

## ISA LDAP server

Generated by Doxygen 1.9.8



<b>1 Teorie</b>	<b>1</b>
1.1 Použití	1
1.1.1 Sestavení	1
1.2 Popis funkce aplikace	1
1.2.1 BIND_REQUEST	1
1.2.2 SEARCH_REQUEST	2
1.2.3 UNBIND_REQUEST	2
1.3 Rozšíření oproti zadání	2
1.3.1 Výběr atributů	2
1.3.2 Notice of Disconnection	2
1.4 Zdrojový kód	2
1.4.1 Třídy	2
1.4.1.1 BerObject	2
1.4.1.2 FilterObject	2
1.4.1.3 DatabaseController	3
1.4.1.4 DatabaseObject	3
1.5 Testování	3
<b>2 Hierarchical Index</b>	<b>5</b>
2.1 Class Hierarchy	5
<b>3 Class Index</b>	<b>7</b>
3.1 Class List	7
<b>4 File Index</b>	<b>9</b>
4.1 File List	9
<b>5 Class Documentation</b>	<b>11</b>
5.1 AndFilter Class Reference	11
5.1.1 Member Function Documentation	12
5.1.1.1 getFilterType()	12
5.2 args_t Struct Reference	12
5.3 BerBoolObject Class Reference	12
5.3.1 Member Function Documentation	13
5.3.1.1 getBerObjectType()	13
5.3.1.2 getBerRepresentation()	13
5.3.1.3 getLenght()	14
5.4 BerEnumObject Class Reference	14
5.4.1 Member Function Documentation	15
5.4.1.1 getBerObjectType()	15
5.4.1.2 getBerRepresentation()	15
5.4.1.3 getLenght()	15
5.5 BerIntObject Class Reference	16
5.5.1 Member Function Documentation	17

5.5.1.1	<a href="#">getBerObjectType()</a>	17
5.5.1.2	<a href="#">getBerRepresentation()</a>	17
5.5.1.3	<a href="#">getLenght()</a>	17
5.6	<a href="#">BerObject Class Reference</a>	18
5.6.1	<a href="#">Detailed Description</a>	18
5.6.2	<a href="#">Member Function Documentation</a>	19
5.6.2.1	<a href="#">getBerObjectType()</a>	19
5.6.2.2	<a href="#">getBerRepresentation()</a>	19
5.6.2.3	<a href="#">getLenght()</a>	19
5.7	<a href="#">BerSequenceObject Class Reference</a>	20
5.7.1	<a href="#">Member Function Documentation</a>	21
5.7.1.1	<a href="#">getBerObjectType()</a>	21
5.7.1.2	<a href="#">getBerRepresentation()</a>	21
5.7.1.3	<a href="#">getLenght()</a>	21
5.8	<a href="#">BerSetObject Class Reference</a>	22
5.8.1	<a href="#">Member Function Documentation</a>	23
5.8.1.1	<a href="#">getBerObjectType()</a>	23
5.8.1.2	<a href="#">getBerRepresentation()</a>	23
5.8.1.3	<a href="#">getLenght()</a>	23
5.9	<a href="#">BerStringObject Class Reference</a>	24
5.9.1	<a href="#">Member Function Documentation</a>	25
5.9.1.1	<a href="#">getBerObjectType()</a>	25
5.9.1.2	<a href="#">getBerRepresentation()</a>	25
5.9.1.3	<a href="#">getLenght()</a>	25
5.10	<a href="#">BerUndefinedObject Class Reference</a>	26
5.10.1	<a href="#">Member Function Documentation</a>	26
5.10.1.1	<a href="#">getBerObjectType()</a>	26
5.10.1.2	<a href="#">getBerRepresentation()</a>	27
5.10.1.3	<a href="#">getLenght()</a>	27
5.11	<a href="#">DatabaseController Class Reference</a>	27
5.11.1	<a href="#">Detailed Description</a>	28
5.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	28
5.11.2.1	<a href="#">DatabaseController()</a>	28
5.11.3	<a href="#">Member Function Documentation</a>	28
5.11.3.1	<a href="#">loadAllRows()</a>	28
5.11.3.2	<a href="#">loadNextRow()</a>	28
5.12	<a href="#">DatabaseObject Class Reference</a>	29
5.12.1	<a href="#">Detailed Description</a>	29
5.13	<a href="#">EqualityMatchFilter Class Reference</a>	30
5.13.1	<a href="#">Member Function Documentation</a>	30
5.13.1.1	<a href="#">getFilterType()</a>	30
5.14	<a href="#">FilterObject Class Reference</a>	31

5.14.1 Detailed Description	31
5.15 NotFilter Class Reference	31
5.15.1 Member Function Documentation	32
5.15.1.1 getFilterType()	32
5.16 OrFilter Class Reference	33
5.16.1 Member Function Documentation	33
5.16.1.1 getFilterType()	33
5.17 searchedAttributes Struct Reference	34
5.18 searchRequest Struct Reference	34
5.19 server Class Reference	35
5.20 SubstringFilter Class Reference	35
5.20.1 Member Function Documentation	36
5.20.1.1 getFilterType()	36
<b>6 File Documentation</b>	<b>37</b>
6.1 AndFilterObject.h	37
6.2 argument_helper_functions.h	37
6.3 ber_constants.h	37
6.4 ber_helper_functions.h	38
6.5 BerBoolObject.h	38
6.6 BerEnumObject.h	39
6.7 BerIntObject.h	39
6.8 BerObject.h	39
6.9 BerParser.h	40
6.10 BerSequenceObject.h	40
6.11 BerSetObject.h	40
6.12 BerStringObject.h	41
6.13 BerUndefinedObject.h	41
6.14 database_helper_functions.h	41
6.15 DatabaseController.h	41
6.16 DatabaseObject.h	42
6.17 EqualityMatchFilterObject.h	42
6.18 filter_helper_functions.h	42
6.19 FilterObject.h	43
6.20 ldap_comunication.h	43
6.21 NotFilterObject.h	44
6.22 OrFilterObject.h	44
6.23 SubstringFilterObject.h	45
6.24 server.h	45
6.25 server.h	46
<b>Index</b>	<b>47</b>



# 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 souboru 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 podprocesu 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 oznámí klientovi, že je vyhledávání dokončeno.

### 1.2.3 UNBIND\_REQUEST

Jakmile přijde 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 atributy jsou ignorovány.

### 1.3.2 Notice of Disconnection

Pokud dojde k chybě, která nejde oznámit přes odpovídající odpověď 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 Doxygen. 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 базovým 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 nimi.

#### 1.4.1.2 FilterObject

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



#### 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:

args_t . . . . .	12
BerObject . . . . .	18
BerBoolObject . . . . .	12
BerEnumObject . . . . .	14
BerIntObject . . . . .	16
BerSequenceObject . . . . .	20
BerSetObject . . . . .	22
BerStringObject . . . . .	24
BerUndefinedObject . . . . .	26
DatabaseController . . . . .	27
DatabaseObject . . . . .	29
FilterObject . . . . .	31
AndFilter . . . . .	11
EqualityMatchFilter . . . . .	30
NotFilter . . . . .	31
OrFilter . . . . .	33
SubstringFilter . . . . .	35
searchedAttributes . . . . .	34
searchRequest . . . . .	34
server . . . . .	35



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">AndFilter</a>	11
<a href="#">args_t</a>	12
<a href="#">BerBoolObject</a>	12
<a href="#">BerEnumObject</a>	14
<a href="#">BerIntObject</a>	16
<a href="#">BerObject</a>	
Base class for all BER objects	18
<a href="#">BerSequenceObject</a>	20
<a href="#">BerSetObject</a>	22
<a href="#">BerStringObject</a>	24
<a href="#">BerUndefinedObject</a>	26
<a href="#">DatabaseController</a>	
Class for loading and parsing database file	27
<a href="#">DatabaseObject</a>	
Object representing one row from database	29
<a href="#">EqualityMatchFilter</a>	30
<a href="#">FilterObject</a>	
Base class for all filter objects	31
<a href="#">NotFilter</a>	31
<a href="#">OrFilter</a>	33
<a href="#">searchedAttributes</a>	34
<a href="#">searchRequest</a>	34
<a href="#">server</a>	35
<a href="#">SubstringFilter</a>	35



# Chapter 4

## File Index

### 4.1 File List

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

<a href="#">server.h</a>	46
<a href="#">inc/AndFilterObject.h</a>	37
<a href="#">inc/argument_helper_functions.h</a>	37
<a href="#">inc/ber_constants.h</a>	37
<a href="#">inc/ber_helper_functions.h</a>	38
<a href="#">inc/BerBoolObject.h</a>	38
<a href="#">inc/BerEnumObject.h</a>	39
<a href="#">inc/BerIntObject.h</a>	39
<a href="#">inc/BerObject.h</a>	39
<a href="#">inc/BerParser.h</a>	40
<a href="#">inc/BerSequenceObject.h</a>	40
<a href="#">inc/BerSetObject.h</a>	40
<a href="#">inc/BerStringObject.h</a>	41
<a href="#">inc/BerUndefinedObject.h</a>	41
<a href="#">inc/database_helper_functions.h</a>	41
<a href="#">inc/DatabaseController.h</a>	41
<a href="#">inc/DatabaseObject.h</a>	42
<a href="#">inc/EqualityMatchFilterObject.h</a>	42
<a href="#">inc/filter_helper_functions.h</a>	42
<a href="#">inc/FilterObject.h</a>	43
<a href="#">inc/ldap_communication.h</a>	43
<a href="#">inc/NotFilterObject.h</a>	44
<a href="#">inc/OrFilterObject.h</a>	44
<a href="#">inc/server.h</a>	45
<a href="#">inc/SubstringFilterObject.h</a>	45



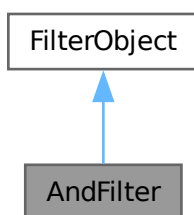


## Chapter 5

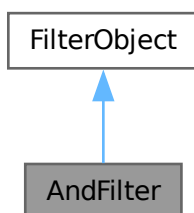
# Class Documentation

### 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

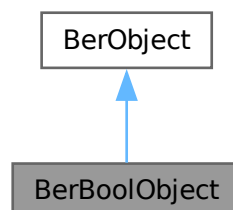
- `char * dbPath`
- `int port`
- `bool err`

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

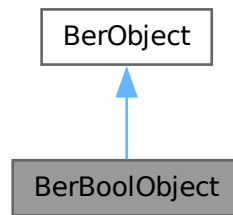
- `inc/argument_helper_functions.h`

## 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

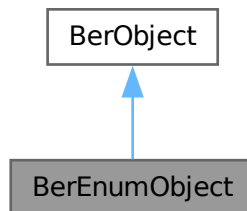
Reimplemented from [BerObject](#).

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

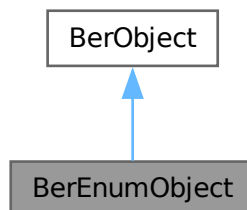
- inc/BerBoolObject.h
- src/BerBoolObject.cpp

## 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

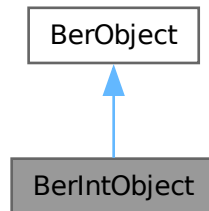
Reimplemented from [BerObject](#).

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

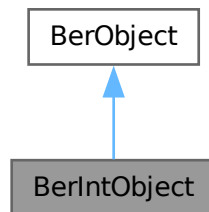
- inc/BerEnumObject.h
- src/BerEnumObject.cpp

## 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:

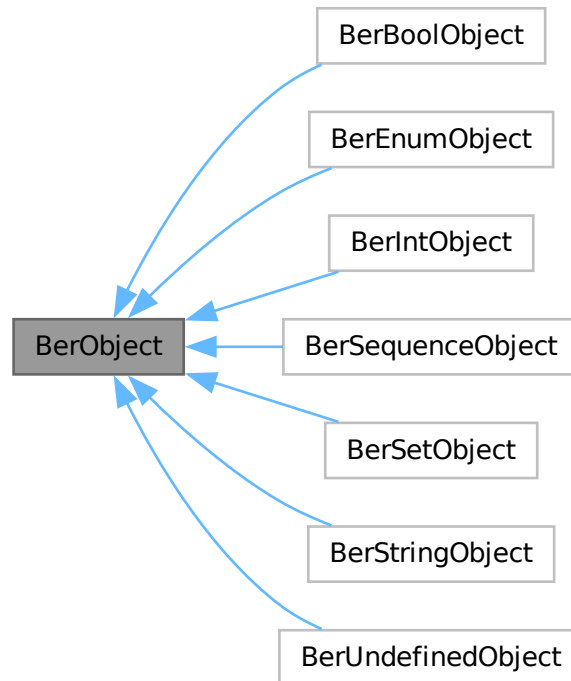
- inc/BerIntObject.h
- src/BerIntObject.cpp

## 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

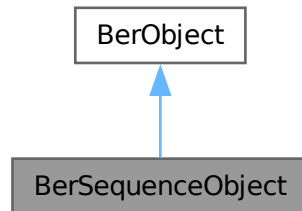
Reimplemented in [BerBoolObject](#), [BerEnumObject](#), [BerIntObject](#), [BerSequenceObject](#), [BerSetObject](#), [BerStringObject](#), and [BerUndefinedObject](#).

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

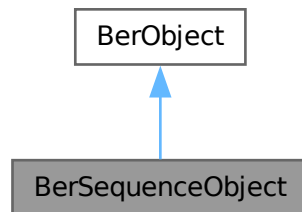
- inc/BerObject.h
- src/BerObject.cpp

## 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

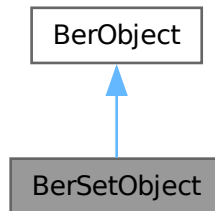
Reimplemented from [BerObject](#).

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

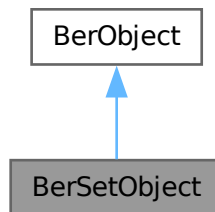
- inc/BerSequenceObject.h
- src/BerSequenceObject.cpp

## 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

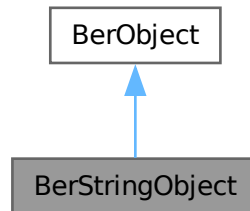
Reimplemented from [BerObject](#).

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

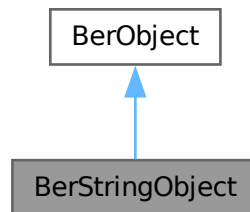
- inc/BerSetObject.h
- src/BerSetObject.cpp

## 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

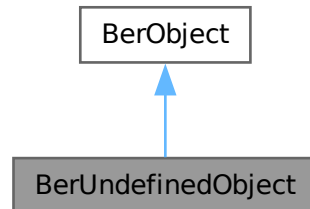
Reimplemented from [BerObject](#).

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

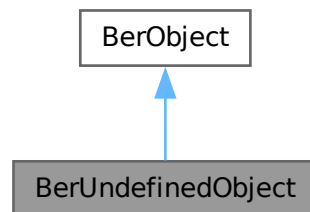
- inc/BerStringObject.h
- src/BerStringObject.cpp

## 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

<i>fileName</i>	path to database csv file
-----------------	---------------------------

### 5.11.3 Member Function Documentation

#### 5.11.3.1 loadAllRows()

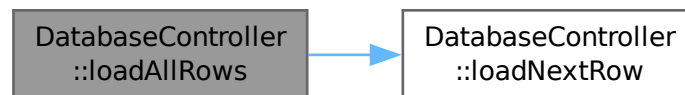
```
std::vector< DatabaseObject > DatabaseController::loadAllRows ( )
```

loads all rows from database file

Returns

```
std::vector<DatabaseObject>
```

Here is the call graph for this function:



#### 5.11.3.2 loadNextRow()

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

loads next row from database file

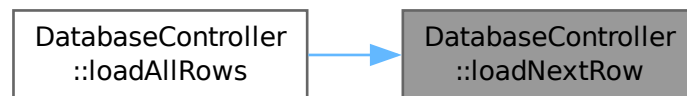
## Parameters

<i>err</i>	1 if EOF, 0 if success
------------	------------------------

## Returns

[DatabaseObject](#)

Here is the caller graph for this function:



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

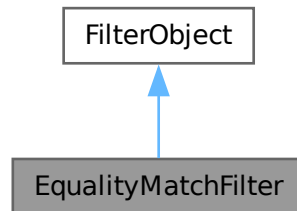
Object representing one row from database.

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

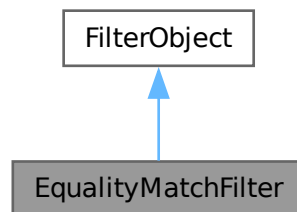
- inc/DatabaseObject.h
- src/DatabaseObject.cpp

## 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:

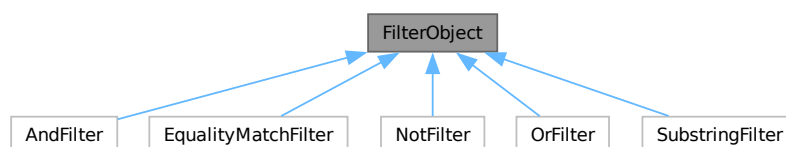
- `inc/EqualityMatchFilterObject.h`
- `src/EqualityMatchFilterObject.cpp`

## 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

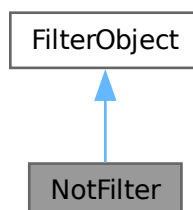
base class for all filter objects

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

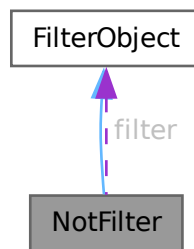
- inc/FilterObject.h
- src/FilterObject.cpp

## 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]
```

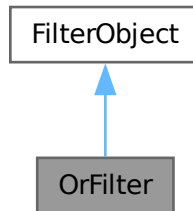
Reimplemented from [FilterObject](#).

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

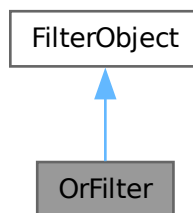
- inc/NotFilterObject.h
- src/NotFilterObject.cpp

## 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

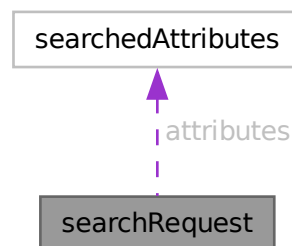
- bool **cn**
- bool **email**
- bool **uid**

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

- inc/ldap\_communication.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\_communication.h
- isa-ldapserver.cpp



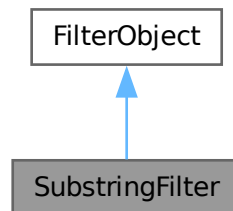
## 5.19 server Class Reference

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

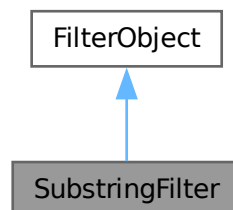
- server.h

## 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

```
00001 #ifndef ANDFILTEROBJECT_H
00002 #define ANDFILTEROBJECT_H
00003 #include "inc/FilterObject.h"
00004 #include "inc/ber_helper_functions.h"
00005
00006 #include <vector>
00007
00008 class AndFilter : public FilterObject {
00009 public:
00010     std::vector<FilterObject *> filters;
00011     filterTypes getFilterType();
00012     ~AndFilter();
00013 };
00014
00015 #endif
```

### 6.2 argument\_helper\_functions.h

```
00001 #ifndef ARGUMENT_HELPER_FUNCTIONS_H
00002 #define ARGUMENT_HELPER_FUNCTIONS_H
00003 #include "string.h"
00004 #include <stdlib.h>
00005 #include <stdio.h>
00006
00007 typedef struct args_t {
00008     char *dbPath;
00009     int port;
00010     bool err;
00011 } argsT;
00012
00013 argsT parseArguments(int argc, const char **argv);
00014
00015 #endif
```

### 6.3 ber\_constants.h

```
00001 #ifndef BER_CONSTANTS_H
00002 #define BER_CONSTANTS_H
00003 const unsigned int BER_TAG_LENGTH = 1;
00004 const unsigned int BER_LENGTH_OF_LENGTH_TAG = 1;
00005 const unsigned int BER_4BYTE_LENGTH_LENGTH = 4;
00006 const unsigned int BER_EXTENDED_RESPONSE_C = 0x78;
00007 const unsigned int BER_BIND_REQUEST_C = 0x60;
00008 const unsigned int BER_BIND_RESPONSE_C = 0x61;
00009 const unsigned int BER_SEARCH_REQUEST_C = 0x63;
00010 const unsigned int BER_SEARCH_RESULT_ENTRY_C = 0x64;
00011 const unsigned int BER_SEARCH_RESULT_DONE_C = 0x65;
00012 const unsigned int BER_UNBIND_REQUEST_C = 0x42;
00013
```

```

00014 const unsigned int BER_BOOL_C = 0x01;
00015 const unsigned int BER_INT_C = 0x02;
00016 const unsigned int BER_INT_4BYTES_C = 0x84;
00017 const unsigned int BER_OCTET_STRING_C = 0x04;
00018 const unsigned int BER_ENUM_C = 0x0A;
00019 const unsigned int BER_SEQUENCE_C = 0x30;
00020 const unsigned int BER_SET_C = 0x31;
00021
00022 //constants --- result codes
00023
00024 const unsigned int BER_LDAP_SUCCESS = 0x00;
00025 const unsigned int BER_LDAP_PROTOCOL_ERROR = 0x02;
00026 const unsigned int BER_LDAP_SIZE_LIMIT_EXCEEDED = 0x04;
00027 const unsigned int BER_LDAP_AUTH_METHOD_NOT_SUPPORTED = 0x07;
00028
00029
00030
00031 #endif

```

## 6.4 ber\_helper\_functions.h

```

00001 #ifndef BER_HELPER_FUNCTIONS_H
00002 #define BER_HELPER_FUNCTIONS_H
00003 #include <vector>
00004 #include "inc/ber_constants.h"
00005 #include <stdio.h>
00006 #include <stdlib.h>
00013 int ParseINT(unsigned char *s, int *err);
00014
00023 typedef enum filterTypes {
00024     AND,
00025     OR,
00026     NOT,
00027     equalityMatch,
00028     substrings,
00029     undefined,
00030 } filterTypes;
00031
00032 typedef enum berObjectTypes {
00033     berSequenceObject,
00034     berIntObject,
00035     berStringObject,
00036     berSetObject,
00037     berEnumObject,
00038     berBoolObject,
00039     berUndefined,
00040     berErr,
00041 } berObjectTypes;
00042
00049 int HowManyBytesWillIntUse(int value);
00050
00058 int WriteIntAppend(std::vector<unsigned char> &s, int value);
00059
00067 void AppendLength4Bytes(std::vector<unsigned char> &start, int value);
00068
00077 int GetLength(std::vector<unsigned char>::iterator start, int *err, std::vector<unsigned
char>::iterator end);
00078
00087 unsigned int ParseINT(std::vector<unsigned char>::iterator s, int *err, std::vector<unsigned
char>::iterator end);
00088
00097 int GetLengthOfLength(std::vector<unsigned char>::iterator start, int *err, std::vector<unsigned
char>::iterator end);
00098
00107 void SkipTags(std::vector<unsigned char>::iterator &start, int n, int *err, std::vector<unsigned
char>::iterator end);
00108
00116 void GoIntoTag(std::vector<unsigned char>::iterator &start, int *err, std::vector<unsigned
char>::iterator end);
00117
00126 void IncreaseLength4Bytes(std::vector<unsigned char>::iterator &start, int n,
int *err, std::vector<unsigned char>::iterator end);
00127
00128
00135 filterTypes getFilterType(std::vector<unsigned char>::iterator start);
00136
00137 #endif

```

## 6.5 BerBoolObject.h

```

00001 #ifndef BERBOOLOBJECT_H

```

```

00002 #define BERBOOLOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006 #include <vector>
00007
00008 class BerBoolObject : public BerObject {
00009 private:
00010     bool value;
00011
00012 public:
00013     berObjectTypes getBerObjectType();
00014     long long int getLenght();
00015     std::vector<unsigned char> getBerRepresentation();
00016     BerBoolObject(char value);
00017     ~BerBoolObject();
00018 };
00019
00020 #endif

```

## 6.6 BerEnumObject.h

```

00001 #ifndef BERENUMOBJECT_H
00002 #define BERENUMOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006 #include <vector>
00007
00008 class BerEnumObject : public BerObject {
00009 private:
00010     int value;
00011
00012 public:
00013     berObjectTypes getBerObjectType();
00014     long long int getLenght();
00015     std::vector<unsigned char> getBerRepresentation();
00016     BerEnumObject(char value);
00017     ~BerEnumObject();
00018 };
00019
00020 #endif

```

## 6.7 BerIntObject.h

```

00001 #ifndef BERINTOBJECT_H
00002 #define BERINTOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006 #include <vector>
00007
00008 class BerIntObject : public BerObject {
00009 private:
00010     int value;
00011
00012 public:
00013     berObjectTypes getBerObjectType();
00014     int getValue();
00015     void setValue(int value);
00016     long long int getLenght();
00017     std::vector<unsigned char> getBerRepresentation();
00018     BerIntObject();
00019     BerIntObject(int value);
00020     ~BerIntObject();
00021 };
00022
00023 #endif

```

## 6.8 BerObject.h

```

00001 #ifndef BER_OBJECT_H
00002 #define BER_OBJECT_H
00003 #include "inc/ber_helper_functions.h"
00004

```

```

00009 class BerObject {
00010 public:
00016     virtual berObjectTypes getBerObjectType();
00023     virtual long long int getLenght();
00029     virtual std::vector<unsigned char> getBerRepresentation();
00030
00031     virtual ~BerObject();
00032 };
00033
00034 #endif

```

## 6.9 BerParser.h

```

00001 #ifndef BERPARSER_H
00002 #define BERPARSER_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006 #include "inc/BerEnumObject.h"
00007 #include "inc/BerStringObject.h"
00008 #include "inc/BerSetObject.h"
00009 #include "inc/BerSequenceObject.h"
00010 #include "inc/BerSetObject.h"
00011 #include "inc/BerIntObject.h"
00012 #include "inc/BerBoolObject.h"
00013 #include "inc/BerUndefinedObject.h"
00014 #include <vector>
00015
00024 BerObject *ParseBerObject(std::vector<unsigned char>::iterator start,
00025                             int *err, std::vector<unsigned char>::iterator end);
00026
00027 #endif

```

## 6.10 BerSequenceObject.h

```

00001 #ifndef BERSEQUENCEOBJECT_H
00002 #define BERSEQUENCEOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006
00007 #include <vector>
00008
00009 class BerSequenceObject : public BerObject {
00010 private:
00011     int tag;
00012
00013 public:
00014     std::vector<BerObject *> objects;
00015     berObjectTypes getBerObjectType();
00016     long long int getLenght();
00017     std::vector<unsigned char> getBerRepresentation();
00018     BerSequenceObject(int tag);
00019     BerSequenceObject();
00020     int GetTag();
00021     ~BerSequenceObject();
00022 };
00023
00024 #endif

```

## 6.11 BerSetObject.h

```

00001 #ifndef BERSETOBJECT_H
00002 #define BERSETOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006
00007 #include <vector>
00008
00009 class BerSetObject : public BerObject {
00010 public:
00011     std::vector<BerObject *> objects;
00012     berObjectTypes getBerObjectType();
00013     long long int getLenght();

```

```

00014     std::vector<unsigned char> getBerRepresentation();
00015     BerSetObject();
00016     ~BerSetObject();
00017 };
00018
00019 #endif

```

## 6.12 BerStringObject.h

```

00001 #ifndef BERSTRINGOBJECT_H
00002 #define BERSTRINGOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_constants.h"
00005 #include "inc/ber_helper_functions.h"
00006 #include <string>
00007 #include <vector>
00008
00009 class BerStringObject : public BerObject {
00010 public:
00011     berObjectTypes getBerObjectType();
00012     std::vector<unsigned char> value;
00013     long long int getLenght();
00014     std::vector<unsigned char> getBerRepresentation();
00015     BerStringObject();
00016     BerStringObject(std::vector<unsigned char> value);
00017     BerStringObject(std::string value);
00018 };
00019
00020 #endif

```

## 6.13 BerUndefinedObject.h

```

00001 #ifndef BERUNDEFINEDOBJECT_H
00002 #define BERUNDEFINEDOBJECT_H
00003 #include "inc/BerObject.h"
00004 #include "inc/ber_helper_functions.h"
00005
00006 class BerUndefinedObject : public BerObject {
00007 private:
00008     std::vector<unsigned char> value;
00009
00010 public:
00011     berObjectTypes getBerObjectType();
00012     long long int getLenght();
00013     std::vector<unsigned char> getBerRepresentation();
00014     BerUndefinedObject(std::vector<unsigned char> value);
00015 };
00016
00017 #endif

```

## 6.14 database\_helper\_functions.h

```

00001 #ifndef DATABASE_HELPER_FUNCTIONS_H
00002 #define DATABASE_HELPER_FUNCTIONS_H
00003 #include <fstream>
00004 #include <iostream>
00005 #include <sstream>
00006 #include <string>
00007 #include <vector>
00008 #include "inc/DatabaseObject.h"
00009
00016 std::vector<DatabaseObject> removeDuplicates(std::vector<DatabaseObject> input);
00017
00018 #endif

```

## 6.15 DatabaseController.h

```

00001 #ifndef DATABASECONTROLLER_H
00002 #define DATABASECONTROLLER_H
00003 #include "inc/DatabaseObject.h"
00004 #include <fstream>

```

```

00005 #include <iostream>
00006 #include <sstream>
00007 #include <string>
00008 #include <vector>
00009
00014 class DatabaseController {
00015 private:
00016     std::ifstream file;
00017     std::vector<unsigned char> sanitaze(std::vector<unsigned char> input);
00018
00019 public:
00026     DatabaseObject loadNextRow(int *err);
00027
00033     std::vector<DatabaseObject> loadAllRows();
00034
00040     DatabaseController(std::string fileName);
00041     ~DatabaseController();
00042 };
00043
00044 #endif

```

## 6.16 DatabaseObject.h

```

00001 #ifndef DATABASE_OBJECT_H
00002 #define DATABASE_OBJECT_H
00003 #include <fstream>
00004 #include <iostream>
00005 #include <sstream>
00006 #include <string>
00007 #include <vector>
00008
00013 class DatabaseObject {
00014 private:
00015     std::vector<unsigned char> name;
00016     std::vector<unsigned char> uid;
00017     std::vector<unsigned char> email;
00018
00019 public:
00020     std::vector<unsigned char> get_name();
00021     std::vector<unsigned char> get_uid();
00022     std::vector<unsigned char> get_email();
00023     DatabaseObject(std::vector<unsigned char> name,
00024                   std::vector<unsigned char> uid,
00025                   std::vector<unsigned char> email);
00026 };
00027
00028
00029 #endif

```

## 6.17 EqualityMatchFilterObject.h

```

00001 #ifndef EQUALITYMATCHFILTEROBJECT_H
00002 #define EQUALITYMATCHFILTEROBJECT_H
00003 #include "inc/FilterObject.h"
00004 #include "inc/ber_helper_functions.h"
00005
00006 #include <vector>
00007
00008 class EqualityMatchFilter : public FilterObject {
00009 private:
00010     std::vector<unsigned char> attributeDescription;
00011     std::vector<unsigned char> assertionValue;
00012
00013 public:
00014     EqualityMatchFilter(std::vector<unsigned char> attributeDescription,
00015                       std::vector<unsigned char> assertionValue);
00016     std::vector<unsigned char> getAttributeDescription();
00017     std::vector<unsigned char> getAssertionValue();
00018     filterTypes getFilterType();
00019 };
00020 #endif

```

## 6.18 filter\_helper\_functions.h

```

00001

```



```

00002 #ifndef INC_FILTER_HELPER_FUNCTIONS_H
00003 #define INC_FILTER_HELPER_FUNCTIONS_H
00004
00005
00006 #include "inc/FilterObject.h"
00007 #include "inc/NotFilterObject.h"
00008 #include "inc/AndFilterObject.h"
00009 #include "inc/OrFilterObject.h"
00010 #include "inc/EqualityMatchFilterObject.h"
00011 #include "inc/SubstringFilterObject.h"
00012 #include "inc/DatabaseObject.h"
00013 #include "inc/DatabaseController.h"
00014 #include "vector"
00015
00025 bool substrFilterHandler(SubstringFilter *sf, int *err,
00026                         std::vector<unsigned char> attribute);
00027
00037 bool equalityMatchHandler(EqualityMatchFilter *emf, int *err,
00038                         std::vector<unsigned char> attribute) ;
00039
00049 bool filterLine(FilterObject *f, int *err, DatabaseObject &databaseEntry) ;
00050
00060 std::vector<DatabaseObject>
00061 filterHandler(FilterObject *f, int *err, const char *dbLocation, int sizeLimit);
00062
00070 FilterObject *convertToFilterObject(std::vector<unsigned char>::iterator BERfilter,
00071                                     std::vector<unsigned char>::iterator end);
00071
00072 #endif

```

## 6.19 FilterObject.h

```

00001 #ifndef FILTER_OBJECT_H
00002 #define FILTER_OBJECT_H
00003 #include "inc/ber_helper_functions.h"
00004
00005 #include <vector>
00006
00011 class FilterObject {
00012 public:
00013     virtual filterTypes getFilterType();
00014     virtual ~FilterObject();
00015 };
00016
00017
00018 #endif

```

## 6.20 ldap\_comunication.h

```

00001 #ifndef LDAP_COMMUNICATION_H
00002 #define LDAP_COMMUNICATION_H
00003 #include "inc/AndFilterObject.h"
00004 #include "inc/BerBoolObject.h"
00005 #include "inc/BerEnumObject.h"
00006 #include "inc/BerIntObject.h"
00007 #include "inc/BerObject.h"
00008 #include "inc/BerParser.h"
00009 #include "inc/BerSequenceObject.h"
00010 #include "inc/BerSetObject.h"
00011 #include "inc/BerStringObject.h"
00012 #include "inc/BerUndefinedObject.h"
00013 #include "inc/DatabaseController.h"
00014 #include "inc/DatabaseObject.h"
00015 #include "inc/EqualityMatchFilterObject.h"
00016 #include "inc/FilterObject.h"
00017 #include "inc/NotFilterObject.h"
00018 #include "inc/OrFilterObject.h"
00019 #include "inc/SubstringFilterObject.h"
00020 #include "inc/ber_constants.h"
00021 #include "inc/ber_helper_functions.h"
00022 #include "inc/database_helper_functions.h"
00023 #include "inc/filter_helper_functions.h"
00024 #include <algorithm>
00025 #include <arpa/inet.h>
00026 #include <netinet/in.h>
00027 #include <stdio.h>
00028 #include <stdlib.h>
00029 #include <string.h>
00030 #include <string>

```

```

00031 #include <sys/resource.h>
00032 #include <sys/socket.h>
00033 #include <sys/time.h>
00034 #include <sys/types.h>
00035 #include <sys/wait.h>
00036 #include <unistd.h>
00037 #include <vector>
00038
00039 typedef struct searchedAttributes {
00040     bool cn;
00041     bool email;
00042     bool uid;
00043 } searchedAttributesType;
00044
00045 // enum for attributes (cn, email, uid)
00046 typedef enum { cn, email, uid } attributeDescriptions;
00047
00048 // sequence - envelope
00049 //     int - message ID
00050 //     application 3 - search request
00051 //     octed string - base object
00052 //     enum - scope
00053 //     enum - derefAliases
00054 //     int - sizeLimit
00055 //     int - timeLimit
00056 //     bool - typesOnly
00057 //     sequence - FilterObject
00058 //     sequence - attributes
00059
00060 typedef struct searchRequest {
00061     int messageIDLength;
00062     unsigned int sizeLimit;
00063     searchedAttributesType attributes;
00064 } searchRequestType;
00065
00073 BerObject *InitSearchResultEntry(BerObject *searchRequest,
00074                                 std::vector<unsigned char> LDAPDN);
00075
00084 int AddToSearchResultEntry(BerObject *envelope,
00085                           std::vector<unsigned char> &attributeDescription,
00086                           std::vector<unsigned char> &attributeValue);
00093 int checkSearchRequest(BerObject *searchRequest);
00094
00102 int sendNoticeOfDisconnection(int comSocket, char errCode);
00103
00111 int searchRequestHandler(BerObject *searchRequest, int comm_socket,
00112                         const char *dbPath);
00113
00121 BerObject *CreateBindResponse(BerObject *bindRequest, int resultCode);
00122
00130 int loadEnvelope(std::vector<unsigned char> &bindRequest, int comm_socket);
00131
00140 int sendSearchResultDone(BerSequenceObject *searchRequest, int comm_socket,
00141                         unsigned int result_code);
00142 #endif

```

## 6.21 NotFilterObject.h

```

00001 #ifndef NOTFILTEROBJECT_H
00002 #define NOTFILTEROBJECT_H
00003 #include "inc/FilterObject.h"
00004 #include "inc/ber_helper_functions.h"
00005
00006 #include <vector>
00007
00008 class NotFilter : public FilterObject {
00009 public:
00010     FilterObject *filter;
00011     filterTypes getFilterType();
00012     ~NotFilter();
00013 };
00014
00015 #endif

```

## 6.22 OrFilterObject.h

```

00001 #ifndef ORFILTEROBJECT_H
00002 #define ORFILTEROBJECT_H
00003 #include "inc/FilterObject.h"

```

```

00004 #include "inc/ber_helper_functions.h"
00005
00006 #include <vector>
00007
00008 class OrFilter : public FilterObject {
00009 public:
00010     std::vector<FilterObject *> filters;
00011     filterTypes getFilterType();
00012     ~OrFilter();
00013 };
00014
00015 #endif

```

## 6.23 SubstringFilterObject.h

```

00001 #ifndef SUBSTRINGFILTER_H
00002 #define SUBSTRINGFILTER_H
00003 #include "inc/ber_helper_functions.h"
00004 #include "inc/FilterObject.h"
00005 #include <vector>
00006
00007 class SubstringFilter : public FilterObject {
00008
00009 private:
00010     std::vector<unsigned char> attributeDescription;
00011     std::vector<unsigned char> subInitial;
00012     std::vector<std::vector<unsigned char>> subAny;
00013     std::vector<unsigned char> subFinal;
00014
00015 public:
00016     SubstringFilter(std::vector<unsigned char> attributeDescription,
00017                     std::vector<unsigned char> subInitial,
00018                     std::vector<std::vector<unsigned char>> subAny,
00019                     std::vector<unsigned char> subFinal);
00020     std::vector<unsigned char> getAttributeDescription();
00021     std::vector<unsigned char> getSubInitial();
00022     std::vector<std::vector<unsigned char>> getSubAny();
00023     std::vector<unsigned char> getSubFinal();
00024     filterTypes getFilterType();
00025 };
00026
00027 #endif

```

## 6.24 server.h

```

00001 #ifndef SERVER_H
00002 #define SERVER_H
00003 #include "inc/BerEnumObject.h"
00004 #include "inc/BerIntObject.h"
00005 #include "inc/BerObject.h"
00006 #include "inc/BerParser.h"
00007 #include "inc/BerSequenceObject.h"
00008 #include "inc/BerSetObject.h"
00009 #include "inc/BerStringObject.h"
00010 #include "inc/DatabaseObject.h"
00011 #include "inc/FilterObject.h"
00012 #include "inc/argument_helper_functions.h"
00013 #include "inc/ldap_communication.h"
00014 #include <arpa/inet.h>
00015 #include <fcntl.h>
00016 #include <netinet/in.h>
00017 #include <stdio.h>
00018 #include <stdlib.h>
00019 #include <string.h>
00020 #include <sys/resource.h>
00021 #include <sys/socket.h>
00022 #include <sys/time.h>
00023 #include <sys/types.h>
00024 #include <sys/wait.h>
00025 #include <unistd.h>
00026
00027 // Macro for printing err message and closing socket when err != 0
00028 #define CHECK_ERR(err, msg)
00029     if (err != 0) {
00030         printf("%s\n", msg);
00031         close(childSocket);
00032         exit(0);
00033     }
00034

```

```
00035 int ldapServer(int port, char *dbPath);
00036
00037 #endif
```

## 6.25 server.h

```
00001 #include <iostream>
00002
00003 class server
00004 {
00005
00006 };
00007
```

# Index

- AndFilter, [11](#)
  - getFilterType, [12](#)
- args\_t, [12](#)
- BerBoolObject, [12](#)
  - getBerObjectType, [13](#)
  - getBerRepresentation, [13](#)
  - getLenght, [13](#)
- BerEnumObject, [14](#)
  - getBerObjectType, [15](#)
  - getBerRepresentation, [15](#)
  - getLenght, [15](#)
- BerIntObject, [16](#)
  - getBerObjectType, [17](#)
  - getBerRepresentation, [17](#)
  - getLenght, [17](#)
- BerObject, [18](#)
  - getBerObjectType, [19](#)
  - getBerRepresentation, [19](#)
  - getLenght, [19](#)
- BerSequenceObject, [20](#)
  - getBerObjectType, [21](#)
  - getBerRepresentation, [21](#)
  - getLenght, [21](#)
- BerSetObject, [22](#)
  - getBerObjectType, [23](#)
  - getBerRepresentation, [23](#)
  - getLenght, [23](#)
- BerStringObject, [24](#)
  - getBerObjectType, [25](#)
  - getBerRepresentation, [25](#)
  - getLenght, [25](#)
- BerUndefinedObject, [26](#)
  - getBerObjectType, [26](#)
  - getBerRepresentation, [27](#)
  - getLenght, [27](#)
- DatabaseController, [27](#)
  - DatabaseController, [28](#)
  - loadAllRows, [28](#)
  - loadNextRow, [28](#)
- DatabaseObject, [29](#)
- EqualityMatchFilter, [30](#)
  - getFilterType, [30](#)
- FilterObject, [31](#)
- getBerObjectType
  - BerBoolObject, [13](#)
  - BerEnumObject, [15](#)
  - BerIntObject, [17](#)
  - BerObject, [19](#)
  - BerSequenceObject, [21](#)
  - BerSetObject, [23](#)
  - BerStringObject, [25](#)
  - BerUndefinedObject, [26](#)
- getBerRepresentation
  - BerBoolObject, [13](#)
  - BerEnumObject, [15](#)
  - BerIntObject, [17](#)
  - BerObject, [19](#)
  - BerSequenceObject, [21](#)
  - BerSetObject, [23](#)
  - BerStringObject, [25](#)
  - BerUndefinedObject, [27](#)
- getFilterType
  - AndFilter, [12](#)
  - EqualityMatchFilter, [30](#)
  - NotFilter, [32](#)
  - OrFilter, [33](#)
  - SubstringFilter, [36](#)
- getLenght
  - BerBoolObject, [13](#)
  - BerEnumObject, [15](#)
  - BerIntObject, [17](#)
  - BerObject, [19](#)
  - BerSequenceObject, [21](#)
  - BerSetObject, [23](#)
  - BerStringObject, [25](#)
  - BerUndefinedObject, [27](#)
- inc/AndFilterObject.h, [37](#)
- inc/argument\_helper\_functions.h, [37](#)
- inc/ber\_constants.h, [37](#)
- inc/ber\_helper\_functions.h, [38](#)
- inc/BerBoolObject.h, [38](#)
- inc/BerEnumObject.h, [39](#)
- inc/BerIntObject.h, [39](#)
- inc/BerObject.h, [39](#)
- inc/BerParser.h, [40](#)
- inc/BerSequenceObject.h, [40](#)
- inc/BerSetObject.h, [40](#)
- inc/BerStringObject.h, [41](#)
- inc/BerUndefinedObject.h, [41](#)
- inc/database\_helper\_functions.h, [41](#)
- inc/DatabaseController.h, [41](#)
- inc/DatabaseObject.h, [42](#)
- inc/EqualityMatchFilterObject.h, [42](#)
- inc/filter\_helper\_functions.h, [42](#)
- inc/FilterObject.h, [43](#)

inc/ldap\_communication.h, [43](#)  
inc/NotFilterObject.h, [44](#)  
inc/OrFilterObject.h, [44](#)  
inc/server.h, [45](#)  
inc/SubstringFilterObject.h, [45](#)

loadAllRows  
    DatabaseController, [28](#)  
loadNextRow  
    DatabaseController, [28](#)

NotFilter, [31](#)  
    getFilterType, [32](#)

OrFilter, [33](#)  
    getFilterType, [33](#)

searchedAttributes, [34](#)  
searchRequest, [34](#)  
server, [35](#)  
SubstringFilter, [35](#)  
    getFilterType, [36](#)

Teorie, [1](#)