



#### Public Attributes

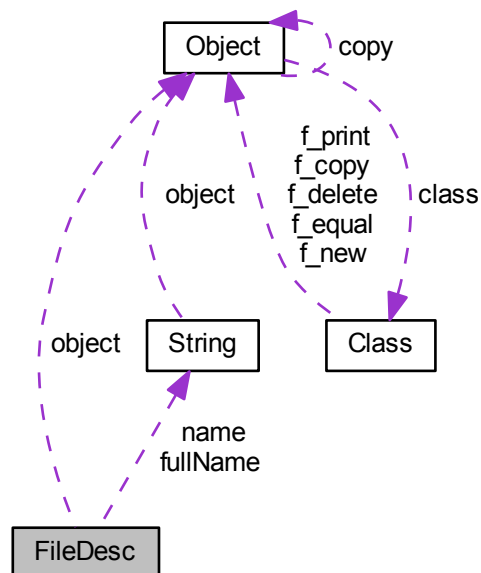
- Constructor **f\_new**
- Destructor **f\_delete**
- Copy\_Operator **f\_copy**
- Equal\_Operator **f\_equal**
- Printer **f\_print**

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

- Class.h

## 3.2 FileDesc Class 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

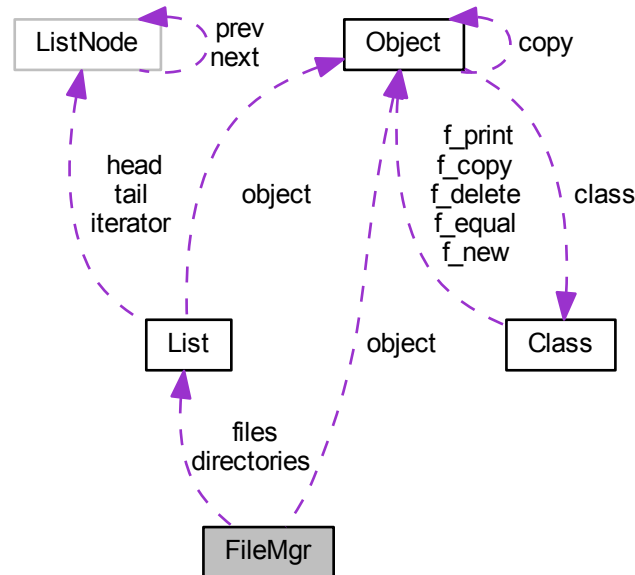
- `Object object`
- `String * name`
- `String * fullName`

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

- `FileDesc.c`

### 3.3 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\\_setRootLocation](#) ([FileMgr](#) \*this, const char \*location)  
*TBD.*
- PUBLIC char \* [FileMgr\\_getRootLocation](#) ([FileMgr](#) \*this)  
*TBD.*
- PUBLIC unsigned int [FileMgr\\_addDirectory](#) ([FileMgr](#) \*this, const char \*directoryName)  
*Add all files in the given directory to the list of managed files.*
- PUBLIC [String](#) \* [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]

## 3.3.1 Member Function Documentation

### 3.3.1.1 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.

### 3.3.1.2 PUBLIC String \* FileMgr\_addFile ( FileMgr \* *this*, const char \* *fileName* )

Add a files to the list of managed files.

#### Returns

Status.

### 3.3.1.3 PUBLIC FileMgr \* FileMgr\_copy ( FileMgr \* *this* )

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

#### Returns

New instance

### 3.3.1.4 PUBLIC List \* FileMgr\_filterFiles ( FileMgr \* *this*, const char \* *pattern* )

TBD.

#### Returns

TBD

### 3.3.1.5 PUBLIC FileMgr \* FileMgr\_getRef ( )

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

#### Returns

Reference to the singleton.



### 3.3.1.6 PUBLIC char \* FileMgr\_getRootLocation ( FileMgr \* *this* )

TBD.

Returns

Status.

### 3.3.1.7 PUBLIC String \* FileMgr\_load ( FileMgr \* *this*, const char \* *fileName* )

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

File Name

Returns

Content of file.

### 3.3.1.8 PUBLIC unsigned int FileMgr\_setRootLocation ( FileMgr \* *this*, const char \* *location* )

TBD.

Returns

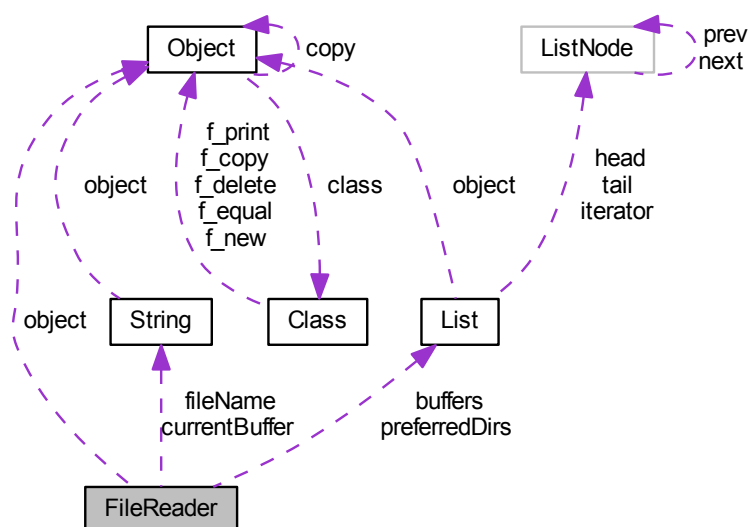
Status.

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

- [FileMgr.c](#)

## 3.4 FileReader Class Reference

Collaboration diagram for FileReader:



## Public Member Functions

- PUBLIC [FileReader](#) \* [FileReader\\_new](#) ([String](#) \*fileName)  
*Create a new [FileReader](#) object.*
- PUBLIC void [FileReader\\_delete](#) ([FileReader](#) \*this)  
*Delete an instance of a [FileReader](#) object.*
- PUBLIC [FileReader](#) \* [FileReader\\_copy](#) ([FileReader](#) \*this)  
*Copy an instance of a [FileReader](#) object.*
- PUBLIC char \* [FileReader\\_getBuffer](#) ([FileReader](#) \*this)  
*Returns the buffer of a [FileReader](#) object.*
- PUBLIC [String](#) \* [FileReader\\_getName](#) ([FileReader](#) \*this)  
*Returns the name of a [FileReader](#) object.*
- PUBLIC char \* [FileReader\\_addFile](#) ([FileReader](#) \*this, [String](#) \*fileName)  
*Add a new file buffer for filename.*

## Public Attributes

- [Object](#) **object**
- [List](#) \* **buffers**
- [String](#) \* **fileName**
- [String](#) \* **currentBuffer**
- [List](#) \* **preferredDirs**

### 3.4.1 Member Function Documentation

#### 3.4.1.1 PUBLIC char \* [FileReader\\_addFile](#) ( [FileReader](#) \* *this*, [String](#) \* *fileName* )

Add a new file buffer for filename.

##### Returns

File buffer

#### 3.4.1.2 PUBLIC [FileReader](#) \* [FileReader\\_copy](#) ( [FileReader](#) \* *this* )

Copy an instance of a [FileReader](#) object.

##### Returns

New instance

#### 3.4.1.3 PUBLIC char \* [FileReader\\_getBuffer](#) ( [FileReader](#) \* *this* )

Returns the buffer of a [FileReader](#) object.

##### Returns

Buffer of characters

#### 3.4.1.4 PUBLIC String \* FileReader\_getName ( FileReader \* this )

Returns the name of a [FileReader](#) object.

Returns

File name

#### 3.4.1.5 PUBLIC FileReader \* FileReader\_new ( String \* fileName )

Create a new [FileReader](#) object.

Returns

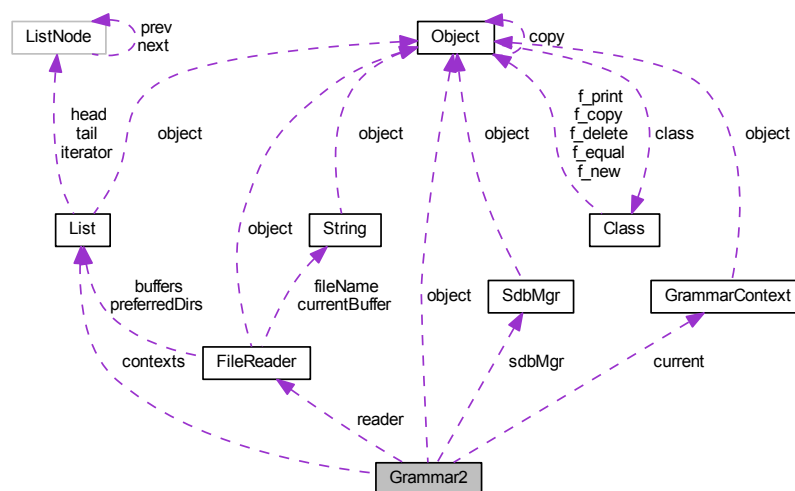
Created [FileReader](#) object.

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

- [FileReader.c](#)

## 3.5 Grammar2 Class Reference

Collaboration diagram for Grammar2:



### Public Member Functions

- PUBLIC [Grammar2](#) \* [Grammar2\\_new](#) ([FileReader](#) \*fr, [SdbMgr](#) \*sdbMgr)  
Create an instance of the class [Grammar2](#).
- PUBLIC void [Grammar2\\_delete](#) ([Grammar2](#) \*this)  
Delete an instance of the class [Grammar2](#).
- PUBLIC [Grammar2](#) \* [Grammar2\\_copy](#) ([Grammar2](#) \*this)  
Copy an instance of the class [Grammar2](#).

## Public Attributes

- [Object](#) **object**
- void \* **scanner**
- [SdbMgr](#) \* **sdbMgr**
- [FileReader](#) \* **reader**
- char **buffer** [MAX\_BUFFER\_SIZE]
- int **node\_text\_position**
- [GrammarContext](#) \* **current**
- [List](#) \* **contexts**

### 3.5.1 Member Function Documentation

#### 3.5.1.1 PUBLIC Grammar2 \* Grammar2\_copy ( Grammar2 \* *this* )

Copy an instance of the class [Grammar2](#).

##### Returns

Copied instance.

#### 3.5.1.2 PUBLIC Grammar2 \* Grammar2\_new ( FileReader \* *fr*, SdbMgr \* *sdbMgr* )

Create an instance of the class [Grammar2](#).

##### Returns

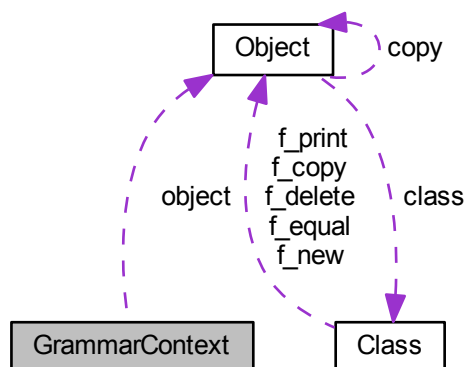
New instance.

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

- [Grammar2.c](#)

## 3.6 GrammarContext Struct Reference

Collaboration diagram for GrammarContext:



## Public Attributes

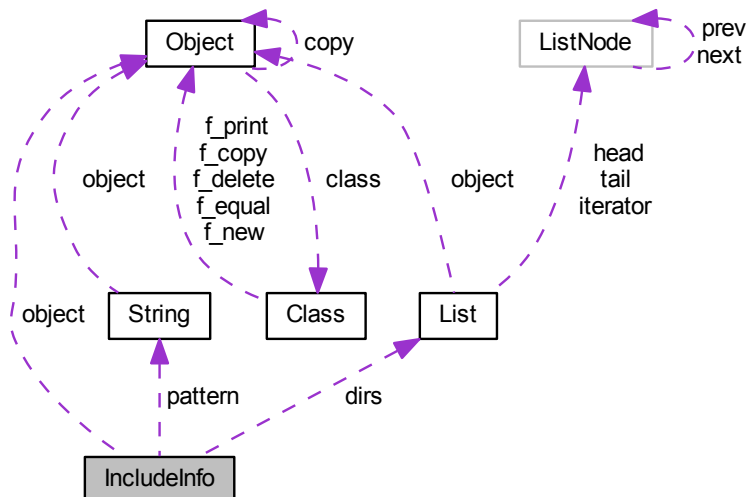
- [Object](#) **object**
- unsigned int **lastNode**

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

- [Grammar2.c](#)

## 3.7 IncludeInfo Struct Reference

Collaboration diagram for IncludeInfo:



## Public Attributes

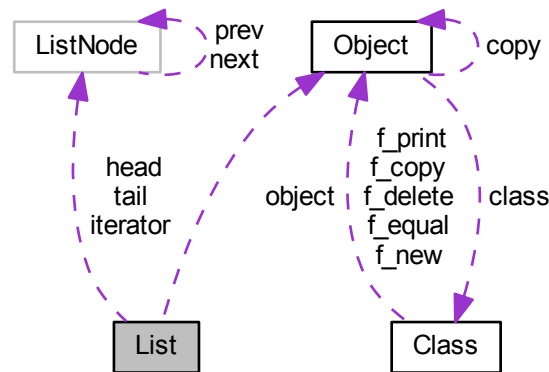
- [Object](#) **object**
- [String](#) \* **pattern**
- [List](#) \* **dirs**

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

- [FileReader.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 unsigned int [List\\_isEqual](#) ([List](#) \*this, [List](#) \*compared)  
*Compare 2 instances of the class [List](#).*
- PUBLIC void [List\\_print](#) ([List](#) \*this)  
*Print 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 void \* [List\\_removeHead](#) ([List](#) \*this)  
*Remove the head item in an instance of [List](#).*
- PUBLIC void \* [List\\_getHead](#) ([List](#) \*this)  
*Get the head item in an insatnce of [List](#).*

## 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 void \* List\_getHead ( List \* *this* )

Get the head item in an insatnce of Llst.

#### 3.8.1.4 PUBLIC unsigned int List\_getSize ( List \* *this* )

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

##### Returns

Number of items.

#### 3.8.1.5 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.
-----------	-------------	--------------------

#### 3.8.1.6 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.
-----------	-------------	--------------------

**3.8.1.7 PUBLIC unsigned int List\_isEqual ( List \* *this*, List \* *compared* )**

Compare 2 instances of the class [List](#).

**Returns**

0 if different, 1 if equal.

**3.8.1.8 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.
-----------	-----------	-----------------------------

**3.8.1.9 PUBLIC List \* List\_new ( )**

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

**Returns**

New instance.

**3.8.1.10 PUBLIC void \* List\_removeHead ( List \* *this* )**

Remove the head item in an instance of Llist.

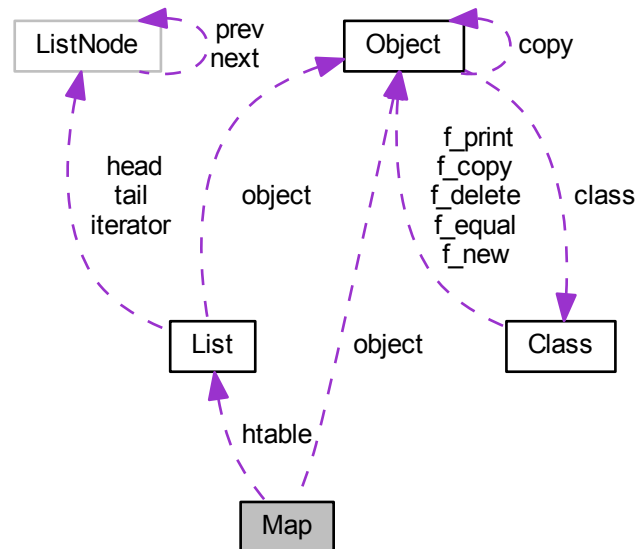
The documentation for this class was generated from the following file:

- [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 List * Map_getAll (Map *this)`  
Get all the entries in an instance of a `Map`.

### 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 List \* Map\_getAll ( Map \* this )**

Get all the entries in an instance of a [Map](#).

**Returns**

[List](#) of map

**3.9.1.3 PUBLIC unsigned int Map\_insert ( Map \* this, String \* s, void \* p )**

TBD.

**Returns**

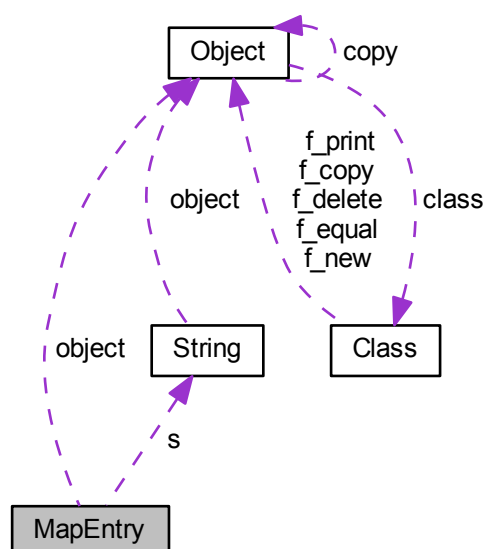
TBD

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

- [Map.c](#)

## 3.10 MapEntry Struct Reference

Collaboration diagram for MapEntry:



### Public Attributes

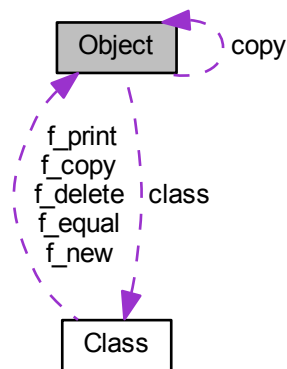
- [Object](#) **object**
- [String](#) \* **s**
- void \* **item**

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

- MapEntry.c

## 3.11 Object Struct Reference

Collaboration diagram for Object:



### Public Member Functions

- PUBLIC [Object](#) \* [Object\\_new](#) (unsigned int size, [Class](#) \*class)  
*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 unsigned int [Object\\_isEqual](#) ([Object](#) \*this, [Object](#) \*compared)  
*Compare 2 instances of the class [Object](#).*
- PUBLIC char \* [Object\\_print](#) ([Object](#) \*this)  
*Print an instance of the class [Object](#) into a buffer of characters.*
- PUBLIC [Object](#) \* [Object\\_getRef](#) ([Object](#) \*this)  
*Get a reference to an instance of the class [Object](#).*

### Public Attributes

- unsigned int **id**
- [Class](#) \* **class**

- void(\* **delete** )(Object \*this)
- Object \*(\* **copy** )(Object \*this)
- unsigned int **refCount**
- unsigned int **size**

### 3.11.1 Member Function Documentation

#### 3.11.1.1 PUBLIC Object \* Object\_copy ( Object \* *this* )

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

##### Returns

New instance

#### 3.11.1.2 PUBLIC Object \* Object\_getRef ( Object \* *this* )

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

##### Returns

Reference to instance

#### 3.11.1.3 PUBLIC unsigned int Object\_isEqual ( Object \* *this*, Object \* *compared* )

Compare 2 instances of the class [Object](#).

##### Returns

1 if equal, 0 else.

#### 3.11.1.4 PUBLIC Object \* Object\_new ( unsigned int *size*, Class \* *class* )

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

##### Parameters

in	<a href="#">Class</a>	to instantiate
----	-----------------------	----------------

#### 3.11.1.5 PUBLIC char \* Object\_print ( Object \* *this* )

Print an instance of the class [Object](#) into a buffer of characters.

##### Returns

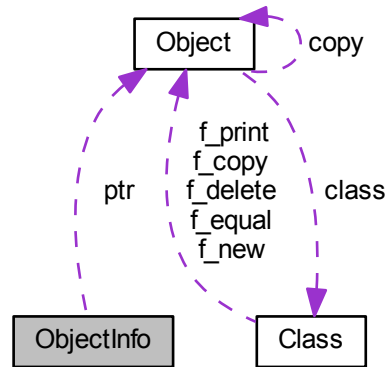
Buffer of characters

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

- [Object.h](#)
- [Object.c](#)

## 3.12 ObjectInfo Struct Reference

Collaboration diagram for ObjectInfo:



### Public Attributes

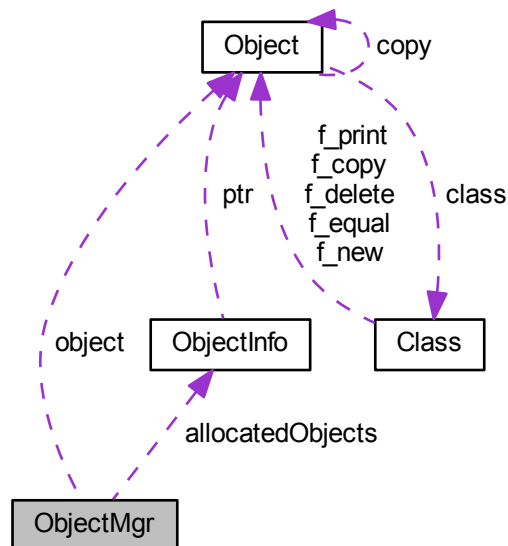
- [Object](#) \* **ptr**
- unsigned int **prevId**
- unsigned int **nextId**

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

- [ObjectMgr.c](#)
- `CommonLib/ObjectMgr/tests/main.c`

### 3.13 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`

### 3.13.1 Member Function Documentation

#### 3.13.1.1 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](#).

#### 3.13.1.2 PUBLIC ObjectMgr \* ObjectMgr\_copy ( ObjectMgr \* *this* )

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

##### Returns

New instance

#### 3.13.1.3 PUBLIC void ObjectMgr\_deallocate ( ObjectMgr \* *this*, Object \* *object* )

De Allocate a given object.

##### Parameters

<i>in</i>	<i>object</i>	Reference to instance of <a href="#">Object</a> .
-----------	---------------	---

#### 3.13.1.4 PUBLIC ObjectMgr \* ObjectMgr\_getRef ( )

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

##### Returns

Reference to the singleton.

### 3.13.2 Member Data Documentation

#### 3.13.2.1 unsigned int ObjectMgr::maxNbObjectAllocated

This is member B

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

- [ObjectMgr.c](#)

## 3.14 OptionDefault Struct Reference

### Public Attributes

- char \* **name**
- char \* **flag**

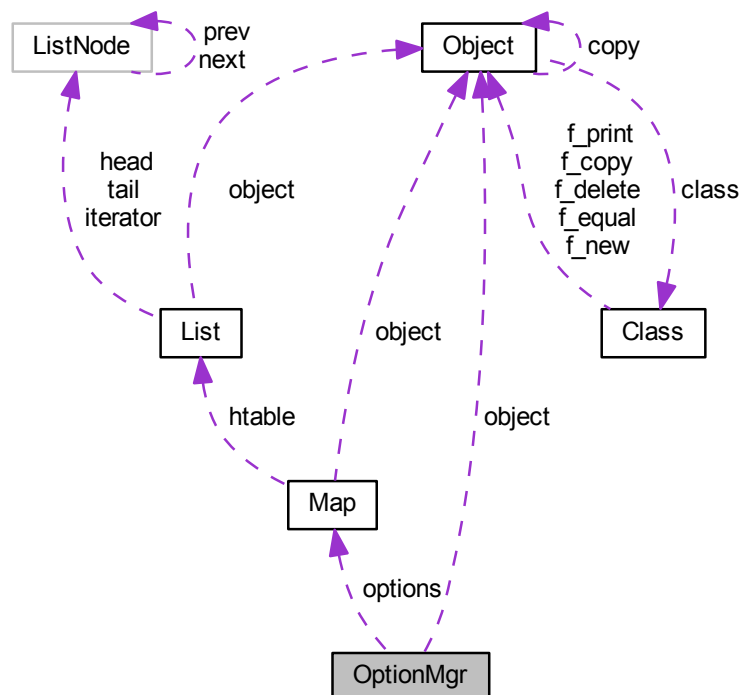
- char \* **value**

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

- [OptionMgr.c](#)

### 3.15 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

### 3.15.1 Member Function Documentation

#### 3.15.1.1 PUBLIC [OptionMgr](#) \* [OptionMgr\\_getRef](#) ( )

TBD.

TBD

#### 3.15.1.2 PUBLIC unsigned int [OptionMgr\\_readFromCmdLine](#) ( [OptionMgr](#) \* this, const int argc, const char \*\* argv )

TBD.

#### Parameters

in	argc	Number of commandline arguments.
in	argv	<a href="#">List</a> os commandline arguments.

#### Returns

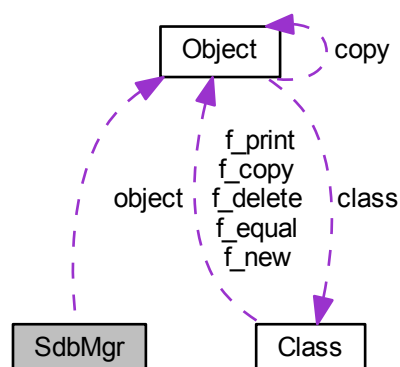
Status of operation.

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

- [OptionMgr.c](#)

## 3.16 SdbMgr Class Reference

Collaboration diagram for SdbMgr:



## Public Member Functions

- PUBLIC void [SdbMgr\\_delete](#) ([SdbMgr](#) \*this)  
*Destroy an instance of the class [SdbMgr](#).*
- PUBLIC [SdbMgr](#) \* [SdbMgr\\_copy](#) ([SdbMgr](#) \*this)  
*Create a copy of an [SdbMgr](#) object.*
- PUBLIC [SdbMgr](#) \* [SdbMgr\\_getRef](#) ()  
*Get a reference to an object.*
- PUBLIC unsigned int [SdbMgr\\_execute](#) ([SdbMgr](#) \*this, const char \*statement, [List](#) \*result)  
*Execute a Sdb request.*

## Public Attributes

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

### 3.16.1 Member Function Documentation

#### 3.16.1.1 PUBLIC [SdbMgr](#) \* [SdbMgr\\_copy](#) ( [SdbMgr](#) \* *this* )

Create a copy of an [SdbMgr](#) object.

##### Returns

A copy of the [SdbMgr](#) object.

#### 3.16.1.2 PUBLIC unsigned int [SdbMgr\\_execute](#) ( [SdbMgr](#) \* *this*, const char \* *statement*, [List](#) \* *result* )

Execute a Sdb request.

##### Returns

status

#### 3.16.1.3 PUBLIC [SdbMgr](#) \* [SdbMgr\\_getRef](#) ( )

Get a reference to an object.

##### Returns

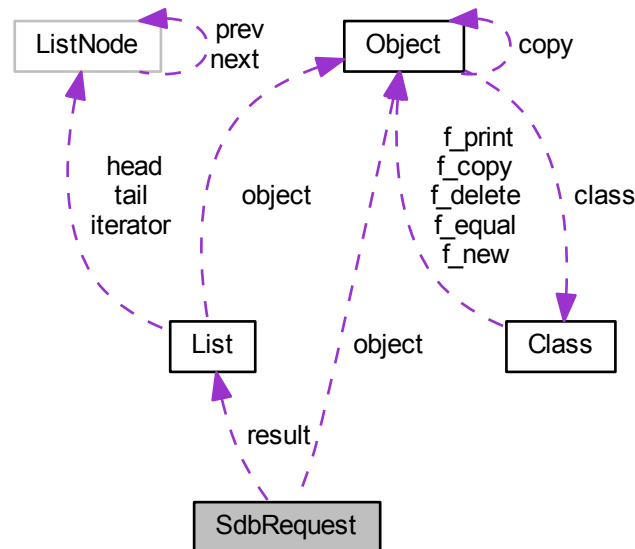
A reference to a [SdbMgr](#) object.

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

- [SdbMgr.c](#)

## 3.17 SdbRequest Class Reference

Collaboration diagram for SdbRequest:



### Public Member Functions

- PUBLIC `SdbRequest * SdbRequest_new` (const char \*fmt)  
Create a new `SdbRequest` instance.
- PUBLIC void `SdbRequest_delete` (`SdbRequest *this`)  
Create a new `SdbRequest` instance.
- PUBLIC void `SdbRequest_execute` (`SdbRequest *this,...`)  
Execute a `SdbRequest`.

### Public Attributes

- `Object object`
- char \* **buffer**
- unsigned int **size**
- const char \* **fmt**
- `List * result`
- unsigned int **nbResults**
- unsigned int **nbColumns**

### 3.17.1 Member Function Documentation

#### 3.17.1.1 PUBLIC void SdbRequest\_delete ( SdbRequest \* this )

Create a new `SdbRequest` instance.

SQL statement template

**Returns**

Instance of an [SdbRequest](#)

**3.17.1.2** `PUBLIC void SdbRequest_execute ( SdbRequest * this, ... )`

Execute a [SdbRequest](#).

Variable list of parameter to use with SQL template

**Returns**

Instance of an [SdbRequest](#)

**3.17.1.3** `PUBLIC SdbRequest * SdbRequest_new ( const char * fmt )`

Create a new [SdbRequest](#) instance.

SQL statement template

**Returns**

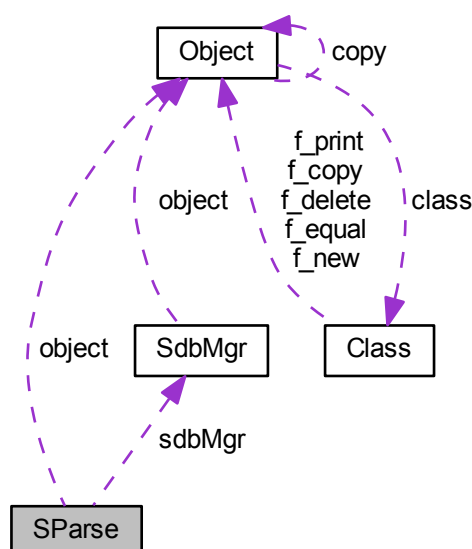
Instance of an [SdbRequest](#)

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

- SdbRequest.c

## 3.18 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**

### 3.18.1 Member Function Documentation

#### 3.18.1.1 PUBLIC void [SParse\\_delete](#) ( [SParse](#) \* *this* )

Delete a [SParse](#) object.

Parameters

<a href="#">Object</a>	to delete.
------------------------	------------

#### 3.18.1.2 PUBLIC [SParse](#) \* [SParse\\_new](#) ( String \* *sdbName* )

Create a new [SParse](#) object.

Returns

New [SParse](#) object.

#### 3.18.1.3 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

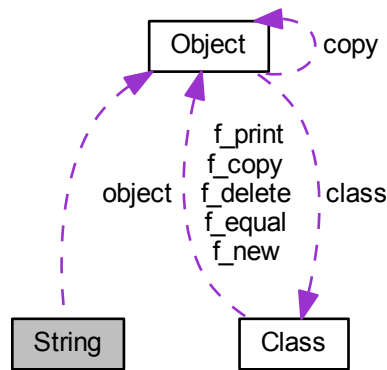
Status of the operation.

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

- [SParse.c](#)

### 3.19 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`

#### 3.19.1 Member Function Documentation

##### 3.19.1.1 PUBLIC `String * String_copy` ( `String * this` )

Copy an instance of class `String`.

## Returns

Copy of instance.

3.19.1.2 `PUBLIC String * String_getRef ( String * this )`

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

## Returns

Copy of instance.

3.19.1.3 `PUBLIC String * String_new ( const char * initString )`

Create a new [String](#) object.

## Returns

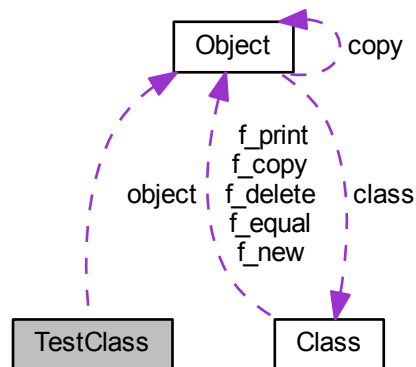
Created [String](#) object.

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

- [String2.c](#)
- ParseLib/Grammar2/main.c

## 3.20 TestClass Struct Reference

Collaboration diagram for TestClass:



## Public Attributes

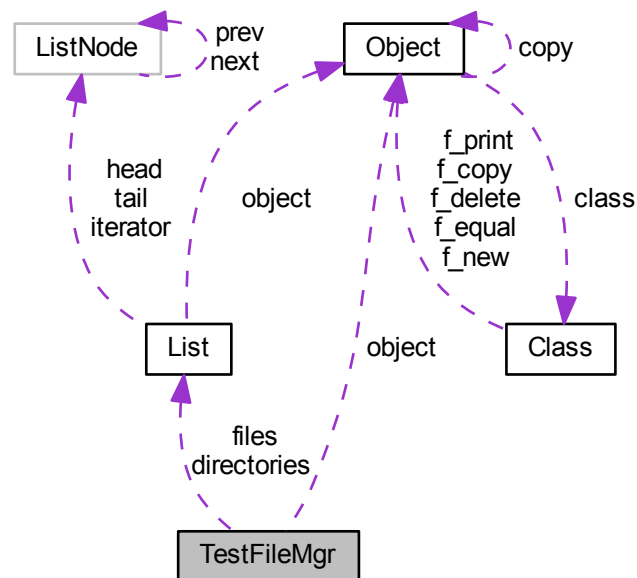
- [Object](#) **object**
- int **testValue**

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

- test1.c

### 3.21 TestFileMgr Struct Reference

Collaboration diagram for TestFileMgr:



#### Public Attributes

- [Object](#) **object**
- [List](#) \* **files**
- [List](#) \* **directories**
- char **rootLocation** [FILEMGR\_MAX\_PATH]

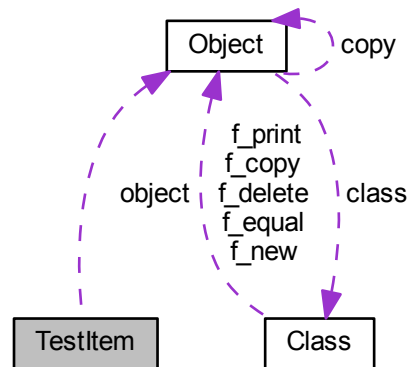
The documentation for this struct was generated from the following file:

- AppliLib/FileMgr/tests/main.c



## 3.22 TestItem Struct Reference

Collaboration diagram for TestItem:



### Public Attributes

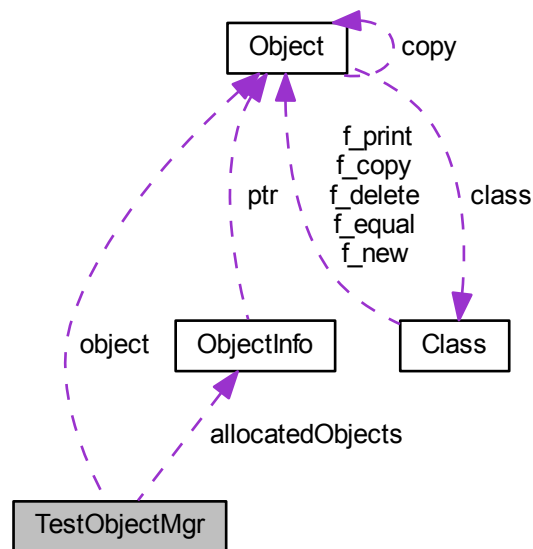
- [Object](#) **object**
- int **x**
- int **y**

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

- CommonLib/List/tests/main.c

### 3.23 TestObjectMgr Struct Reference

Collaboration diagram for TestObjectMgr:



#### Public Attributes

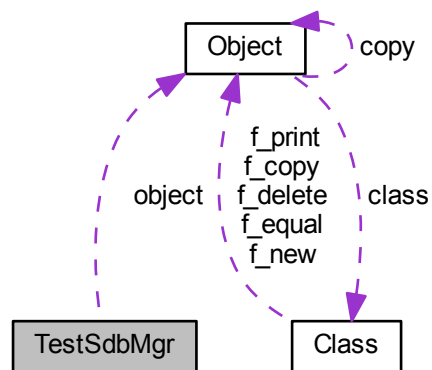
- [Object](#) `object`
- unsigned int `nbBytesAllocated`
- unsigned int `maxNbBytesAllocated`
- unsigned int `allocRequestId`
- unsigned int `freeRequestId`
- unsigned int `nbAllocatedObjects`
- [ObjectInfo](#) `allocatedObjects` [MAX\_NB\_OBJECTS]
- unsigned int `freeSpace`
- unsigned int `usedSpace`

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

- `CommonLib/ObjectMgr/tests/main.c`

## 3.24 TestSdbMgr Struct Reference

Collaboration diagram for TestSdbMgr:



### Public Attributes

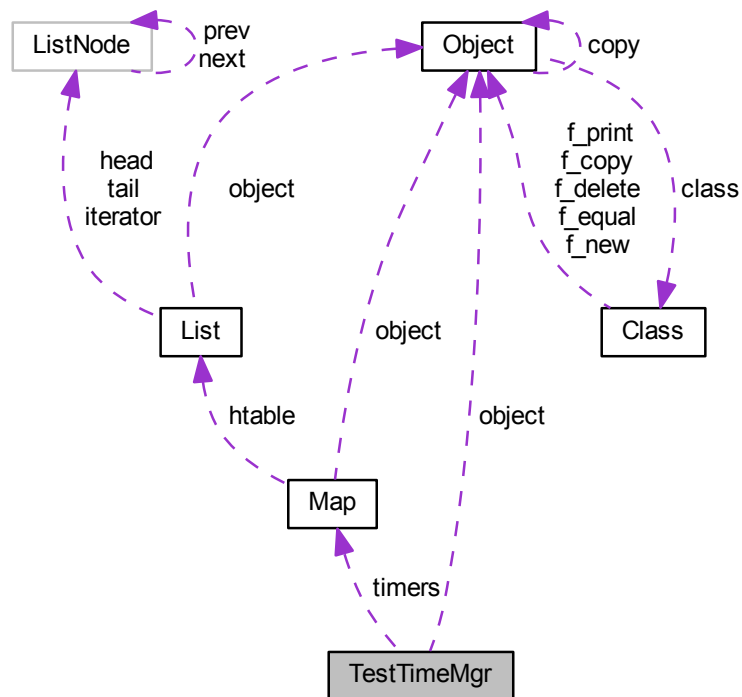
- [Object](#) `object`

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

- `AppliLib/SdbMgr/tests/main.c`

### 3.25 TestTimeMgr Struct Reference

Collaboration diagram for TestTimeMgr:



#### Public Attributes

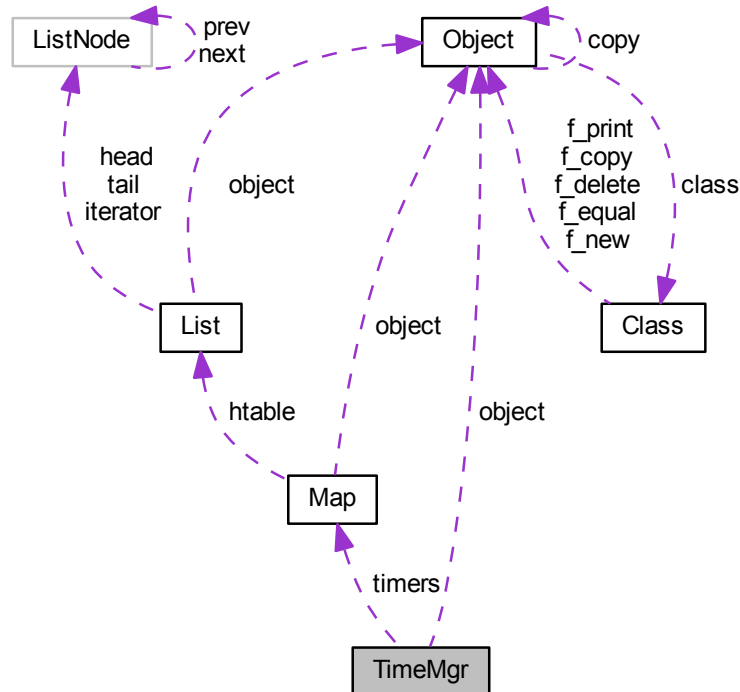
- [Object](#) **object**
- [Map](#) \* **timers**

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

- AppliLib/TimeMgr/tests/main.c

## 3.26 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\\_latchTime](#) (TimeMgr \*this, String \*s)  
*TBD.*
- PUBLIC void [TimeMgr\\_report](#) (TimeMgr \*this)  
*TBD.*

### Public Attributes

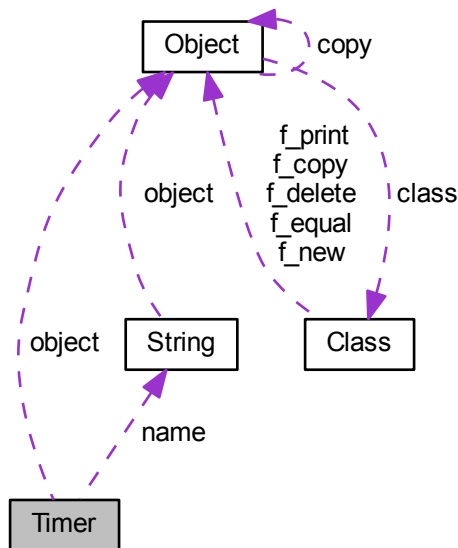
- [Object](#) **object**
- [Map](#) \* **timers**

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

- [TimeMgr.c](#)

### 3.27 Timer Class Reference

Collaboration diagram for Timer:



#### Public Member Functions

- PUBLIC [Timer](#) \* [Timer\\_new](#) ([String](#) \*name)  
*Create an instance of the class [Timer](#).*
- PUBLIC void [Timer\\_delete](#) ([Timer](#) \*this)  
*Delete an instance of the class [Timer](#).*
- PUBLIC [Timer](#) \* [Timer\\_copy](#) ([Timer](#) \*this)  
*Copy an instance of the class [Timer](#).*
- PUBLIC unsigned int [Timer\\_isEqual](#) ([Timer](#) \*this, [Timer](#) \*compared)  
*TBD.*
- PUBLIC char \* [Timer\\_print](#) ([Timer](#) \*this)  
*TBD.*
- PUBLIC void [Timer\\_latchTime](#) ([Timer](#) \*this, double timeS)  
*TBD.*

#### Public Attributes

- [Object](#) **object**
- [String](#) \* **name**
- unsigned int **state**
- unsigned int **nbCalls**
- double **durationS**
- double **latchedTime**

### 3.27.1 Member Function Documentation

#### 3.27.1.1 PUBLIC Timer \* Timer\_copy ( Timer \* *this* )

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

Returns

Copied instance.

#### 3.27.1.2 PUBLIC Timer \* Timer\_new ( String \* *name* )

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

Returns

New instance.

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

- Timer.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 file:

- 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 file:

- Grammar2.lex.c

### 3.30 yyalloc Union Reference

#### Public Attributes

- yytype\_int16 **yyss\_alloc**
- **YYSTYPE** **yyvs\_alloc**

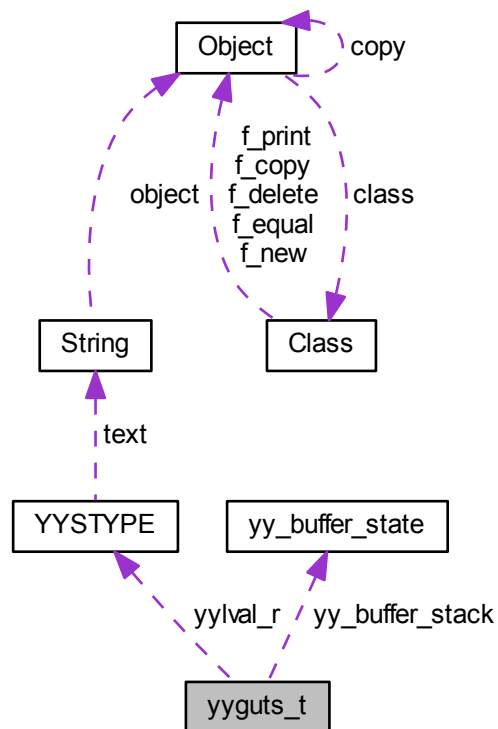
The documentation for this union was generated from the following files:

- Grammar1.parse.c
- 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 yyleng_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

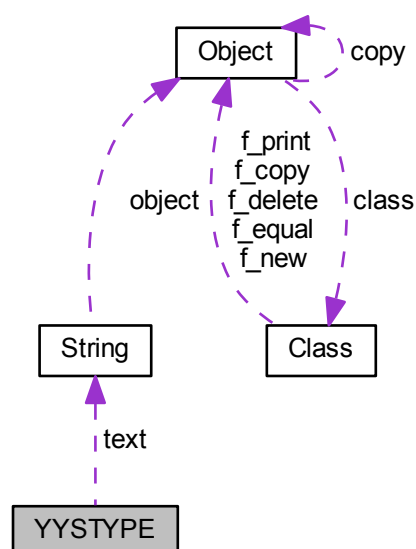
index of top of stack.

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

- Grammar2.lex.c

## 3.32 YYSTYPE Union Reference

Collaboration diagram for YYSTYPE:



### Public Attributes

- [String](#) \* **text**

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

- Grammar2.parse.c
- Grammar2.parse.h



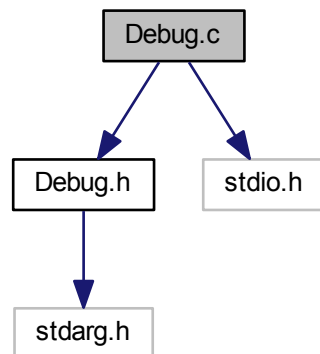
## Chapter 4

# File Documentation

### 4.1 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,...)

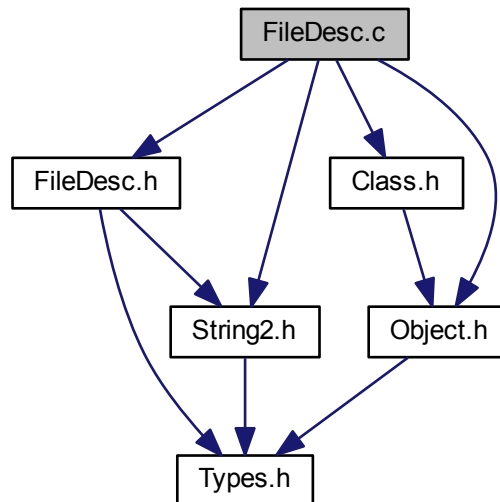
#### 4.1.1 Detailed Description

This file contains debugging functions. The debugging function are TBD

### 4.2 FileDesc.c File Reference

The [FileDesc](#) class describe a File in the FileMgr.

```
#include "FileDesc.h"
#include "String2.h"
#include "Class.h"
#include "Object.h"
Include dependency graph for FileDesc.c:
```



## Classes

- class [FileDesc](#)

## Functions

- PRIVATE [String](#) \* **FileDesc\_getBasename** ([FileDesc](#) \*this)

## Variables

- PRIVATE [Class](#) [fileDescClass](#)

### 4.2.1 Detailed Description

The [FileDesc](#) class describe a File in the FileMgr. The class [FileDesc](#) is TBD

### 4.2.2 Variable Documentation

#### 4.2.2.1 PRIVATE Class [fileDescClass](#)

**Initial value:**

```
=
{
```

```

.f_new = (Constructor)0,
.f_delete = (Destructor)&FileDesc_delete,
.f_copy = (Copy_Operator)&FileDesc_copy,
.f_equal = (Equal_Operator)0,
.f_print = (Printer)0
}

```

## Class Description

## 4.3 FileMgr.c File Reference

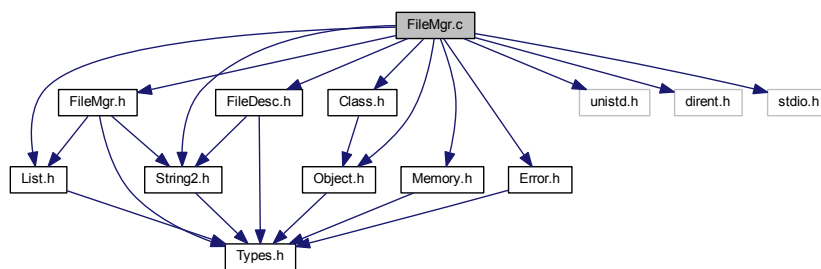
The `FileMgr` class manages a list of files contained in a group of locations.

```

#include "FileMgr.h"
#include "String2.h"
#include "Class.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)
- PUBLIC `String` \* `FileMgr_searchFile` (`FileMgr` \*this, `String` \*name, `List` \*preferredDir)

## Variables

- PRIVATE Class `fileMgrClass`
- 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

### 4.3.2 Variable Documentation

#### 4.3.2.1 PRIVATE Class `fileMgrClass`

Initial value:

```
=
{
    .f_new = 0,
    .f_delete = (Destructor)&FileMgr_delete,
    .f_copy = (Copy_Operator)&FileMgr_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = (Printer)0
}
```

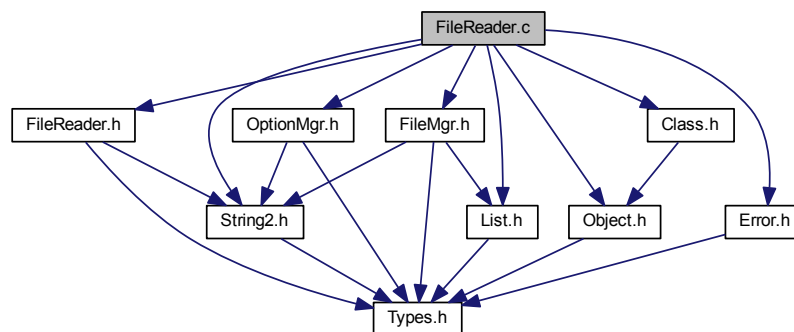
[Class Description](#)

## 4.4 FileReader.c File Reference

This file contains the implementation for the class `FileReader`.

```
#include "FileReader.h"
#include "Class.h"
#include "Object.h"
#include "String2.h"
#include "FileMgr.h"
#include "OptionMgr.h"
#include "List.h"
#include "Error.h"
```

Include dependency graph for `FileReader.c`:





## Classes

- struct [IncludeInfo](#)
- class [FileReader](#)

## Functions

- PRIVATE void **FileReader\_getListPreferredDir** ([FileReader](#) \*this)
- PRIVATE void **FileReader\_deleteListPreferredDir** ([FileReader](#) \*this)
- PRIVATE void **FileReader\_printListPreferredDir** ([FileReader](#) \*this)

## Variables

- PRIVATE Class [fileReaderClass](#)

### 4.4.1 Detailed Description

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

### 4.4.2 Variable Documentation

#### 4.4.2.1 PRIVATE Class [fileReaderClass](#)

##### Initial value:

```
=
{
    .f_new = (Constructor)0,
    .f_delete = (Destructor)&FileReader_delete,
    .f_copy = (Copy_Operator)&FileReader_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = 0
}
```

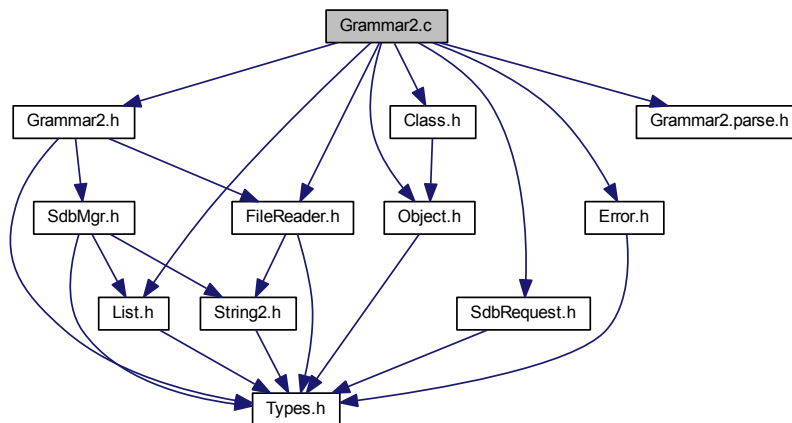
##### [Class](#) Description

## 4.5 Grammar2.c File Reference

A test grammar class. The [Grammar2](#) class implements a test grammar parsing comments code blocks and allowing includes files.

```
#include "Grammar2.h"
#include "Class.h"
#include "Object.h"
#include "FileReader.h"
#include "SdbRequest.h"
#include "Error.h"
#include "List.h"
#include "Grammar2.parse.h"
```

Include dependency graph for Grammar2.c:



## Classes

- struct [GrammarContext](#)
- class [Grammar2](#)

## Macros

- `#define MAX_BUFFER_SIZE (16384)`

## Typedefs

- `typedef struct GrammarContext GrammarContext`

## Functions

- `int Grammar2_parse (void *scanner, Grammar2 *this)`
- `void * Grammar2_scan_string (const char *yystr, void *yyscanner)`
- `int Grammar2lex_init (void *scanner)`
- `int Grammar2lex_destroy (void *yyscanner)`
- `PRIVATE void Grammar2_initSdbTables (Grammar2 *this)`
- `PRIVATE unsigned int Grammar2_isFileToBelgnored (Grammar2 *this, String *fileName)`
- `PRIVATE unsigned int Grammar2_isIncludeNodeProcessed (Grammar2 *this, String *name)`
- `PUBLIC void Grammar2_process (Grammar2 *this)`
- `PUBLIC FileReader * Grammar2_getFileReader (Grammar2 *this)`
- `PUBLIC SdbMgr * Grammar2_getSdbMgr (Grammar2 *this)`
- `PUBLIC void Grammar2_addToBuffer (Grammar2 *this, char *text)`
- `PUBLIC void Grammar2_addNode (Grammar2 *this, unsigned int type, int nodePtr)`
- `PUBLIC void Grammar2_addComment (Grammar2 *this)`
- `PUBLIC void Grammar2_addCodeNode (Grammar2 *this)`
- `PUBLIC void Grammar2_addIncludeNode (Grammar2 *this, char *name)`
- `PUBLIC char * Grammar2_processNewFile (Grammar2 *this, String *fileName)`
- `PUBLIC void Grammar2_returnToFile (Grammar2 *this)`

## Variables

- PRIVATE unsigned int **nodeId** = 0
- PRIVATE unsigned int **codeNodeId** = 0
- PRIVATE unsigned int **commentNodeId** = 0
- PRIVATE unsigned int **includeNodeId** = 0
- PRIVATE unsigned int **unitId** = 0
- PRIVATE unsigned int **isInitialised** = 0
- PRIVATE [Class grammar2Class](#)

### 4.5.1 Detailed Description

A test grammar class. The [Grammar2](#) class implements a test grammar parsing comments code blocks and allowing includes files.

### 4.5.2 Variable Documentation

#### 4.5.2.1 PRIVATE Class grammar2Class

**Initial value:**

```
=
{
    .f_new = (Constructor)0,
    .f_delete = (Destructor)&Grammar2_delete,
    .f_copy = (Copy_Operator)&Grammar2_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = (Printer)0
}
```

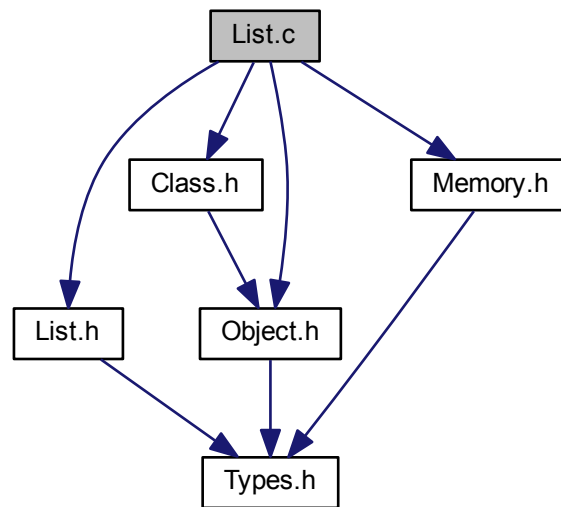
[Class](#) Description

## 4.6 List.c File Reference

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

```
#include "List.h"
#include "Class.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)
- PUBLIC void **List\_resetIterator** ([List](#) \*this)

## Variables

- PRIVATE [Class](#) listClass

### 4.6.1 Detailed Description

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

- init
- add

## 4.6.2 Variable Documentation

### 4.6.2.1 PRIVATE Class listClass

Initial value:

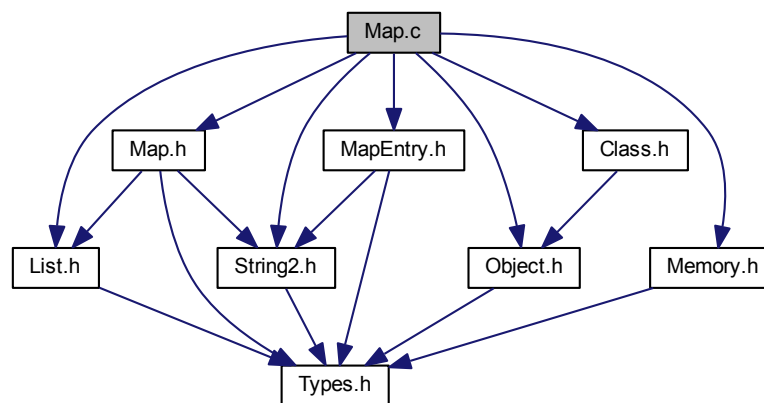
```
=
{
    .f_new = 0,
    .f_delete = (Destructor)&List_delete,
    .f_copy = (Copy_Operator)&List_copy,
    .f_equal = (Equal_Operator)&List_isEqual,
    .f_print = (Printer)&List_print
}
```

[Class](#) Description

## 4.7 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 "Class.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)`
- PUBLIC void `Map_print (Map *this)`

## Variables

- PRIVATE Class `mapClass`

### 4.7.1 Detailed Description

A `Map` class. This class provides a container indexed by a string. A support class for the `Map` class.

### 4.7.2 Variable Documentation

#### 4.7.2.1 PRIVATE Class `mapClass`

**Initial value:**

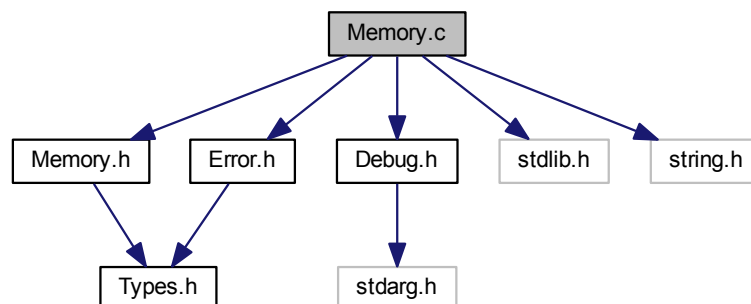
```
=
{
    .f_new = 0,
    .f_delete = (Destructor)&Map_delete,
    .f_copy = (Copy_Operator)&Map_copy,
    .f_equal = 0,
    .f_print = 0
}
```

`Class` Description

## 4.8 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 (const 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`

### 4.8.1 Detailed Description

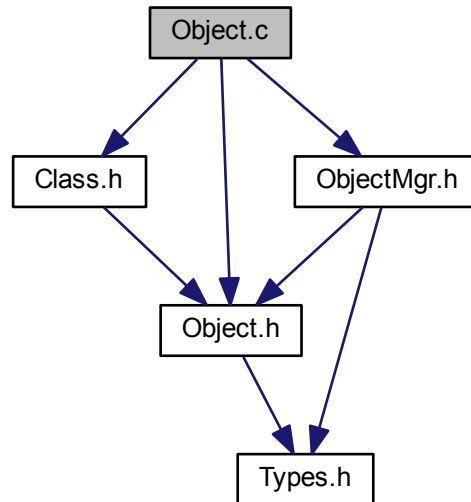
This file provides the implementation of the memory functions. TBD

## 4.9 Object.c File Reference

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

```
#include "Class.h"
#include "Object.h"
#include "ObjectMgr.h"
```

Include dependency graph for Object.c:



## Variables

- PRIVATE `ObjectMgr` \* `Object_objMgrPtr` = 0

### 4.9.1 Detailed Description

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

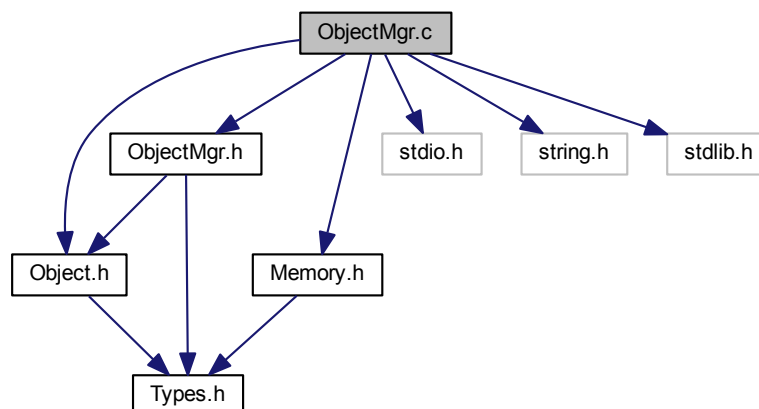
## 4.10 ObjectMgr.c File Reference

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

### 4.10.1 Detailed Description

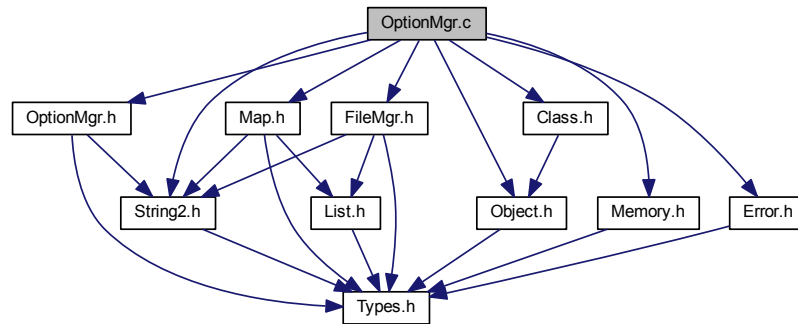
An object management class. This class provides an object allocation and de-allocation service. Only one instance of this class can be created.

## 4.11 OptionMgr.c File Reference

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

```
#include "OptionMgr.h"
#include "Class.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 [Class](#) [optionMgrClass](#)
- PRIVATE [OptionMgr](#) \* **optionMgr** = 0
- PRIVATE struct [OptionDefault](#) **optionDefault** []

### 4.11.1 Detailed Description

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

### 4.11.2 Variable Documentation

#### 4.11.2.1 PRIVATE struct [OptionDefault](#) [optionDefault](#) []

#### Initial value:

```
=
{
    {"DB Name", "-o", "test.db"},
```

```

{"Input Directory", "-d", "."},
{"Config file name", "-c", "sparse.txt"}
}

```

#### 4.11.2.2 PRIVATE Class optionMgrClass

Initial value:

```

=
{
    .f_new = (Constructor)0,
    .f_delete = (Destructor)&OptionMgr_delete,
    .f_copy = (Copy_Operator)&OptionMgr_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = (Printer)0
}

```

[Class Description](#)

## 4.12 SdbMgr.c File Reference

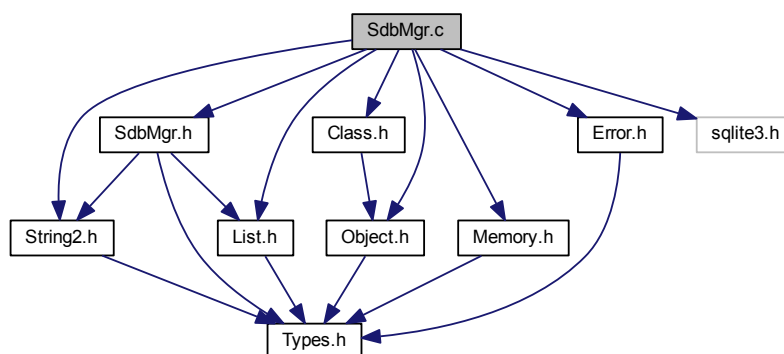
TBD.

```

#include "SdbMgr.h"
#include "Class.h"
#include "Object.h"
#include "String2.h"
#include "Memory.h"
#include "Error.h"
#include "List.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)

## Variables

- PRIVATE [Class sdbMgrClass](#)
- PRIVATE [SdbMgr](#) \* **sdbMgr** = 0

### 4.12.1 Detailed Description

TBD. TBD

### 4.12.2 Variable Documentation

#### 4.12.2.1 PRIVATE Class sdbMgrClass

**Initial value:**

```
=
{
    .f_new = (Constructor)0,
    .f_delete = (Destructor)&SdbMgr_delete,
    .f_copy = (Copy_Operator)&SdbMgr_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = (Printer)0
}
```

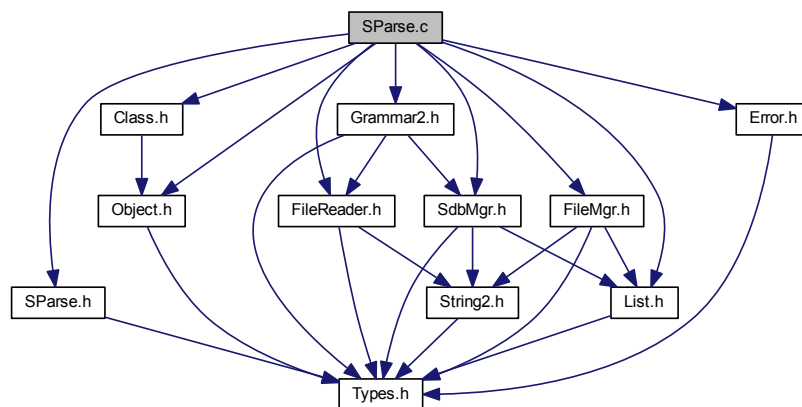
[Class](#) Description

## 4.13 SParse.c File Reference

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

```
#include "SParse.h"
#include "Class.h"
#include "Object.h"
#include "FileReader.h"
#include "SdbMgr.h"
#include "Error.h"
#include "Grammar2.h"
#include "FileMgr.h"
#include "List.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)
- PRIVATE void **SParse\_buildPreferredDirList** ([SParse](#) \*this, const char \*extension)
- PUBLIC [SParse](#) \* **SParse\_copy** ([SParse](#) \*this)

## Variables

- PRIVATE Class [sparseClass](#)

### 4.13.1 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.

### 4.13.2 Variable Documentation

#### 4.13.2.1 PRIVATE Class [sparseClass](#)

##### Initial value:

```
=
{
    .f_new = (Constructor)0,
    .f_delete = (Destructor)&SParse\_delete,
    .f_copy = (Copy_Operator)&SParse\_copy,
    .f_equal = (Equal_Operator)0,
    .f_print = (Printer)0
}
```

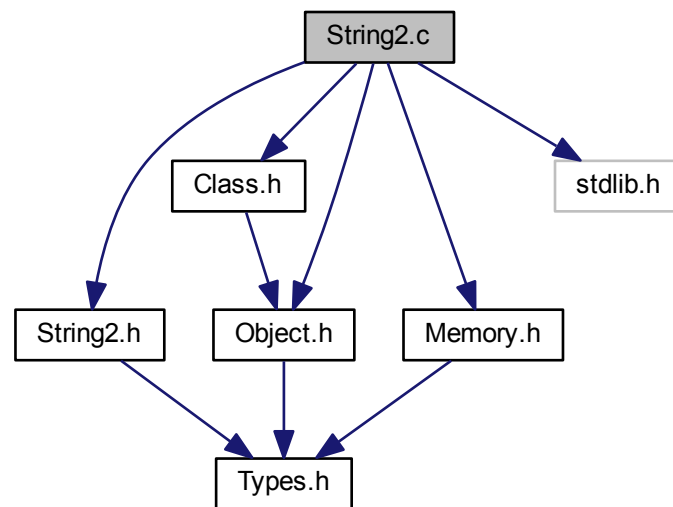
##### [Class](#) Description

## 4.14 String2.c File Reference

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

```
#include "String2.h"
#include "Class.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)

## Variables

- PRIVATE [Class](#) [stringClass](#)

### 4.14.1 Detailed Description

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

### 4.14.2 Variable Documentation

#### 4.14.2.1 PRIVATE Class [stringClass](#)

**Initial value:**

```

=
{
    .f_new = NULL,
    .f_delete = (Destructor)&String_delete,
    .f_copy = (Copy_Operator)&String_copy,
    .f_equal = (Equal_Operator)NULL,
    .f_print = (Printer)NULL
}

```

## Class Description

## 4.15 TimeMgr.c File Reference

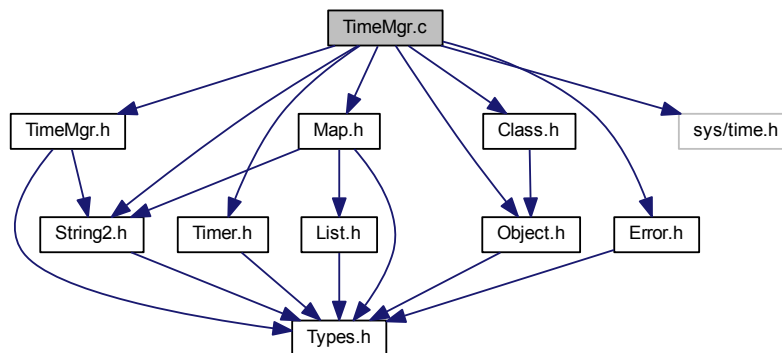
This file implements the class [TimeMgr](#).

```

#include "TimeMgr.h"
#include "Timer.h"
#include "String2.h"
#include "Map.h"
#include "Class.h"
#include "Object.h"
#include "Error.h"
#include <sys/time.h>

```

Include dependency graph for TimeMgr.c:



## Classes

- class [TimeMgr](#)

## Macros

- `#define USEC_IN_SEC (1.0E6)`

## Variables

- `PRIVATE Class timeMgrClass`
- `PRIVATE TimeMgr * timeMgr = 0`

### 4.15.1 Detailed Description

This file implements the class [TimeMgr](#). The class [TimeMgr](#) is TBD

### 4.15.2 Variable Documentation

#### 4.15.2.1 PRIVATE Class timeMgrClass

**Initial value:**

```
=  
{  
    .f_new = (Constructor)0,  
    .f_delete = (Destructor)&TimeMgr_delete,  
    .f_copy = (Copy_Operator)&TimeMgr_copy,  
    .f_equal = (Equal_Operator)0,  
    .f_print = (Printer)0  
}
```

[Class](#) Description



# Index

Class, [5](#)

Debug.c, [45](#)

FileDesc, [6](#)

FileDesc.c, [45](#)

fileDescClass, [46](#)

fileDescClass

FileDesc.c, [46](#)

FileMgr, [7](#)

FileMgr\_addDirectory, [8](#)

FileMgr\_addFile, [8](#)

FileMgr\_copy, [8](#)

FileMgr\_filterFiles, [8](#)

FileMgr\_getRef, [8](#)

FileMgr\_getRootLocation, [8](#)

FileMgr\_load, [9](#)

FileMgr\_setRootLocation, [9](#)

FileMgr.c, [47](#)

fileMgrClass, [48](#)

FileMgr\_addDirectory

FileMgr, [8](#)

FileMgr\_addFile

FileMgr, [8](#)

FileMgr\_copy

FileMgr, [8](#)

FileMgr\_filterFiles

FileMgr, [8](#)

FileMgr\_getRef

FileMgr, [8](#)

FileMgr\_getRootLocation

FileMgr, [8](#)

FileMgr\_load

FileMgr, [9](#)

FileMgr\_setRootLocation

FileMgr, [9](#)

fileMgrClass

FileMgr.c, [48](#)

FileReader, [9](#)

FileReader\_addFile, [10](#)

FileReader\_copy, [10](#)

FileReader\_getBuffer, [10](#)

FileReader\_getName, [10](#)

FileReader\_new, [11](#)

FileReader.c, [48](#)

fileReaderClass, [49](#)

FileReader\_addFile

FileReader, [10](#)

FileReader\_copy

FileReader, [10](#)

FileReader\_getBuffer

FileReader, [10](#)

FileReader\_getName

FileReader, [10](#)

FileReader\_new

FileReader, [11](#)

fileReaderClass

FileReader.c, [49](#)

Grammar2, [11](#)

Grammar2\_copy, [12](#)

Grammar2\_new, [12](#)

Grammar2.c, [49](#)

grammar2Class, [51](#)

Grammar2\_copy

Grammar2, [12](#)

Grammar2\_new

Grammar2, [12](#)

grammar2Class

Grammar2.c, [51](#)

GrammarContext, [12](#)

IncludeInfo, [13](#)

List, [14](#)

List\_copy, [15](#)

List\_forEach, [15](#)

List\_getHead, [15](#)

List\_getSize, [15](#)

List\_insertHead, [15](#)

List\_insertTail, [15](#)

List\_isEqual, [16](#)

List\_merge, [16](#)

List\_new, [16](#)

List\_removeHead, [16](#)

List.c, [51](#)

listClass, [53](#)

List\_copy

List, [15](#)

List\_forEach

List, [15](#)

List\_getHead

List, [15](#)

List\_getSize

List, [15](#)

List\_insertHead

List, [15](#)

List\_insertTail

List, [15](#)

List\_isEqual

- List, 16
- List\_merge
  - List, 16
- List\_new
  - List, 16
- List\_removeHead
  - List, 16
- listClass
  - List.c, 53
- Map, 17
  - Map\_copy, 17
  - Map\_getAll, 18
  - Map\_insert, 18
- Map.c, 53
  - mapClass, 54
- Map\_copy
  - Map, 17
- Map\_getAll
  - Map, 18
- Map\_insert
  - Map, 18
- mapClass
  - Map.c, 54
- MapEntry, 18
- maxNbObjectAllocated
  - ObjectMgr, 23
- Memory.c, 54
- Object, 19
  - Object\_copy, 20
  - Object\_getRef, 20
  - Object\_isEqual, 20
  - Object\_new, 20
  - Object\_print, 20
- Object.c, 55
- Object\_copy
  - Object, 20
- Object\_getRef
  - Object, 20
- Object\_isEqual
  - Object, 20
- Object\_new
  - Object, 20
- Object\_print
  - Object, 20
- ObjectInfo, 21
- ObjectMgr, 22
  - maxNbObjectAllocated, 23
  - ObjectMgr\_allocate, 23
  - ObjectMgr\_copy, 23
  - ObjectMgr\_deallocate, 23
  - ObjectMgr\_getRef, 23
- ObjectMgr.c, 56
- ObjectMgr\_allocate
  - ObjectMgr, 23
- ObjectMgr\_copy
  - ObjectMgr, 23
- ObjectMgr\_deallocate
- ObjectMgr, 23
- ObjectMgr\_getRef
  - ObjectMgr, 23
- OptionDefault, 23
- optionDefault
  - OptionMgr.c, 58
- OptionMgr, 24
  - OptionMgr\_getRef, 25
  - OptionMgr\_readFromCmdLine, 25
- OptionMgr.c, 57
  - optionDefault, 58
  - optionMgrClass, 59
- OptionMgr\_getRef
  - OptionMgr, 25
- OptionMgr\_readFromCmdLine
  - OptionMgr, 25
- optionMgrClass
  - OptionMgr.c, 59
- SParse, 28
  - SParse\_delete, 29
  - SParse\_new, 29
  - SParse\_parse, 29
- SParse.c, 60
  - sparseClass, 61
- SParse\_delete
  - SParse, 29
- SParse\_new
  - SParse, 29
- SParse\_parse
  - SParse, 29
- SdbMgr, 25
  - SdbMgr\_copy, 26
  - SdbMgr\_execute, 26
  - SdbMgr\_getRef, 26
- SdbMgr.c, 59
  - sdbMgrClass, 60
- SdbMgr\_copy
  - SdbMgr, 26
- SdbMgr\_execute
  - SdbMgr, 26
- SdbMgr\_getRef
  - SdbMgr, 26
- sdbMgrClass
  - SdbMgr.c, 60
- SdbRequest, 27
  - SdbRequest\_delete, 27
  - SdbRequest\_execute, 28
  - SdbRequest\_new, 28
- SdbRequest\_delete
  - SdbRequest, 27
- SdbRequest\_execute
  - SdbRequest, 28
- SdbRequest\_new
  - SdbRequest, 28
- sparseClass
  - SParse.c, 61
- String, 30
  - String\_copy, 30

- String\_getRef, [31](#)
  - String\_new, [31](#)
- String2.c, [61](#)
  - stringClass, [62](#)
- String\_copy
  - String, [30](#)
- String\_getRef
  - String, [31](#)
- String\_new
  - String, [31](#)
- stringClass
  - String2.c, [62](#)
- TestClass, [31](#)
- TestFileMgr, [32](#)
- TestItem, [33](#)
- TestObjectMgr, [34](#)
- TestSdbMgr, [35](#)
- TestTimeMgr, [36](#)
- TimeMgr, [37](#)
- TimeMgr.c, [63](#)
  - timeMgrClass, [64](#)
- timeMgrClass
  - TimeMgr.c, [64](#)
- Timer, [38](#)
  - Timer\_copy, [39](#)
  - Timer\_new, [39](#)
- Timer\_copy
  - Timer, [39](#)
- Timer\_new
  - Timer, [39](#)
- YYSTYPE, [42](#)
- yy\_bs\_column
  - yy\_buffer\_state, [39](#)
- yy\_bs\_lineno
  - yy\_buffer\_state, [39](#)
- yy\_buffer\_stack
  - yyguts\_t, [42](#)
- yy\_buffer\_stack\_max
  - yyguts\_t, [42](#)
- yy\_buffer\_stack\_top
  - yyguts\_t, [42](#)
- yy\_buffer\_state, [39](#)
  - yy\_bs\_column, [39](#)
  - yy\_bs\_lineno, [39](#)
- yy\_trans\_info, [40](#)
- yyalloc, [40](#)
- yyguts\_t, [41](#)
  - yy\_buffer\_stack, [42](#)
  - yy\_buffer\_stack\_max, [42](#)
  - yy\_buffer\_stack\_top, [42](#)