

NMD: Numerical Metadata library Reference Manual

Victor Eijkhout

Version 2

Abstract

The NMD library offers a convenient storage mechanism for numerical data that is organised in a two-level structure of categories and components. It has routines for creating, deleting, and manipulating metadata objects, as well as routines for output to various formats.

Contents

1	NMD: the Numerical MetaData library	i
2	Installing NMD	iii
3	Metadata object manipulation	iii
4	Viewing objects	iii
5	Value handling	iv
6	NMD Data Types	iv
7	Metadata category manipulation	iv
8	Metadata component manipulation	iv
9	NMD String Library	v
10	Data Structure Index	v
11	File Index	v
12	Data Structure Documentation	vii
13	File Documentation	xix

1 NMD: the Numerical MetaData library

The NMD library serves to store two-level data structures of numerical metadata. - While this library was intended to be used with AnaMod in the Salsa project, it can be used on its own. Unlike other libraries in the Salsa project, it does not rely on Petsc.

1.1 General notes

All routines in NMD return an error code of type `NMDErrorCode`. This is zero for success and anything else for failure. Return codes can be caught with `NMD_ERR_RETURN(ierr)` and generate with `NMD_ERR_REPORT(msg)`.

[Metadata object manipulation](#)

[Metadata category manipulation](#)

[Metadata component manipulation](#)

[Value handling](#)

[Viewing objects](#)

[Installing NMD](#)

Author

Victor Eijkhout

Version

2.5

Date

unreleased

1.2 history**2.5 changes**

- added routines: [NMDUnsetValue\(\)](#), [NMDCategoryGetComponents\(\)](#), [NMDTypeGetMySQLName\(\)](#)
- [NMDReportObject\(\)](#) now takes delimiter arguments. This is useful for generating MySQL strings and such.

2.4 changes

- minor

2.3 changes

- [NMDGetValue](#) now returns failure (instead of aborting) if cat/cmp do not exist
- removed occurrences of Petsc's `CHKERRQ`
- introduced `NMDErrorCode`
- new files [nmdcat.c](#) [nmdcmp.c](#)
- [NMDGetValue](#) and [GetArrayValue](#) now use `NMDTruth`
- [NMDSetValue](#) and [NMDSetArrayValue](#) are now analogous: use ampersand for all types of data
- `NMDTrue` and `NMDFalse` instead of 1 and 0

- unit tests added

2.3 bug fixes

- NMDGetValue missing case of string value added
- NMDTryGetCategory (and various other Get routines) were able to find non-existing category names. Fixed.
- lots of memory leaks plugged

2.2

- Completely revamped array handling; watch out for prototype changes
- CFLAGS is now NMD_CFLAGS

2 Installing NMD

Installing NMD takes the following steps:

- edit the [Make.inc](#) file for:
 - compiler options, and settings for your `ar` and `ranlib` program
 - add `"-DNMD_HAVE_PETSC"` to the compile line if you are using NMD with `Petsc` (see [NMDReportObject\(\)](#))
 - `NMD_LIB_DIR` is the location where the library will be installed
- do `"make install"` to generate the binaries

3 Metadata object manipulation

Top level functions for manipulation metadata objects.

See [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDBuildObjectStructure\(\)](#), [NMDCloneObject\(\)](#), [NMDCloneObjectStructure\(\)](#), [NMDViewObject\(\)](#)

4 Viewing objects

[NMDViewObject\(\)](#) gives informal output; [NMDReportObject\(\)](#) can be used for database records and such.

5 Value handling

NMD handles scalar and array values slightly differently. For scalar values use [NMDSetValue\(\)](#) and [NMDGetValue\(\)](#); for array values use [NMDSetArrayValue\(\)](#), [NMDGetArrayValue\(\)](#), [NMDCopyArrayValue\(\)](#). The array routines take an extra parameter denoting the length of the array.

Note that scalar values have to be passed by reference:

```
int i,*ii;
NMDSetValue(nmd,...,&i);           // use an ampersand here
NMDSetValue(nmd,...,ii,length);    // no ampersand needed here!
NMDGetValue(nmd,...,&i);
NMDGetValue(nmd,...,&ii,&length);
```

Furthermore, see [NMDCopyItemValues\(\)](#), [NMDGetDataType\(\)](#), [NMDIsArrayType\(\)](#).

6 NMD Data Types

Most NMD datatypes have two different definitions, depending on whether Petsc is used or not.

NMDTruth is only used for success parameters in routines such as [NMDGetValue\(\)](#). It has possible values NMDTrue and NMDFalse.

7 Metadata category manipulation

A metadata object contains a number of categories, each containing multiple component which store the actual metadata. Here are the routines for manipulation the categories.

See [NMDObjectTryGetCategory\(\)](#), [NMDObjectGetCategory\(\)](#), [NMDAllocateCategory\(\)](#), [NMDObjectAllocateNewCategory\(\)](#), [NMDObjectGetOrCreateCategory\(\)](#), [NMDRemoveCategory\(\)](#), [NMDGetCategories\(\)](#), [NMDCopyCategory\(\)](#).

8 Metadata component manipulation

Categories have components, much like metadata objects have categories. Most of the component functions work on a metadata object, and specify both category and component name.

See [NMDCategoryAllocateNewComponent\(\)](#), [NMDCategoryGetOrCreateComponent\(\)](#), [NMDObjectHasCategoryComponent\(\)](#), [NMDCategoryTryGetComponent\(\)](#), [NMDCategoryGetComponent\(\)](#), [NMDGetCategoryGetComponent\(\)](#)

9 NMD String Library

We have some routines for string handling.

10 Data Structure Index

10.1 Data Structures

Here are the data structures with brief descriptions:

NMD_intarray_struct	vii
NMD_metadata_	viii
NMD_metadata_category_	xi
NMD_metadata_item_	xiv
NMD_object_	xvii
NMD_realarray_struct	xviii
NMD_string_	xix

11 File Index

11.1 File List

Here is a list of all files with brief descriptions:

Make.inc	xx
nmd.c	xx
nmd.h	xxx
nmd5.c	lv
nmd_impl.h	lvii
nmdcat.c	lviii
nmdcmp.c	lxiii

nmdmysql.c	lxx
nmdreport.c	lxxi
nmdtest.c	lxxii
nmdutil.c	lxxiii
u1.c	lxxvi
u10.c	lxxvii
u11.c	lxxviii
u12.c	lxxx
u13.c	lxxxi
u14.c	lxxxii
u15.c	lxxxiii
u16.c	lxxxv
u18.c	lxxxvi
u19.c	lxxxvii
u2.c	lxxxviii
u21.c	lxxxix
u27.c	xci
u3.c	xciii
u4.c	xciv
u5.c	xcv
u6.c	xcvii
u7.c	xcviii
u8.c	xcix
u9.c	ci

12 Data Structure Documentation

12.1 NMD_intarray_struct Struct Reference

```
#include <nmd_impl.h>
```

Data Fields

- int [length](#)
- int [unique](#)
- int * [data](#)

12.1.1 Detailed Description

Definition at line 42 of file `nmd_impl.h`.

12.1.2 Field Documentation

12.1.2.1 int* NMD_intarray_struct::data

Definition at line 44 of file `nmd_impl.h`.

Referenced by `NMDComponentDestroy()`, `NMDComponentSetArrayValue()`, `NMDComponentUnsetValue()`, `NMDCopyArrayValue()`, `NMDCopyItemValues()`, `NMDGetArrayValue()`, and `NMDReportObject()`.

12.1.2.2 int NMD_intarray_struct::length

Definition at line 43 of file `nmd_impl.h`.

Referenced by `NMDComponentDestroy()`, `NMDComponentSetArrayValue()`, `NMDCopyArrayValue()`, `NMDCopyItemValues()`, `NMDGetArrayValue()`, and `NMDReportObject()`.

12.1.2.3 int NMD_intarray_struct::unique

Definition at line 43 of file `nmd_impl.h`.

Referenced by `NMDComponentDestroy()`, `NMDComponentSetArrayValue()`, `NMDCopyArrayValue()`, and `NMDCopyItemValues()`.

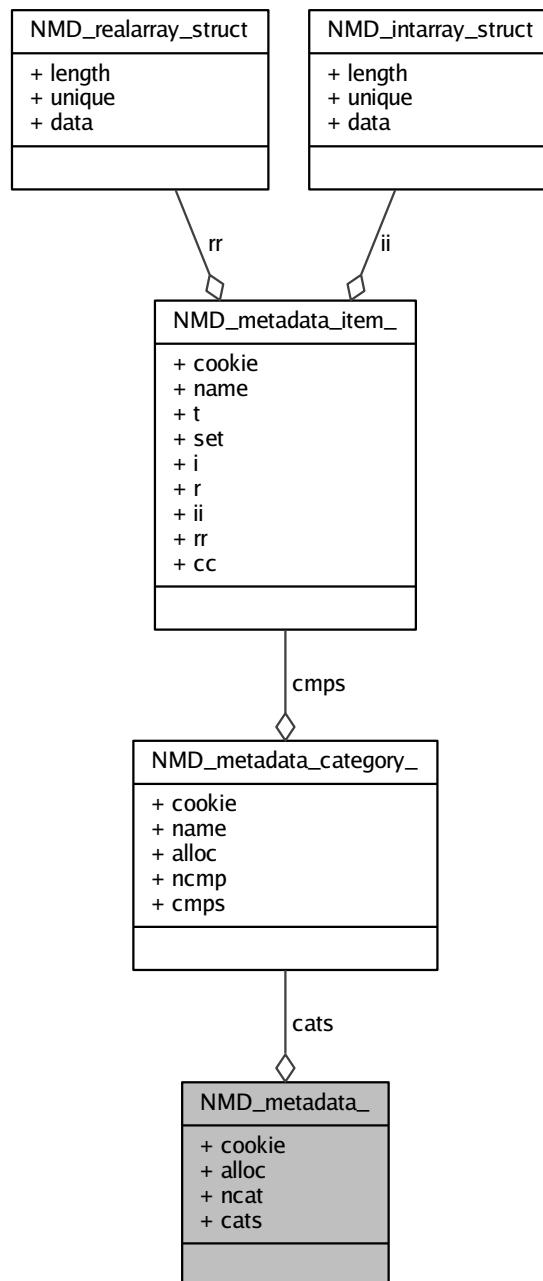
The documentation for this struct was generated from the following file:

- [nmd_impl.h](#)

12.2 NMD_metadata_Struct Reference

```
#include <nmd_impl.h>
```

Collaboration diagram for NMD_metadata_:



Data Fields

- int [cookie](#)
- int [alloc](#)
- int [ncat](#)
- [NMD_metadata_category](#) * [cats](#)

12.2.1 Detailed Description

Definition at line 27 of file [nmd_impl.h](#).

12.2.2 Field Documentation

12.2.2.1 int NMD_metadata_::alloc

Definition at line 29 of file [nmd_impl.h](#).

Referenced by [NMDCreateObject\(\)](#), and [NMDOBJECTAllocateNewCategory\(\)](#).

12.2.2.2 NMD_metadata_category* NMD_metadata_::cats

Definition at line 30 of file [nmd_impl.h](#).

Referenced by [NMDCloneObject\(\)](#), [NMDCloneObjectStructure\(\)](#), [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDGetCategories\(\)](#), [NMDGetCategoryIGetComponents\(\)](#), - [NMDOBJECTAllocateNewCategory\(\)](#), [NMDOBJECTDumpToMySQL\(\)](#), [NMDOBJECTTryGetCategory\(\)](#), [NMDReportObject\(\)](#), and [NMDViewObject\(\)](#).

12.2.2.3 int NMD_metadata_::cookie

Definition at line 28 of file [nmd_impl.h](#).

Referenced by [NMDCreateObject\(\)](#).

12.2.2.4 int NMD_metadata_::ncat

Definition at line 29 of file [nmd_impl.h](#).

Referenced by [NMDCloneObject\(\)](#), [NMDCloneObjectStructure\(\)](#), [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDGetCategories\(\)](#), [NMDOBJECTAllocateNewCategory\(\)](#), - [NMDOBJECTDumpToMySQL\(\)](#), [NMDOBJECTTryGetCategory\(\)](#), [NMDReportObject\(\)](#), and [NMDViewObject\(\)](#).

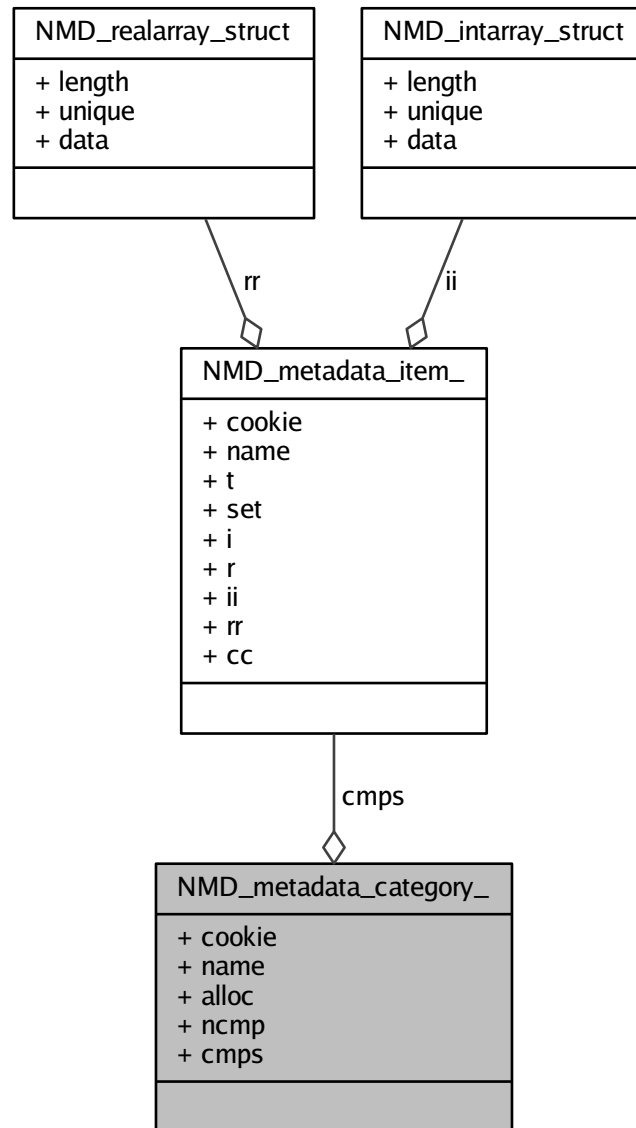
The documentation for this struct was generated from the following file:

- [nmd_impl.h](#)

12.3 NMD_metadata_category_ Struct Reference

```
#include <nmd_impl.h>
```

Collaboration diagram for NMD_metadata_category_:



Data Fields

- int `cookie`
- char * `name`
- int `alloc`
- int `ncmp`
- NMD_metadata_item * `cmps`

12.3.1 Detailed Description

Definition at line 20 of file `nmd_impl.h`.

12.3.2 Field Documentation

12.3.2.1 int NMD_metadata_category_::alloc

Definition at line 23 of file `nmd_impl.h`.

Referenced by `NMDAllocateCategory()`, and `NMDCategoryAllocateNewComponent()`.

12.3.2.2 NMD_metadata_item* NMD_metadata_category_::cmps

Definition at line 24 of file `nmd_impl.h`.

Referenced by `NMDAllocateCategory()`, `NMDCategoryAllocateNewComponent()`, `NMDCategoryGetComponent()`, `NMDCategoryTryGetComponent()`, `NMDCloneObject()`, `NMDCloneObjectStructure()`, `NMDCopyCategory()`, `NMDDestroyObject()`, `NMDGetComponent()`, `NMDGetComponent()`, `NMDObjectDumpToMySQL()`, `NMDReportObject()`, and `NMDViewObject()`.

12.3.2.3 int NMD_metadata_category_::cookie

Definition at line 21 of file `nmd_impl.h`.

Referenced by `NMDAllocateCategory()`.

12.3.2.4 char* NMD_metadata_category_::name

Definition at line 22 of file `nmd_impl.h`.

Referenced by `NMDCategoryGetComponent()`, `NMDCloneObject()`, `NMDCloneObjectStructure()`, `NMDDestroyObject()`, `NMDGetCategories()`, `NMDObjectAllocateNewCategory()`, `NMDObjectDumpToMySQL()`, `NMDObjectTryGetCategory()`, `NMDRemoveCategory()`, `NMDReportObject()`, and `NMDViewObject()`.

12.3.2.5 int NMD_metadata_category_::ncmp

Definition at line 23 of file `nmd_impl.h`.

Referenced by NMDAllocateCategory(), NMDCategoryAllocateNewComponent(), NMDCategoryGetComponents(), NMDCategoryTryGetComponent(), NMDCloneObject(), NMDCloneObjectStructure(), NMDCopyCategory(), NMDDestroyObject(), NMDGetCategoryIGetComponents(), NMDOBJECTDumpToMySQL(), NMDReportObject(), and NMDViewObject().

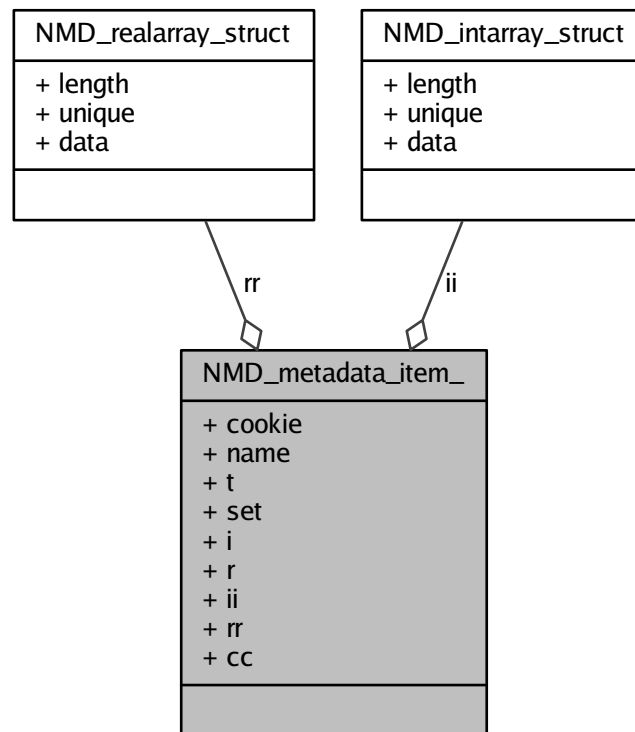
The documentation for this struct was generated from the following file:

- [nmd_impl.h](#)

12.4 NMD_metadata_item_ Struct Reference

```
#include <nmd_impl.h>
```

Collaboration diagram for NMD_metadata_item_:



Data Fields

- int [cookie](#)
- char * [name](#)
- [NMDDataType](#) t
- [NMDTruth](#) set
- int i
- double r
- struct [NMD_intarray_struct](#) * ii
- struct [NMD_realarray_struct](#) * rr
- char * cc

12.4.1 Detailed Description

Definition at line 10 of file nmd_impl.h.

12.4.2 Field Documentation

12.4.2.1 char* NMD_metadata_item::cc

Definition at line 17 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetValue(), NMDComponentUnsetValue(), NMDCopyItemValues(), NMDGetValue(), NMDReportObject(), and NMDViewObject().

12.4.2.2 int NMD_metadata_item::cookie

Definition at line 11 of file nmd_impl.h.

Referenced by NMDAllocateComponent().

12.4.2.3 int NMD_metadata_item::i

Definition at line 13 of file nmd_impl.h.

Referenced by NMDComponentSetValue(), NMDCopyItemValues(), NMDGetValue(), NMDObjectDumpToMySQL(), NMDReportObject(), and NMDViewObject().

12.4.2.4 struct NMD_intarray_struct* NMD_metadata_item::ii

Definition at line 15 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDComponentUnsetValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDGetArrayValue(), and NMDReportObject().

12.4.2.5 char* NMD_metadata_item::name

Definition at line 12 of file nmd_impl.h.

Referenced by NMDCategoryAllocateNewComponent(), NMDCategoryGetComponents(), NMDCategoryTryGetComponent(), NMDCloneObject(), NMDCloneObjectStructure(), NMDComponentDestroy(), NMDCopyCategory(), NMDGetCategoryIGetComponents(), NMDObjectDumpToMySQL(), NMDReportObject(), and NMDViewObject().

12.4.2.6 double NMD_metadata_item::r

Definition at line 14 of file nmd_impl.h.

Referenced by NMDComponentSetValue(), NMDCopyItemValues(), NMDGetValue(), NMDObjectDumpToMySQL(), NMDReportObject(), and NMDViewObject().

12.4.2.7 struct NMD_realarray_struct* NMD_metadata_item_::rr

Definition at line 16 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDComponentUnsetValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDGetArrayValue(), and NMDReportObject().

12.4.2.8 NMDTruth NMD_metadata_item_::set

Definition at line 12 of file nmd_impl.h.

Referenced by NMDAllocateComponent(), NMDCategoryAllocateNewComponent(), NMDComponentSetArrayValue(), NMDComponentSetValue(), NMDComponentUnsetValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDGetArrayValue(), NMDGetValue(), NMDReportObject(), and NMDViewObject().

12.4.2.9 NMDDataType NMD_metadata_item_::t

Definition at line 12 of file nmd_impl.h.

Referenced by NMDCategoryAllocateNewComponent(), NMDCategoryGetComponents(), NMDCategoryGetOrCreateComponent(), NMDCloneObject(), NMDCloneObjectStructure(), NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDComponentSetValue(), NMDComponentUnsetValue(), NMDCopyArrayValue(), NMDCopyCategory(), NMDCopyItemValues(), NMDGetArrayValue(), NMDGetCategoryIGetComponents(), NMDGetDataType(), NMDGetValue(), NMDObjectDumpToMySQL(), NMDObjectEnsureCategoryComponent(), NMDReportObject(), and NMDViewObject().

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

- [nmd_impl.h](#)

12.5 NMD_object_ Struct Reference

```
#include <nmd_impl.h>
```

Data Fields

- int [cookie](#)

12.5.1 Detailed Description

Definition at line 38 of file nmd_impl.h.

12.5.2 Field Documentation

12.5.2.1 int NMD_object_::cookie

Definition at line 39 of file nmd_impl.h.

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

- [nmd_impl.h](#)

12.6 NMD_realarray_struct Struct Reference

```
#include <nmd_impl.h>
```

Data Fields

- int [length](#)
- int [unique](#)
- [NMDRealtype](#) * [data](#)

12.6.1 Detailed Description

Definition at line 47 of file nmd_impl.h.

12.6.2 Field Documentation

12.6.2.1 NMDRealtype* NMD_realarray_struct::data

Definition at line 49 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDComponentUnsetValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDGetArrayValue(), and NMDReportObject().

12.6.2.2 int NMD_realarray_struct::length

Definition at line 48 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDGetArrayValue(), and NMDReportObject().

12.6.2.3 int NMD_realarray_struct::unique

Definition at line 48 of file nmd_impl.h.

Referenced by NMDComponentDestroy(), NMDComponentSetArrayValue(), NMDCopyArrayValue(), and NMDCopyItemValues().

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

- [nmd_impl.h](#)

12.7 NMD_string Struct Reference

Data Fields

- int [cookie](#)
- int [n](#)
- char * [t](#)

12.7.1 Detailed Description

Definition at line 10 of file nmdutil.c.

12.7.2 Field Documentation

12.7.2.1 int NMD_string::cookie

Definition at line 11 of file nmdutil.c.

Referenced by NMDStringCreateOfSize().

12.7.2.2 int NMD_string::n

Definition at line 12 of file nmdutil.c.

Referenced by NMDStringConcat(), and NMDStringCreateOfSize().

12.7.2.3 char* NMD_string::t

Definition at line 12 of file nmdutil.c.

Referenced by NMDStringConcat(), NMDStringCreate(), NMDStringCreateOfSize(), NMDStringDestroy(), and NMDStringGetString().

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

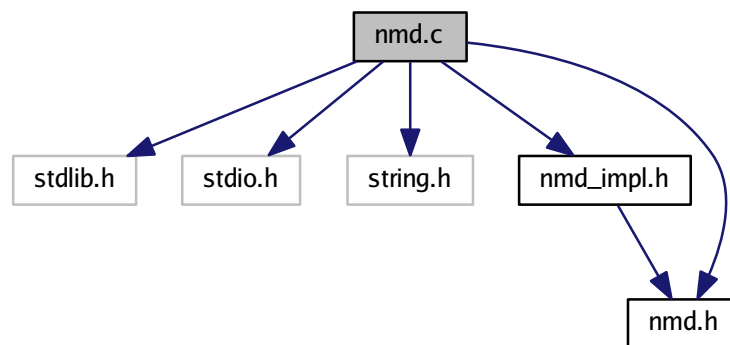
- [nmdutil.c](#)

13 File Documentation

13.1 Make.inc File Reference

13.2 nmd.c File Reference

#include <stdlib.h> #include <stdio.h> #include <string.h> #include "nmd_impl.h" #include "nmd.h" Include dependency graph for nmd.c:



Defines

- #define [CATCHUNK](#) 10
- #define [CHKLEN](#) 500
- #define [CHKSPACEFOR](#)(ar, arlen, fmt, str) { int nr; memset(scratch,0,[CHKLEN](#)); sprintf(scratch,fmt,str); nr=strlen(scratch); if (write==0) { arlen = arlen+nr+2; } else { int l = strlen(ar); if (l+nr<arlen) { sprintf(ar+l,fmt,str);} else SETERRQ(-MPI_COMM_WORLD,1,"sprintf would overflow allocated buffer");} }

Functions

- [NMDErrorCode NMDCreateObject](#) (NMD_metadata *obj)
- [NMDErrorCode NMDDestroyObject](#) (NMD_metadata obj)
- [NMDErrorCode NMDCloneObjectStructure](#) (NMD_metadata old, NMD_metadata *ret)
- [NMDErrorCode NMDCloneObject](#) (NMD_metadata old, NMD_metadata nnew)
- [NMDErrorCode NMDViewObject](#) (NMD_metadata obj)

- [NMDErrorCode NMDReportObject](#) ([NMD_metadata](#) obj, [NMDTruth](#) arrays, const char **rkey, const char **rval, const char delim, const char itemdelim1, const char itemdelim2)
- [NMDErrorCode NMDSetValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) t, void *v)
- [NMDErrorCode NMDUnsetValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp)
- [NMDErrorCode NMDSetArrayValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) t, void *v, int l)
- [NMDErrorCode NMDCopyArrayValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) t, void *v, int l)
- [NMDErrorCode NMDGetValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) *t, void *v, [NMDTruth](#) *f)
- [NMDErrorCode NMDGetArrayValue](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) *t, void *v, int *len, [NMDTruth](#) *f)
- [NMDErrorCode NMDGetDataType](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) *t)
- [NMDErrorCode NMDIsArrayType](#) ([NMDDataType](#) type, [NMDTruth](#) *flg)
- [PetscErrorCode NMDGetTypeMySQLName](#) ([NMDDataType](#) type, const char **name)

Variables

- const char * [typenames](#) []
- const char * [mysqltypenames](#) []
- static const int [nmdtypenames](#) = 6

13.2.1 Define Documentation

13.2.1.1 #define CATCHUNK 10

Definition at line 91 of file nmd.c.

Referenced by [NMDCreateObject\(\)](#).

13.2.1.2 #define CHKLEN 500

Referenced by [NMDReportObject\(\)](#).

```
13.2.1.3 #define CHKSPACEFOR( ar, arlen, fmt, str ) {int nr;
memset(scratch,0,CHKLEN); sprintf(scratch,fmt,str); nr=strlen(scratch);
if (write==0) {arlen = arlen+nr+2; } else {int l = strlen(ar); if (l+nr<arlen)
{sprintf(ar+l,fmt,str);} else SETERRQ(MPI_COMM_WORLD,1,"sprintf would
overflow allocated buffer");}}
```

Referenced by [NMDReportObject\(\)](#).

13.2.2 Function Documentation

13.2.2.1 NMDErrorCode NMDCloneObject (NMD_metadata *old*, NMD_metadata *nnew*)

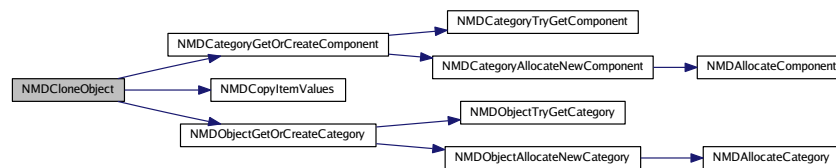
Given an already created NMD_metadata object, fill it with the data of a template object. See also [NMDCloneObjectStructure\(\)](#).

Definition at line 210 of file nmd.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_metadata_category_::ncmp, NMDCategoryGetOrCreateComponent(), NMDCopyItemValues(), NMDOBJECTGetOrCreateCategory(), and NMD_metadata_item_::t.

Referenced by main().

Here is the call graph for this function:



13.2.2.2 NMDErrorCode NMDCloneObjectStructure (NMD_metadata *old*, NMD_metadata * *ret*)

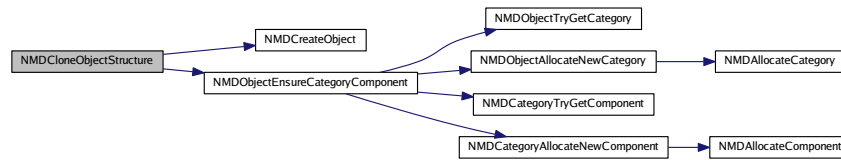
This routine creates an NMD_metadata object, and fills it in with the categories and components of a template object. Data is not copied; for that, see [NMDCloneObject\(\)](#) and [NMDCopyCategory\(\)](#).

Definition at line 181 of file nmd.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_metadata_category_::ncmp, NMDCreateObject(), NMDOBJECTEnsureCategoryComponent(), and NMD_metadata_item_::t.

Referenced by main().

Here is the call graph for this function:



13.2.2.3 NMDErrorCode NMDCopyArrayValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType *t*, void * *v*, int *l*)

Set a metadata array value; the user array is copied.

This call can be used to create categories and components; there is no checking of slight misspellings.

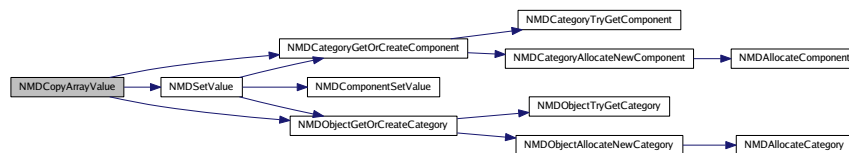
See also [Value handling](#).

Definition at line 518 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_intarray_struct::data, - NMD_rearray_struct::data, NMD_metadata_item_::ii, NMD_intarray_struct::length, NMD_rearray_struct::length, NMD_MALLOC, NMDCategoryGetOrCreateComponent(), NMDInt, NMDIntarray, NMDObjectGetOrCreateCategory(), NMDReal, NMDRearray, NMDSetValue(), NMDString, NMDTrue, NMD_metadata_item_::rr, NMD_metadata_item_::set, NMD_metadata_item_::t, NMD_intarray_struct::unique, and NMD_rearray_struct::unique.

Referenced by main().

Here is the call graph for this function:



13.2.2.4 NMDErrorCode NMDCreateObject (NMD_metadata * *obj*)

This routine create an NMD_metadata object, and allocates enough space in it for 10 categories of 20 elements each. Currently this can not be reallocated. In the future we want to be a bit more flexible.

Definition at line 108 of file nmd.c.

References NMD_metadata_::alloc, CATCHUNK, NMD_metadata_::cats, CHKMEMQ, NMD_metadata_::cookie, NMD_metadata_::ncat, NMD_MALLOC, and NMD_COOKIE.

Referenced by main(), and NMDCloneObjectStructure().

13.2.2.5 NMDErrorCode NMDDestroyObject (NMD_metadata *obj*)

Deallocate all the data in a metadata object.

Definition at line 130 of file nmd.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_metadata_category_::ncmp, NMD_FREE, and NMDComponentDestroy().

Referenced by main().

Here is the call graph for this function:



13.2.2.6 NMDErrorCode NMDGetArrayValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType * *t*, void * *v*, int * *len*, NMDTruth * *f*)

Retrieve a stored value. If no value has been stored under the specified category and component, a zero flag is returned. The flag parameter can be null.

Null pointers can be passed for the datatype or value, for instance to test only for the existence of a set value.

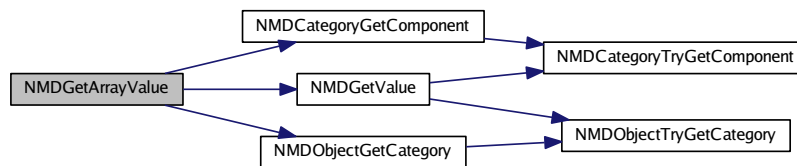
See also [Value handling](#).

Definition at line 632 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_intarray_struct::data, - NMD_realarray_struct::data, NMD_metadata_item_::ii, NMD_intarray_struct::length, NMD_realarray_struct::length, NMDCategoryGetComponent(), NMDFalse, NMDGetValue(), NMDInt, NMDIntarray, NMDObjectGetCategory(), NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::rr, NMD_metadata_item_::set, and - NMD_metadata_item_::t.

Referenced by main().

Here is the call graph for this function:

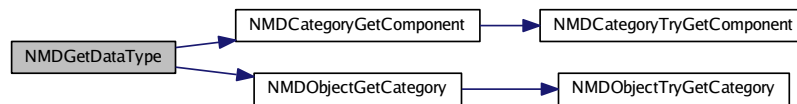


13.2.2.7 NMDErrorCode NMDGetDataType (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType * *t*)

Definition at line 720 of file nmd.c.

References CHECKHASNMDCOOKIE, NMDCategoryGetComponent(), NMDObjectGetCategory(), and NMD_metadata_item_::t.

Here is the call graph for this function:



13.2.2.8 PetscErrorCode NMDGetTypeMySQLName (NMDDataType *type*, const char ** *name*)

Definition at line 745 of file nmd.c.

References mysqltypenames, and nmmdtypenames.

Referenced by main().

13.2.2.9 NMDErrorCode NMDGetValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType * *t*, void * *v*, NMDTruth * *f*)

Retrieve a stored scalar value. If no value has been stored under the specified category and component, a zero flag is returned. The flag parameter can be null.

Null pointers can be passed for the datatype or value, for instance to test only for the existence of a set value.

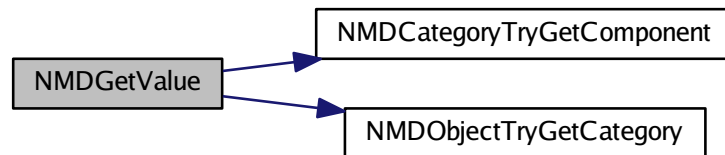
See also [Value handling](#).

Definition at line 571 of file nmd.c.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_item_::i, NMDCategoryTryGetComponent(), NMDFalse, NMDInt, NMDIntArray, NMDOBJECTTryGetCategory(), NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::r, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by main(), NMDGetArrayValue(), and NMDTabReportData().

Here is the call graph for this function:



13.2.2.10 NMDErrorCode NMDIsArrayType (NMDDataType *type*, NMDTruth * *flg*)

Test whether a data type is an array type

Definition at line 737 of file nmd.c.

References NMDFalse, NMDIntArray, NMDRealarray, and NMDTrue.

Referenced by main().

13.2.2.11 `NMDErrorCode NMDReportObject (NMD_metadata obj,
NMDTruth arrays, const char ** rkey, const char ** rval, const char
delim, const char itemdelim1, const char itemdelim2)`

Generate a delimited representation of a metadata object.

The returned strings are allocated in this routine and it is the user's responsibility to free them with [NMD_FREE\(\)](#).

Arguments:

- *obj* : the metadata object
- *ar* : boolean to indicate whether arrays need to be written out in full. If this is false, only the first and last couple of elements are given.
- *rkey* : a string containing the names of the metadata items
- *rval* : the metadata items
- *delim* : delimiter character used in *rkey* and *rval*
- *itemdelim1* : an optional opening quote, used for both keys and values. (A NULL value will cause no delimiter to be printed, rather than a null character.) For instance, use the backquote when generating MySQL strings.
- *itemdelim2* : an optional closing quote

Definition at line 297 of file `nmd.c`.

References `NMD_metadata_::cats`, `NMD_metadata_item_::cc`, `CHECKHASNMDC-OOKIE`, `CHKLEN`, `CHKMEMQ`, `CHKSPACEFOR`, `NMD_metadata_category_::cmps`, `NMD_intarray_struct::data`, `NMD_rearray_struct::data`, `NMD_metadata_item_::i`, `NMD_metadata_item_::ii`, `NMD_intarray_struct::length`, `NMD_rearray_struct::length`, `NMD_metadata_item_::name`, `NMD_metadata_category_::name`, `NMD_metadata_::ncat`, `NMD_metadata_category_::ncmp`, `NMD_FREE`, `NMD_MALLOC`, `NMDInt`, `NMDIntarray`, `NMDReal`, `NMDRearray`, `NMDString`, `NMD_metadata_item_::r`, `NMD_metadata_item_::rr`, `NMD_metadata_item_::set`, and `NMD_metadata_item_::t`.

Referenced by `main()`.

13.2.2.12 `NMDErrorCode NMDSetArrayValue (NMD_metadata obj, const
char * cat, const char * cmp, NMDDataType t, void * v, int i)`

Set a metadata value, if it is an array type.

The arrays are not copied, so the user is responsible for freeing the array. Use [NMDCopyArrayValue\(\)](#) to have the array copied; NMD will then free the array when the metadata object is freed.

This call can be used to create categories and components; there is no checking of slight misspellings.

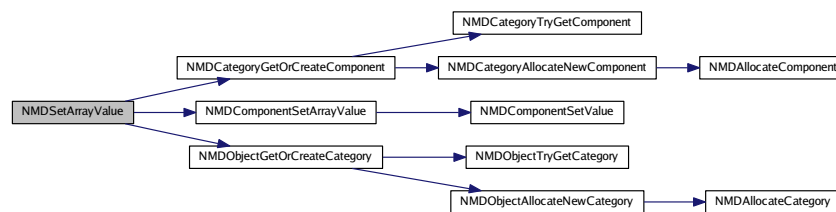
See also [Value handling](#).

Definition at line 494 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryGetOrCreateComponent(), NMDComponentSetArrayValue(), and NMDOBJECTGetOrCreateCategory().

Referenced by main().

Here is the call graph for this function:



13.2.2.13 NMDErrorCode NMDSetValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType *t*, void * *v*)

Set a metadata value, indexed by category and component name.

The value has to be passed by reference

String values are copied. (Reason: literal strings are treated differently from allocated, and Petsc has its own way of doing strings.)

This call can be used to create categories and components; there is no checking of slight misspellings.

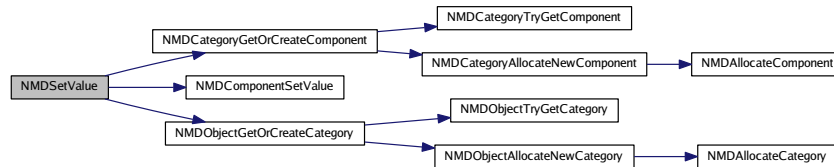
See also [Value handling](#).

Definition at line 451 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryGetOrCreateComponent(), NMDComponentSetValue(), and NMDOBJECTGetOrCreateCategory().

Referenced by main(), and NMDCopyArrayValue().

Here is the call graph for this function:

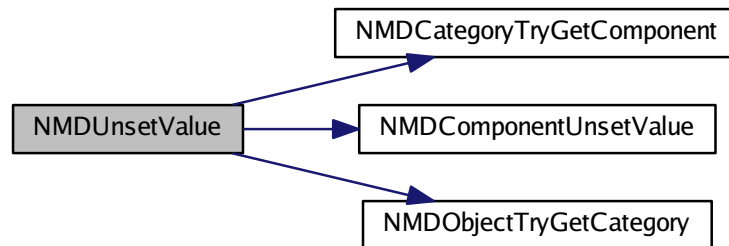


13.2.2.14 NMDErrorCode NMDUnsetValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*)

Definition at line 464 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryTryGetComponent(), NMDComponentUnsetValue(), and NMDObjectTryGetCategory().

Here is the call graph for this function:



13.2.2.15 NMDErrorCode NMDViewObject (NMD_metadata *obj*)

Print out an NMD object.

Currently only int, real, string fields are displayed, others are displayed as "***".

Definition at line 245 of file nmd.c.

References NMD_metadata_::cats, NMD_metadata_item_::cc, CHECKHASNMDC-

13.2.3 Variable Documentation

Initial value:

Definition at line 86 of file nmd.c.

Referenced by NMDGetTypeMySQLName().

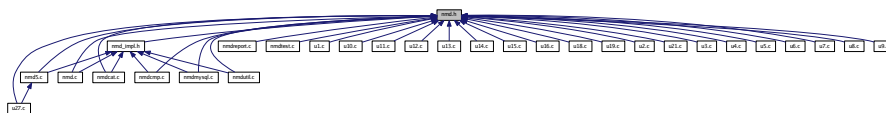
Definition at line 89 of file nmd.c.

Referenced by NMDGetTypeMySQLName().

Initial value:

Definition at line 84 of file nmd.c.

This graph shows which files directly or indirectly include this file:



- #define NMDTrue 1

- #define `NMDFalse` 0
- #define `NMDCOOKIE` 32897432
- #define `CHECKHASNMDCOOKIE`(obj) { if (!obj) NMD_ERR_REPORT("Null object"); if (((NMD_object)(obj))->cookie!=`NMDCOOKIE`) NMD_ERR_REPORTi("Object has invalid cookie",((NMD_object)(obj))->cookie); }
- #define `NMD_MALLOC`(a, b, c, d)
- #define `NMD_FREE`(a) {free(a);}
- #define `NMD_STRDUP`(a, b) b = strdup(a);

Typedefs

- typedef double `NMDRealtype`
- typedef int `NMDTruth`
- typedef int `NMDErrorCode`
- typedef struct `NMD_metadata_item_` * `NMD_metadata_item`
- typedef struct `NMD_metadata_category_` * `NMD_metadata_category`
- typedef struct `NMD_metadata_` * `NMD_metadata`
- typedef struct `NMD_object_` * `NMD_object`
- typedef struct `NMD_string_` * `NMD_string`

Enumerations

- enum `NMDDataType` { `NMDInvalid`, `NMDInt`, `NMDReal`, `NMDString`, `NMDIntArray`, `NMDRealarray` }

Functions

- `NMDErrorCode` `NMDCreateObject` (`NMD_metadata` *)
- `NMDErrorCode` `NMDDestroyObject` (`NMD_metadata`)
- `NMDErrorCode` `NMDViewObject` (`NMD_metadata`)
- `NMDErrorCode` `NMDBuildObjectStructure` (`NMD_metadata`)
- `NMDErrorCode` `NMDDestroyObjectStructure` (`NMD_metadata`)
- `NMDErrorCode` `NMDCloneObjectStructure` (`NMD_metadata`, `NMD_metadata` *)
- `NMDErrorCode` `NMDCloneObject` (`NMD_metadata`, `NMD_metadata`)
- `NMDErrorCode` `NMDReportObject` (`NMD_metadata`, `NMDTruth`, const char **, const char **, const char, const char, const char)
- `NMDErrorCode` `NMDObjectAllocateNewCategory` (`NMD_metadata`, const char *, `NMD_metadata_category` *)
- `NMDErrorCode` `NMDObjectTryGetCategory` (`NMD_metadata`, const char *, `NMD_metadata_category` *, `NMDTruth` *)
- `NMDErrorCode` `NMDObjectGetCategory` (`NMD_metadata`, const char *, `NMD_metadata_category` *)

- [NMDErrorCode NMDOBJECTGetOrCreateCategory](#) (NMD_metadata obj, const char *cat, NMD_metadata_category *ctg)
- [NMDErrorCode NMDRemoveCategory](#) (NMD_metadata, const char *)
- [NMDErrorCode NMDCopyCategory](#) (NMD_metadata_category, NMD_metadata_category)
- [NMDErrorCode NMDGetCategories](#) (NMD_metadata, int *, char ***)
- [NMDErrorCode NMDCategoryAllocateNewComponent](#) (NMD_metadata_category, const char *, NMDDataType, NMD_metadata_item *)
- [NMDErrorCode NMDComponentDestroy](#) (NMD_metadata_item)
- [NMDErrorCode NMDCategoryCreateComponent](#) (NMD_metadata, char *, char *)
- [NMDErrorCode NMDCategoryGetComponents](#) (NMD_metadata, const char *, int *, const char ***, NMDDataType **)
- [NMDErrorCode NMDCategoryGetOrCreateComponent](#) (NMD_metadata_category, const char *, NMDDataType, NMD_metadata_item *)
- [NMDErrorCode NMDCategoryTryGetComponent](#) (NMD_metadata_category, const char *, NMD_metadata_item *, NMDTruth *)
- [NMDErrorCode NMDOBJECTHasCategoryComponent](#) (NMD_metadata, const char *, const char *, NMDTruth *)
- [NMDErrorCode NMDOBJECTEnsureCategoryComponent](#) (NMD_metadata, const char *, const char *, NMDDataType, NMDTruth *)
- [NMDErrorCode NMDCategoryGetComponent](#) (NMD_metadata_category, const char *, NMD_metadata_item *)
- [NMDErrorCode NMDGetCategoryIGetComponents](#) (NMD_metadata, int, int *, char ***, NMDDataType **)
- [NMDErrorCode NMDSetValue](#) (NMD_metadata, const char *, const char *, NMDDataType, void *)
- [NMDErrorCode NMDComponentSetValue](#) (NMD_metadata_item, NMDDataType, void *)
- [NMDErrorCode NMDComponentUnsetValue](#) (NMD_metadata_item)
- [NMDErrorCode NMDSetArrayValue](#) (NMD_metadata, const char *, const char *, NMDDataType, void *, int)
- [NMDErrorCode NMDComponentSetArrayValue](#) (NMD_metadata_item, NMDDataType, void *, int)
- [NMDErrorCode NMDCopyArrayValue](#) (NMD_metadata, const char *, const char *, NMDDataType, void *, int)
- [NMDErrorCode NMDGetValue](#) (NMD_metadata, const char *, const char *, NMDDataType *, void *, NMDTruth *)
- [NMDErrorCode NMDGetArrayValue](#) (NMD_metadata, const char *, const char *, NMDDataType *, void *, int *, NMDTruth *)
- [NMDErrorCode NMDCopyItemValues](#) (NMD_metadata_item, NMD_metadata_item)
- [NMDErrorCode NMDGetDataType](#) (NMD_metadata, const char *, const char *, NMDDataType *t)

- [NMDErrorCode NMDIsArrayType](#) (NMDDataType type, NMDTruth *)
- [NMDErrorCode NMDUnsetValue](#) (NMD_metadata, const char *, const char *)
- [NMDErrorCode NMDGetTypeMySQLName](#) (NMDDataType, const char **)
- [NMDErrorCode NMDObjectDumpToMySQL](#) (NMD_metadata obj)
- [NMDErrorCode NMDStringCreate](#) (const char *, NMD_string *)
- [NMDErrorCode NMDStringDestroy](#) (NMD_string)
- [NMDErrorCode NMDStringGetString](#) (NMD_string str, const char **t)
- [NMDErrorCode NMDStringConcat](#) (char, NMD_string, char, NMD_string, char, NMD_string *)
- [NMDErrorCode NMDStringAppend](#) (char, NMD_string *, char, NMD_string, char)

Variables

- const char * [typenames](#) []

13.3.1 Define Documentation

13.3.1.1 `#define CHECKHASNMDCOOKIE(obj) { if (!obj)
NMD_ERR_REPORT("Null object"); if (((NMD_object)(obj))-
>cookie!=NMDCOOKIE) NMD_ERR_REPORTi("Object has invalid
cookie",((NMD_object)(obj))->cookie); }`

Definition at line 39 of file nmd.h.

Referenced by `NMDCategoryGetComponent()`, `NMDCategoryGetComponents()`, `NMDCategoryGetOrCreateComponent()`, `NMDCategoryTryGetComponent()`, `NMDCloneObject()`, `NMDCloneObjectStructure()`, `NMDComponentSetArrayValue()`, `NMDComponentSetValue()`, `NMDComponentUnsetValue()`, `NMDCopyArrayValue()`, `NMDCopyCategory()`, `NMDDestroyObject()`, `NMDGetArrayValue()`, `NMDGetCategories()`, `NMDGetCategoryIGetComponents()`, `NMDGetDataType()`, `NMDGetValue()`, `NMDObjectDumpToMySQL()`, `NMDObjectEnsureCategoryComponent()`, `NMDObjectGetCategory()`, `NMDObjectGetOrCreateCategory()`, `NMDObjectHasCategoryComponent()`, `NMDObjectTryGetCategory()`, `NMDRemoveCategory()`, `NMDReportObject()`, `NMDSetArrayValue()`, `NMDSetValue()`, `NMDStringAppend()`, `NMDStringConcat()`, `NMDStringDestroy()`, `NMDStringGetString()`, `NMDUnsetValue()`, and `NMDViewObject()`.

13.3.1.2 `#define NMD_FREE(a) {free(a);}`

Definition at line 145 of file nmd.h.

Referenced by `main()`, `NMDComponentDestroy()`, `NMDComponentUnsetValue()`, `NMDDestroyObject()`, `NMDRemoveCategory()`, `NMDReportObject()`, and `NMDStringDestroy()`.

13.3.1.3 #define NMD_MALLOC(a, b, c, d)

Value:

```
{ a = (c*)malloc((b)*sizeof(c)); \
    if (!a) NMD_ERR_REPORTs("Could not allocate",d); \
    memset(a,0,(b)*sizeof(c)); }
```

Definition at line 141 of file nmd.h.

Referenced by main(), NMDAllocateCategory(), NMDAllocateComponent(), NMDCategoryGetComponent(), NMDComponentSetArrayValue(), NMDCopyArrayValue(), NMDCopyItemValues(), NMDCreateObject(), NMDGetCategories(), NMDGetCategory-IGetComponents(), NMDReportObject(), NMDStringCreateOfSize(), and NMDTab-ReportData().

13.3.1.4 #define NMD_STRDUP(a, b) b = strdup(a);

Definition at line 146 of file nmd.h.

Referenced by NMDCategoryAllocateNewComponent(), NMDComponentSetValue(), NMDCopyItemValues(), NMDObjectAllocateNewCategory(), and NMDRemoveCategory().

13.3.1.5 #define NMDCOOKIE 32897432

Definition at line 38 of file nmd.h.

Referenced by NMDAllocateCategory(), NMDAllocateComponent(), NMDCreateObject(), and NMDStringCreateOfSize().

13.3.1.6 #define NMDFalse 0

Definition at line 24 of file nmd.h.

Referenced by main(), NMDAllocateComponent(), NMDCategoryAllocateNewComponent(), NMDCategoryTryGetComponent(), NMDComponentUnsetValue(), NMDGetArrayValue(), NMDGetValue(), NMDIsArrayType(), NMDObjectEnsureCategoryComponent(), and NMDObjectTryGetCategory().

13.3.1.7 #define NMDTrue 1

Definition at line 23 of file nmd.h.

Referenced by main(), NMDCategoryTryGetComponent(), NMDComponentSetArrayValue(), NMDComponentSetValue(), NMDCopyArrayValue(), NMDGetArrayValue(), NMDGetValue(), NMDIsArrayType(), NMDObjectEnsureCategoryComponent(), and NMDObjectTryGetCategory().

13.3.2 Typedef Documentation

13.3.2.1 typedef struct NMD_metadata_* NMD_metadata

Definition at line 35 of file nmd.h.

13.3.2.2 typedef struct NMD_metadata_category_* NMD_metadata_category

Definition at line 34 of file nmd.h.

13.3.2.3 typedef struct NMD_metadata_item_* NMD_metadata_item

Definition at line 33 of file nmd.h.

13.3.2.4 typedef struct NMD_object_* NMD_object

Definition at line 36 of file nmd.h.

13.3.2.5 typedef struct NMD_string_* NMD_string

Definition at line 119 of file nmd.h.

13.3.2.6 typedef int NMDErrorCode

Definition at line 25 of file nmd.h.

13.3.2.7 typedef double NMDRealtype

Definition at line 21 of file nmd.h.

13.3.2.8 typedef int NMDTruth

Definition at line 22 of file nmd.h.

13.3.3 Enumeration Type Documentation**13.3.3.1 enum NMDDataType**

Enumerator:

NMDInvalid

NMDInt

NMDReal

NMDString

NMDIntArray

NMDRealarray

Definition at line 28 of file nmd.h.

13.3.4 Function Documentation

13.3.4.1 NMDErrorCode NMDBuildObjectStructure (NMD_metadata)

13.3.4.2 NMDErrorCode NMDCategoryAllocateNewComponent (NMD_metadata_category *cat*, const char * *cmp*, NMDDataType *type*, NMD_metadata_item * *rcpt*)

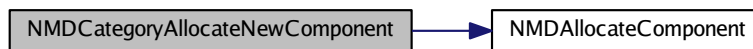
Create a new component by name in an existing category object. If a component pointer is supplied, the new component object is returned, but this pointer is allowed to be NULL.

Definition at line 41 of file nmdcmp.c.

References NMD_metadata_category_::alloc, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMD_STRDUP, NMDAllocateComponent(), NMDFalse, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by main(), NMDCategoryGetOrCreateComponent(), and NMDObjectEnsureCategoryComponent().

Here is the call graph for this function:



13.3.4.3 NMDErrorCode NMDCategoryCreateComponent (NMD_metadata , char * , char *)

13.3.4.4 NMDErrorCode NMDCategoryGetComponent (NMD_metadata_category *cat*, const char * *cmp*, NMD_metadata_item * *cpt*)

Test whether a metadata category has a certain component. The component has to exist.

Definition at line 234 of file nmdcmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::name, and NMDCategoryTryGetComponent().

Referenced by NMDGetArrayValue(), and NMDGetDataType().

Here is the call graph for this function:



13.3.4.5 NMDErrorCode NMDCategoryGetComponents (NMD_metadata *obj*, const char * *cat*, int * *ncmp*, const char *** *cmps*, NMDDataType ** *typs*)

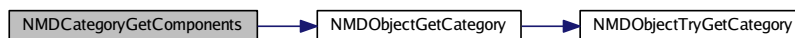
Get a list of all component names and types in a category. All three output arguments are optional. The names and types arrays are allocated and should be freed by the user by [NMD_FREE\(\)](#). The names in the name array points to the strings in the database object, so they do not need to be freed.

Definition at line 205 of file `nmdcmp.c`.

References `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::name`, `NMD_metadata_category_::ncmp`, `NMD_MALLOC`, `-NMDObjectGetCategory()`, and `NMD_metadata_item_::t`.

Referenced by `main()`.

Here is the call graph for this function:



13.3.4.6 NMDErrorCode NMDCategoryGetOrCreateComponent (NMD_metadata_category *cat*, const char * *cmp*, NMDDataType *type*, NMD_metadata_item * *cpt*)

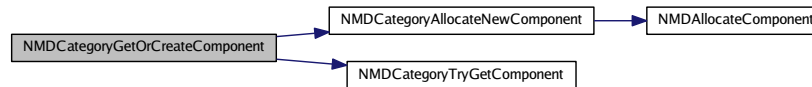
Retrieve a component, creating it if it doesn't already exist.

Definition at line 108 of file `nmdcmp.c`.

References `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMDCategoryAllocateNewComponent()`, `NMDCategoryTryGetComponent()`, and `NMD_metadata_item_::t`.

Referenced by `main()`, `NMDCloneObject()`, `NMDCopyArrayValue()`, `NMDCopyCategory()`, `NMDSetArrayValue()`, and `NMDSetValue()`.

Here is the call graph for this function:



13.3.4.7 NMDErrorCode NMDCategoryTryGetComponent (NMD_metadata_category *cat*, const char * *cmp*, NMD_metadata_item * *rcpt*, NMDTruth * *f*)

Test whether a metadata category has a certain component.

Definition at line 178 of file `nmdcmp.c`.

References `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::name`, `NMD_metadata_category_::ncmp`, `NMDFalse`, and `NMDTrue`.

Referenced by `main()`, `NMDCategoryGetComponent()`, `NMDCategoryGetOrCreateComponent()`, `NMDGetValue()`, `NMDObjectEnsureCategoryComponent()`, `NMDObjectHasCategoryComponent()`, and `NMDUnsetValue()`.

13.3.4.8 NMDErrorCode NMDCloneObject (NMD_metadata *old*, NMD_metadata *nnew*)

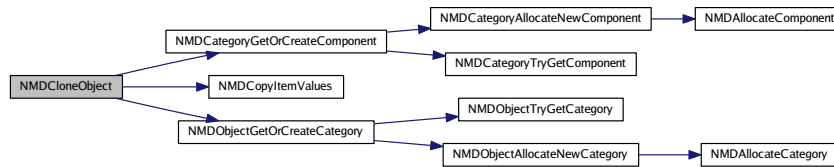
Given an already created `NMD_metadata` object, fill it with the data of a template object. See also [NMDCloneObjectStructure\(\)](#).

Definition at line 210 of file `nmd.c`.

References `NMD_metadata_::cats`, `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::name`, `NMD_metadata_category_::name`, `NMD_metadata_::ncat`, `NMD_metadata_category_::ncmp`, `NMDCategoryGetOrCreateComponent()`, `NMDCopyItemValues()`, `NMDObjectGetOrCreateCategory()`, and `NMD_metadata_item_::t`.

Referenced by `main()`.

Here is the call graph for this function:



13.3.4.9 NMDErrorCode NMDCloneObjectStructure (NMD_metadata *old*, NMD_metadata * *ret*)

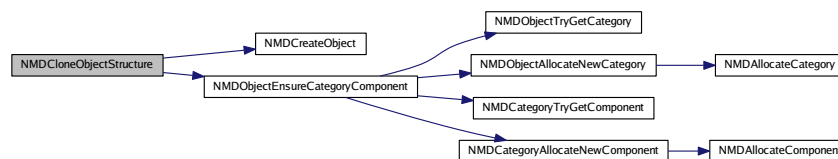
This routine creates an NMD_metadata object, and fills it in with the categories and components of a template object. Data is not copied; for that, see [NMDCloneObject\(\)](#) and [NMDCopyCategory\(\)](#).

Definition at line 181 of file nmd.c.

References `NMD_metadata::cats`, `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_metadata_category::cmps`, `NMD_metadata_item::name`, `NMD_metadata_category::name`, `NMD_metadata::ncat`, `NMD_metadata_category::ncmp`, `NMDCreateObject()`, `NMDObjectEnsureCategoryComponent()`, and `NMD_metadata_item::t`.

Referenced by `main()`.

Here is the call graph for this function:



13.3.4.10 NMDErrorCode NMDComponentDestroy (NMD_metadata_item)

Definition at line 72 of file nmdcmp.c.

References `NMD_metadata_item::cc`, `NMD_intarray_struct::data`, `NMD_rearray_struct::data`, `NMD_metadata_item::ii`, `NMD_intarray_struct::length`, `NMD_rearray_struct::length`, `NMD_metadata_item::name`, `NMD_FREE`, `NMDIntarray`, `NMDRearray`,

NMDString, NMD_metadata_item_::rr, NMD_metadata_item_::t, NMD_intarray_struct_::unique, and NMD_realarray_struct_::unique.

Referenced by NMDDestroyObject().

13.3.4.11 NMDErrorCode NMDComponentSetArrayValue (NMD_metadata_item , NMDDataType , void * , int)

Definition at line 323 of file nmncmp.c.

References CHECKHASNMDCOOKIE, NMD_intarray_struct_::data, NMD_realarray_struct_::data, NMD_metadata_item_::ii, NMD_intarray_struct_::length, NMD_realarray_struct_::length, NMD_MALLOC, NMDComponentSetValue(), NMDInt, NMDIntarray, NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::rr, NMD_metadata_item_::set, NMD_metadata_item_::t, NMD_intarray_struct_::unique, and NMD_realarray_struct_::unique.

Referenced by NMDSetArrayValue().

Here is the call graph for this function:



13.3.4.12 NMDErrorCode NMDComponentSetValue (NMD_metadata_item , NMDDataType , void *)

Definition at line 278 of file nmncmp.c.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, NMD_metadata_item_::i, NMD_STRDUP, NMDInt, NMDIntarray, NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::r, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by NMDComponentSetArrayValue(), and NMDSetValue().

13.3.4.13 NMDErrorCode NMDComponentUnsetValue (NMD_metadata_item)

Definition at line 297 of file nmncmp.c.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, NMD_intarray_struct_::data, NMD_realarray_struct_::data, NMD_metadata_item_::ii, NMD_FREE, -

NMDFalse, NMDInt, NMDIntarray, NMDReal, NMDRealarray, NMDString, NMD_metadata_item_::rr, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by NMDUnsetValue().

13.3.4.14 NMDErrorCode NMDCopyArrayValue (NMD_metadata_obj, const char * cat, const char * cmp, NMDDataType t, void * v, int l)

Set a metadata array value; the user array is copied.

This call can be used to create categories and components; there is no checking of slight misspellings.

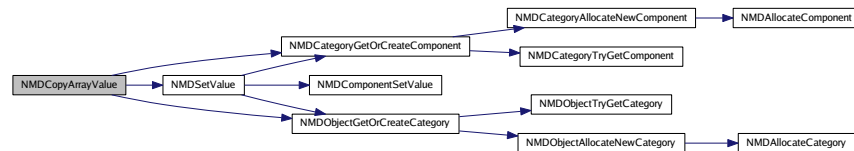
See also [Value handling](#).

Definition at line 518 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_intarray_struct::data, - NMD_realarray_struct::data, NMD_metadata_item_::ii, NMD_intarray_struct::length, NMD_realarray_struct::length, NMD_MALLOC, NMDCategoryGetOrCreateComponent(), NMDInt, NMDIntarray, NMDOBJECTGetOrCreateCategory(), NMDReal, NMDRealarray, NMDSetValue(), NMDString, NMDTrue, NMD_metadata_item_::rr, NMD_metadata_item_::set, NMD_metadata_item_::t, NMD_intarray_struct::unique, and NMD_realarray_struct::unique.

Referenced by main().

Here is the call graph for this function:



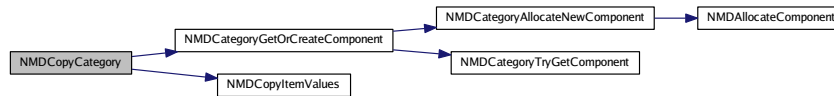
13.3.4.15 NMDErrorCode NMDCopyCategory (NMD_metadata_category incat, NMD_metadata_category outcat)

Copy category data from one metadata structure into another. This assumes that the category already exists in the target; see for instance NMDHasCategory(), [NMDCloneObject\(\)](#), [NMDCloneObjectStructure\(\)](#).

Definition at line 180 of file nmdcat.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMDCategoryGetOrCreateComponent(), NMDCopyItemValues(), and NMD_metadata_item_::t.

Here is the call graph for this function:



13.3.4.16 NMDErrorCode NMDCopyItemValues (NMD_metadata_item *src*, NMD_metadata_item *tar*)

Copy data between two item structures. If the original has unique data, so does the clone.

Definition at line 359 of file nmdcmp.c.

References NMD_metadata_item::cc, CHKMEMQ, NMD_intarray_struct::data, NMD_realarray_struct::data, NMD_metadata_item::i, NMD_metadata_item::ii, NMD_intarray_struct::length, NMD_realarray_struct::length, NMD_MALLOC, NMD_STRDUP, NMDInt, NMDIntarray, NMDReal, NMDRealarray, NMDString, NMD_metadata_item::r, NMD_metadata_item::rr, NMD_metadata_item::set, NMD_metadata_item::t, NMD_intarray_struct::unique, and NMD_realarray_struct::unique.

Referenced by NMDCloneObject(), and NMDCopyCategory().

13.3.4.17 NMDErrorCode NMDCreateObject (NMD_metadata * *obj*)

This routine create an NMD_metadata object, and allocates enough space in it for 10 categories of 20 elements each. Currently this can not be reallocated. In the future we want to be a bit more flexible.

Definition at line 108 of file nmd.c.

References NMD_metadata::alloc, CATCHUNK, NMD_metadata::cats, CHKMEMQ, NMD_metadata::cookie, NMD_metadata::ncat, NMD_MALLOC, and NMD_COOKIE.

Referenced by main(), and NMDCloneObjectStructure().

13.3.4.18 NMDErrorCode NMDDestroyObject (NMD_metadata *obj*)

Deallocate all the data in a metadata object.

Definition at line 130 of file nmd.c.

References NMD_metadata::cats, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category::cmps, NMD_metadata_category::name, NMD_metadata::ncat, NMD_metadata_category::ncmp, NMD_FREE, and NMDComponentDestroy().

Referenced by `main()`.

Here is the call graph for this function:



13.3.4.19 `NMDErrorCode NMDDestroyObjectStructure (NMD_metadata)`

13.3.4.20 `NMDErrorCode NMDGetArrayValue (NMD_metadata obj, const char * cat, const char * cmp, NMDDataType * t, void * v, int * len, NMDDTruth * f)`

Retrieve a stored value. If no value has been stored under the specified category and component, a zero flag is returned. The flag parameter can be null.

Null pointers can be passed for the datatype or value, for instance to test only for the existence of a set value.

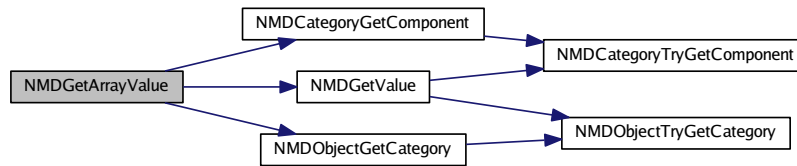
See also [Value handling](#).

Definition at line 632 of file `nmd.c`.

References `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_intarray_struct::data`, `-NMD_realarray_struct::data`, `NMD_metadata_item_::ii`, `NMD_intarray_struct::length`, `NMD_realarray_struct::length`, `NMDCategoryGetComponent()`, `NMDFalse`, `NMDGetValue()`, `NMDInt`, `NMDIntArray`, `NMDObjectGetCategory()`, `NMDReal`, `NMDRealarray`, `NMDString`, `NMDTrue`, `NMD_metadata_item_::rr`, `NMD_metadata_item_::set`, and `-NMD_metadata_item_::t`.

Referenced by `main()`.

Here is the call graph for this function:



13.3.4.21 `NMDErrorCode NMDGetCategories (NMD_metadata obj, int * ncat, char *** cats)`

Get the number of categories and their names. Both arguments can be NULL. The names array is allocated; the user needs to free it. The names themselves are pointers to the strings in the metadata object, so they do not need to be freed.

Definition at line 137 of file `nmdcat.c`.

References `NMD_metadata_::cats`, `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMD_metadata_category_::name`, `NMD_metadata_::ncat`, and `NMD_MALLOC`.

Referenced by `main()`, and `NMDTabReportData()`.

13.3.4.22 `NMDErrorCode NMDGetCategoryIGetComponents (NMD_metadata obj, int icat, int * ncmp, char *** cmps, NMDDataType ** typs)`

For a given category, get the number of components and their names.

All output arguments can be NULL. The names array is allocated; the user needs to free it. The names themselves are pointers to the strings in the metadata object, so they do not need to be freed. The types array is also allocated and needs to be freed.

Definition at line 255 of file `nmdcmp.c`.

References `NMD_metadata_::cats`, `CHECKHASNMDCOOKIE`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::name`, `NMD_metadata_category_::ncmp`, `NMD_MALLOC`, and `NMD_metadata_item_::t`.

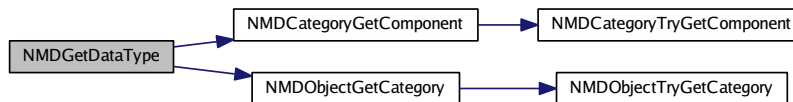
13.3.4.23 `NMDErrorCode NMDGetDataType (NMD_metadata , const char * , const char * , NMDDataType * t)`

Definition at line 720 of file `nmd.c`.

References `CHECKHASNMDCOOKIE`, `NMDCategoryGetComponent()`, `NMDObject-`

GetCategory(), and NMD_metadata_item_::t.

Here is the call graph for this function:



13.3.4.24 NMDErrorCode NMDGetTypeMySQLName (NMDDataType , const char **)

Definition at line 745 of file nmd.c.

References mysqltypenames, and nnmdtypenames.

Referenced by main().

13.3.4.25 NMDErrorCode NMDGetValue (NMD_metadata_obj, const char * cat, const char * cmp, NMDDataType * t, void * v, NMDTruth * f)

Retrieve a stored scalar value. If no value has been stored under the specified category and component, a zero flag is returned. The flag parameter can be null.

Null pointers can be passed for the datatype or value, for instance to test only for the existence of a set value.

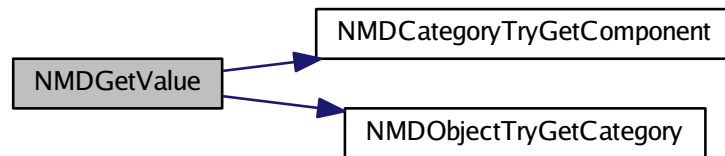
See also [Value handling](#).

Definition at line 571 of file nmd.c.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_item_::i, NMDCategoryTryGetComponent(), NMDFalse, NMDInt, NMDIntArray, NMDOBJECTTryGetCategory(), NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::r, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by main(), NMDGetArrayValue(), and NMDTabReportData().

Here is the call graph for this function:



13.3.4.26 NMDErrorCode NMDIsArrayType (NMDDataType *type*, NMDTruth * *flg*)

Test whether a data type is an array type

Definition at line 737 of file nmd.c.

References NMDFalse, NMDIntarray, NMDRealarray, and NMDTrue.

Referenced by main().

13.3.4.27 NMDErrorCode NMDObjectAllocateNewCategory (NMD_metadata_obj, const char * *cat*, NMD_metadata_category * *rctg*)

Allocate a category in a metadata object. There is no testing whether the category name is already in use.

If a category pointer is supplied, the category is returned, but this pointer is allowed to be null.

Definition at line 84 of file nmdcat.c.

References NMD_metadata_::alloc, NMD_metadata_::cats, CHKMEMQ, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_STRDUP, and NMDAllocateCategory().

Referenced by main(), NMDObjectEnsureCategoryComponent(), and NMDObject-GetOrCreateCategory().

Here is the call graph for this function:



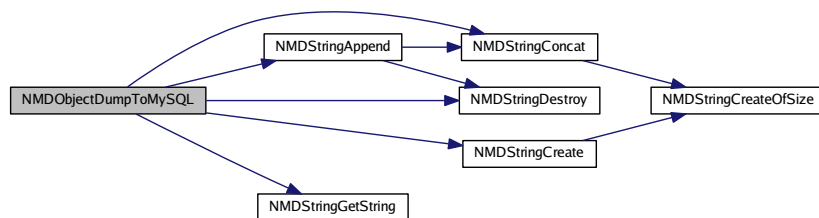
13.3.4.28 NMDErrorCode NMDObjectDumpToMySQL (NMD_metadata *obj*)

Generate an mysql dump of an object

Definition at line 11 of file `nmdmysql.c`.

References `NMD_metadata_::cats`, `CHECKHASNMDCOOKIE`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::i`, `NMD_metadata_item_::name`, `NMD_metadata_category_::name`, `NMD_metadata_::ncat`, `NMD_metadata_category_::ncmp`, `NMDInt`, `NMDReal`, `NMDStringAppend()`, `NMDStringConcat()`, `NMDStringCreate()`, `NMDStringDestroy()`, `NMDStringGetString()`, `NMD_metadata_item_::r`, and `NMD_metadata_item_::t`.

Here is the call graph for this function:



13.3.4.29 NMDErrorCode NMDObjectEnsureCategoryComponent (NMD_metadata , const char * , const char * , NMDDataType , NMDTruth *)

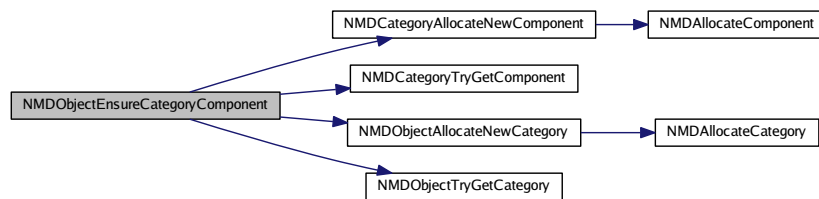
Definition at line 134 of file `nmdcmp.c`.

References `CHECKHASNMDCOOKIE`, `CHKMEMQ`, `NMDCategoryAllocateNewComponent()`, `NMDCategoryTryGetComponent()`, `NMDFalse`, `NMDObjectAllocate-`

NewCategory(), NMDOBJECTTRYGETCATEGORY(), NMDTRUE, and NMD_metadata_item_::t.

Referenced by main(), and NMDCcloneObjectStructure().

Here is the call graph for this function:



13.3.4.30 NMDErrorCode NMDOBJECT_GET_CATEGORY (NMD_metadata_obj, const char * cat, NMD_metadata_category * ctg)

Retrieve a category from a metadata object. The category has to exist.

Definition at line 49 of file nmdcat.c.

References CHECKHASNMDCOOKIE, and NMDOBJECTTRYGETCATEGORY().

Referenced by NMDCATEGORY_GET_COMPONENTS(), NMDGETARRAYVALUE(), NMDGETDATATYPE(), and NMDREMOVECATEGORY().

Here is the call graph for this function:



13.3.4.31 NMDErrorCode NMDOBJECT_GET_OR_CREATE_CATEGORY (NMD_metadata_obj, const char * cat, NMD_metadata_category * ctg)

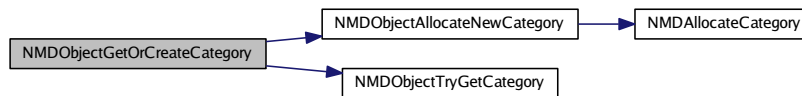
Retrieve a category from a metadata object, or create it if it doesn't exist yet.

Definition at line 118 of file nmdcat.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDObjectAllocateNewCategory(), and NMDObjectTryGetCategory().

Referenced by NMDCloneObject(), NMDCopyArrayValue(), NMDSetArrayValue(), and NMDSetValue().

Here is the call graph for this function:



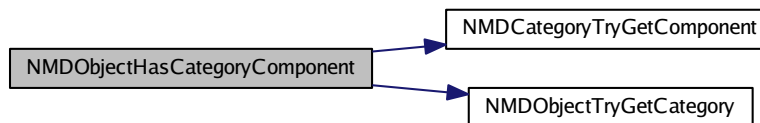
13.3.4.32 NMDErrorCode NMDObjectHasCategoryComponent (NMD_metadata , const char * , const char * , NMDTruth *)

Definition at line 161 of file nmdcmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryTryGetComponent(), and NMDObjectTryGetCategory().

Referenced by main().

Here is the call graph for this function:



13.3.4.33 NMDErrorCode NMDObjectTryGetCategory (NMD_metadata obj, const char * cat, NMD_metadata_category * rctg, NMDTruth * f)

Test whether a metadata object has a certain category, if so yield up its pointer.

The category pointer parameter can be null, in which case only existence is tested.

Definition at line 29 of file nmdcat.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, NMD_metadata_category_::name, NMD_metadata_::ncat, NMDFalse, and NMDTrue.

Referenced by main(), NMDGetValue(), NMDObjectEnsureCategoryComponent(), NMDObjectGetCategory(), NMDObjectGetOrCreateCategory(), NMDObjectHasCategoryComponent(), and NMDUnsetValue().

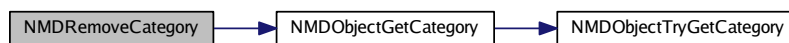
13.3.4.34 NMDErrorCode NMDRemoveCategory (NMD_metadata , const char *)

Definition at line 160 of file nmecat.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::name, NMD_FREE, NMD_STRDUP, and NMDObjectGetCategory().

Referenced by main().

Here is the call graph for this function:



13.3.4.35 NMDErrorCode NMDReportObject (NMD_metadata *obj*, NMDTruth *arrays*, const char ** *rkey*, const char ** *rval*, const char *delim*, const char *itemdelim1*, const char *itemdelim2*)

Generate a delimited representation of a metadata object.

The returned strings are allocated in this routine and it is the user's responsibility to free them with [NMD_FREE\(\)](#).

Arguments:

- *obj* : the metadata object
- *ar* : boolean to indicate whether arrays need to be written out in full. If this is false, only the first and last couple of elements are given.
- *rkey* : a string containing the names of the metadata items
- *rval* : the metadata items
- *delim* : delimiter character used in *rkey* and *rval*
- *itemdelim1* : an optional opening quote, used for both keys and values. (A NULL value will cause no delimiter to be printed, rather than a null character.) For instance, use the backquote when generating MySQL strings.

- itemdelim2 : an optional closing quote

Definition at line 297 of file nmd.c.

References NMD_metadata_::cats, NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, CHKLEN, CHKMEMQ, CHKSPACEFOR, NMD_metadata_category_::cmps, NMD_intarray_struct::data, NMD_realarray_struct::data, NMD_metadata_item_::i, - NMD_metadata_item_::ii, NMD_intarray_struct::length, NMD_realarray_struct::length, NMD_metadata_item_::name, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_metadata_category_::ncmp, NMD_FREE, NMD_MALLOC, NMDInt, NMDIntarray, NMDReal, NMDRealarray, NMDString, NMD_metadata_item_::r, NMD_metadata_item_::rr, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by main().

13.3.4.36 NMDErrorCode NMDSetArrayValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType *t*, void * *v*, int *i*)

Set a metadata value, if it is an array type.

The arrays are not copied, so the user is responsible for freeing the array. Use [NMDCopyArrayValue\(\)](#) to have the array copied; NMD will then free the array when the metadata object is freed.

This call can be used to create categories and components; there is no checking of slight misspellings.

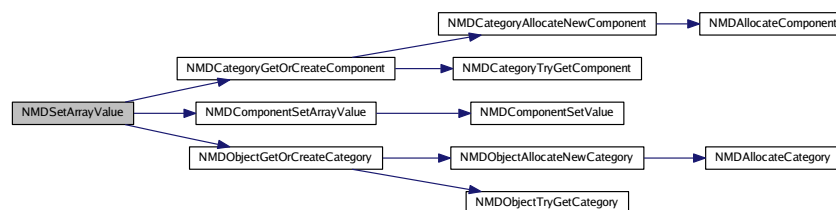
See also [Value handling](#).

Definition at line 494 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryGetOrCreateComponent(), NMDComponentSetArrayValue(), and NMDOBJECTGetOrCreateCategory().

Referenced by main().

Here is the call graph for this function:



13.3.4.37 NMDErrorCode NMDSetValue (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType *t*, void * *v*)

Set a metadata value, indexed by category and component name.

The value has to be passed by reference

String values are copied. (Reason: literal strings are treated differently from allocated, and Petsc has its own way of doing strings.)

This call can be used to create categories and components; there is no checking of slight misspellings.

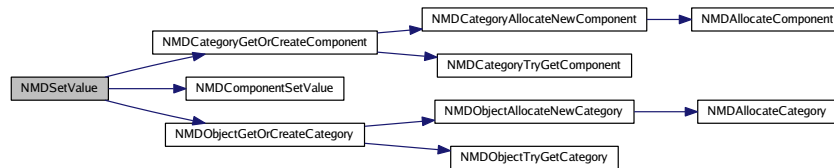
See also [Value handling](#).

Definition at line 451 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryGetOrCreateComponent(), NMDComponentSetValue(), and NMDObjectGetOrCreateCategory().

Referenced by main(), and NMDCopyArrayValue().

Here is the call graph for this function:



13.3.4.38 NMDErrorCode NMDStringAppend (char *s1*, NMD_string * *str1*, char *s2*, NMD_string *str2*, char *s3*)

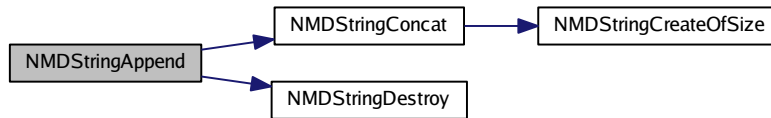
A version of [NMDStringConcat\(\)](#) that appends to string 1, rather than creating a new string.

Definition at line 121 of file nmdutil.c.

References CHECKHASNMDCOOKIE, NMDStringConcat(), and NMDStringDestroy().

Referenced by NMDObjectDumpToMySQL().

Here is the call graph for this function:



13.3.4.39 `NMDErrorCode NMDStringConcat (char s1, NMD_string str1, char s2, NMD_string str2, char s3, NMD_string * r_str)`

Concatenate string objects, with delimiter characters before, after, in between. All delimiters, and the second string, can be null.

Definition at line 74 of file `nmdutil.c`.

References `CHECKHASNMDCOOKIE`, `NMD_string_::n`, `NMDStringCreateOfSize()`, and `NMD_string_::t`.

Referenced by `NMDObjectDumpToMySQL()`, and `NMDStringAppend()`.

Here is the call graph for this function:



13.3.4.40 `NMDErrorCode NMDStringCreate (const char * txt, NMD_string * r_str)`

Create a string object around a C string; the C string is copied, so it can be freed by the calling environment.

Definition at line 36 of file `nmdutil.c`.

References `NMDStringCreateOfSize()`, and `NMD_string_::t`.

Referenced by NMDObjectDumpToMySQL().

Here is the call graph for this function:



13.3.4.41 NMDErrorCode NMDStringDestroy (NMD_string *str*)

Destroy a string object, and free the stored string.

Definition at line 49 of file nmdutil.c.

References CHECKHASNMDCOOKIE, NMD_FREE, and NMD_string_::t.

Referenced by NMDObjectDumpToMySQL(), and NMDStringAppend().

13.3.4.42 NMDErrorCode NMDStringGetString (NMD_string *str*, const char ** *t*)

Return a pointer to the string in a string object

Definition at line 60 of file nmdutil.c.

References CHECKHASNMDCOOKIE, and NMD_string_::t.

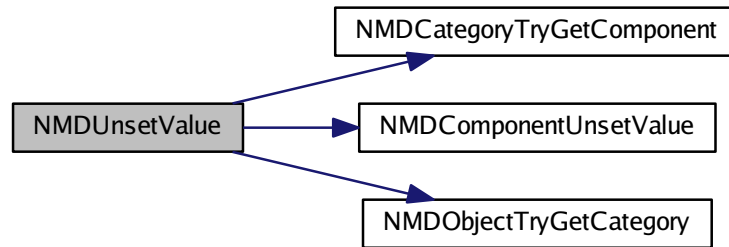
Referenced by NMDObjectDumpToMySQL().

13.3.4.43 NMDErrorCode NMDUnsetValue (NMD_metadata , const char * , const char *)

Definition at line 464 of file nmd.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryTryGetComponent(), NMDComponentUnsetValue(), and NMDObjectTryGetCategory().

Here is the call graph for this function:



13.3.4.44 NMDErrorCode NMDViewObject (NMD_metadata *obj*)

Print out an NMD object.

Currently only int, real, string fields are displayed, others are displayed as "***".

Definition at line 245 of file `nmd.c`.

References `NMD_metadata::cats`, `NMD_metadata_item::cc`, `CHECKHASNMDC-OOKIE`, `CHKMEMQ`, `NMD_metadata_category::cmps`, `NMD_metadata_item::i`, `NMD_metadata_item::name`, `NMD_metadata_category::name`, `NMD_metadata::ncat`, `NMD_metadata_category::ncmp`, `NMDInt`, `NMDReal`, `NMDString`, `NMD_metadata_item::r`, `NMD_metadata_item::set`, and `NMD_metadata_item::t`.

13.3.5 Variable Documentation

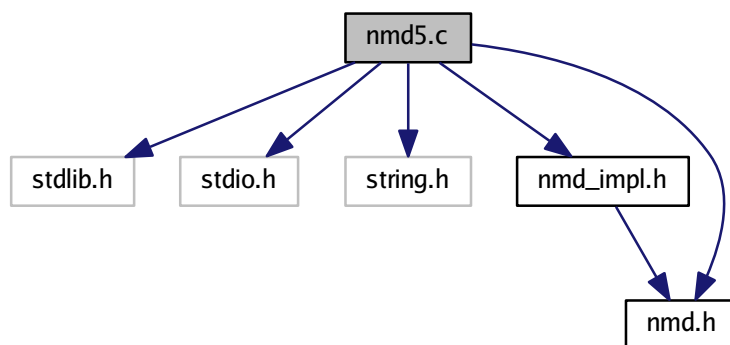
13.3.5.1 const char* typenames[]

Definition at line 84 of file `nmd.c`.

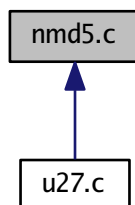
13.4 nmd5.c File Reference

```
#include <stdlib.h> #include <stdio.h> #include <string.-
h> #include "nmd_impl.h" #include "nmd.h" Include dependency graph
```


for nmd5.c:

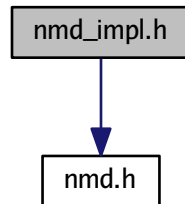


This graph shows which files directly or indirectly include this file:

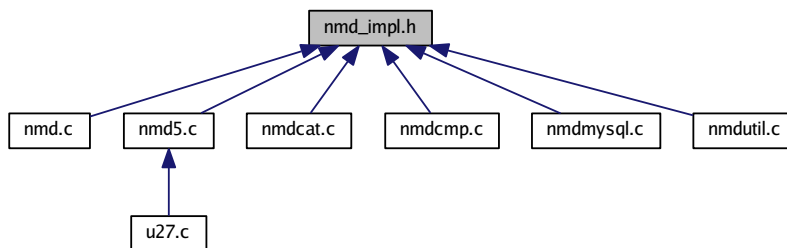


13.5 nmd_impl.h File Reference

#include "nmd.h" Include dependency graph for nmd_impl.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [NMD_metadata_item_](#)
- struct [NMD_metadata_category_](#)
- struct [NMD_metadata_](#)
- struct [NMD_object_](#)
- struct [NMD_intarray_struct](#)
- struct [NMD_realarray_struct](#)

Defines

- #define [CHKMEMQ](#)

13.5.1 Define Documentation

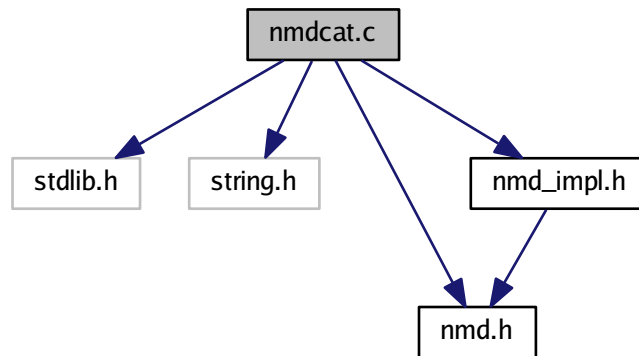
13.5.1.1 #define [CHKMEMQ](#)

Definition at line 7 of file `nmd_impl.h`.

Referenced by `NMDAllocateCategory()`, `NMDAllocateComponent()`, `NMDCategoryAllocateNewComponent()`, `NMDCategoryGetComponent()`, `NMDCategoryGetComponents()`, `NMDCategoryGetOrCreateComponent()`, `NMDCategoryTryGetComponent()`, `NMDCloneObject()`, `NMDCloneObjectStructure()`, `NMDCopyArrayValue()`, `NMDCopyCategory()`, `NMDCopyItemValues()`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetArrayValue()`, `NMDGetCategories()`, `NMDGetValue()`, `NMDObjectAllocateNewCategory()`, `NMDObjectEnsureCategoryComponent()`, `NMDObjectGetOrCreateCategory()`, `NMDObjectHasCategoryComponent()`, `NMDRemoveCategory()`, `NMDReportObject()`, `NMDSetArrayValue()`, `NMDSetValue()`, `NMDUnsetValue()`, and `NMDViewObject()`.

13.6 nmdcat.c File Reference

```
#include <stdlib.h> #include "string.h" #include "nmd.h" ×
#include "nmd_impl.h" Include dependency graph for nmdcat.c:
```



Defines

- `#define CMPCHUNK 30`

Functions

- `NMDErrorCode NMDObjectTryGetCategory (NMD_metadata obj, const char *cat, NMD_metadata_category *rctg, NMDTruth *f)`
- `NMDErrorCode NMDObjectGetCategory (NMD_metadata obj, const char *cat, NMD_metadata_category *ctg)`
- `static NMDErrorCode NMDAllocateCategory (NMD_metadata_category *rctg)`
- `NMDErrorCode NMDObjectAllocateNewCategory (NMD_metadata obj, const char *cat, NMD_metadata_category *rctg)`
- `NMDErrorCode NMDObjectGetOrCreateCategory (NMD_metadata obj, const char *cat, NMD_metadata_category *ctg)`
- `NMDErrorCode NMDGetCategories (NMD_metadata obj, int *ncat, char ***cats)`
- `NMDErrorCode NMDRemoveCategory (NMD_metadata obj, const char *cat)`
- `NMDErrorCode NMDCopyCategory (NMD_metadata_category incat, NMD_metadata_category outcat)`

13.6.1 Define Documentation

13.6.1.1 `#define CMPCHUNK 30`

Definition at line 6 of file nmecat.c.

Referenced by `NMDAllocateCategory()`.

13.6.2 Function Documentation

13.6.2.1 `static NMDErrorCode NMDAllocateCategory (NMD_metadata_category * rcat) [static]`

This is an internal routine that merely allocates the data structure for storing a category.

Definition at line 63 of file nmecat.c.

References `NMD_metadata_category_::alloc`, `CHKMEMQ`, `CMPCHUNK`, `NMD_metadata_category_::cmps`, `NMD_metadata_category_::cookie`, `NMD_metadata_category_::ncmp`, `NMD_MALLOC`, and `NMDCOOKIE`.

Referenced by `NMDObjectAllocateNewCategory()`.

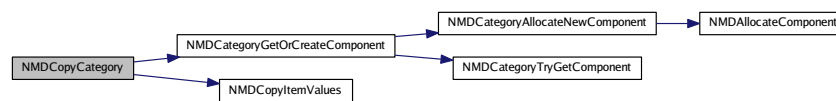
13.6.2.2 NMDErrorCode NMDCopyCategory (NMD_metadata_category *incat*, NMD_metadata_category *outcat*)

Copy category data from one metadata structure into another. This assumes that the category already exists in the target; see for instance NMDHasCategory(), [NMDCloneObject\(\)](#), [NMDCloneObjectStructure\(\)](#).

Definition at line 180 of file nmecat.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMDCategoryGetOrCreateComponent(), NMDCopyItemValues(), and NMD_metadata_item_::t.

Here is the call graph for this function:



13.6.2.3 NMDErrorCode NMDGetCategories (NMD_metadata *obj*, int * *ncat*, char *** *cats*)

Get the number of categories and their names. Both arguments can be NULL. The names array is allocated; the user needs to free it. The names themselves are pointers to the strings in the metadata object, so they do not need to be freed.

Definition at line 137 of file nmecat.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::name, NMD_metadata_::ncat, and NMD_MALLOC.

Referenced by main(), and NMDTabReportData().

13.6.2.4 NMDErrorCode NMDObjectAllocateNewCategory (NMD_metadata *obj*, const char * *cat*, NMD_metadata_category * *rctg*)

Allocate a category in a metadata object. There is no testing whether the category name is already in use.

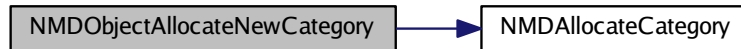
If a category pointer is supplied, the category is returned, but this pointer is allowed to be null.

Definition at line 84 of file nmecat.c.

References NMD_metadata_::alloc, NMD_metadata_::cats, CHKMEMQ, NMD_metadata_category_::name, NMD_metadata_::ncat, NMD_STRDUP, and NMDAllocateCategory().

Referenced by `main()`, `NMDOBJECT_ENSURE_CATEGORY_COMPONENT()`, and `NMDOBJECT_GET_OR_CREATE_CATEGORY()`.

Here is the call graph for this function:



13.6.2.5 NMDErrorCode NMDOBJECT_GET_CATEGORY (NMD_metadata *obj*, const char * *cat*, NMD_metadata_category * *ctg*)

Retrieve a category from a metadata object. The category has to exist.

Definition at line 49 of file `nmecat.c`.

References `CHECK_HAS_NMD_COOKIE`, and `NMDOBJECT_TRY_GET_CATEGORY()`.

Referenced by `NMDCATEGORY_GET_COMPONENTS()`, `NMD_GET_ARRAY_VALUE()`, `NMD_GET_DATA_TYPE()`, and `NMD_REMOVE_CATEGORY()`.

Here is the call graph for this function:



13.6.2.6 NMDErrorCode NMDOBJECT_GET_OR_CREATE_CATEGORY (NMD_metadata *obj*, const char * *cat*, NMD_metadata_category * *ctg*)

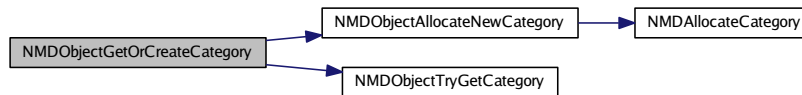
Retrieve a category from a metadata object, or create it if it doesn't exist yet.

Definition at line 118 of file `nmecat.c`.

References `CHECK_HAS_NMD_COOKIE`, `CHKMEMQ`, `NMDOBJECT_ALLOCATE_NEW_CATEGORY()`, and `NMDOBJECT_TRY_GET_CATEGORY()`.

Referenced by NMDCloneObject(), NMDCopyArrayValue(), NMDSetArrayValue(), and NMDSetValue().

Here is the call graph for this function:



13.6.2.7 NMDErrorCode NMDObjectTryGetCategory (NMD_metadata obj, const char * cat, NMD_metadata_category * rctg, NMDTruth * f)

Test whether a metadata object has a certain category, if so yield up its pointer.

The category pointer parameter can be null, in which case only existence is tested.

Definition at line 29 of file nmecat.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, NMD_metadata_category_::name, NMD_metadata_::ncat, NMDFalse, and NMDTrue.

Referenced by main(), NMDGetValue(), NMDObjectEnsureCategoryComponent(), NMDObjectGetCategory(), NMDObjectGetOrCreateCategory(), NMDObjectHasCategoryComponent(), and NMDUnsetValue().

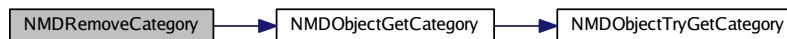
13.6.2.8 NMDErrorCode NMDRemoveCategory (NMD_metadata obj, const char * cat)

Definition at line 160 of file nmecat.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::name, NMD_FREE, NMD_STRDUP, and NMDObjectGetCategory().

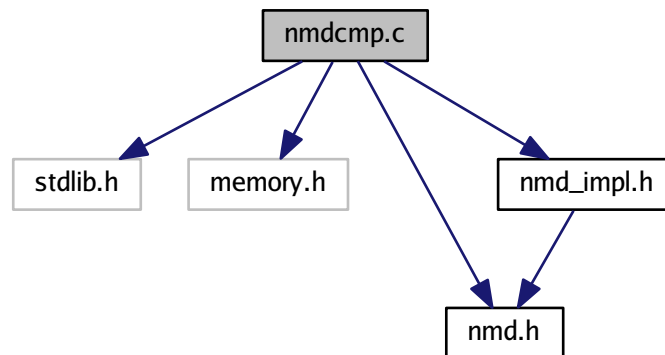
Referenced by main().

Here is the call graph for this function:



13.7 nmdcmp.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
#include "nmd_impl.h" Include dependency graph for nmdcmp.c:
```



Functions

- static [NMDErrorCode](#) [NMDAllocateComponent](#) ([NMD_metadata_item](#) *rcmp)
- [NMDErrorCode](#) [NMDCategoryAllocateNewComponent](#) ([NMD_metadata_category](#) cat, const char *cmp, [NMDDataType](#) type, [NMD_metadata_item](#) *rcpt)
- [NMDErrorCode](#) [NMDCategoryDestroy](#) ([NMD_metadata_item](#) cmp)
- [NMDErrorCode](#) [NMDCategoryGetOrCreateComponent](#) ([NMD_metadata_category](#) cat, const char *cmp, [NMDDataType](#) type, [NMD_metadata_item](#) *cpt)
- [NMDErrorCode](#) [NMDObjEnsureCategoryComponent](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDDataType](#) type, [NMDTruth](#) *nnew)
- [NMDErrorCode](#) [NMDObjHasCategoryComponent](#) ([NMD_metadata](#) obj, const char *cat, const char *cmp, [NMDTruth](#) *f)
- [NMDErrorCode](#) [NMDCategoryTryGetComponent](#) ([NMD_metadata_category](#) cat, const char *cmp, [NMD_metadata_item](#) *rcpt, [NMDTruth](#) *f)
- [NMDErrorCode](#) [NMDCategoryGetComponents](#) ([NMD_metadata](#) obj, const char *cat, int *ncmp, const char ***cmps, [NMDDataType](#) **types)
- [NMDErrorCode](#) [NMDCategoryGetComponent](#) ([NMD_metadata_category](#) cat, const char *cmp, [NMD_metadata_item](#) *cpt)
- [NMDErrorCode](#) [NMDGetCategoryGetComponent](#) ([NMD_metadata](#) obj, int icat, int *ncmp, char ***cmps, [NMDDataType](#) **types)

- [NMDErrorCode NMDComponentSetValue](#) ([NMD_metadata_item](#) cpt, [NMDDataType](#) t, void *v)
- [NMDErrorCode NMDComponentUnsetValue](#) ([NMD_metadata_item](#) cpt)
- [NMDErrorCode NMDComponentSetArrayValue](#) ([NMD_metadata_item](#) cpt, [NMDDataType](#) t, void *v, int l)
- [PetscErrorCode NMDCopyItemValues](#) ([NMD_metadata_item](#) src, [NMD_metadata_item](#) tar)

13.7.1 Function Documentation

13.7.1.1 static NMDErrorCode NMDAllocateComponent (NMD_metadata_item *rcmp) [static]

An internal routine that only allocates the component data structure

Definition at line 24 of file nmdcmp.c.

References [CHKMEMQ](#), [NMD_metadata_item::cookie](#), [NMD_MALLOC](#), [NMDCOOKIE](#), [NMDFalse](#), and [NMD_metadata_item::set](#).

Referenced by [NMDCategoryAllocateNewComponent\(\)](#).

13.7.1.2 NMDErrorCode NMDCategoryAllocateNewComponent (NMD_metadata_category cat, const char *cmp, NMDDataType type, NMD_metadata_item *rcpt)

Create a new component by name in an existing category object. If a component pointer is supplied, the new component object is returned, but this pointer is allowed to be NULL.

Definition at line 41 of file nmdcmp.c.

References [NMD_metadata_category::alloc](#), [CHKMEMQ](#), [NMD_metadata_category::cmps](#), [NMD_metadata_item::name](#), [NMD_metadata_category::ncmp](#), [NMD_STRDUP](#), [NMDAllocateComponent\(\)](#), [NMDFalse](#), [NMD_metadata_item::set](#), and [NMD_metadata_item::t](#).

Referenced by [main\(\)](#), [NMDCategoryGetOrCreateComponent\(\)](#), and [NMDObjectEnsureCategoryComponent\(\)](#).

Here is the call graph for this function:



13.7.1.3 NMDErrorCode NMDCategoryGetComponent (NMD_metadata_category *cat*, const char * *cmp*, NMD_metadata_item * *cpt*)

Test whether a metadata category has a certain component. The component has to exist.

Definition at line 234 of file nmncmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::name, and NMDCategoryTryGetComponent().

Referenced by NMDGetArrayValue(), and NMDGetDataType().

Here is the call graph for this function:



13.7.1.4 NMDErrorCode NMDCategoryGetComponents (NMD_metadata_obj, const char * *cat*, int * *ncmp*, const char *** *cmps*, NMDDataType ** *typs*)

Get a list of all component names and types in a category. All three output arguments are optional. The names and types arrays are allocated and should be freed by the user by [NMD_FREE\(\)](#). The names in the name array points to the strings in the database object, so they do not need to be freed.

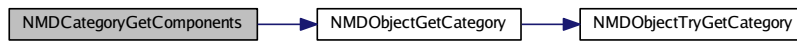
Definition at line 205 of file nmncmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps,

NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMD_MALLOC, - NMDObjectGetCategory(), and NMD_metadata_item_::t.

Referenced by main().

Here is the call graph for this function:



13.7.1.5 NMDErrorCode NMDCategoryGetOrCreateComponent (NMD_metadata_category *cat*, const char * *cmp*, NMDDataType *type*, NMD_metadata_item * *cpt*)

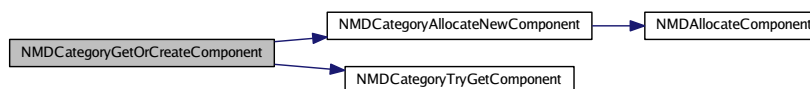
Retrieve a component, creating it if it doesn't already exist.

Definition at line 108 of file nmdcmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryAllocateNewComponent(), NMDCategoryTryGetComponent(), and NMD_metadata_item_::t.

Referenced by main(), NMDCloneObject(), NMDCopyArrayValue(), NMDCopyCategory(), NMDSetArrayValue(), and NMDSetValue().

Here is the call graph for this function:



13.7.1.6 NMDErrorCode NMDCategoryTryGetComponent (NMD_metadata_category *cat*, const char * *cmp*, NMD_metadata_item * *rcpt*, NMDTruth * *f*)

Test whether a metadata category has a certain component.

Definition at line 178 of file nmdcmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMDFalse, and N-

MDTrue.

Referenced by `main()`, `NMDCategoryGetComponent()`, `NMDCategoryGetOrCreateComponent()`, `NMDGetValue()`, `NMDObjectEnsureCategoryComponent()`, `NMDObjectHasCategoryComponent()`, and `NMDUnsetValue()`.

13.7.1.7 NMDErrorCode NMDComponentDestroy (NMD_metadata_item *cmp*)

Definition at line 72 of file `nmncmp.c`.

References `NMD_metadata_item::cc`, `NMD_intarray_struct::data`, `NMD_realarray_struct::data`, `NMD_metadata_item::ii`, `NMD_intarray_struct::length`, `NMD_realarray_struct::length`, `NMD_metadata_item::name`, `NMD_FREE`, `NMDIntarray`, `NMDRealarray`, `NMDString`, `NMD_metadata_item::rr`, `NMD_metadata_item::t`, `NMD_intarray_struct::unique`, and `NMD_realarray_struct::unique`.

Referenced by `NMDDestroyObject()`.

13.7.1.8 NMDErrorCode NMDComponentSetArrayValue (NMD_metadata_item *cpt*, NMDDataType *t*, void * *v*, int *l*)

Definition at line 323 of file `nmncmp.c`.

References `CHECKHASNMDCOOKIE`, `NMD_intarray_struct::data`, `NMD_realarray_struct::data`, `NMD_metadata_item::ii`, `NMD_intarray_struct::length`, `NMD_realarray_struct::length`, `NMD_MALLOC`, `NMDComponentSetValue()`, `NMDInt`, `NMDIntarray`, `NMDReal`, `NMDRealarray`, `NMDString`, `NMDTrue`, `NMD_metadata_item::rr`, `NMD_metadata_item::set`, `NMD_metadata_item::t`, `NMD_intarray_struct::unique`, and `NMD_realarray_struct::unique`.

Referenced by `NMDSetArrayValue()`.

Here is the call graph for this function:



13.7.1.9 NMDErrorCode NMDComponentSetValue (NMD_metadata_item *cpt*, NMDDataType *t*, void * *v*)

Definition at line 278 of file `nmncmp.c`.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, NMD_metadata_item_::i, NMD_STRDUP, NMDInt, NMDIntArray, NMDReal, NMDRealarray, NMDString, NMDTrue, NMD_metadata_item_::r, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by NMDComponentSetArrayValue(), and NMDSetValue().

13.7.1.10 NMDErrorCode NMDComponentUnsetValue (NMD_metadata_item *cpt*)

Definition at line 297 of file nmncmp.c.

References NMD_metadata_item_::cc, CHECKHASNMDCOOKIE, NMD_intarray_struct::data, NMD_realarray_struct::data, NMD_metadata_item_::ii, NMD_FREE, NMDFalse, NMDInt, NMDIntArray, NMDReal, NMDRealarray, NMDString, NMD_metadata_item_::rr, NMD_metadata_item_::set, and NMD_metadata_item_::t.

Referenced by NMDUnsetValue().

13.7.1.11 PetscErrorCode NMDCopyItemValues (NMD_metadata_item *src*, NMD_metadata_item *tar*)

Copy data between two item structures. If the original has unique data, so does the clone.

Definition at line 359 of file nmncmp.c.

References NMD_metadata_item_::cc, CHKMEMQ, NMD_intarray_struct::data, NMD_realarray_struct::data, NMD_metadata_item_::i, NMD_metadata_item_::ii, NMD_intarray_struct::length, NMD_realarray_struct::length, NMD_MALLOC, NMD_STRDUP, NMDInt, NMDIntArray, NMDReal, NMDRealarray, NMDString, NMD_metadata_item_::r, NMD_metadata_item_::rr, NMD_metadata_item_::set, NMD_metadata_item_::t, NMD_intarray_struct::unique, and NMD_realarray_struct::unique.

Referenced by NMDCloneObject(), and NMDCopyCategory().

13.7.1.12 NMDErrorCode NMDGetCategoryIGetComponents (NMD_metadata *obj*, int *icat*, int * *ncmp*, char *** *cmps*, NMDDataType ** *types*)

For a given category, get the number of components and their names.

All output arguments can be NULL. The names array is allocated; the user needs to free it. The names themselves are pointers to the strings in the metadata object, so they do not need to be freed. The types array is also allocated and needs to be freed.

Definition at line 255 of file nmncmp.c.

References NMD_metadata_::cats, CHECKHASNMDCOOKIE, NMD_metadata_category_::cmps, NMD_metadata_item_::name, NMD_metadata_category_::ncmp, NMD_MALLOC, and NMD_metadata_item_::t.

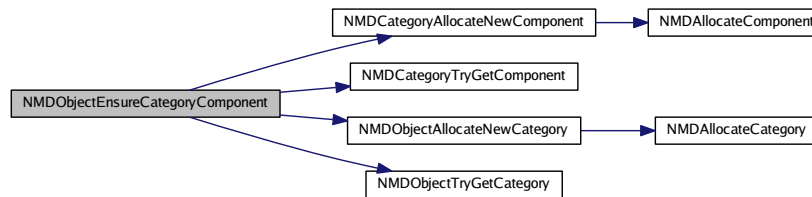
13.7.1.13 NMDErrorCode NMDOBJECTEnsureCategoryComponent (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDDataType *type*, NMDTruth * *nnew*)

Definition at line 134 of file nmncmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryAllocateNewComponent(), NMDCategoryTryGetComponent(), NMDFalse, NMDOBJECTAllocateNewCategory(), NMDOBJECTTryGetCategory(), NMDTrue, and NMD_metadata_item_::t.

Referenced by main(), and NMDCCloneObjectStructure().

Here is the call graph for this function:



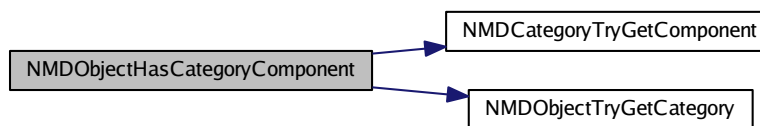
13.7.1.14 NMDErrorCode NMDOBJECTHasCategoryComponent (NMD_metadata *obj*, const char * *cat*, const char * *cmp*, NMDTruth * *f*)

Definition at line 161 of file nmncmp.c.

References CHECKHASNMDCOOKIE, CHKMEMQ, NMDCategoryTryGetComponent(), and NMDOBJECTTryGetCategory().

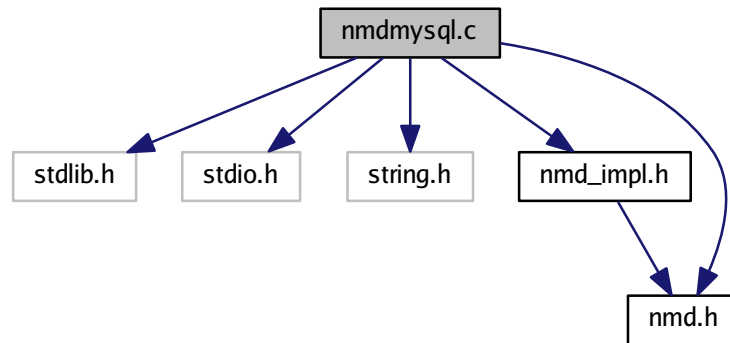
Referenced by main().

Here is the call graph for this function:



13.8 nmdmysql.c File Reference

```
#include <stdlib.h> #include <stdio.h> #include <string.-
h> #include "nmd_impl.h" #include "nmd.h" Include dependency graph
for nmdmysql.c:
```



Functions

- [NMDErrorCode NMDObjectDumpToMySQL \(NMD_metadata obj\)](#)

13.8.1 Function Documentation

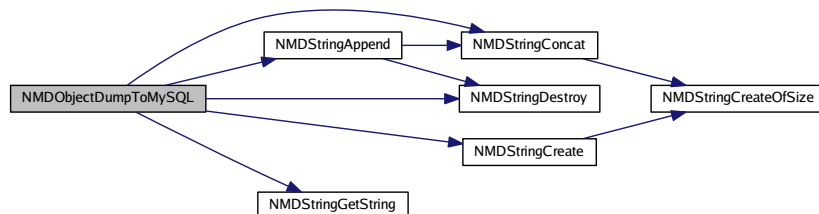
13.8.1.1 NMDErrorCode NMDObjectDumpToMySQL (NMD_metadata *obj*)

Generate an mysql dump of an object

Definition at line 11 of file nmdmysql.c.

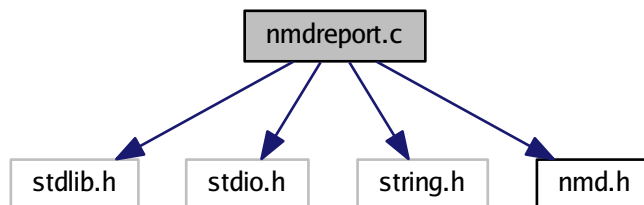
References `NMD_metadata_::cats`, `CHECKHASNMDCOOKIE`, `NMD_metadata_category_::cmps`, `NMD_metadata_item_::i`, `NMD_metadata_item_::name`, `NMD_metadata_category_::name`, `NMD_metadata_::ncat`, `NMD_metadata_category_::ncmp`, `NMDInt`, `NMDReal`, `NMDStringAppend()`, `NMDStringConcat()`, `NMDStringCreate()`, `NMDStringDestroy()`, `NMDStringGetString()`, `NMD_metadata_item_::r`, and `NMD_metadata_item_::t`.

Here is the call graph for this function:



13.9 nmdreport.c File Reference

```
#include <stdlib.h> #include <stdio.h> #include <string.-
h> #include "nmd.h" Include dependency graph for nmdreport.c:
```



Functions

- int [NMDTabReportData](#) (NMD_metadata nmd, char **rkey, char **rval, int separator)

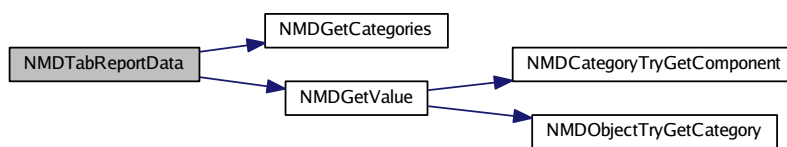
13.9.1 Function Documentation

13.9.1.1 int NMDTabReportData (NMD_metadata *nmd*, char ** *rkey*, char ** *rval*, int *separator*)

Definition at line 8 of file nmdreport.c.

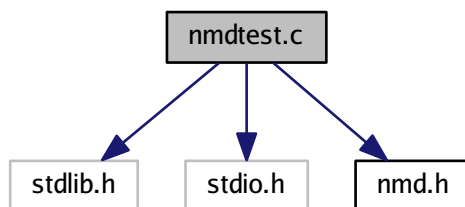
References NMD_MALLOC, NMDGetCategories(), NMDGetValue(), NMDIntarray, and NMDRearray.

Here is the call graph for this function:



13.10 nmdtest.c File Reference

`#include <stdlib.h> #include <stdio.h> #include "nmd.h"` Include dependency graph for `nmdtest.c`:



Functions

- int `main` (int argc, char **argv)

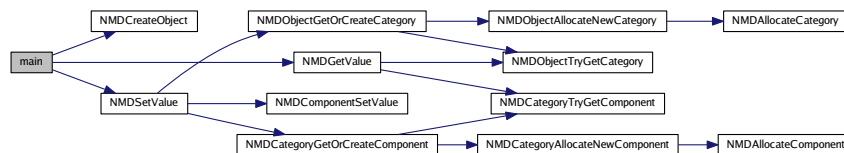
13.10.1 Function Documentation

13.10.1.1 `int main (int argc, char ** argv)`

Definition at line 5 of file nmdtest.c.

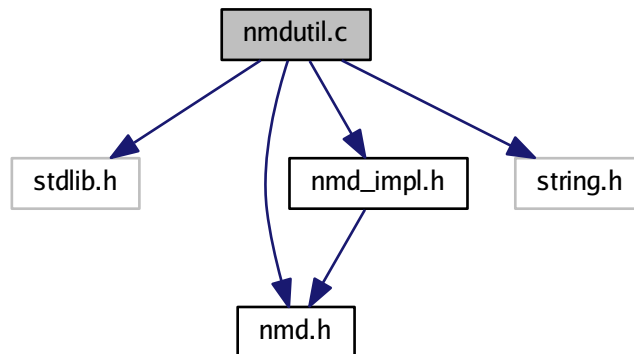
References `NMDCreateObject()`, `NMDGetValue()`, `NMDInt`, `NMDReal`, and `NMDSetValue()`.

Here is the call graph for this function:



13.11 nmdutil.c File Reference

```
#include <stdlib.h> #include "nmd.h" #include "nmd_impl.h" ×
#include "string.h" Include dependency graph for nmdutil.c:
```



Data Structures

- struct [NMD_string_](#)

Functions

- static [NMDErrorCode](#) [NMDStringCreateOfSize](#) (int n, [NMD_string](#) *r_str)
- [NMDErrorCode](#) [NMDStringCreate](#) (const char *txt, [NMD_string](#) *r_str)
- [NMDErrorCode](#) [NMDStringDestroy](#) ([NMD_string](#) str)
- [NMDErrorCode](#) [NMDStringGetString](#) ([NMD_string](#) str, const char **t)
- [NMDErrorCode](#) [NMDStringConcat](#) (char s1, [NMD_string](#) str1, char s2, [NMD_string](#) str2, char s3, [NMD_string](#) *r_str)
- [NMDErrorCode](#) [NMDStringAppend](#) (char s1, [NMD_string](#) *str1, char s2, [NMD_string](#) str2, char s3)

13.11.1 Function Documentation

13.11.1.1 [NMDErrorCode](#) [NMDStringAppend](#) (char s1, [NMD_string](#) * str1, char s2, [NMD_string](#) str2, char s3)

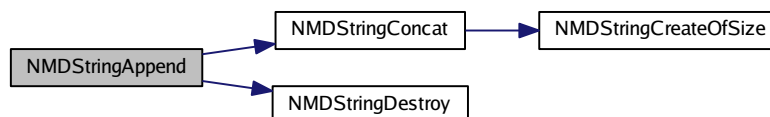
A version of [NMDStringConcat\(\)](#) that appends to string 1, rather than creating a new string.

Definition at line 121 of file nmdutil.c.

References [CHECKHASNMDCOOKIE](#), [NMDStringConcat\(\)](#), and [NMDStringDestroy\(\)](#).

Referenced by [NMDObjectDumpToMySQL\(\)](#).

Here is the call graph for this function:



13.11.1.2 [NMDErrorCode](#) [NMDStringConcat](#) (char s1, [NMD_string](#) str1, char s2, [NMD_string](#) str2, char s3, [NMD_string](#) * r_str)

Concatenate string objects, with delimiter characters before, after, in between. All delimiters, and the second string, can be null.

Definition at line 74 of file nmdutil.c.

References CHECKHASNMDCOOKIE, NMD_string_::n, NMDStringCreateOfSize(), and NMD_string_::t.

Referenced by NMDObjectDumpToMySQL(), and NMDStringAppend().

Here is the call graph for this function:



13.11.1.3 NMDErrorCode NMDStringCreate (const char * *txt*, NMD_string * *r_str*)

Create a string object around a C string; the C string is copied, so it can be freed by the calling environment.

Definition at line 36 of file nmdutil.c.

References NMDStringCreateOfSize(), and NMD_string_::t.

Referenced by NMDObjectDumpToMySQL().

Here is the call graph for this function:



13.11.1.4 static NMDErrorCode NMDStringCreateOfSize (int *n*, NMD_string * *r_str*) [static]

Internal auxiliary function for creating a string object of a given length. Zero length is allowed.

Definition at line 21 of file nmdutil.c.

References NMD_string_::cookie, NMD_string_::n, NMD_MALLOC, NMDCOOKIE, and NMD_string_::t.

Referenced by NMDStringConcat(), and NMDStringCreate().

13.11.1.5 NMDErrorCode NMDStringDestroy (NMD_string *str*)

Destroy a string object, and free the stored string.

Definition at line 49 of file nmdutil.c.

References CHECKHASNMDCOOKIE, NMD_FREE, and NMD_string_::t.

Referenced by NMDObjectDumpToMySQL(), and NMDStringAppend().

13.11.1.6 NMDErrorCode NMDStringGetString (NMD_string *str*, const char ** *t*)

Return a pointer to the string in a string object

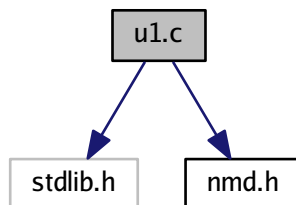
Definition at line 60 of file nmdutil.c.

References CHECKHASNMDCOOKIE, and NMD_string_::t.

Referenced by NMDObjectDumpToMySQL().

13.12 u1.c File Reference

#include <stdlib.h> #include "nmd.h" Include dependency graph for u1.c:



Functions

- int `main` (int argc, char **argv)

13.12.1 Function Documentation

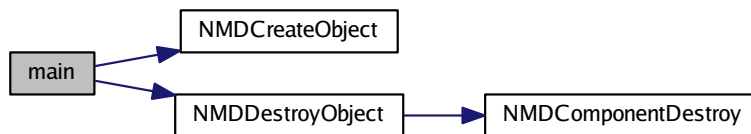
13.12.1.1 int main (int argc, char ** argv)

Test setting and getting values

Definition at line 5 of file u1.c.

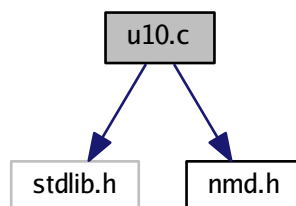
References `NMDCreateObject()`, and `NMDDestroyObject()`.

Here is the call graph for this function:



13.13 u10.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for `u10.c`:



Functions

- int `main` (int argc, char **argv)

13.13.1 Function Documentation

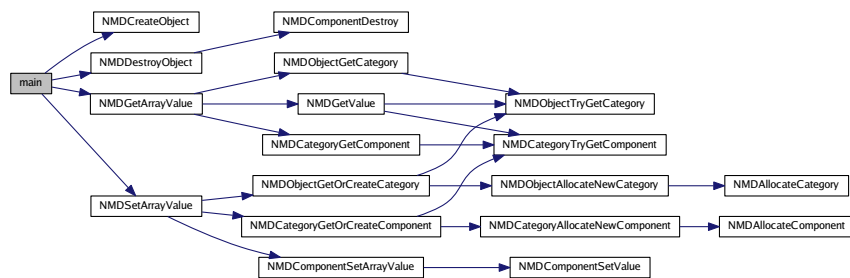
13.13.1.1 int main (int argc, char ** argv)

Test setting and getting of array values

Definition at line 5 of file u10.c.

References NMD_FREE, NMD_MALLOC, NMDCreateObject(), NMDDestroyObject(), NMDGetArrayValue(), NMDIntarray, NMDRealarray, and NMDSetArrayValue().

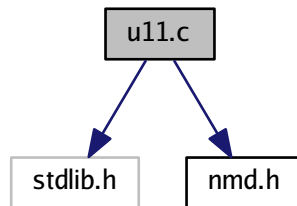
Here is the call graph for this function:



13.14 u11.c File Reference

#include <stdlib.h> #include "nmd.h" Include dependency graph for

u11.c:



Functions

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

13.14.1 Function Documentation

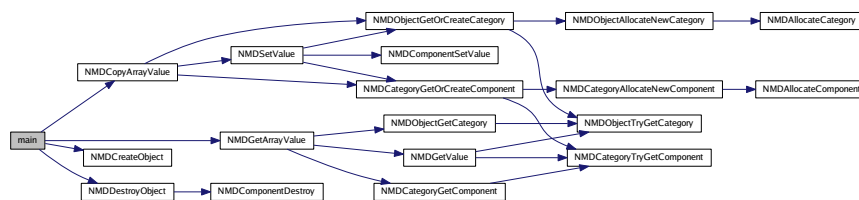
13.14.1.1 int main (int argc, char ** argv)

Test setting and getting of array values with internal copy

Definition at line 5 of file u11.c.

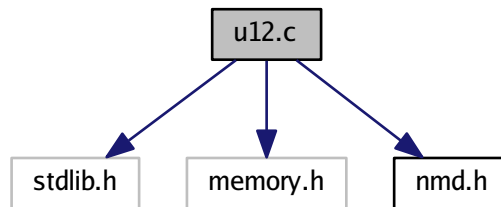
References `NMD_FREE`, `NMD_MALLOC`, `NMDCopyArrayValue()`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetArrayValue()`, and `NMDIntarray`.

Here is the call graph for this function:



13.15 u12.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×  
Include dependency graph for u12.c:
```



Functions

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

13.15.1 Function Documentation

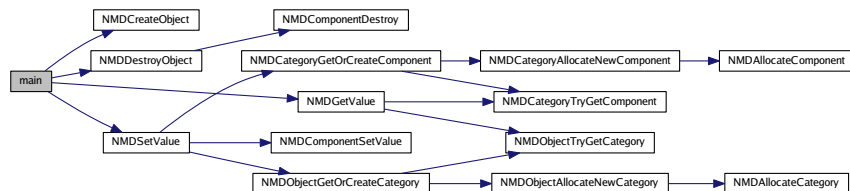
13.15.1.1 int main (int *argc*, char ** *argv*)

Stress test

Definition at line 6 of file u12.c.

References [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDGetValue\(\)](#), [NMDInt](#), and [NMDSetValue\(\)](#).

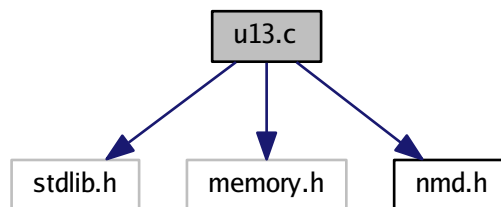
Here is the call graph for this function:



13.16 u13.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
```

Include dependency graph for u13.c:



Defines

- #define [ILEN](#) 4
- #define [RLEN](#) 6

Functions

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

13.16.1 Define Documentation

13.16.1.1 #define ILEN 4

Referenced by [main\(\)](#).

13.16.1.2 #define RLEN 6

Referenced by [main\(\)](#).

13.16.2 Function Documentation

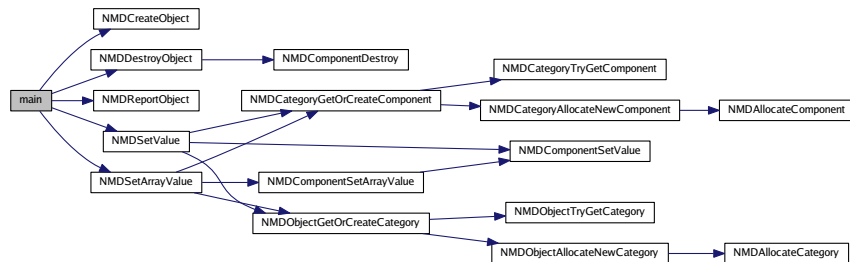
13.16.2.1 int main (int *argc*, char ** *argv*)

Object reporting

Definition at line 6 of file u13.c.

References ILEN, NMD_FREE, NMD_MALLOC, NMDCreateObject(), NMDDestroyObject(), NMDFalse, NMDInt, NMDIntArray, NMDReal, NMDRealarray, NMDReportObject(), NMDSetArrayValue(), NMDSetValue(), NMDTrue, and RLEN.

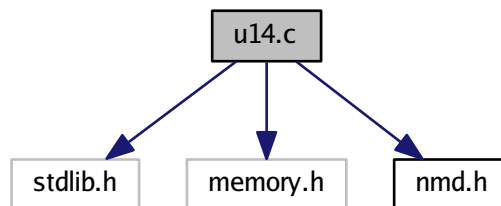
Here is the call graph for this function:



13.17 u14.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
```

Include dependency graph for u14.c:



Functions

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

13.17.1 Function Documentation

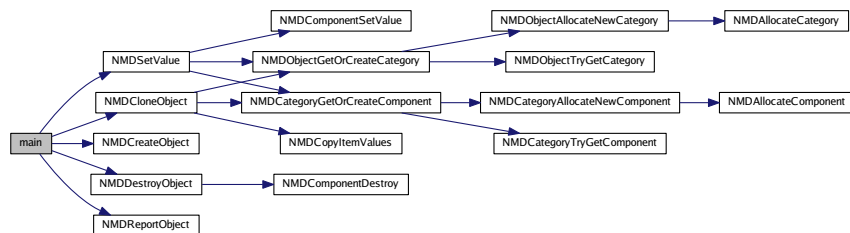
13.17.1.1 `int main (int argc, char ** argv)`

Object cloning with scalars

Definition at line 6 of file u14.c.

References `NMD_FREE`, `NMDCloneObject()`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDFalse`, `NMDInt`, `NMDReal`, `NMDReportObject()`, and `NMDSetValue()`.

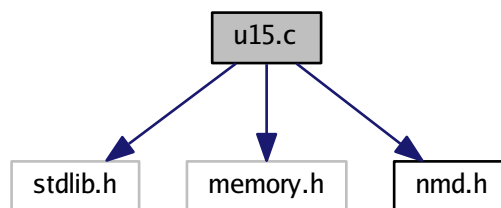
Here is the call graph for this function:



13.18 u15.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
```

Include dependency graph for u15.c:



Defines

- `#define ILEN 4`
- `#define RLEN 6`

Functions

- `int main (int argc, char **argv)`

13.18.1 Define Documentation

13.18.1.1 `#define ILEN 4`

13.18.1.2 `#define RLEN 6`

13.18.2 Function Documentation

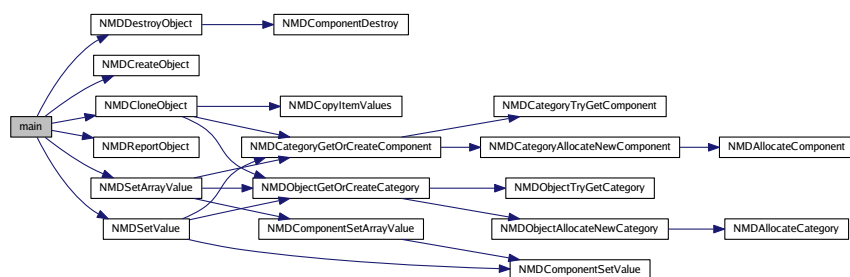
13.18.2.1 `int main (int argc, char ** argv)`

Object cloning including arrays

Definition at line 6 of file u15.c.

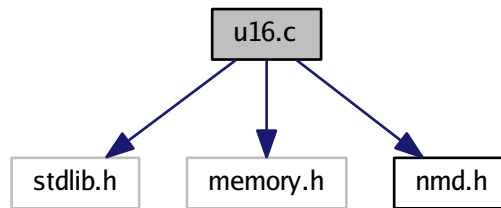
References ILEN, NMD_FREE, NMD_MALLOC, NMDCloneObject(), NMDCreateObject(), NMDDestroyObject(), NMDFalse, NMDInt, NMDIntArray, NMDReal, NMDRealarray, NMDReportObject(), NMDSetArrayValue(), NMDSetValue(), NMDTrue, and RLEN.

Here is the call graph for this function:



13.19 u16.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×  
Include dependency graph for u16.c:
```



Functions

- int `main` (int argc, char **argv)

13.19.1 Function Documentation

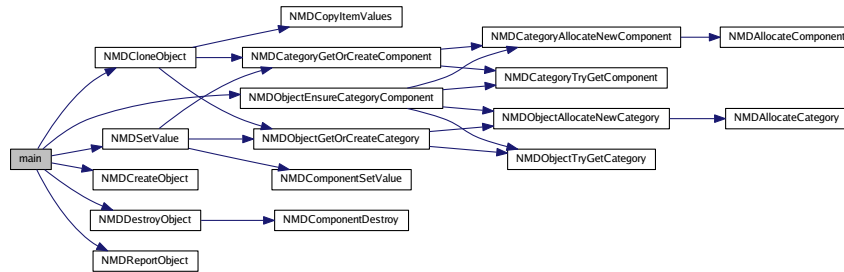
13.19.1.1 int main (int argc, char ** argv)

Object cloning and reporting in the presence of place holder empty components

Definition at line 7 of file u16.c.

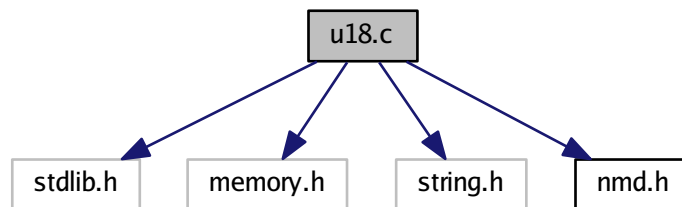
References NMD_FREE, NMDCloneObject(), NMDCreateObject(), NMDDestroyObject(), NMDFalse, NMDInt, NMDOBJECTEnsureCategoryComponent(), NMDReal, NMDReportObject(), and NMDSetValue().

Here is the call graph for this function:



13.20 u18.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "string.-
h" #include "nmd.h" Include dependency graph for u18.c:
```



Functions

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

13.20.1 Function Documentation

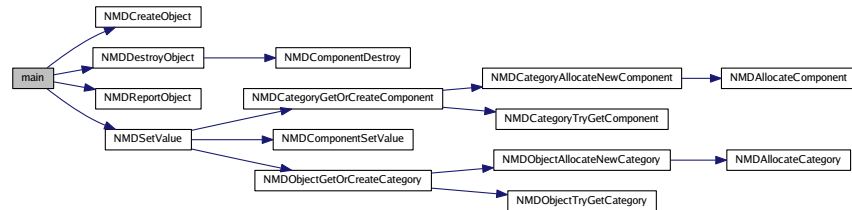
13.20.1.1 int main (int argc, char ** argv)

Object reporting

Definition at line 7 of file u18.c.

References NMD_FREE, NMDCreateObject(), NMDDestroyObject(), NMDFalse, NMDInt, NMDReportObject(), and NMDSetValue().

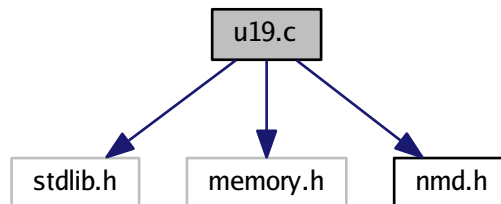
Here is the call graph for this function:



13.21 u19.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
```

Include dependency graph for u19.c:



Functions

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

13.21.1 Function Documentation

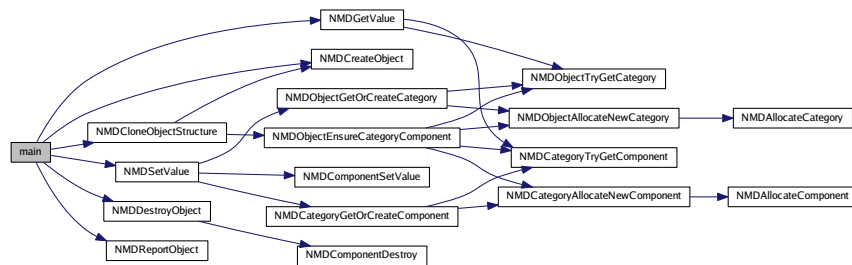
13.21.1.1 `int main (int argc, char ** argv)`

Object structure cloning

Definition at line 6 of file u19.c.

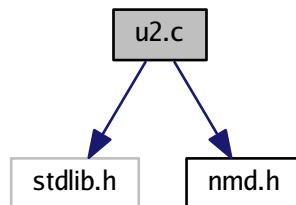
References `NMD_FREE`, `NMDCloneObjectStructure()`, `NMDCreateObject()`, `NMD_DestroyObject()`, `NMDFalse`, `NMDGetValue()`, `NMDInt`, `NMDReportObject()`, and `NMDSetValue()`.

Here is the call graph for this function:



13.22 u2.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for u2.c:



Functions

- int `main` (int argc, char **argv)

13.22.1 Function Documentation

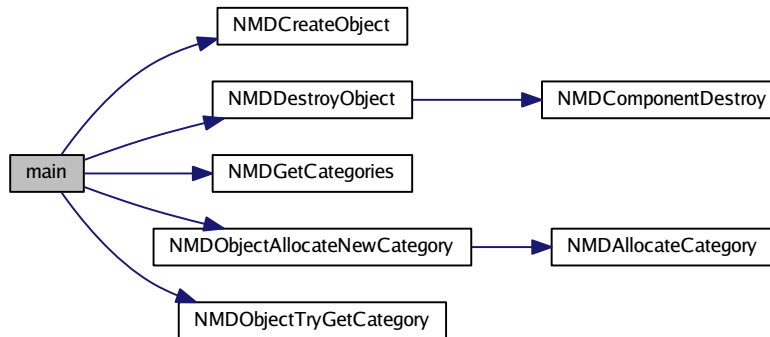
13.22.1.1 int main (int argc, char ** argv)

Test different ways of creating a category

Definition at line 5 of file u2.c.

References `NMD_FREE`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetCategories()`, `NMDOBJECTAllocateNewCategory()`, and `NMDOBJECTTryGetCategory()`.

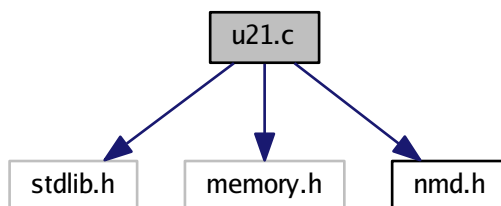
Here is the call graph for this function:



13.23 u21.c File Reference

```
#include <stdlib.h> #include "memory.h" #include "nmd.h" ×
```

Include dependency graph for u21.c:



Defines

- #define [ILEN](#) 4
- #define [RLEN](#) 6

Functions

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

13.23.1 Define Documentation

13.23.1.1 #define ILEN 4

13.23.1.2 #define RLEN 6

13.23.2 Function Documentation

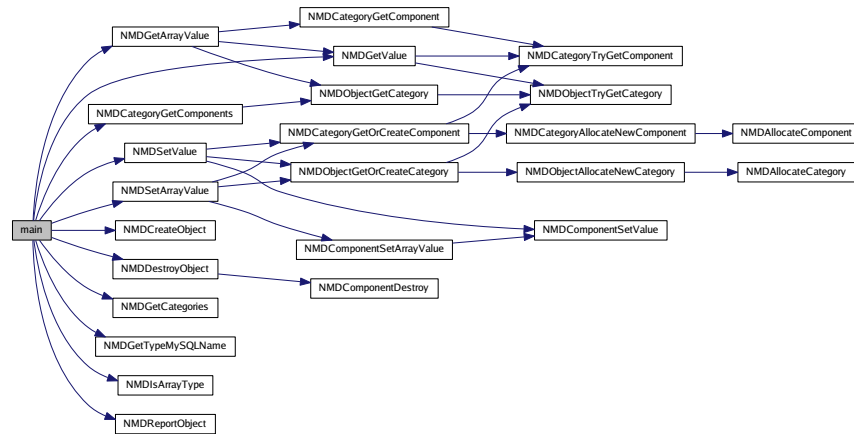
13.23.2.1 int main (int argc, char ** argv)

Object database output

Definition at line 10 of file u21.c.

References [ILEN](#), [NMD_FREE](#), [NMD_MALLOC](#), [NMDCategoryGetComponents\(\)](#), [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDFalse](#), [NMDGetArrayValue\(\)](#), [NMDGetCategories\(\)](#), [NMDGetTypeMySQLName\(\)](#), [NMDGetValue\(\)](#), [NMDInt](#), [NMDIntArray](#), [NMDIsArrayType\(\)](#), [NMDReal](#), [NMDRealarray](#), [NMDReportObject\(\)](#), [NMDSetArrayValue\(\)](#), [NMDSetValue\(\)](#), [NMDDTrue](#), and [RLEN](#).

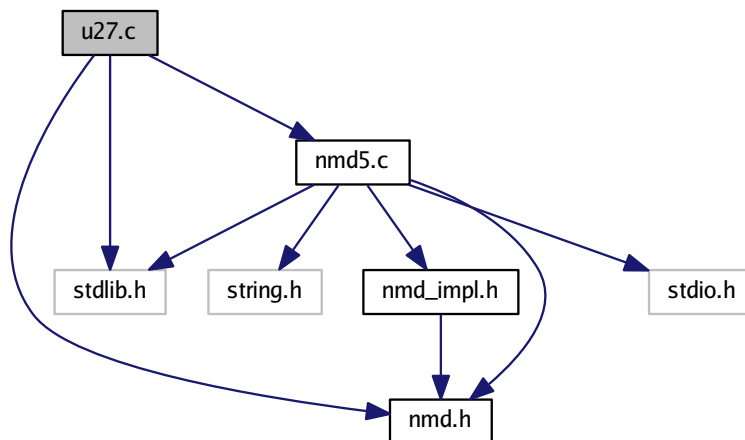
Here is the call graph for this function:



13.24 u27.c File Reference

```
#include <stdlib.h> #include "nmd.h" #include "nmd5.c" Include
```

dependency graph for u27.c:



Functions

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

13.24.1 Function Documentation

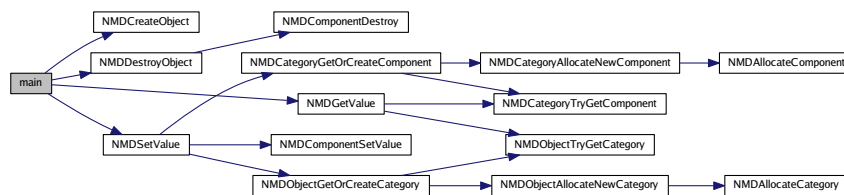
13.24.1.1 int main (int argc, char ** argv)

Test hdf5 dumping of scalar values

Definition at line 6 of file u27.c.

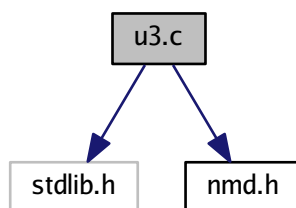
References [NMDCreateObject\(\)](#), [NMDDestroyObject\(\)](#), [NMDGetValue\(\)](#), [NMDInt](#), - [NMDReal](#), and [NMDSetValue\(\)](#).

Here is the call graph for this function:



13.25 u3.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for u3.c:



Functions

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

13.25.1 Function Documentation

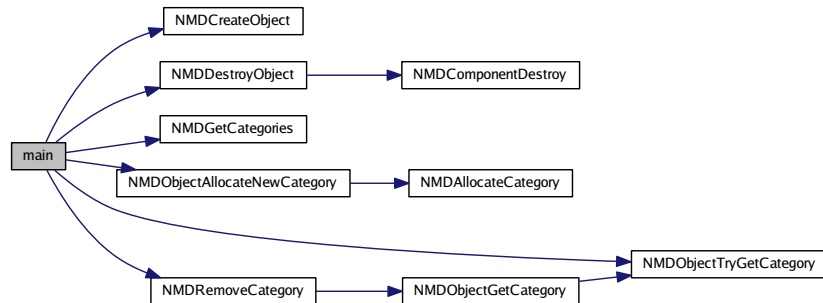
13.25.1.1 int main (int argc, char ** argv)

Test removal of a category

Definition at line 5 of file u3.c.

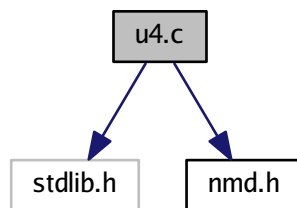
References NMD_FREE, NMDCreateObject(), NMDDestroyObject(), NMDGetCategories(), NMDObjectAllocateNewCategory(), NMDObjectTryGetCategory(), and NMDRemoveCategory().

Here is the call graph for this function:



13.26 u4.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for `u4.c`:



Functions

- `int main (int argc, char **argv)`

13.26.1 Function Documentation

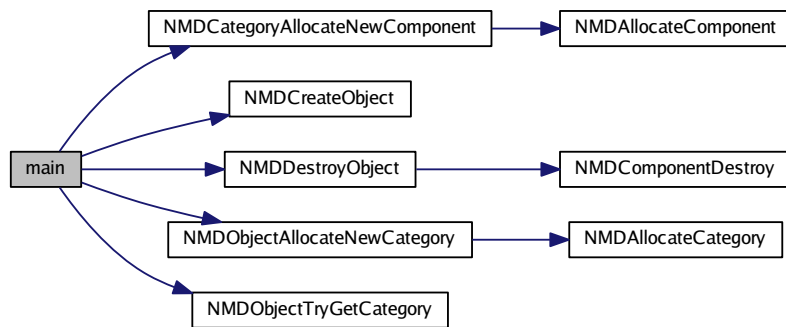
13.26.1.1 `int main (int argc, char ** argv)`

Test creation of components

Definition at line 5 of file u4.c.

References `NMDCategoryAllocateNewComponent()`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDInt`, `NMDIntArray`, `NMDOBJECTAllocateNewCategory()`, `NMDOBJECTTryGetCategory()`, `NMDReal`, `NMDRealarray`, and `NMDString`.

Here is the call graph for this function:



13.27 u5.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for

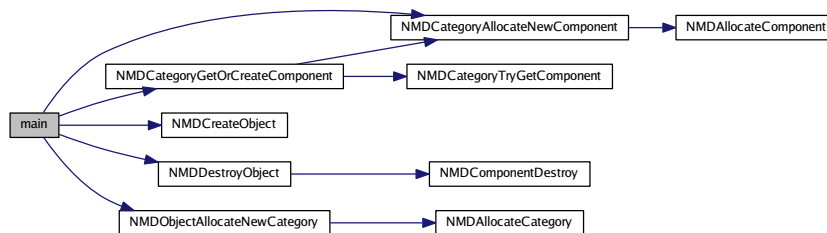

```
graph TD; u5c[u5.c] --> stdlibh[stdlib.h]; u5c --> nmdh[nmd.h]
```

- `int main (int argc, char **argv)`

13.27.1.1 int main (int argc, char ** argv)

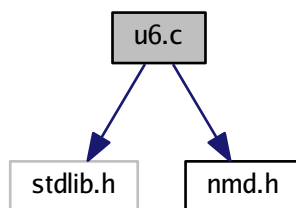
Definition at line 5 of file u5.c.

Here is the call graph for this function:



13.28 u6.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for u6.c:



Functions

- int `main` (int argc, char **argv)

13.28.1 Function Documentation

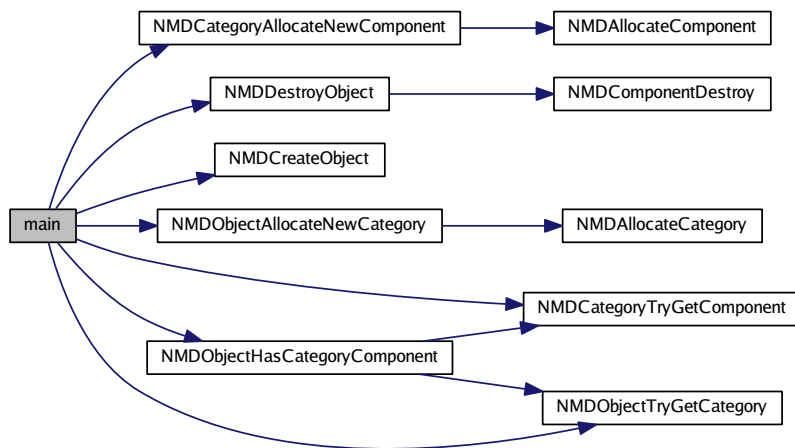
13.28.1.1 int main (int argc, char ** argv)

Test existence tests of components

Definition at line 5 of file u6.c.

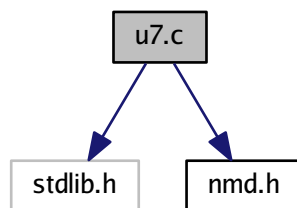
References `NMDCategoryAllocateNewComponent()`, `NMDCategoryTryGetComponent()`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDInt`, `NMDIntArray`, `NMDOBJECTAllocateNewCategory()`, `NMDOBJECTHasCategoryComponent()`, `NMDOBJECTTryGetCategory()`, `NMDReal`, `NMDRealarray`, and `NMDString`.

Here is the call graph for this function:



13.29 u7.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for `u7.c`:



Functions

- int `main` (int argc, char **argv)

13.29.1 Function Documentation

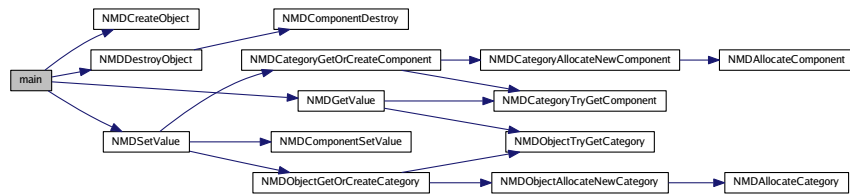
13.29.1.1 int main (int argc, char ** argv)

Test setting and getting of scalar values

Definition at line 5 of file u7.c.

References `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetValue()`, `NMDInt`, and `NMDSetValue()`.

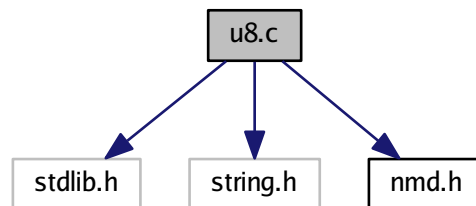
Here is the call graph for this function:



13.30 u8.c File Reference

```
#include <stdlib.h> #include "string.h" #include "nmd.h" ×
```

Include dependency graph for u8.c:



Functions

- `int main (int argc, char **argv)`

13.30.1 Function Documentation

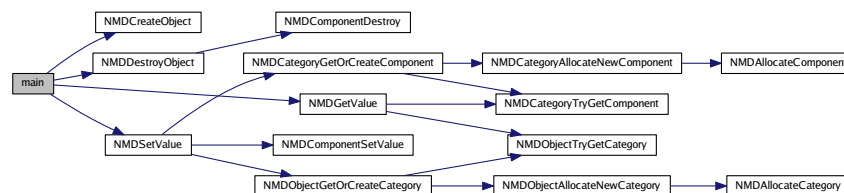
13.30.1.1 `int main (int argc, char ** argv)`

Test setting and getting of array values

Definition at line 6 of file `u8.c`.

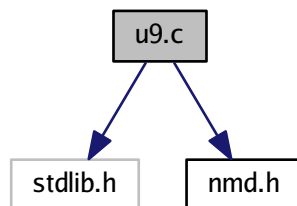
References `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetValue()`, `NMDSetValue()`, and `NMDString`.

Here is the call graph for this function:



13.31 u9.c File Reference

`#include <stdlib.h> #include "nmd.h"` Include dependency graph for u9.c:



Functions

- `int main (int argc, char **argv)`

13.31.1 Function Documentation

13.31.1.1 `int main (int argc, char ** argv)`

Test setting and getting of array values

Definition at line 5 of file u9.c.

References `NMD_FREE`, `NMD_MALLOC`, `NMDCreateObject()`, `NMDDestroyObject()`, `NMDGetArrayValue()`, `NMDIntArray`, `NMDRealarray`, and `NMDSetArrayValue()`.

Here is the call graph for this function:

