ISA LDAP server

Generated by Doxygen 1.9.8

Chapter 1

Teorie

1.1 Použití

./isa-ldapserver {-p <port>} -f <soubor> Význam parametrů a jejich hodnot:

-p <port>: Umožňuje specifikovat konkrétní port, na kterém začne server naslouchat požadavkům klientů. Výchozí hodnota čísla portu je 389. -f <soubor>: Cesta k textovému soubor ve formátu CSV.

1.1.1 Sestavení

Sestavení probíhá pomocí příkazu `make`. Výsledkem je spustitelný soubor `isa-ldapserver`.

1.2 Popis funkce aplikace

Hlavní smyčka aplikace je v souboru server.cpp, ve kterém se prvně nastaví poslouchání na uživatelem zadaném portu. Následně aplikace čeká dokud nepříjde požadavek od ldap klienta. Následně dojde k forku, a v dceřiném procesu se zpracuje požadavek klienta.

Na požadavek podprocese reaguje dle jeho typu. Podporuje 3 druhy požadavků od klienta. Tyto požadavky jsou:

- BIND_REQUEST
- SEARCH_REQUEST
- UNBIND_REQUEST

1.2.1 BIND_REQUEST

Bind request může přijít kdykoli a nemusí být prvním požadavkem. Zde se ověřuje jestli client žádá o správný typ přihlášení. Aplikace podporuje pouze simple, pokud klient zažádá o jinou, je mu vrácena chyba, a komunikace ukončena. Pokud je vše v pořádku, je klientovi vrácen úspěšný BIND_RESPONSE.

1.2.2 SEARCH_REQUEST

Search request může přijít kdykoli a není třeba aby navazoval na bindrequest. Podproces následně zpracuje tento požadavek, vyhledá v databázi odpovídající záznamy a pomocí odpovědi SearchResultEntry je vrátí klientovi. Nakonec pošle SearchResultDone, kterým oznámý klientovi, že je vyhledávání dokončeno.

1.2.3 UNBIND_REQUEST

Jakmile příjde tento požadavek, podproces ukončí komunikaci s klientem a ukončí se.

1.3 Rozšíření oproti zadání

1.3.1 Výběr atributů

Aplikace podporuje možnost vybrat si jaké atributy chce uživatel vyhledat. V případě, že uživatel nevybere žádný atribut, jsou mu vráceny všechny atributy. Neexistující attributy jsou ignorovány.

1.3.2 Notice of Disconnection

Pokud dojde k chybě, která nejde oznámit přes odpovídající odpověd na požadavek, je klientovi oznámeno ukončení komunikace pomocí Notice of Disconnection.

1.4 Zdrojový kód

Program byl psát v jazyku CPP s objektovým přístupem. Zdrojový kód byl dokumentován pomocí Doxygenu. Výsledná dokumentace je k dispozici v adresáři docs v souboru refman.pdf.

Zde je pouze stručný popis zajímavých tříd, jejich významu a použití.

1.4.1 Třídy

1.4.1.1 BerObject

Je bázovým objektem pro objekty reprezentující BER struktury. Obsahuje základní metody pro ostatní objekty.

Pomocí funkce `getBerRepresentation` lze získat reprezentaci objektu v podobě BER bytového pole. Které lze snadno odeslat clientovy. Pro deserializaci je třeba využít funkce `ParseBerObject` která vrací ukazatel na nově vytvořený objekt.

Z těchto objektů lze snadno vytvářet BER struktury a pracovat s nimy.

1.4.1.2 FilterObject

Je bázovým objektem pro objekty reprezentující LDAP filtry. Jeho podtřídy jsou obsahují metody pro snadnou práci s nimy.

1.4.1.3 DatabaseController

Třída DatabaseController slouží pro práci s csv databází. Obsahuje metody pro načítání řádků z databáze a vrací je v podobě objektů třídy DatabaseObject.

1.4.1.4 DatabaseObject

Třída DatabaseObject slouží pro reprezentaci řádku v databázi. Obsahuje metody pro získání hodnot atributů.

1.5 Testování

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

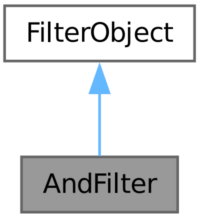
Here is a list of all documented files with brief descriptions:

Chapter 5

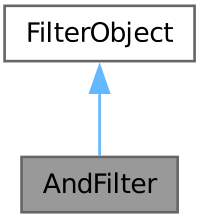
Class Documentation

5.1 AndFilter Class Reference

Inheritance diagram for AndFilter:



Collaboration diagram for AndFilter:



Public Member Functions

- filterTypes getFilterType ()

Public Attributes

- std::vector< FilterObject ∗ > **filters**

5.1.1 Member Function Documentation

5.1.1.1 getFilterType()

```
filterTypes AndFilter::getFilterType ( )  [virtual]
```

Reimplemented from FilterObject.

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

- inc/AndFilterObject.h
- src/AndFilterObject.cpp

5.2 args_t Struct Reference

Public Attributes

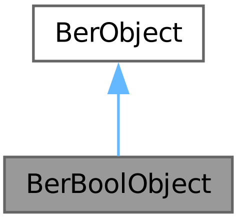
- char ∗ **dbPath**
- int **port**
- bool **err**

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

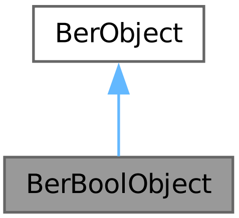
- inc/argument_helper_functions.h

5.3 BerBoolObject Class Reference

Inheritance diagram for BerBoolObject:



Collaboration diagram for BerBoolObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

  Get type of BerObject.
- long long int getLenght ()

  Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

  Returns the BER representation of BerObject.
- **BerBoolObject** (char value)

5.3.1 Member Function Documentation

5.3.1.1 getBerObjectType()

```
berObjectTypes BerBoolObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.3.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerBoolObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.3.1.3 getLenght()

```
long long int BerBoolObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

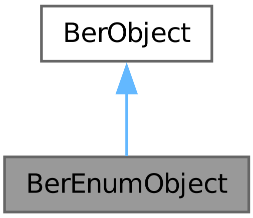
Reimplemented from BerObject.

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

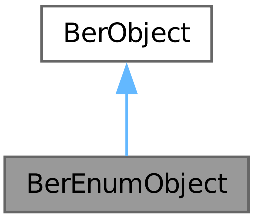
- inc/BerBoolObject.h
- src/BerBoolObject.cpp

5.4 BerEnumObject Class Reference

Inheritance diagram for BerEnumObject:



Collaboration diagram for BerEnumObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

    Get type of BerObject.
- long long int getLenght ()

    Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

    Returns the BER representation of BerObject.
- **BerEnumObject** (char value)

5.4.1 Member Function Documentation

5.4.1.1 getBerObjectType()

```
berObjectTypes BerEnumObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.4.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerEnumObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.4.1.3 getLenght()

```
long long int BerEnumObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

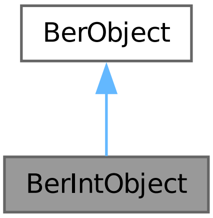
Reimplemented from BerObject.

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

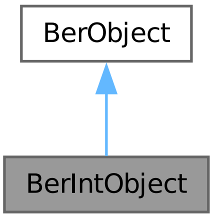
- inc/BerEnumObject.h
- src/BerEnumObject.cpp

5.5 BerIntObject Class Reference

Inheritance diagram for BerIntObject:



Collaboration diagram for BerIntObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

  Get type of BerObject.
- int **getValue** ()
- void **setValue** (int value)
- long long int getLenght ()

  Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

  Returns the BER representation of BerObject.
- **BerIntObject** (int value)

5.5.1 Member Function Documentation

5.5.1.1 getBerObjectType()

```
berObjectTypes BerIntObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.5.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerIntObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.5.1.3 getLenght()

```
long long int BerIntObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

Reimplemented from BerObject.

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

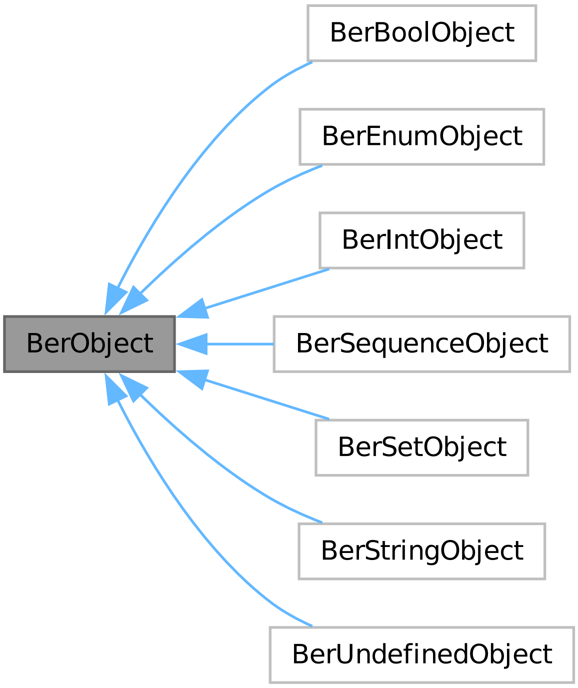
- inc/BerIntObject.h
- src/BerIntObject.cpp

5.6 BerObject Class Reference

Base class for all BER objects.

```
#include <BerObject.h>
```

Inheritance diagram for BerObject:



Public Member Functions

- virtual berObjectTypes getBerObjectType ()

  Get type of BerObject.
- virtual long long int getLenght ()

  Get the Lenght of BerObject representation in BER (including tag and lenght)
- virtual std::vector< unsigned char > getBerRepresentation ()

  Returns the BER representation of BerObject.

5.6.1 Detailed Description

Base class for all BER objects.

5.6.2 Member Function Documentation

5.6.2.1 getBerObjectType()

```
berObjectTypes BerObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented in BerBoolObject, BerEnumObject, BerIntObject, BerSequenceObject, BerSetObject, BerStringObject, and BerUndefinedObject.

5.6.2.2 getBerRepresentation()

```
std::vector< unsigned char > BerObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented in BerBoolObject, BerEnumObject, BerIntObject, BerSequenceObject, BerSetObject, BerStringObject, and BerUndefinedObject.

5.6.2.3 getLenght()

```
long long int BerObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

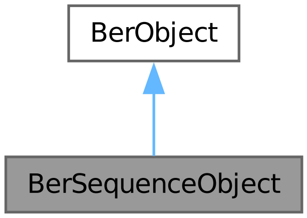
Reimplemented in BerBoolObject, BerEnumObject, BerIntObject, BerSequenceObject, BerSetObject, BerStringObject, and BerUndefinedObject.

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

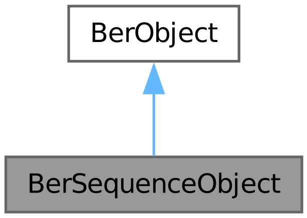
- inc/BerObject.h
- src/BerObject.cpp

5.7 BerSequenceObject Class Reference

Inheritance diagram for BerSequenceObject:



Collaboration diagram for BerSequenceObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

    Get type of BerObject.
- long long int getLenght ()

    Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

    Returns the BER representation of BerObject.
- **BerSequenceObject** (int tag)
- int **GetTag** ()

Public Attributes

- std::vector< BerObject ∗ > **objects**

5.7.1 Member Function Documentation

5.7.1.1 getBerObjectType()

```
berObjectTypes BerSequenceObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.7.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerSequenceObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.7.1.3 getLenght()

```
long long int BerSequenceObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

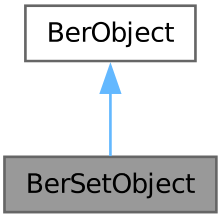
Reimplemented from BerObject.

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

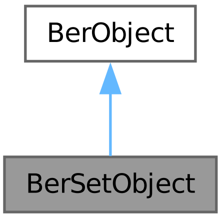
- inc/BerSequenceObject.h
- src/BerSequenceObject.cpp

5.8 BerSetObject Class Reference

Inheritance diagram for BerSetObject:



Collaboration diagram for BerSetObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

  Get type of BerObject.
- long long int getLenght ()

  Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

  Returns the BER representation of BerObject.

Public Attributes

- std::vector< BerObject ∗ > **objects**

5.8.1 Member Function Documentation

5.8.1.1 getBerObjectType()

```
berObjectTypes BerSetObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.8.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerSetObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.8.1.3 getLenght()

```
long long int BerSetObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

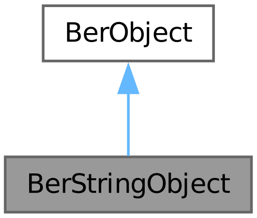
Reimplemented from BerObject.

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

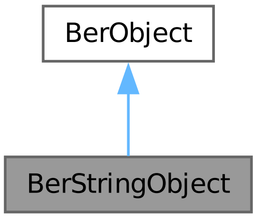
- inc/BerSetObject.h
- src/BerSetObject.cpp

5.9 BerStringObject Class Reference

Inheritance diagram for BerStringObject:



Collaboration diagram for BerStringObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

    Get type of BerObject.
- long long int getLenght ()

    Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

    Returns the BER representation of BerObject.
- **BerStringObject** (std::vector< unsigned char > value)
- **BerStringObject** (std::string value)

Public Attributes

- std::vector< unsigned char > **value**

5.9.1 Member Function Documentation

5.9.1.1 getBerObjectType()

```
berObjectTypes BerStringObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.9.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerStringObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.9.1.3 getLenght()

```
long long int BerStringObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

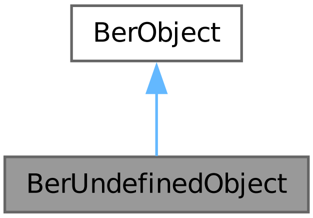
Reimplemented from BerObject.

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

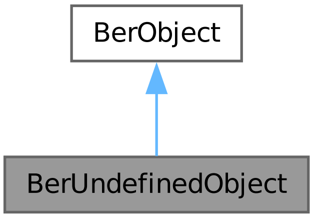
- inc/BerStringObject.h
- src/BerStringObject.cpp

5.10 BerUndefinedObject Class Reference

Inheritance diagram for BerUndefinedObject:



Collaboration diagram for BerUndefinedObject:



Public Member Functions

- berObjectTypes getBerObjectType ()

  Get type of BerObject.
- long long int getLenght ()

  Get the Lenght of BerObject representation in BER (including tag and lenght)
- std::vector< unsigned char > getBerRepresentation ()

  Returns the BER representation of BerObject.
- **BerUndefinedObject** (std::vector< unsigned char > value)

5.10.1 Member Function Documentation

5.10.1.1 getBerObjectType()

```
berObjectTypes BerUndefinedObject::getBerObjectType ( )  [virtual]
```

Get type of BerObject.

Returns

berObjectTypes

Reimplemented from BerObject.

5.10.1.2 getBerRepresentation()

```
std::vector< unsigned char > BerUndefinedObject::getBerRepresentation ( )  [virtual]
```

Returns the BER representation of BerObject.

Returns

std::vector<unsigned char>

Reimplemented from BerObject.

5.10.1.3 getLenght()

```
long long int BerUndefinedObject::getLenght ( )  [virtual]
```

Get the Lenght of BerObject representation in BER (including tag and lenght)

Returns

long long int

Reimplemented from BerObject.

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

- inc/BerUndefinedObject.h
- src/BerUndefinedObject.cpp

5.11 DatabaseController Class Reference

class for loading and parsing database file

```
#include <DatabaseController.h>
```

Public Member Functions

- DatabaseObject loadNextRow (int ∗err)

  loads next row from database file
- std::vector< DatabaseObject > loadAllRows ()

  loads all rows from database file
- DatabaseController (std::string fileName)

  Construct a new Database Controller object.

5.11.1 Detailed Description

class for loading and parsing database file

5.11.2 Constructor & Destructor Documentation

5.11.2.1 DatabaseController()

```
DatabaseController::DatabaseController (
            std::string fileName )
```

Construct a new Database Controller object.

Parameters

fileName	path to database csv file

5.11.3 Member Function Documentation

5.11.3.1 loadAllRows()

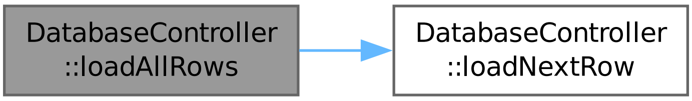
```
std::vector< DatabaseObject > DatabaseController::loadAllRows ( )
```

loads all rows from database file

Returns

std::vector<DatabaseObject>

Here is the call graph for this function:


DatabaseController ::loadAllRows
DatabaseController ::loadNextRow

5.11.3.2 loadNextRow()

```
DatabaseObject DatabaseController::loadNextRow (
            int * err )
```

loads next row from database file

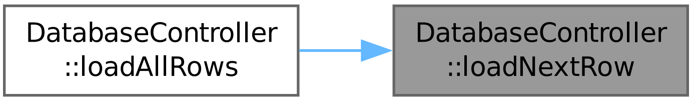
Parameters

err	1 if EOF, 0 if success

Returns

DatabaseObject

Here is the caller graph for this function:


DatabaseController ::loadAllRows
DatabaseController ::loadNextRow

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

- inc/DatabaseController.h
- src/DatabaseController.cpp

5.12 DatabaseObject Class Reference

Object representing one row from database.

```
#include <DatabaseObject.h>
```

Public Member Functions

- std::vector< unsigned char > **get_name** ()
- std::vector< unsigned char > **get_uid** ()
- std::vector< unsigned char > **get_email** ()
- **DatabaseObject** (std::vector< unsigned char > name, std::vector< unsigned char > uid, std::vector< unsigned char > email)

5.12.1 Detailed Description

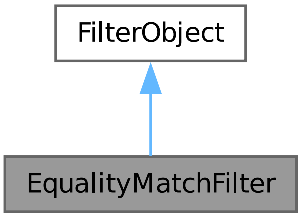
Object representing one row from database.

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

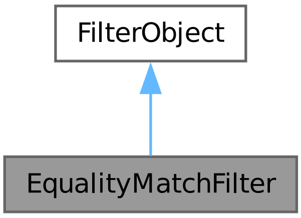
- inc/DatabaseObject.h
- src/DatabaseObject.cpp

5.13 EqualityMatchFilter Class Reference

Inheritance diagram for EqualityMatchFilter:



Collaboration diagram for EqualityMatchFilter:



Public Member Functions

- **EqualityMatchFilter** (std::vector< unsigned char > attributeDescription, std::vector< unsigned char > assertionValue)
- std::vector< unsigned char > **getAttributeDescription** ()
- std::vector< unsigned char > **getAssertionValue** ()
- filterTypes getFilterType ()

5.13.1 Member Function Documentation

5.13.1.1 getFilterType()

```
filterTypes EqualityMatchFilter::getFilterType ( )  [virtual]
```

Reimplemented from FilterObject.

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

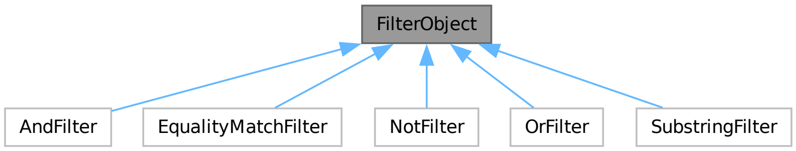
- inc/EqualityMatchFilterObject.h
- src/EqualityMatchFilterObject.cpp

5.14 FilterObject Class Reference

base class for all filter objects

```
#include <FilterObject.h>
```

Inheritance diagram for FilterObject:



Public Member Functions

- virtual filterTypes **getFilterType** ()

5.14.1 Detailed Description

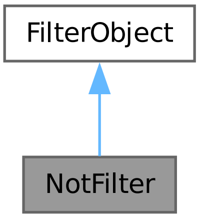
base class for all filter objects

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

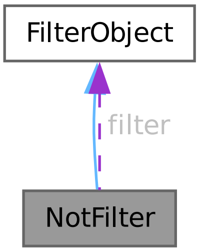
- inc/FilterObject.h
- src/FilterObject.cpp

5.15 NotFilter Class Reference

Inheritance diagram for NotFilter:



Collaboration diagram for NotFilter:



Public Member Functions

- filterTypes getFilterType ()

Public Attributes

- FilterObject ∗ **filter**

5.15.1 Member Function Documentation

5.15.1.1 getFilterType()

```
filterTypes NotFilter::getFilterType ( )  [virtual]
```

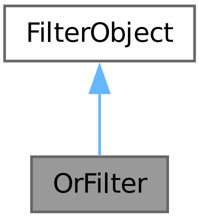
Reimplemented from FilterObject.

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

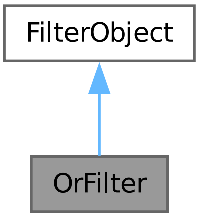
- inc/NotFilterObject.h
- src/NotFilterObject.cpp

5.16 OrFilter Class Reference

Inheritance diagram for OrFilter:



Collaboration diagram for OrFilter:



Public Member Functions

- filterTypes getFilterType ()

Public Attributes

- std::vector< FilterObject * > **filters**

5.16.1 Member Function Documentation

5.16.1.1 getFilterType()

```
filterTypes OrFilter::getFilterType ( )  [virtual]
```

Reimplemented from FilterObject.

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

- inc/OrFilterObject.h
- src/OrFilterObject.cpp

5.17 searchedAttributes Struct Reference

Public Attributes

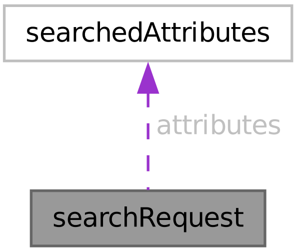
- bool **cn**
- bool **email**
- bool **uid**

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

- inc/ldap_comunication.h
- isa-ldapserver.cpp

5.18 searchRequest Struct Reference

Collaboration diagram for searchRequest:



Public Attributes

- int **messageIDLength**
- unsigned int **sizeLimit**
- searchedAttributesType **attributes**
- char ∗ **messageID**
- int **sizeLimit**

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

- inc/ldap_comunication.h
- isa-ldapserver.cpp

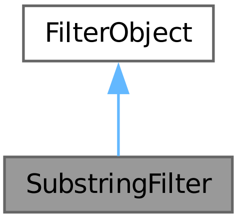
5.19 server Class Reference

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

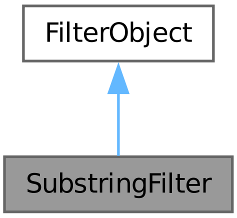
- server.h

5.20 SubstringFilter Class Reference

Inheritance diagram for SubstringFilter:



Collaboration diagram for SubstringFilter:



Public Member Functions

- **SubstringFilter** (std::vector< unsigned char > attributeDescription, std::vector< unsigned char > subInitial, std::vector< std::vector< unsigned char > > subAny, std::vector< unsigned char > subFinal)
- std::vector< unsigned char > **getAttributeDescription** ()
- std::vector< unsigned char > **getSubInitial** ()
- std::vector< std::vector< unsigned char > > **getSubAny** ()
- std::vector< unsigned char > **getSubFinal** ()
- filterTypes getFilterType ()

5.20.1 Member Function Documentation

5.20.1.1 getFilterType()

```
filterTypes SubstringFilter::getFilterType ( )  [virtual]
```

Reimplemented from FilterObject.

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

- inc/SubstringFilterObject.h
- src/SubstringFilterObject.cpp

Chapter 6

File Documentation

6.1 AndFilterObject.h

```
#ifndef ANDFILTEROBJECT_H
#define ANDFILTEROBJECT_H
#include "inc/FilterObject.h"
#include "inc/ber_helper_functions.h"

#include <vector>

class AndFilter : public FilterObject {
public:
  std::vector<FilterObject *> filters;
  filterTypes getFilterType();
  ~AndFilter();
};

#endif
```

6.2 argument_helper_functions.h

```
#ifndef ARGUMENT_HELPER_FUNCTIONS_H
#define ARGUMENT_HELPER_FUNCTIONS_H
#include "string.h"
#include <stdlib.h>
#include <stdio.h>

typedef struct args_t {
  char *dbPath;
  int port;
  bool err;
} argsT;

argsT parseArguments(int argc,const char **argv);

#endif
```

6.3 ber_constants.h

```
#ifndef BER_CONSTANTS_H
#define BER_CONSTANTS_H
const unsigned int BER_TAG_LENGTH = 1;
const unsigned int BER_LENGTH_OF_LENGTH_TAG = 1;
const unsigned int BER_4BYTE_LENGTH_LENGTH = 4;
const unsigned int BER_EXTENDED_RESPONSE_C = 0x78;
const unsigned int BER_BIND_REQUEST_C = 0x60;
const unsigned int BER_BIND_RESPONSE_C = 0x61;
const unsigned int BER_SEARCH_REQUEST_C = 0x63;
const unsigned int BER_SEARCH_RESULT_ENTRY_C = 0x64;
const unsigned int BER_SEARCH_RESULT_DONE_C = 0x65;
const unsigned int BER_UNBIND_REQUEST_C = 0x42;

const unsigned int BER_BOOL_C = 0x01;
const unsigned int BER_INT_C = 0x02;
const unsigned int BER_INT_4BYTES_C = 0x84;
const unsigned int BER_OCTET_STRING_C = 0x04;
const unsigned int BER_ENUM_C = 0x0A;
const unsigned int BER_SEQUENCE_C = 0x30;
const unsigned int BER_SET_C = 0x31;

//constants --- result codes

const unsigned int BER_LDAP_SUCCESS = 0x00;
const unsigned int BER_LDAP_PROTOCOL_ERROR = 0x02;
const unsigned int BER_LDAP_SIZE_LIMIT_EXCEEDED = 0x04;
const unsigned int BER_LDAP_AUTH_METHOD_NOT_SUPPORTED = 0x07;



#endif
```

6.4 ber_helper_functions.h

```
#ifndef BER_HELPER_FUNCTIONS_H
#define BER_HELPER_FUNCTIONS_H
#include <vector>
#include "inc/ber_constants.h"
#include <stdio.h>
#include <stdlib.h>
int ParseINT(unsigned char *s, int *err);

typedef enum filterTypes {
  AND,
  OR,
  NOT,
  equalityMatch,
  substrings,
  undefined,
} filterTypes;

typedef enum berObjectTypes {
  berSequenceObject,
  berIntObject,
  berStringObject,
  berSetObject,
  berEnumObject,
  berBoolObject,
  berUndefined,
  berErr,
} berObjectTypes;

int HowManyBytesWillIntUse(int value);

int WriteIntAppend(std::vector<unsigned char> &s, int value);

void AppendLenght4Bytes(std::vector<unsigned char> &start, int value);

int GetLength(std::vector<unsigned char>::iterator start, int *err,std::vector<unsigned char>::iterator end);

unsigned int ParseINT(std::vector<unsigned char>::iterator s, int *err,std::vector<unsigned char>::iterator end);

int GetLengthOfLength(std::vector<unsigned char>::iterator start, int *err,std::vector<unsigned char>::iterator end);

void SkipTags(std::vector<unsigned char>::iterator &start, int n, int *err, std::vector<unsigned char>::iterator end);

void GoIntoTag(std::vector<unsigned char>::iterator &start, int *err, std::vector<unsigned char>::iterator end);

void IncreaseLength4Bytes(std::vector<unsigned char>::iterator &start, int n,
                          int *err,std::vector<unsigned char>::iterator end);

filterTypes getFilterType(std::vector<unsigned char>::iterator start);

#endif
```

6.5 BerBoolObject.h

```
#ifndef BERBOOLOBJECT_H
```

```
#define BERBOOLOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include <vector>

class BerBoolObject : public BerObject {
private:
  bool value;

public:
  berObjectTypes getBerObjectType();
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerBoolObject(char value);
  ~BerBoolObject();
};

#endif
```

6.6 BerEnumObject.h

```
#ifndef BERENUMOBJECT_H
#define BERENUMOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include <vector>

class BerEnumObject : public BerObject {
private:
  int value;

public:
  berObjectTypes getBerObjectType();
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerEnumObject(char value);
  ~BerEnumObject();
};

#endif
```

6.7 BerIntObject.h

```
#ifndef BERINTOBJECT_H
#define BERINTOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include <vector>

class BerIntObject : public BerObject {
private:
  int value;

public:
  berObjectTypes getBerObjectType();
  int getValue();
  void setValue(int value);
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerIntObject();
  BerIntObject(int value);
  ~BerIntObject();
};

#endif
```

6.8 BerObject.h

```
#ifndef BER_OBJECT_H
#define BER_OBJECT_H
#include "inc/ber_helper_functions.h"

```

```
class BerObject {
public:
  virtual berObjectTypes getBerObjectType();
  virtual long long int getLenght();
  virtual std::vector<unsigned char> getBerRepresentation();

  virtual ~BerObject();
};

#endif
```

6.9 BerParser.h

```
#ifndef BERPARSER_H
#define BERPARSER_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include "inc/BerEnumObject.h"
#include "inc/BerStringObject.h"
#include "inc/BerSetObject.h"
#include "inc/BerSequenceObject.h"
#include "inc/BerSetObject.h"
#include "inc/BerIntObject.h"
#include "inc/BerBoolObject.h"
#include "inc/BerUndefinedObject.h"
#include <vector>

BerObject *ParseBerObject(std::vector<unsigned char>::iterator start,
                          int *err,std::vector<unsigned char>::iterator end);

#endif
```

6.10 BerSequenceObject.h

```
#ifndef BERSEQUENCEOBJECT_H
#define BERSEQUENCEOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"

#include <vector>

class BerSequenceObject : public BerObject {
private:
  int tag;

public:
  std::vector<BerObject *> objects;
  berObjectTypes getBerObjectType();
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerSequenceObject(int tag);
  BerSequenceObject();
  int GetTag();
  ~BerSequenceObject();
};

#endif
```

6.11 BerSetObject.h

```
#ifndef BERSETOBJECT_H
#define BERSETOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"

#include <vector>

class BerSetObject : public BerObject {
public:
  std::vector<BerObject *> objects;
  berObjectTypes getBerObjectType();
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerSetObject();
  ~BerSetObject();
};

#endif
```

6.12 BerStringObject.h

```
#ifndef BERSTRINGOBJECT_H
#define BERSTRINGOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include <string>
#include <vector>

class BerStringObject : public BerObject {
public:
  berObjectTypes getBerObjectType();
  std::vector<unsigned char> value;
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerStringObject();
  BerStringObject(std::vector<unsigned char> value);
BerStringObject(std::string value);
};

#endif
```

6.13 BerUndefinedObject.h

```
#ifndef BERUNDEFINEDOBJECT_H
#define BERUNDEFINEDOBJECT_H
#include "inc/BerObject.h"
#include "inc/ber_helper_functions.h"

class BerUndefinedObject : public BerObject {
private:
  std::vector<unsigned char> value;

public:
  berObjectTypes getBerObjectType();
  long long int getLenght();
  std::vector<unsigned char> getBerRepresentation();
  BerUndefinedObject(std::vector<unsigned char> value);
};

#endif
```

6.14 database_helper_functions.h

```
#ifndef DATABASE_HELPER_FUNCTIONS_H
#define DATABASE_HELPER_FUNCTIONS_H
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
#include "inc/DatabaseObject.h"

std::vector<DatabaseObject> removeDuplicates(std::vector<DatabaseObject> input);

#endif
```

6.15 DatabaseController.h

```
#ifndef DATABASECONTROLLER_H
#define DATABASECONTROLLER_H
#include "inc/DatabaseObject.h"
#include <fstream>
```

```
#include <iostream>
#include <sstream>
#include <string>
#include <vector>

class DatabaseController {
private:
  std::ifstream file;
  std::vector<unsigned char> sanitaze(std::vector<unsigned char> input);

public:
  DatabaseObject loadNextRow(int *err);

  std::vector<DatabaseObject> loadAllRows();

  DatabaseController(std::string fileName);
  ~DatabaseController();
};

#endif
```

6.16 DatabaseObject.h

```
#ifndef DATABASE_OBJECT_H
#define DATABASE_OBJECT_H
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <vector>

class DatabaseObject {
private:
  std::vector<unsigned char> name;
  std::vector<unsigned char> uid;
  std::vector<unsigned char> email;

public:
  std::vector<unsigned char> get_name();
  std::vector<unsigned char> get_uid();
  std::vector<unsigned char> get_email();
  DatabaseObject(std::vector<unsigned char> name,
                 std::vector<unsigned char> uid,
                 std::vector<unsigned char> email);
};


#endif
```

6.17 EqualityMatchFilterObject.h

```
#ifndef EQUALITYMATCHFILTEROBJECT_H
#define EQUALITYMATCHFILTEROBJECT_H
#include "inc/FilterObject.h"
#include "inc/ber_helper_functions.h"

#include <vector>

class EqualityMatchFilter : public FilterObject {
private:
  std::vector<unsigned char> attributeDescription;
  std::vector<unsigned char> assertionValue;

public:
  EqualityMatchFilter(std::vector<unsigned char> attributeDescription,
                      std::vector<unsigned char> assertionValue);
  std::vector<unsigned char> getAttributeDescription();
  std::vector<unsigned char> getAssertionValue();
  filterTypes getFilterType();
};
#endif
```

6.18 filter_helper_functions.h

```
#ifndef INC_FILTER_HELPER_FUNCTIONS_H
#define INC_FILTER_HELPER_FUNCTIONS_H


#include "inc/FilterObject.h"
#include "inc/NotFilterObject.h"
#include "inc/AndFilterObject.h"
#include "inc/OrFilterObject.h"
#include "inc/EqualityMatchFilterObject.h"
#include "inc/SubstringFilterObject.h"
#include "inc/DatabaseObject.h"
#include "inc/DatabaseController.h"
#include "vector"

bool substrFilterHandler(SubstringFilter *sf, int *err,
                         std::vector<unsigned char> attribute);

bool equalityMatchHandler(EqualityMatchFilter *emf, int *err,
                          std::vector<unsigned char> attribute) ;

bool filterLine(FilterObject *f, int *err, DatabaseObject &databaseEntry) ;

std::vector<DatabaseObject>
filterHandler(FilterObject *f, int *err, const char *dbLocation, int sizeLimit);

FilterObject *convertToFilterObject(std::vector<unsigned char>::iterator BERfilter,
std::vector<unsigned char>::iterator end);

#endif
```

6.19 FilterObject.h

```
#ifndef FILTER_OBJECT_H
#define FILTER_OBJECT_H
#include "inc/ber_helper_functions.h"

#include <vector>

class FilterObject {
public:
  virtual filterTypes getFilterType();
  virtual ~FilterObject();
};


#endif
```

6.20 ldap_comunication.h

```
#ifndef LDAP_COMUNICATION_H
#define LDAP_COMUNICATION_H
#include "inc/AndFilterObject.h"
#include "inc/BerBoolObject.h"
#include "inc/BerEnumObject.h"
#include "inc/BerIntObject.h"
#include "inc/BerObject.h"
#include "inc/BerParser.h"
#include "inc/BerSequenceObject.h"
#include "inc/BerSetObject.h"
#include "inc/BerStringObject.h"
#include "inc/BerUndefinedObject.h"
#include "inc/DatabaseController.h"
#include "inc/DatabaseObject.h"
#include "inc/EqualityMatchFilterObject.h"
#include "inc/FilterObject.h"
#include "inc/NotFilterObject.h"
#include "inc/OrFilterObject.h"
#include "inc/SubstringFilterObject.h"
#include "inc/ber_constants.h"
#include "inc/ber_helper_functions.h"
#include "inc/database_helper_functions.h"
#include "inc/filter_helper_functions.h"
#include <algorithm>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <string>
```

```
#include <sys/resource.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <vector>

typedef struct searchedAttributes {
  bool cn;
  bool email;
  bool uid;
} searchedAttributesType;

// enum for attributes (cn, email, uid)
typedef enum { cn, email, uid } atributeDescriptions;

// sequence - envelope
//     int   - message ID
//     application 3 - search request
//     octed string - base object
//     enum - scope
//     enum - derefAliases
//     int - sizeLimit
//     int - timeLimit
//     bool - typesOnly
//     sequence - FilterObject
//     sequence - attributes

typedef struct searchRequest {
  int messageIDLength;
  unsigned int sizeLimit;
  searchedAttributesType attributes;
} searchRequestType;

BerObject *InitSearchResultEntry(BerObject *searchRequest,
                                 std::vector<unsigned char> LDAPDN);

int AddToSearchResultEntry(BerObject *envelope,
                           std::vector<unsigned char> &attributeDescription,
                           std::vector<unsigned char> &attributeValue);
int checkSearchRequest(BerObject *searchRequest);

int sendNoticeOfDisconnection(int comSocket, char errCode);

int searchRequestHandler(BerObject *searchRequest, int comm_socket,
                         const char *dbPath);

BerObject *CreateBindResponse(BerObject *bindRequest, int resultCode);

int loadEnvelope(std::vector<unsigned char> &bindRequest, int comm_socket) ;

int sendSearchResultDone(BerSequenceObject *searchRequest, int comm_socket,
                         unsigned int result_code);
#endif
```

6.21 NotFilterObject.h

```
#ifndef NOTFILTEROBJECT_H
#define NOTFILTEROBJECT_H
#include "inc/FilterObject.h"
#include "inc/ber_helper_functions.h"

#include <vector>

class NotFilter : public FilterObject {
public:
  FilterObject *filter;
  filterTypes getFilterType();
  ~NotFilter();
};

#endif
```

6.22 OrFilterObject.h

```
#ifndef ORFILTEROBJECT_H
#define ORFILTEROBJECT_H
#include "inc/FilterObject.h"
```

```
#include "inc/ber_helper_functions.h"

#include <vector>

class OrFilter : public FilterObject {
public:
  std::vector<FilterObject *> filters;
  filterTypes getFilterType();
  ~OrFilter();
};

#endif
```

6.23 SubstringFilterObject.h

```
#ifndef SUBSTRINGFILTER_H
#define SUBSTRINGFILTER_H
#include "inc/ber_helper_functions.h"
#include "inc/FilterObject.h"
#include <vector>

class SubstringFilter : public FilterObject {

private:
  std::vector<unsigned char> attributeDescription;
  std::vector<unsigned char> subInitial;
  std::vector<std::vector<unsigned char» subAny;
  std::vector<unsigned char> subFinal;

public:
  SubstringFilter(std::vector<unsigned char> attributeDescription,
                  std::vector<unsigned char> subInitial,
                  std::vector<std::vector<unsigned char» subAny,
                  std::vector<unsigned char> subFinal);
  std::vector<unsigned char> getAttributeDescription();
  std::vector<unsigned char> getSubInitial();
  std::vector<std::vector<unsigned char» getSubAny();
  std::vector<unsigned char> getSubFinal();
  filterTypes getFilterType();
};

#endif
```

6.24 server.h

```
#ifndef SERVER_H
#define SERVER_H
#include "inc/BerEnumObject.h"
#include "inc/BerIntObject.h"
#include "inc/BerObject.h"
#include "inc/BerParser.h"
#include "inc/BerSequenceObject.h"
#include "inc/BerSetObject.h"
#include "inc/BerStringObject.h"
#include "inc/DatabaseObject.h"
#include "inc/FilterObject.h"
#include "inc/argument_helper_functions.h"
#include "inc/ldap_comunication.h"
#include <arpa/inet.h>
#include <fcntl.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/resource.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

// Macro for printing err message and closing socket when err != 0
#define CHECK_ERR(err, msg)                                                    \
  if (err != 0) {                                                              \
    printf("%s\n", msg);                                                       \
    close(childSocket);                                                        \
    exit(0);                                                                   \
  }

```

```
int ldapServer(int port, char *dbPath);

#endif
```

6.25 server.h

```
#include <iostream>

class server
{

};

```

Index