SParse

Generated by Doxygen 1.8.5

Fri Sep 13 2019 10:16:00

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

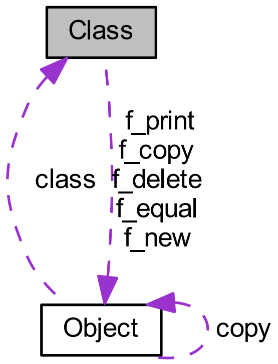
Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 Class Struct Reference

Collaboration diagram for Class:



Public Attributes

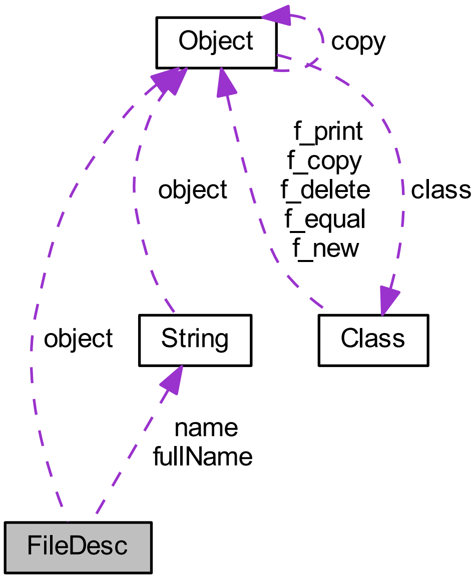
- Constructor **f_new**
- Destructor **f_delete**
- Copy_Operator **f_copy**
- Equal_Operator **f_equal**
- Printer **f_print**

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

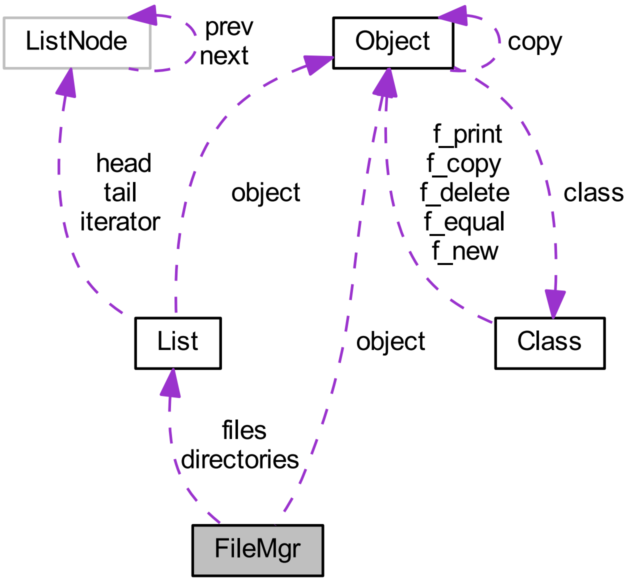
- Object **object**
- String ∗ **name**
- String ∗ **fullName**

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

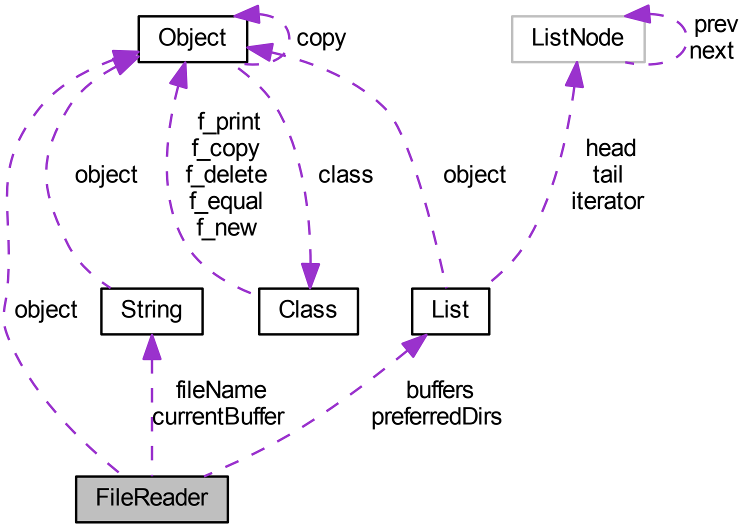
Status.

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

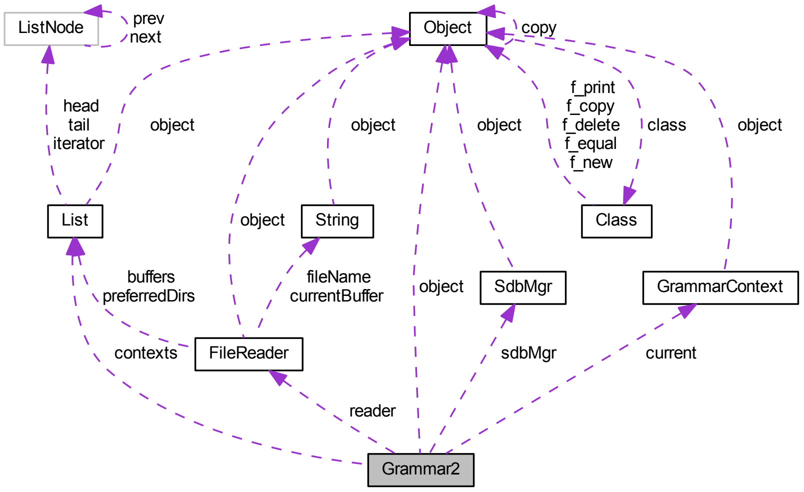
Created FileReader object.

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

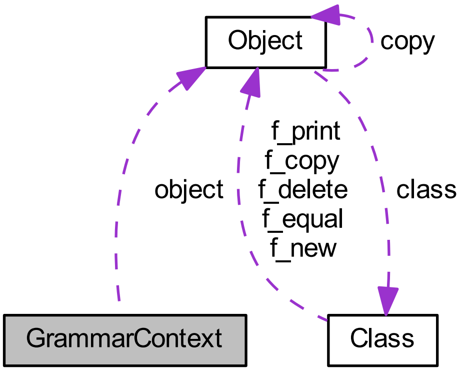
New instance.

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

- Grammar2.c

3.6 GrammarContext Struct Reference

Collaboration diagram for GrammarContext:



Public Attributes

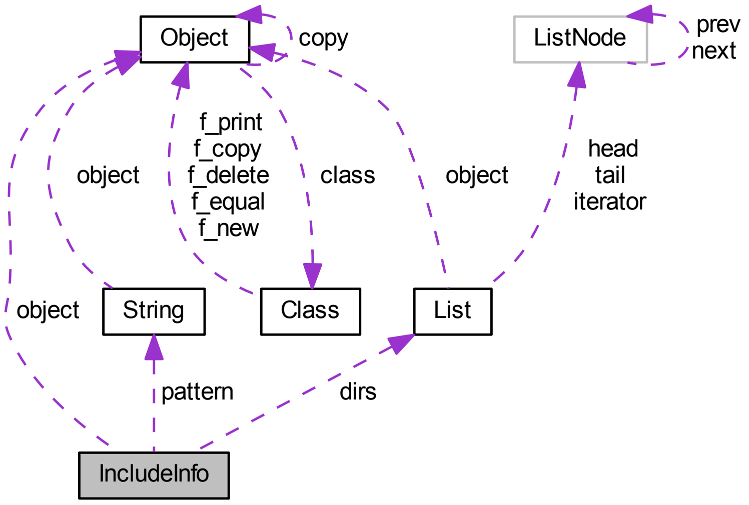
- Object **object**
- unsigned int **lastNode**

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

- Grammar2.c

3.7 IncludeInfo Struct Reference

Collaboration diagram for IncludeInfo:



Public Attributes

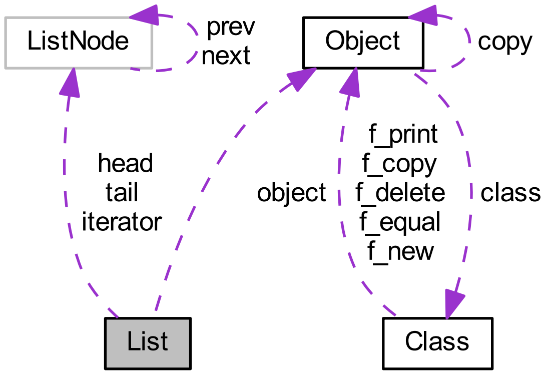
- Object **object**
- String ∗ **pattern**
- List ∗ **dirs**

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

in	*f*	Pointer to function.

3.8.1.3 PUBLIC void ∗ List_getHead (List ∗ *this*)

Get the head item in an insatnce of LIst.

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

in	*item*	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

in	*item*	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

in	*l1*	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*)

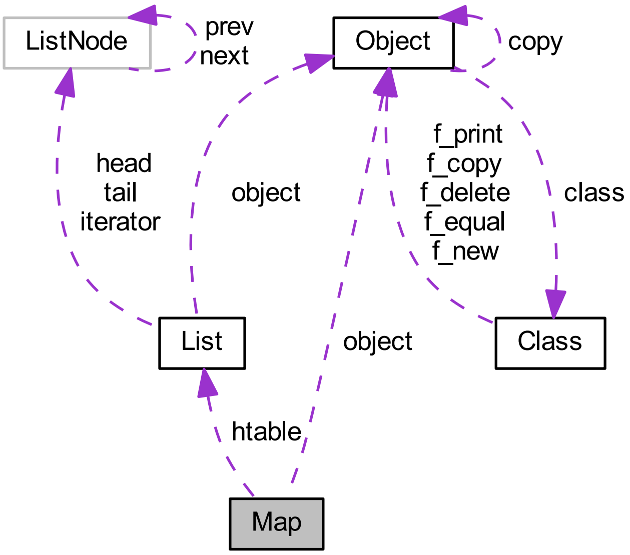
Remove the head item in an instance of LIst.

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

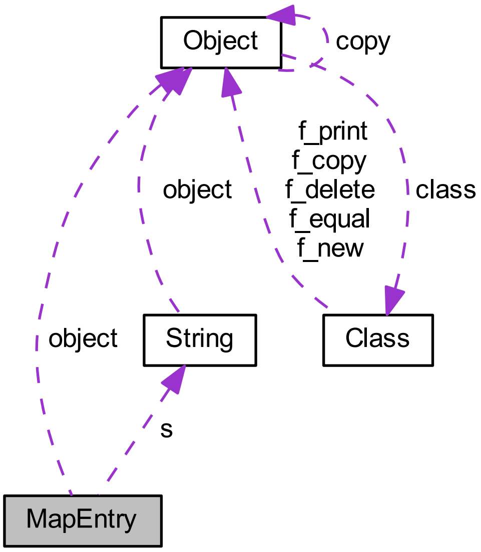
TBD

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

- Map.c

3.10 MapEntry Struct Reference

Collaboration diagram for MapEntry:



Public Attributes

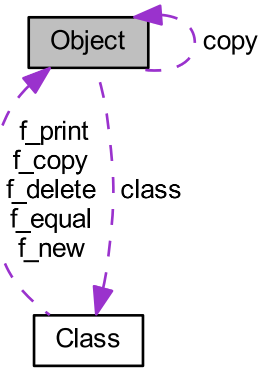
- Object **object**
- String ∗ **s**
- void ∗ **item**

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	*Class*	to instanciate

3.11.1.5 PUBLIC char ∗ Object_print (Object ∗ *this*)

Print an instance of the class Object into a buffer of characters.

Returns

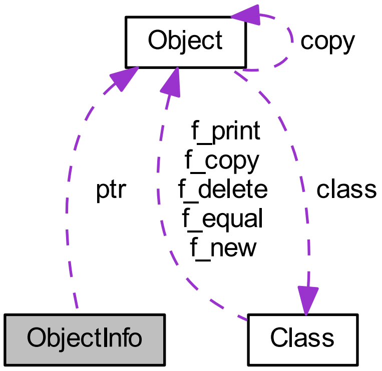
Buffer of characters

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

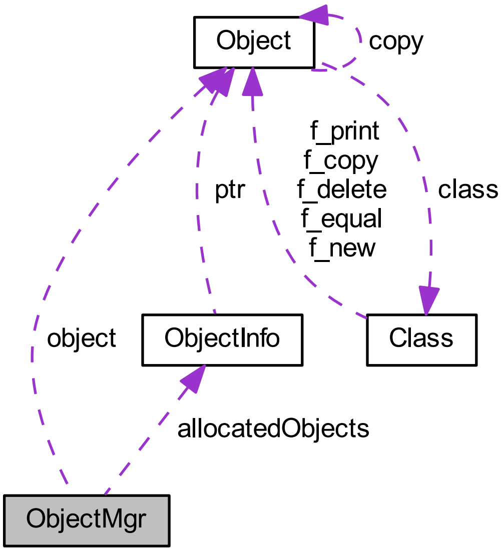
- Object ∗ **ptr**
- unsigned int **prevId**
- unsigned int **nextId**

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

in	*size*	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

in	*object*	Reference to instance of Object.

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**

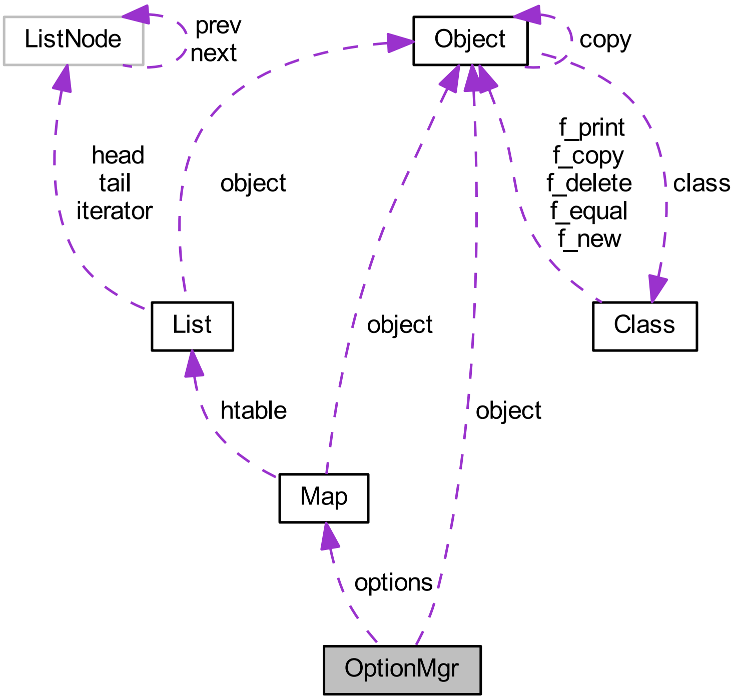
- char ∗ **value**

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*	List os commandline arguments.

Returns

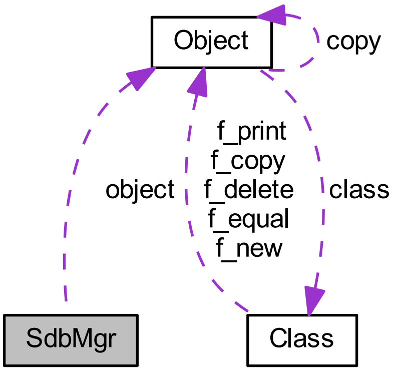
Status of operation.

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

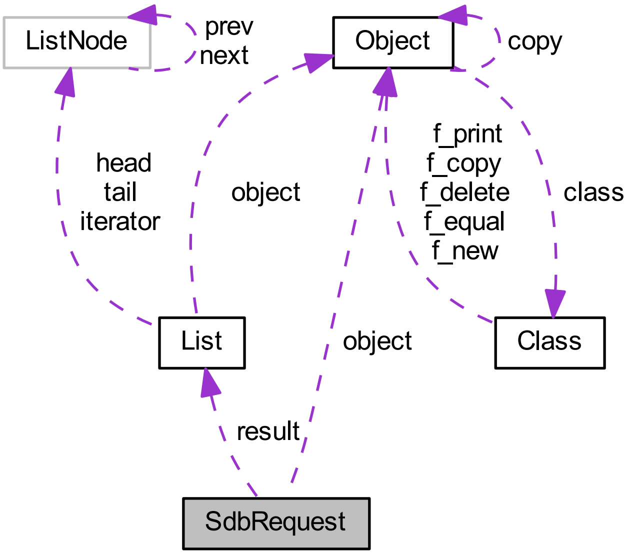
A reference to a SdbMgr object.

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

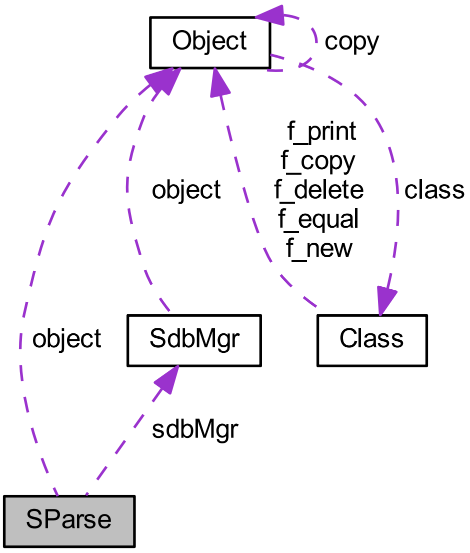
Instance of an SdbRequest

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

Object	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

in	*extension*	Extension of the files to parse.

Returns

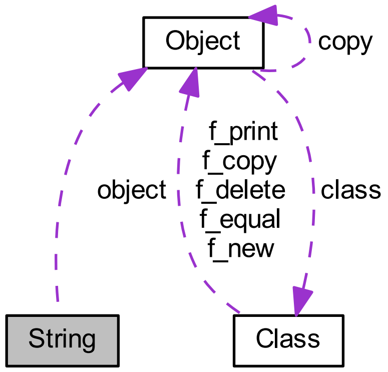
Status of the operation.

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

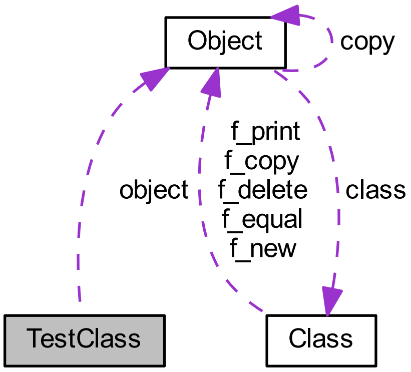
Created String object.

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

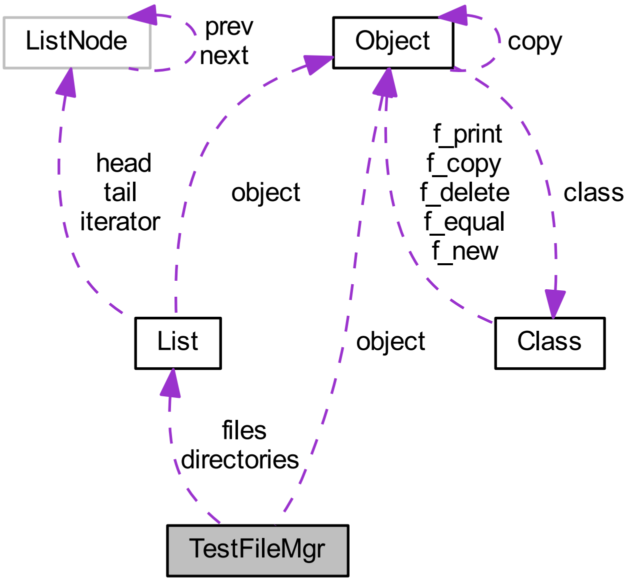
- Object **object**
- int **testValue**

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

- test1.c

3.21 TestFileMgr Struct Reference

Collaboration diagram for TestFileMgr:



Public Attributes

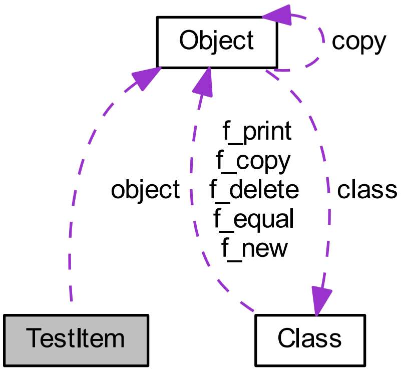
- Object **object**
- List ∗ **files**
- List ∗ **directories**
- char **rootLocation** [FILEMGR_MAX_PATH]

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

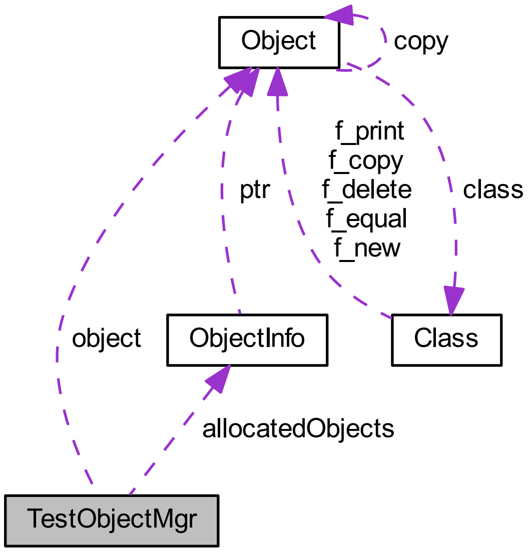
- Object **object**
- int **x**
- int **y**

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

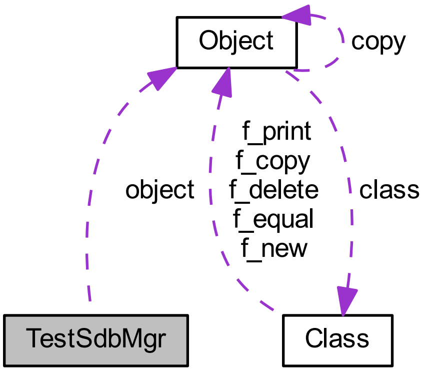
- Object **object**
- unsigned int **nbBytesAllocated**
- unsigned int **maxNbBytesAllocated**
- unsigned int **allocRequestId**
- unsigned int **freeRequestId**
- unsigned int **nbAllocatedObjects**
- ObjectInfo **allocatedObjects** [MAX_NB_OBJECTS]
- unsigned int **freeSpace**
- unsigned int **usedSpace**

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

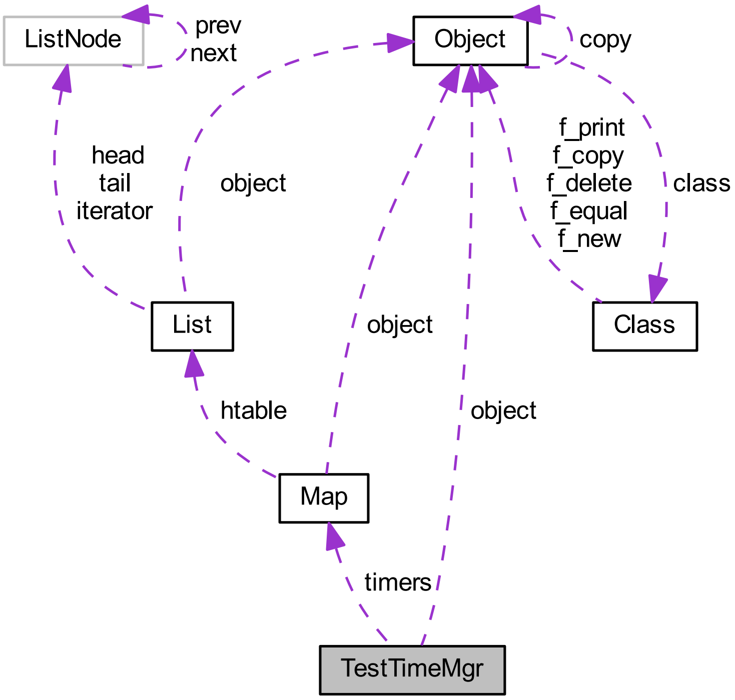
- Object **object**

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

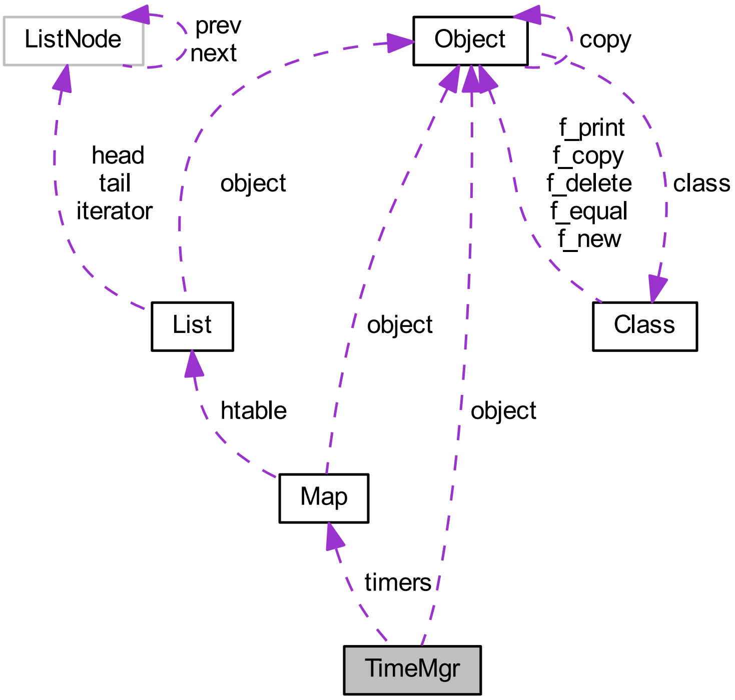
- Object **object**
- Map ∗ **timers**

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

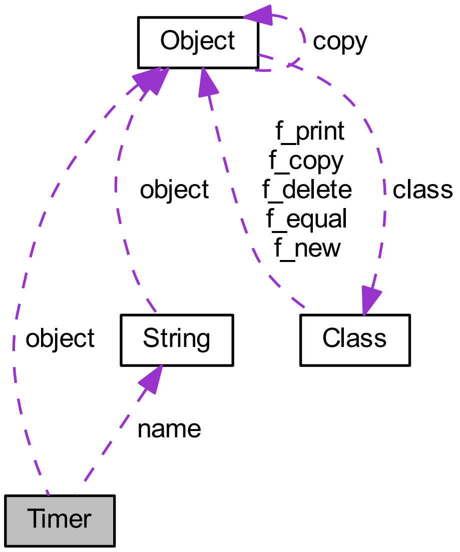
- Object **object**
- Map ∗ **timers**

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

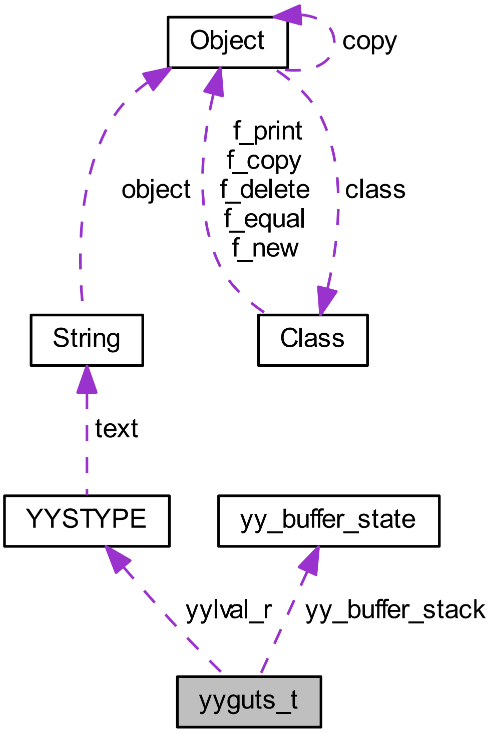
- yytype_int16 **yyss_alloc**
- YYSTYPE **yyvs_alloc**

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 ∗ **yylval_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

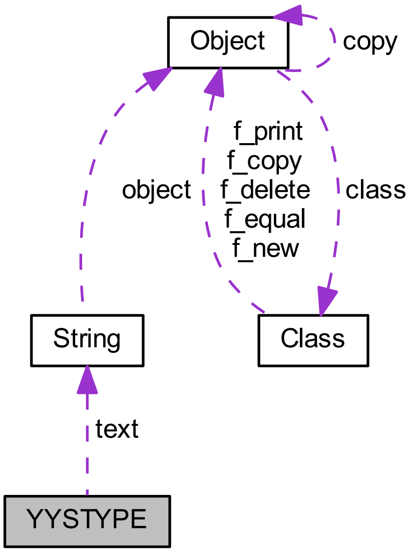
index of top of stack.

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

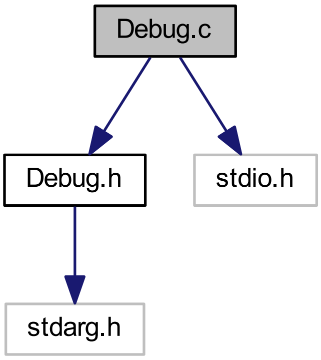
File Documentation

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

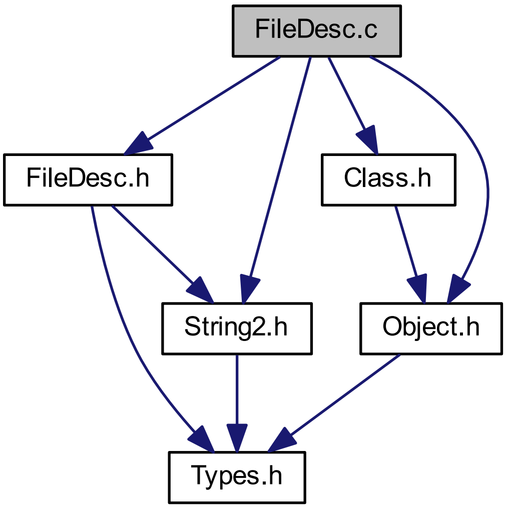
This file contains debugging functions. The debugging function are TBD

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
}
```

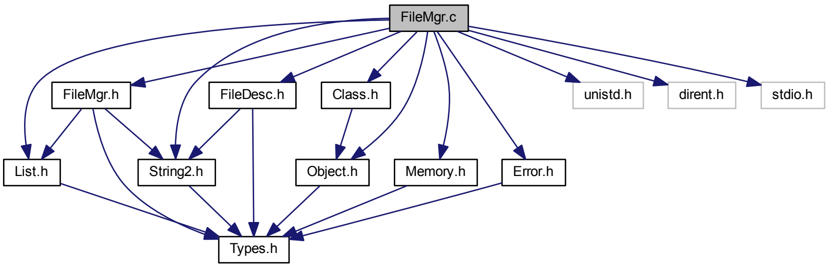
Class Description

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
}
```

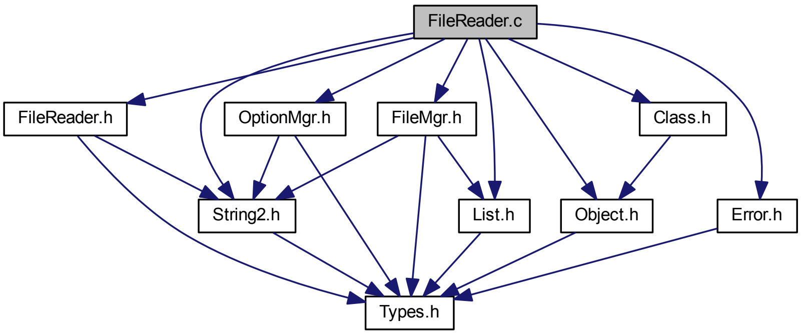
Class Description

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
}
```

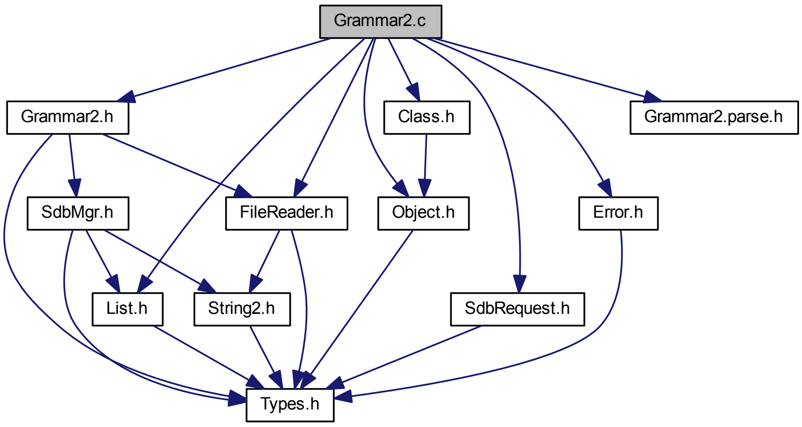
Class Description

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_isFileToBeIgnored** (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
}
```

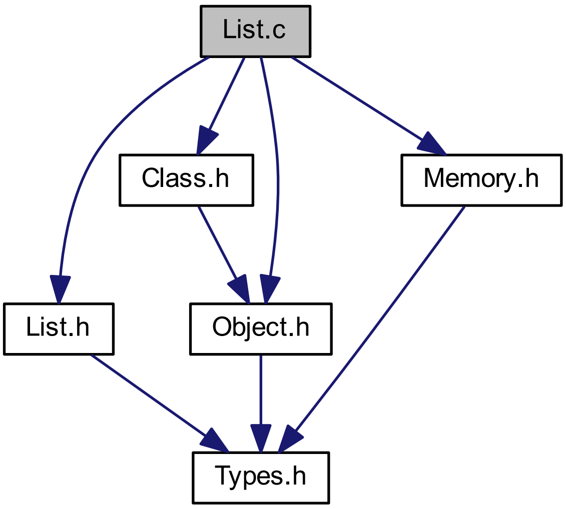
Class Description

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
}
```

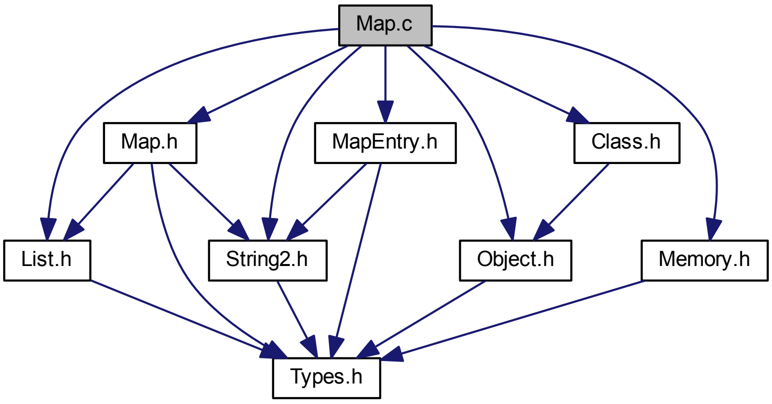
Class Description

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 theMap 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
}
```

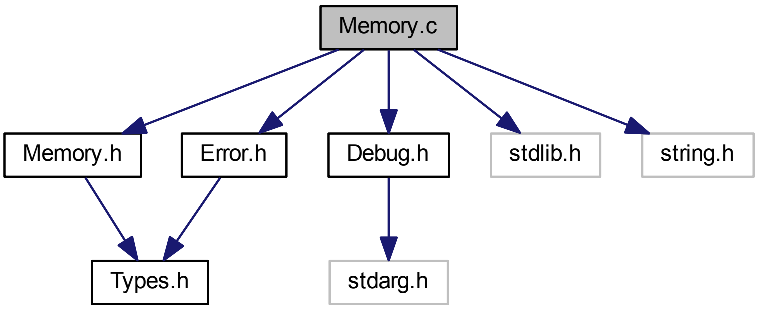
Class Description

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

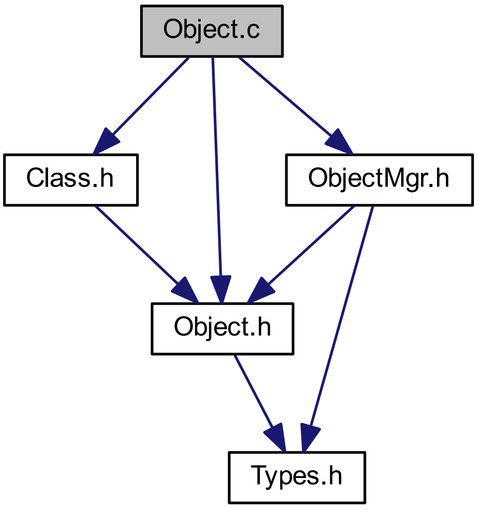
This file provides the implementation of the memory functions. TBD

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

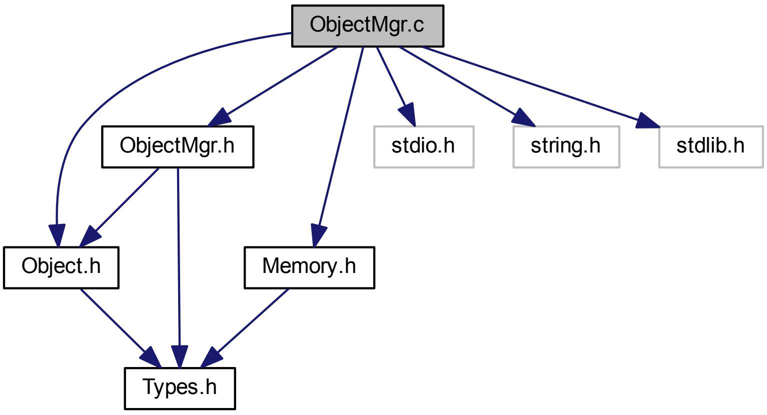
This file contains the implementation for the class Object. The class Object is TBD

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

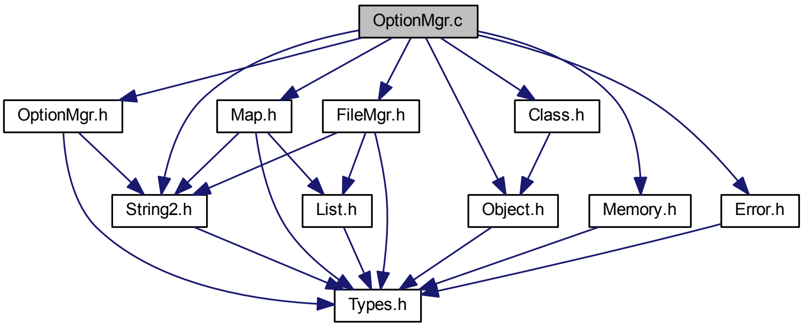
An object management class. This class provides an object allocation and de-allocation service. Only one instance of this class can be created.

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
}
```

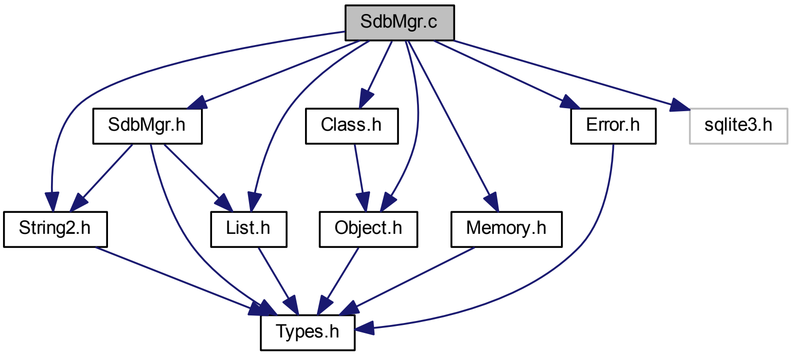
Class Description

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
}
```

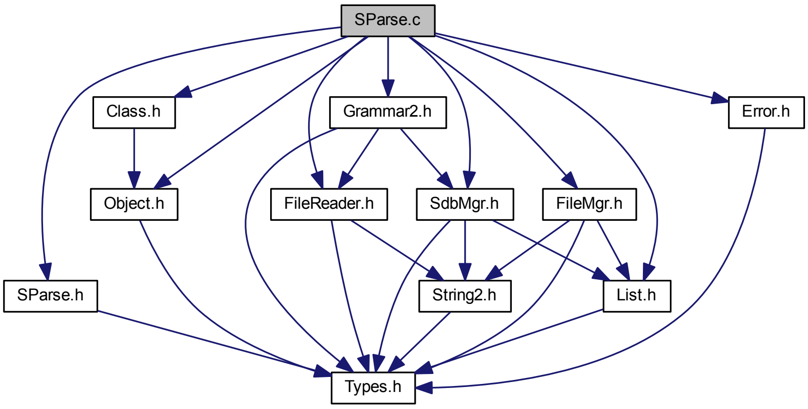
Class Description

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
}
```

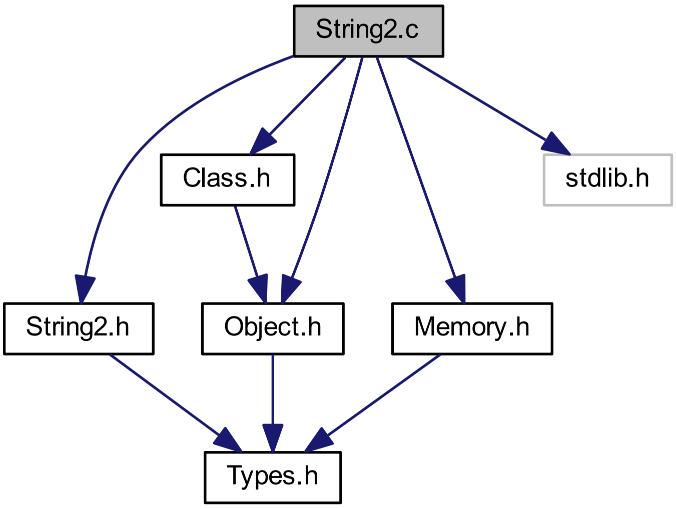
Class Description

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
}
```

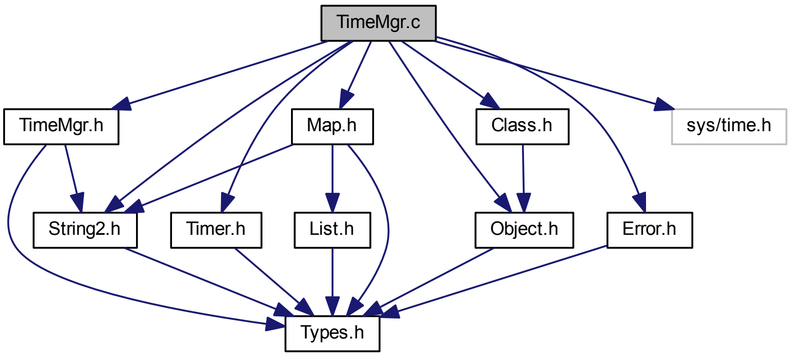
Class Description

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