Project 4

Generated by Doxygen 1.9.6

1 Class Index 1

1.1 Class List 1

2 File Index 3

2.1 File List 3

3 Class Documentation 5

3.1 DelimTextBuffer Class Reference 5

3.1.1 Constructor & Destructor Documentation 6

3.1.1.1 DelimTextBuffer() 6

3.1.2 Member Function Documentation 7

3.1.2.1 Clear() 7

3.1.2.2 Init() 7

3.1.2.3 Pack() 8

3.1.2.4 PackHeader() 8

3.1.2.5 Print() 9

3.1.2.6 Read() 9

3.1.2.7 ReadHeader() 10

3.1.2.8 Unpack() 11

3.1.2.9 UnpackHeader() 11

3.1.2.10 Write() 12

3.1.2.11 WriteHeader() 13

3.1.3 Member Data Documentation 13

3.1.3.1 Buffer 13

3.1.3.2 BufferSize 13

3.1.3.3 count 13

3.1.3.4 Delim 13

3.1.3.5 DelimStr 14

3.1.3.6 MaxBytes 14

3.1.3.7 NextByte 14

3.1.3.8 Rbuffer 14

3.2 PKIStruct Struct Reference 14

3.2.1 Member Data Documentation 15

3.2.1.1 byteOffset 15

3.2.1.2 zipcode 15

3.3 State Class Reference 15

3.3.1 Constructor & Destructor Documentation 16

3.3.1.1 State() 16

3.3.2 Member Data Documentation 16

3.3.2.1 easternZipcode 16

3.3.2.2 largestLat 16

3.3.2.3 largestLong 16

3.3.2.4 northernZipcode 17

3.3.2.5 smallestLat 17
3.3.2.6 smallestLong 17
3.3.2.7 southernZipcode 17
3.3.2.8 stateName 17
3.3.2.9 westernZipcode 17
3.4 Zipcode Class Reference 18
3.4.1 Constructor & Destructor Documentation 19
3.4.1.1 Zipcode() 19
3.4.2 Member Function Documentation 19
3.4.2.1 Clear() 19
3.4.2.2 InitBuffer() 20
3.4.2.3 Pack() 20
3.4.2.4 Print() 21
3.4.2.5 Size() 22
3.4.2.6 Unpack() 22
3.4.3 Member Data Documentation 23
3.4.3.1 Code 23
3.4.3.2 County 23
3.4.3.3 Lat 24
3.4.3.4 Long 24
3.4.3.5 Placename 24
3.4.3.6 State 24

4 File Documentation **25**
4.1 deltext.cpp File Reference 25
4.2 deltext.h File Reference 26
4.2.1 Macro Definition Documentation 27
4.2.1.1 FALSE 27
4.2.1.2 TRUE 27
4.3 deltext.h 28
4.4 Proj3_group5.cpp File Reference 28
4.4.1 Detailed Description 29
4.4.2 Function Documentation 29
4.4.2.1 application() 30
4.4.2.2 compareStates() 31
4.4.2.3 constructStateArray() 31
4.4.2.4 main() 31
4.4.2.5 outputTable() 32
4.4.2.6 setZipCodes() 33
4.5 Proj4_group5.cpp File Reference 33
4.5.1 Detailed Description 35
4.5.2 Function Documentation 35

4.5.2.1 application() 35
4.5.2.2 compareStates() 36
4.5.2.3 constructStateArray() 36
4.5.2.4 generateColumnOrder() 36
4.5.2.5 generatePKI() 37
4.5.2.6 generatePKIHeader() 37
4.5.2.7 main() 38
4.5.2.8 myrandom() 38
4.5.2.9 outputCSV() 38
4.5.2.10 outputRandomColumnCSV() 40
4.5.2.11 outputTable() 41
4.5.2.12 packHeader() 41
4.5.2.13 readFileNoHeaderLength() 42
4.5.2.14 readFileWithHeaderLength() 43
4.5.2.15 readPKI() 44
4.5.2.16 searchPKI() 45
4.5.2.17 setZipCodes() 46
4.5.3 Variable Documentation 46
4.5.3.1 columnOrderArray 46
4.5.3.2 headerArray 47
4.6 State.h File Reference 47
4.6.1 Detailed Description 47
4.7 State.h 48
4.8 zipcode.cpp File Reference 48
4.8.1 Detailed Description 49
4.9 zipcode.h File Reference 50
4.9.1 Detailed Description 51
4.10 zipcode.h 51

Index **53**

Chapter 1

Class Index

1.1 Class List

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

DelimTextBuffer 5
PKIStruct 14
State 15
Zipcode 18

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

deltext.cpp 25
deltext.h 26
Proj3_group5.cpp
Takes a csv file containing US postal codes as inputs and generates an output table consiting of each of the state's easternmost, westernmost, northernmost and southernmost zipcodes based on latitude and longitude comparisions 28
Proj4_group5.cpp
Driver file for Zipcode class 33
State.h
Declaration file for State class 47
zipcode.cpp
Header file for Zipcode class 48
zipcode.h
Header file for Zipcode class 50

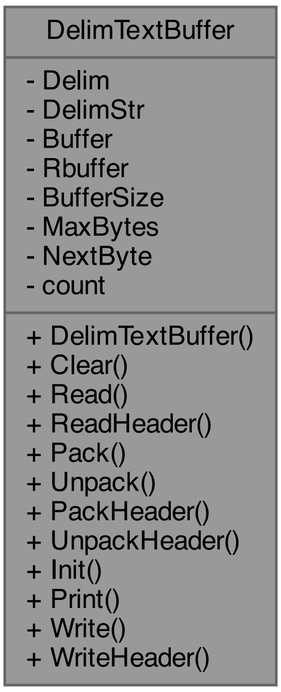
Chapter 3

Class Documentation

3.1 DelimTextBuffer Class Reference

```
#include <deltext.h>
```

Collaboration diagram for DelimTextBuffer:



Public Member Functions

- DelimTextBuffer (char Delim=',', int maxBytes=10000)
- void Clear ()
- int Read (std::istream &)
- int ReadHeader (std::istream &)
- int Pack (const char ∗, int size=-1)
- int Unpack (char ∗)
- int PackHeader (const char ∗, int size=-1)
- int UnpackHeader (char ∗)
- int Init (char delim, int maxBytes=10000)
- void Print (std::ostream &) const
- int Write (std::ostream &) const
- int WriteHeader (std::ostream &) const

Private Attributes

- char Delim
- char DelimStr [3]
- char ∗ Buffer
- std::string Rbuffer
- int BufferSize
- int MaxBytes
- int NextByte
- int count

3.1.1 Constructor & Destructor Documentation

3.1.1.1 DelimTextBuffer()

```
DelimTextBuffer::DelimTextBuffer (
            char Delim = ',',
            int maxBytes = 10000 )
```

Constructor;

Parameters

Delim	the delimiter character to be used
maxBytes	maximum number of characters in the buffer

Precondition

none

Postcondition

a buffer of size maxBytes is created with Delim as the delimiter

3.1.2 Member Function Documentation

3.1.2.1 Clear()

```
void DelimTextBuffer::Clear ( )
```

MODIFICATION MEMBER FUNCTIONS Clear; Clear fields from buffer

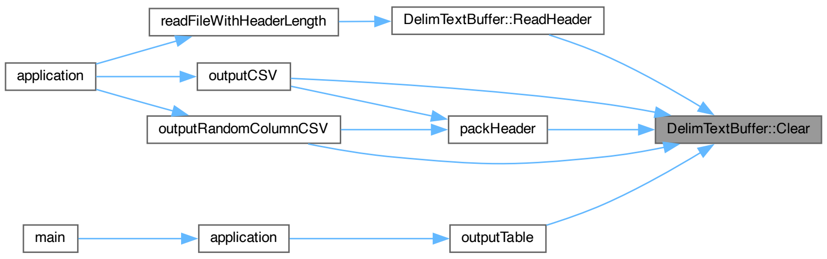
Precondition

a DelimTextBuffer must exist

Postcondition

the fields in the buffer are empty

Here is the caller graph for this function:



3.1.2.2 Init()

```
int DelimTextBuffer::Init (
             char delim,
             int maxBytes = 10000 )
```

Init Initalize the buffer

Parameters

delim	a character delimiter for the buffer
maxBytes	the maximum number of characters in the buffer(default 10000)

Precondition

a DelimTextBuffer must exist

Postcondition

the buffer is inialized to have delim as the delimiter and a maxBytes of maximum characters

3.1.2.3 Pack()

```
int DelimTextBuffer::Pack (
            const char * str,
            int size = -1 )
```

Pack; Packs the next value into a c style string

Parameters

a	c style string
size	of the c style string

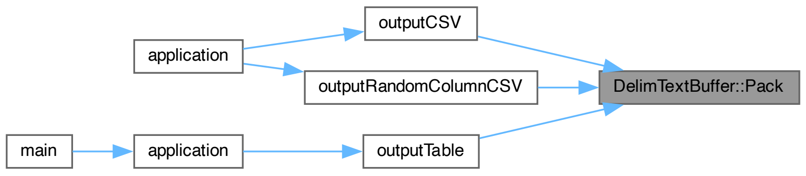
Precondition

a DelimTextBuffer must exist

Postcondition

a c style string is packed from the buffer

Here is the caller graph for this function:



3.1.2.4 PackHeader()

```
int DelimTextBuffer::PackHeader (
            const char * str,
            int size = -1 )
```

PackHeader Packs the header data into a character buffer.

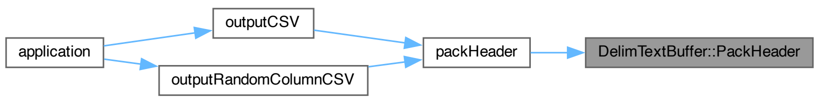
Parameters

data	a pointer to the character buffer to store the packed data
size	the size of the buffer, defaults to -1 to indicate that the buffer size should be calculated

Returns

an integer value indicating the number of bytes written to the buffer

Here is the caller graph for this function:



3.1.2.5 Print()

```
void DelimTextBuffer::Print (
             std::ostream & stream ) const
```

NONMODIFICATION MEMBER FUNCTIONS Print; Prints the maximum size and characters for the buffer

Parameters

ostream	object

Precondition

a DelimTextBuffer must exist

Postcondition

the maximum size and characters for the buffer are written to an ostream object

3.1.2.6 Read()

```
int DelimTextBuffer::Read (
             std::istream & stream )
```

Read; Reads into the buffer from an istream object

Parameters

istream	object

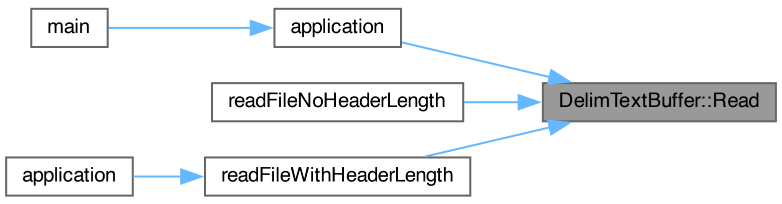
Precondition

a DelimTextBuffer must exsist

Postcondition

a single line from an istream object is read into the buffer

Here is the caller graph for this function:



3.1.2.7 ReadHeader()

```
int DelimTextBuffer::ReadHeader (
             std::istream & stream )
```

Read Header Reads the header of an input stream.

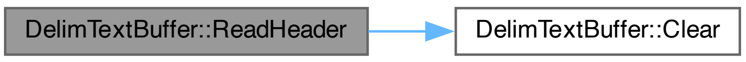
Parameters

input	the input stream to read from

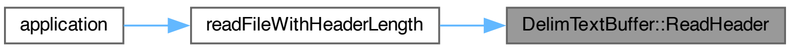
Returns

an integer value indicating success or failure

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.8 Unpack()

```
int DelimTextBuffer::Unpack (
             char * str )
```

Unpack; Unpacks a c style string into the buffer

Parameters

a	c style string

Precondition

a DelimTextBuffer must exist

Postcondition

a c style string is unpacked into the buffer

3.1.2.9 UnpackHeader()

```
int DelimTextBuffer::UnpackHeader (
             char * str )
```

UnpackHeader; Unpacks the header data from a character buffer.

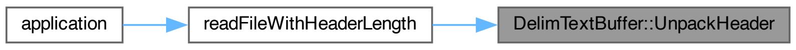
Parameters

buffer	a pointer to the character buffer containing the packed data

Returns

an integer value indicating the number of bytes read from the buffer

Here is the caller graph for this function:



3.1.2.10 Write()

```
int DelimTextBuffer::Write (
             std::ostream & stream ) const
```

Write; Writes the entire buffer to an ostream object

Parameters

ostream	object

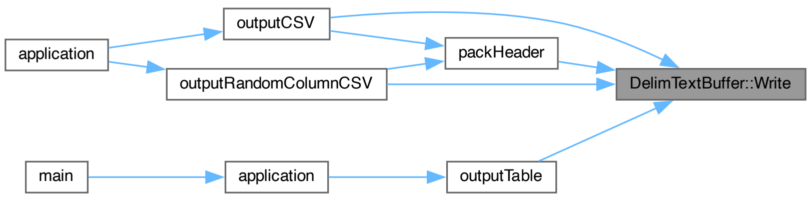
Precondition

a DelimTextBuffer must exist

Postcondition

the entire buffer is written into an ostream object with delimiters

Here is the caller graph for this function:



3.1.2.11 WriteHeader()

```
int DelimTextBuffer::WriteHeader (
            std::ostream & stream ) const
```

WriteHeader; Writes the header to an output stream.

Parameters

output	the output stream to write to

Returns

an integer value indicating success or failure

Postcondition

a single line from an istream object is read into the buffer

3.1.3 Member Data Documentation

3.1.3.1 Buffer

```
char* DelimTextBuffer::Buffer  [private]
```

character array to hold field values

3.1.3.2 BufferSize

```
int DelimTextBuffer::BufferSize  [private]
```

size of packed fields

3.1.3.3 count

```
int DelimTextBuffer::count  [private]
```

count if it is the end

3.1.3.4 Delim

```
char DelimTextBuffer::Delim  [private]
```

delimiter character

3.1.3.5 DelimStr

```
char DelimTextBuffer::DelimStr[3]  [private]
```

zero terminated string for Delim

3.1.3.6 MaxBytes

```
int DelimTextBuffer::MaxBytes  [private]
```

maximum number of characters in the buffer

3.1.3.7 NextByte

```
int DelimTextBuffer::NextByte  [private]
```

packing/unpacking position in buffer

3.1.3.8 Rbuffer

```
std::string DelimTextBuffer::Rbuffer  [private]
```

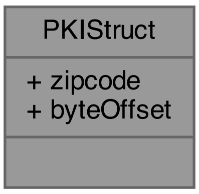
string buffer

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

- deltext.h
- deltext.cpp

3.2 PKIStruct Struct Reference

Collaboration diagram for PKIStruct:



Public Attributes

- char zipcode [10]
- char byteOffset [10]

3.2.1 Member Data Documentation

3.2.1.1 byteOffset

```
char PKIStruct::byteOffset[10]
```

3.2.1.2 zipcode

```
char PKIStruct::zipcode[10]
```

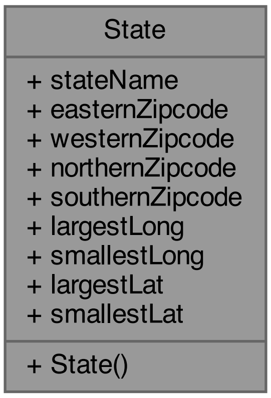
The documentation for this struct was generated from the following file:

- Proj4_group5.cpp

3.3 State Class Reference

```
#include <State.h>
```

Collaboration diagram for State:



Public Member Functions

- State ()

Public Attributes

- char stateName [5]
- char easternZipcode [10]
- char westernZipcode [10]
- char northernZipcode [10]
- char southernZipcode [10]
- char largestLong [10]
- char smallestLong [10]
- char largestLat [10]
- char smallestLat [10]

3.3.1 Constructor & Destructor Documentation

3.3.1.1 State()

```
State::State ( )
```

3.3.2 Member Data Documentation

3.3.2.1 easternZipcode

```
char State::easternZipcode[10]
```

3.3.2.2 largestLat

```
char State::largestLat[10]
```

3.3.2.3 largestLong

```
char State::largestLong[10]
```

3.3.2.4 northernZipcode

```
char State::northernZipcode[10]
```

3.3.2.5 smallestLat

```
char State::smallestLat[10]
```

3.3.2.6 smallestLong

```
char State::smallestLong[10]
```

3.3.2.7 southernZipcode

```
char State::southernZipcode[10]
```

3.3.2.8 stateName

```
char State::stateName[5]
```

Data members

3.3.2.9 westernZipcode

```
char State::westernZipcode[10]
```

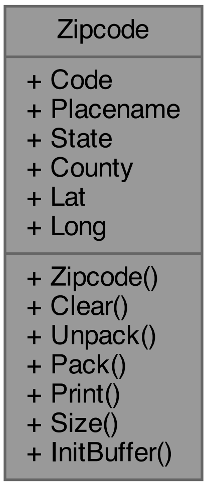
The documentation for this class was generated from the following file:

- State.h

3.4 Zipcode Class Reference

```
#include <zipcode.h>
```

Collaboration diagram for Zipcode:



Public Member Functions

- Zipcode ()
- void Clear ()
- int Unpack (DelimTextBuffer &)
- int Pack (DelimTextBuffer &) const

    the Zipcode object into a DelimTextBuffer object
- void Print (std::ostream &)
- int Size ()

Static Public Member Functions

- static int InitBuffer (DelimTextBuffer &)

Public Attributes

- char Code [10]
- char Placename [30]
- char State [5]
- char County [25]
- char Lat [10]
- char Long [10]

3.4.1 Constructor & Destructor Documentation

3.4.1.1 Zipcode()

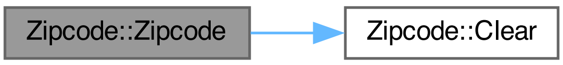
```
Zipcode::Zipcode ( )
```

Constructor

Postcondition

Initializes an empty Zipcode object

Here is the call graph for this function:



3.4.2 Member Function Documentation

3.4.2.1 Clear()

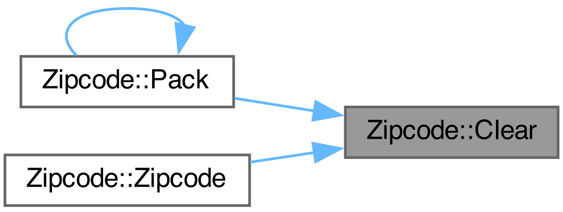
```
Zipcode::Clear ( )
```

Sets all fields to empty strings

Postcondition

All fields set to empty strings

Here is the caller graph for this function:



3.4.2.2 InitBuffer()

```
Zipcode::InitBuffer (
            DelimTextBuffer & Buffer )  [static]
```

Initializes a DelimTextBuffer object

Parameters

the	DelimTextBuffer to be initialized

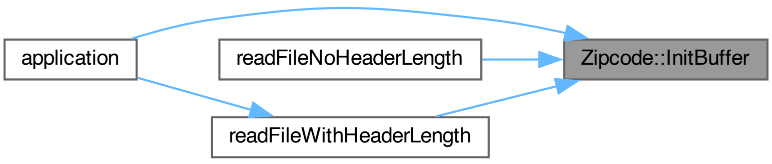
Precondition

a Zipcode object must exist

Postcondition

the DelimTextBuffer object is initialized

Here is the caller graph for this function:



3.4.2.3 Pack()

```
Zipcode::Pack (
            DelimTextBuffer & Buffer ) const
```

the Zipcode object into a DelimTextBuffer object

Parameters

The	DelimTextBuffer to pack

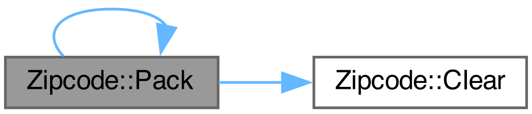
Precondition

DelimTextBuffer must exist and be initialized

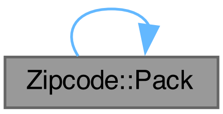
Postcondition

the Zipcode object is packed into a DelimTextBuffer

Set Code to '' if writing the header.Here is the call graph for this function:



Here is the caller graph for this function:



3.4.2.4 Print()

```
Zipcode::Print (
            std::ostream & stream )
```

Prints Zipcode object into an ostream object

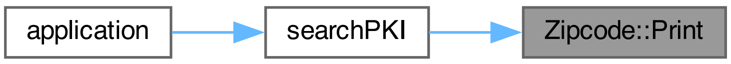
Parameters

ostream	object to print to

Postcondition

Fields from Zipcode are written into ostream object

Here is the caller graph for this function:



3.4.2.5 Size()

```
int Zipcode::Size ( )
```

Size;

Precondition

a zipcode object must exist

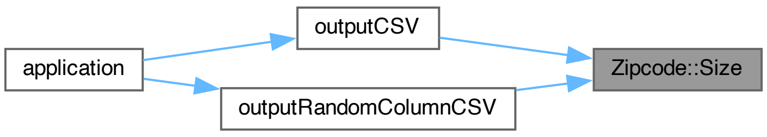
Postcondition

returns the length of all fields in a zipcode including the commas

Returns

int

Here is the caller graph for this function:



3.4.2.6 Unpack()

```
Zipcode::Unpack (
            DelimTextBuffer & Buffer )
```

Unpacks DelimTextBuffer into Zipcode object

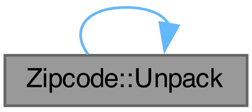
Parameters

the	DelimTextBuffer to be unpacked

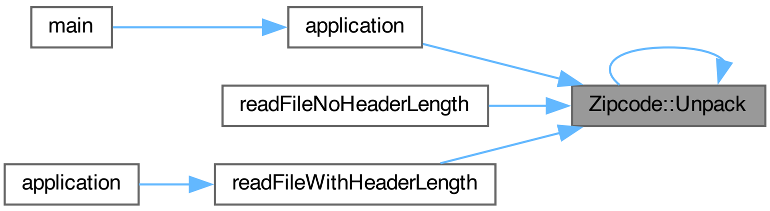
Postcondition

the DelimTextBuffer is unpacked into a Zipcode object

Here is the call graph for this function:



Here is the caller graph for this function:



3.4.3 Member Data Documentation

3.4.3.1 Code

```
char Zipcode::Code[10]
```

3.4.3.2 County

```
char Zipcode::County[25]
```

3.4.3.3 Lat

```
char Zipcode::Lat[10]
```

3.4.3.4 Long

```
char Zipcode::Long[10]
```

3.4.3.5 Placename

```
char Zipcode::Placename[30]
```

3.4.3.6 State

```
char Zipcode::State[5]
```

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

- zipcode.h
- zipcode.cpp

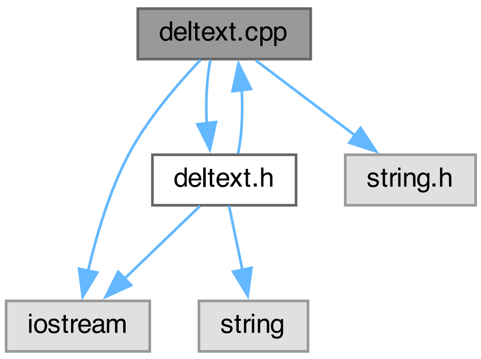
Chapter 4

File Documentation

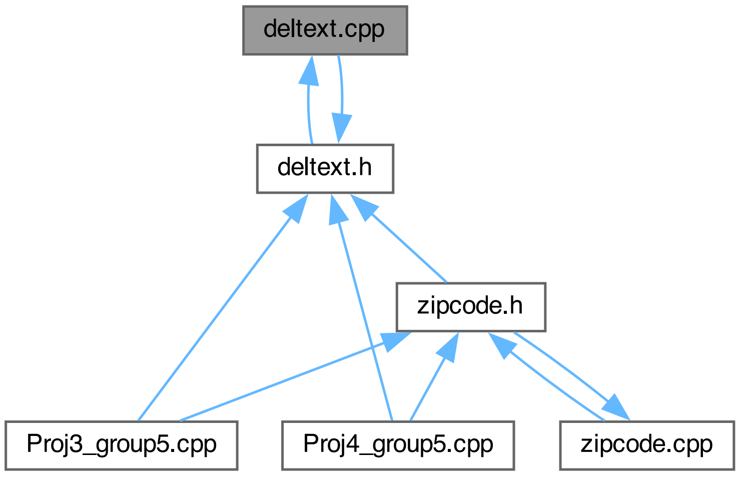
4.1 deltext.cpp File Reference

```
#include "deltext.h"
#include <string.h>
#include <iostream>
```

Include dependency graph for deltext.cpp:



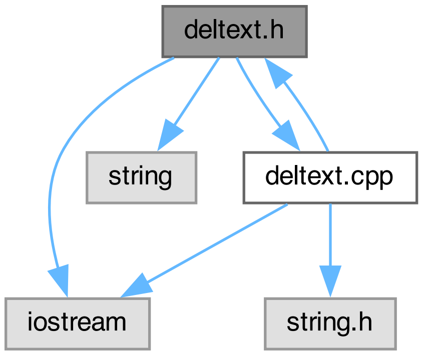
This graph shows which files directly or indirectly include this file:



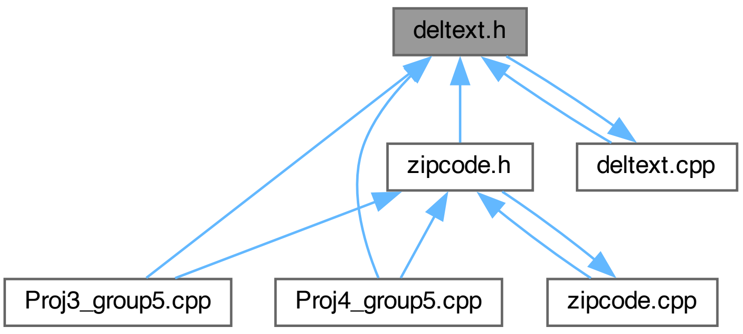
4.2 deltext.h File Reference

```
#include <iostream>
#include <string>
#include "deltext.cpp"
```

Include dependency graph for deltext.h:



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



Classes

- class DelimTextBuffer

Macros

- #define FALSE (0)
- #define TRUE (1)

4.2.1 Macro Definition Documentation

4.2.1.1 FALSE

```
#define FALSE (0)
```

4.2.1.2 TRUE

```
#define TRUE (1)
```

4.3 deltext.h

Go to the documentation of this file.

```
#ifndef DELTEXT_H
#define DELTEXT_H

/*provides ostream and istream*/
#include <iostream>
/*provides use of c++ strings*/
#include <string>

#ifndef FALSE
#define FALSE (0)
#define TRUE (1)
#endif

class DelimTextBuffer
// a buffer which holds delimited text fields.
// Record variables can be packed into and extracted from a buffer.
{
public:
    DelimTextBuffer(char Delim = ',', int maxBytes = 10000);

    void Clear(); // clear fields from buffer

    int Read(std::istream &);

    int ReadHeader(std::istream &);

    int Pack(const char *, int size = -1);

    int Unpack(char *);

    int PackHeader(const char *, int size = -1);

    int UnpackHeader(char *);

    int Init(char delim, int maxBytes = 10000);

    void Print(std::ostream &) const;

    int Write(std::ostream &) const;

    int WriteHeader(std::ostream &) const;

private:
    char Delim;
    char DelimStr[3];
    char *Buffer;
    std::string Rbuffer;
    int BufferSize;
    int MaxBytes;
    int NextByte;
    int count;
};

#include "deltext.cpp"
#endif
```

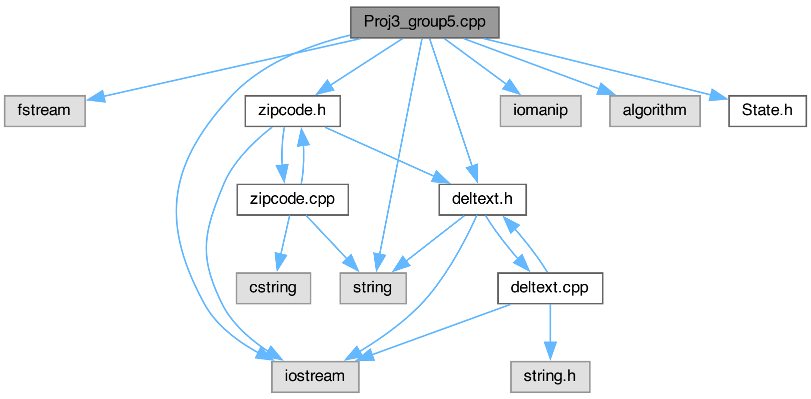
4.4 Proj3_group5.cpp File Reference

Takes a csv file containing US postal codes as inputs and generates an output table consiting of each of the state's easternmost, westernmost, northernmost and southernmost zipcodes based on latitude and longitude comparisons.

```
#include <fstream>
#include <iostream>
#include <string>
#include <iomanip>
#include <algorithm>
#include "deltext.h"
#include "zipcode.h"
```

```
#include "State.h"
```

Include dependency graph for Proj3_group5.cpp:



Functions

- bool compareStates (State a, State b)
- void constructStateArray (State sArray[ ], Zipcode zArray[ ], int zArraySize)
- void setZipCodes (State sArray[ ], Zipcode zArray[ ], int zArraySize)
- void outputTable (std::string outputFileName, DelimTextBuffer OutBuff, State sArray[ ])
- void application ()
- int main ()

4.4.1 Detailed Description

Takes a csv file containing US postal codes as inputs and generates an output table consiting of each of the state's easternmost, westernmost, northernmost and southernmost zipcodes based on latitude and longitude comparisons.

Author

Steven Kraus

Emily Yang

Tyler Knudtson

Ashesh Nepal

4.4.2 Function Documentation

4.4.2.1 application()

```
application ( )
```

Contains the code for controlling the Zipcode class and generating output file.

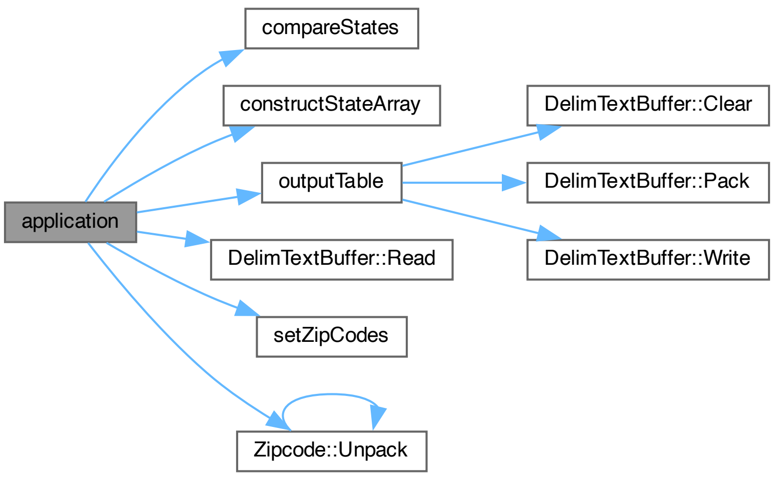
Precondition

specified InFile must be present

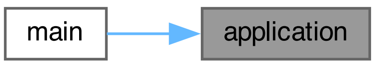
Postcondition

sorted OutFile with zip codes from each state will be created

Here is the call graph for this function:



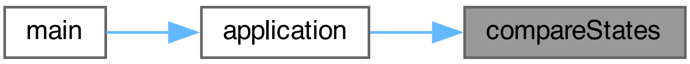
Here is the caller graph for this function:



4.4.2.2 compareStates()

```
bool compareStates (
            State a,
            State b )
```

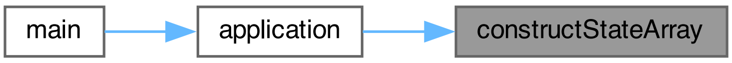
Here is the caller graph for this function:



4.4.2.3 constructStateArray()

```
void constructStateArray (
            State sArray[ ],
            Zipcode zArray[ ],
            int zArraySize )
```

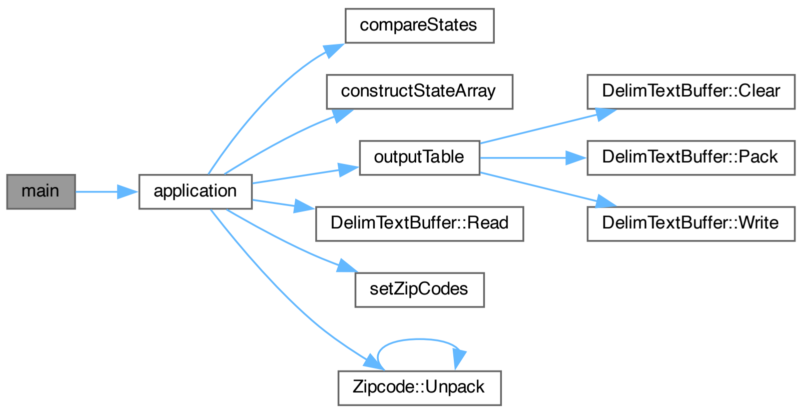
Here is the caller graph for this function:



4.4.2.4 main()

```
main ( )
```

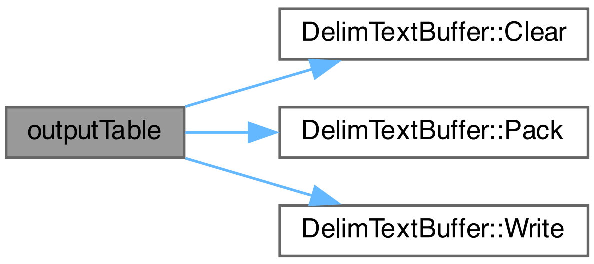
Executes the code present in application() Here is the call graph for this function:



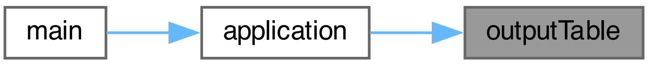
4.4.2.5 outputTable()

```
void outputTable (
            std::string outputFileName,
            DelimTextBuffer OutBuff,
            State sArray[] )
```

Here is the call graph for this function:



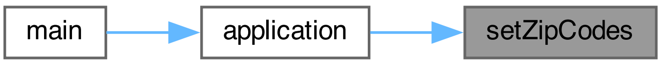
Here is the caller graph for this function:



4.4.2.6 setZipCodes()

```
void setZipCodes (
            State sArray[ ],
            Zipcode zArray[ ],
            int zArraySize )
```

Here is the caller graph for this function:



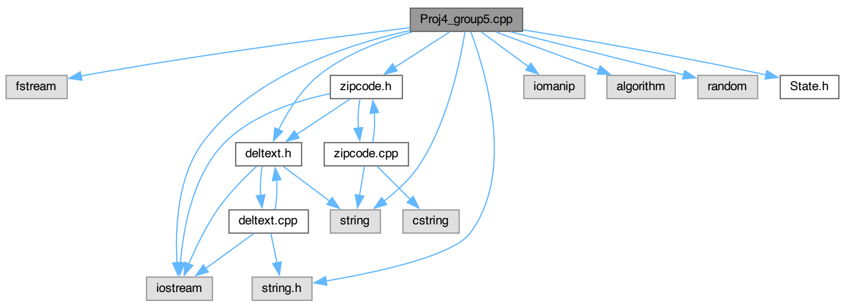
4.5 Proj4_group5.cpp File Reference

Driver file for Zipcode class.

```
#include <fstream>
#include <iostream>
#include <string>
#include <string.h>
#include <iomanip>
#include <algorithm>
#include <random>
#include "deltext.h"
#include "zipcode.h"
```

```
#include "State.h"
```

Include dependency graph for Proj4_group5.cpp:



Classes

- struct PKIStruct

Functions

- bool compareStates (State a, State b)
- void constructStateArray (State sArray[ ], Zipcode zArray[ ], int zArraySize)
- void setZipCodes (State sArray[ ], Zipcode zArray[ ], int zArraySize)
- void outputTable (std::string outputFileName, DelimTextBuffer OutBuff, State sArray[ ], int sArraySize)
- int myrandom (int i)
- int packHeader (std::string outputFileName, DelimTextBuffer OutBuff, std::string hArray[ ])
- std::string generatePKIHeader (std::string PKIheader)
- void generatePKI (std::ofstream &PKIOutFile, int currentPos, char ∗zipcode, int index)
- void outputCSV (std::string outputFileName, std::string PKIoutputFileName, DelimTextBuffer OutBuff, Zipcode zArray[ ], int zSize, std::string hArray[ ])
- int readFileWithHeaderLength (Zipcode zArray[ ], DelimTextBuffer InBuff, std::string InFileName)
- int readFileNoHeaderLength (Zipcode zArray[ ], DelimTextBuffer InBuff, std::string InFileName)
- int readPKI (PKIStruct pArray[ ], std::string PKIFileName)
- void searchPKI (int zIndex, int pIndex, int argc, char ∗∗argv, Zipcode zArray[ ], PKIStruct pArray[ ])
- void generateColumnOrder (std::string hArray[ ])
- void outputRandomColumnCSV (std::string outputFileName, std::string PKIoutputFileName, DelimTextBuffer OutBuff, Zipcode zArray[ ], int zSize, std::string hArray[ ])
- void application (int argc, char ∗∗argv)
- int main (int argc, char ∗∗argv)

Variables

- std::string headerArray [HEADER_FIELDS]
- std::string columnOrderArray [6]

4.5.1 Detailed Description

Driver file for Zipcode class.

Version

0.1

Date

2023-04-10

Author

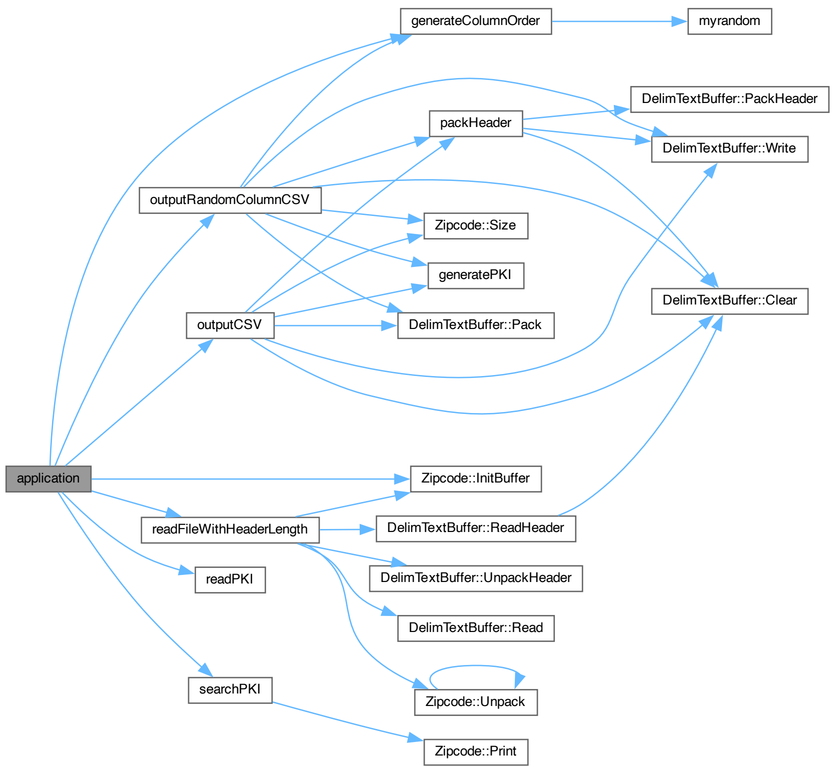
Steven Kraus
Emily Yang
Tyler Knudtson
Ashesh Nepal

4.5.2 Function Documentation

4.5.2.1 application()

```
void application (
            int argc,
            char ** argv )
```

Here is the call graph for this function:



4.5.2.2 compareStates()

```
bool compareStates (
            State a,
            State b )
```

4.5.2.3 constructStateArray()

```
void constructStateArray (
            State sArray[ ],
            Zipcode zArray[ ],
            int zArraySize )
```

4.5.2.4 generateColumnOrder()

```
void generateColumnOrder (
            std::string hArray[ ] )
```

generateColumOrder

Parameters

hArray	headerArray

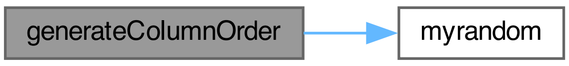
Precondition

none

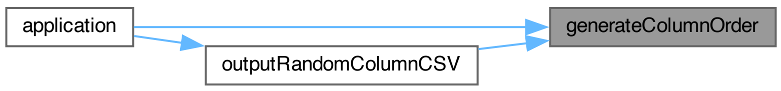
Postcondition

the header array has its 1-6 items shuffled and the column order is randomized

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.5 generatePKI()

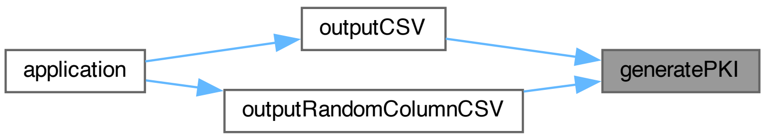
```
void generatePKI (
             std::ofstream & PKIOutFile,
             int currentPos,
             char * zipcode,
             int index )
```

generatePKI

Parameters

ofStream	PKIOutFile
int	currentPos
cstyle	string zipcode
int	index

Here is the caller graph for this function:



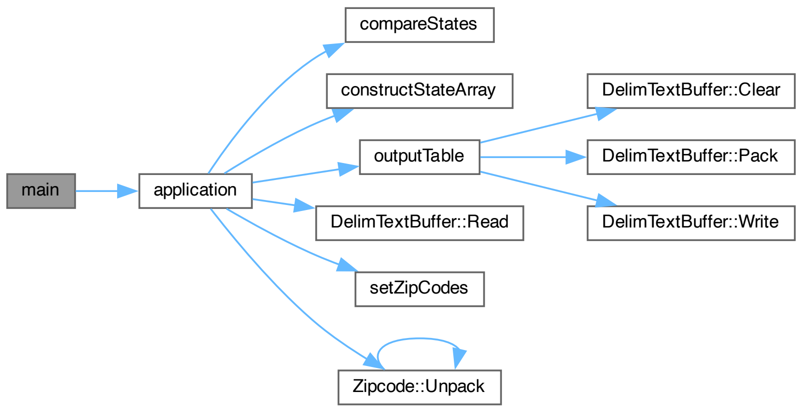
4.5.2.6 generatePKIHeader()

```
std::string generatePKIHeader (
             std::string PKIheader )
```

4.5.2.7 main()

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

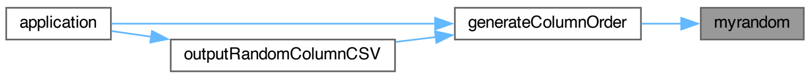
Here is the call graph for this function:



4.5.2.8 myrandom()

```
int myrandom (
            int i )
```

Here is the caller graph for this function:



4.5.2.9 outputCSV()

```
void outputCSV (
            std::string outputFileName,
            std::string PKIoutputFileName,
            DelimTextBuffer OutBuff,
            Zipcode zArray[ ],
            int zSize,
            std::string hArray[ ] )
```

outputCSV

Parameters

string	outputFileName
string	PKIoutputFileName
DelimTextBuffer	OutBuff
Zipcode	array zArray
int	zSize
string	array hArray

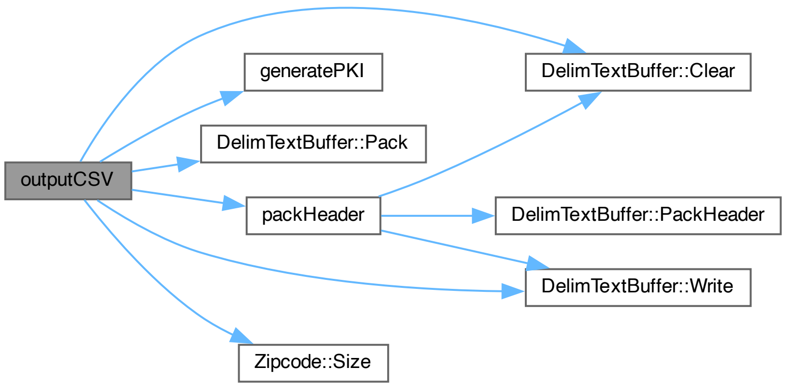
Precondition

a DelimtTextBuffer, and a filled array of zipcodes must exist

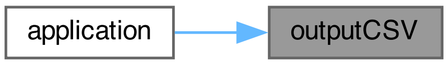
Postcondition

the zipcode array is packed into the buffer and written to the given outputFileName file after the given hArray header record is written. A pki file is generated with the name PKIoutputFileName.

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.10 outputRandomColumnCSV()

```
void outputRandomColumnCSV (
            std::string outputFileName,
            std::string PKIoutputFileName,
            DelimTextBuffer OutBuff,
            Zipcode zArray[ ],
            int zSize,
            std::string hArray[ ] )
```

outputRandomColumnCSV

Parameters

string	outputFileName
string	PKIoutputFileName
DelimTextBuffer	OutBuff
Zipcode	array zArray
int	zSize
string	array hArray

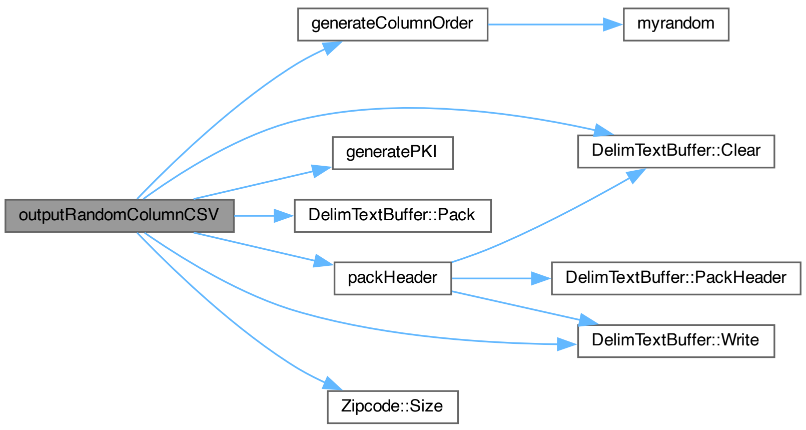
Precondition

a DelimtTextBuffer, and a filled array of zipcodes must exist

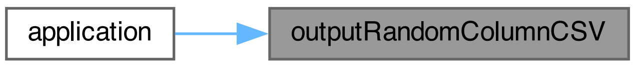
Postcondition

the zipcode array is packed into the buffer randomly and written to the given outputFileName file after the given hArray header record is written. A pki file is generated with the name PKIoutputFileName.

Here is the call graph for this function:



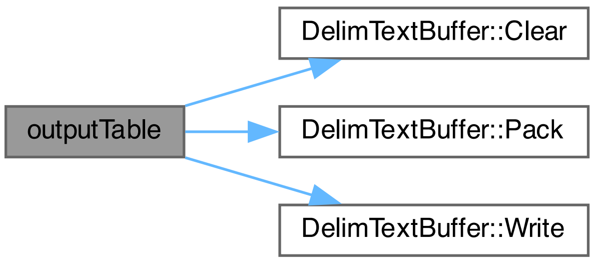
Here is the caller graph for this function:



4.5.2.11 outputTable()

```
void outputTable (
            std::string outputFileName,
            DelimTextBuffer OutBuff,
            State sArray[ ],
            int sArraySize )
```

Here is the call graph for this function:



4.5.2.12 packHeader()

```
int packHeader (
            std::string outputFileName,
            DelimTextBuffer OutBuff,
            std::string hArray[ ] )
```

packHeader

Parameters

string	outputFileName
DelimTextBuffer	OutBuff
string	array hArray

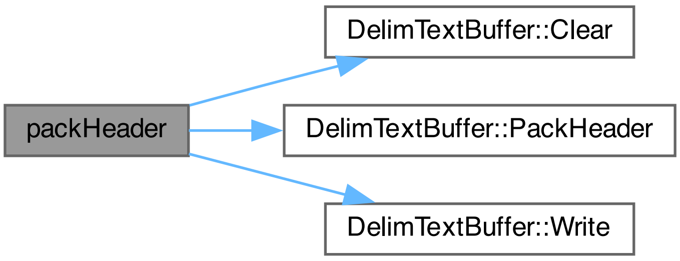
Precondition

a DelimTextBuffer object must exist

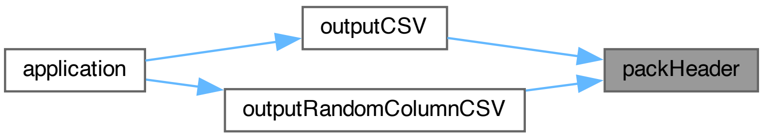
Postcondition

the given hArray is packed with header record from given outputFileName file

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.13 readFileNoHeaderLength()

```
int readFileNoHeaderLength (
            Zipcode zArray[],
            DelimTextBuffer InBuff,
            std::string InFileName )
```

readFileNoHeaderLength

Parameters

Zipcode	array zArray
DelimTextBuffer	InBuff
string	InFileName

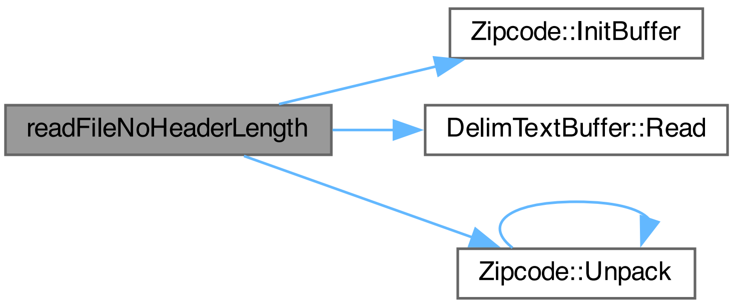
Precondition

a DelimTextBuffer object must exist

Postcondition

the given file InFileName is opened and the file is read into the InBuff buffer. The contents of the buffer are then unpacked into zArray

Here is the call graph for this function:



4.5.2.14 readFileWithHeaderLength()

```
int readFileWithHeaderLength (
            Zipcode zArray[ ],
            DelimTextBuffer InBuff,
            std::string InFileName )
```

readFileWithHeaderLength

Parameters

Zipcode	array zArray
DelimTextBuffer	InBuff
string	InFileName

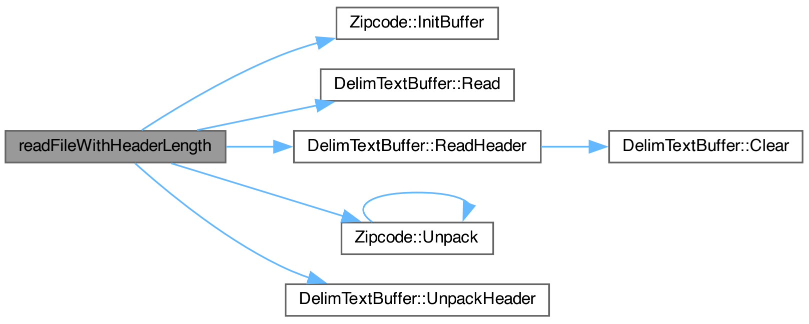
Precondition

a DelimTextBuffer object must exist

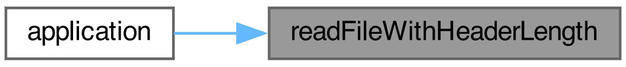
Postcondition

the given file InFileName is opened and the header is unpacked into headerArray then the rest of the file is read into the InBuff buffer. The contents of the buffer are then unpacked into zArray

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.15 readPKI()

```
int readPKI (
             PKIStruct pArray[],
             std::string PKIFileName )
```

readPKI

Parameters

PKIStruct	array pArray
string	PKIFileName

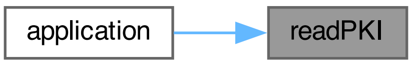
Precondition

none

Postcondition

the given PKIFileName file is opened and the header is read in and the rest of the contents are read and put into pArray

Here is the caller graph for this function:



4.5.2.16 searchPKI()

```
void searchPKI (
            int zIndex,
            int pIndex,
            int argc,
            char ** argv,
            Zipcode zArray[ ],
            PKIStruct pArray[ ] )
```

searchPKI

Parameters

int	zIndex
int	pIndex
int	argc
cstyle	string argv
Zipcode	array zArray
PKIStruct	array pArray

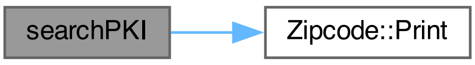
Precondition

a filled Zipcode array and filled PKIStruct array must exist

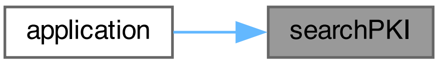
Postcondition

the given pArray is sequntially search for given argv values. If found it outputs a message to the CLI and if not found it outputs a message to the CLI.

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.17 setZipCodes()

```
void setZipCodes (
             State sArray[ ],
             Zipcode zArray[ ],
             int zArraySize )
```

4.5.3 Variable Documentation

4.5.3.1 columnOrderArray

```
std::string columnOrderArray[6]
```

Initial value:

```
={
    "ZipCode",
    "PlaceName",
    "State",
    "County",
    "Lat",
    "Long",
}
```

4.5.3.2 headerArray

```
std::string headerArray[HEADER_FIELDS]
```

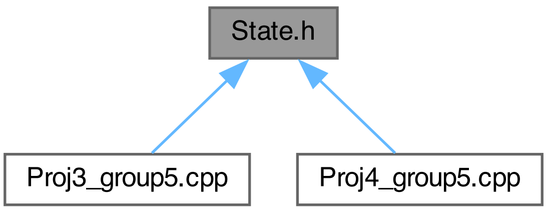
Initial value:

```
= {
    "RecordSize",
    "ZipCode",
    "PlaceName",
    "State",
    "County",
    "Lat",
    "Long",
    "HeaderRecordSize",
    "FileType=CSV",
    "Version=1.0",
    "SizeType=ASCII",
    "IndexFileName",
    "PKISchema=CSV",
    "RecordCount=10",
    "FieldsPerRecord=5",
    "PKIFormat=PKI,Index"
    "EndOfHeaderRecord"
}
```

4.6 State.h File Reference

Declaration file for State class.

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



Classes

- class State

4.6.1 Detailed Description

Declaration file for State class.

Author

Steven Kraus

Emily Yang

Tyler Knudtson

Ashesh Nepal

4.7 State.h

Go to the documentation of this file.

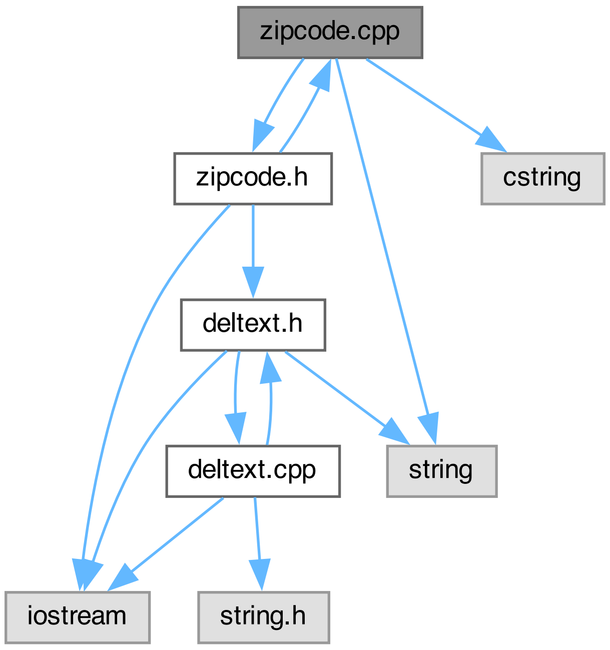
```

#ifndef STATE_H
#define STATE_H

class State
{
    //Data members set as public for easy access
public:
    State();
    char stateName [5]; //State name abbriviation
    char easternZipcode [10]; //Easternmost Zipcode
    char westernZipcode [10]; // Westernmost Zipcode
    char northernZipcode [10]; //Northernmost Zipcode
    char southernZipcode [10]; //Southernmost Zipcode
    char largestLong [10]; //Largest longitude
    char smallestLong [10]; // Smallest longitude
    char largestLat [10]; //Largest Latitude
    char smallestLat [10]; // Smallest Latitude
};


State::State()
{
    // Set each field to an empty string
     stateName[0]  = 0;
     easternZipcode[0]  = 0;
     westernZipcode[0]  = 0;
     northernZipcode[0]  = 0;
     southernZipcode[0]  = 0;
     largestLong[0]  = 0;
     smallestLong[0]  = 0;
     largestLat[0]  = 0;
     smallestLat[0]  = 0;
}
#endif
```

4.8 zipcode.cpp File Reference

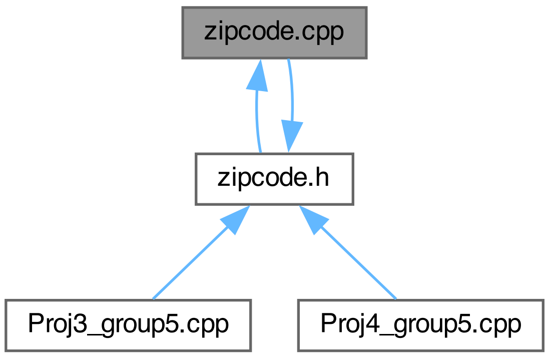
Header file for Zipcode class.

```
#include "zipcode.h"
#include <string>
#include <cstring>
```

Include dependency graph for zipcode.cpp:



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



4.8.1 Detailed Description

Header file for Zipcode class.

Version

0.1

Date

2023-04-10

Author

Steven Kraus

Emily Yang

Tyler Knudtson

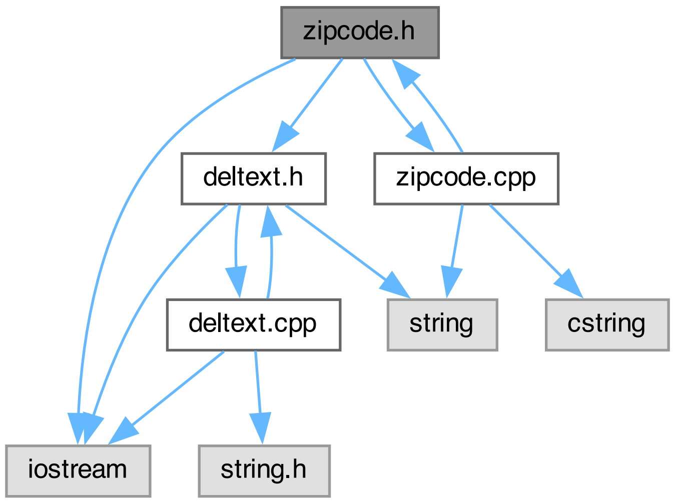
Ashesh Nepal

4.9 zipcode.h File Reference

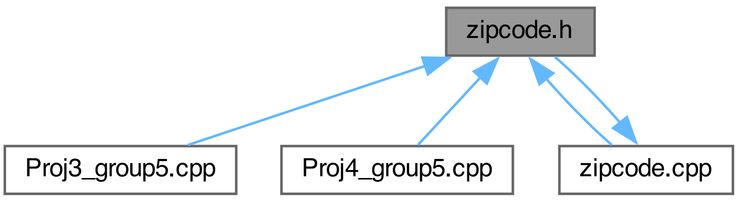
Header file for Zipcode class.

```
#include <iostream>
#include "deltext.h"
#include "zipcode.cpp"
```

Include dependency graph for zipcode.h:



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



Classes

- class Zipcode

4.9.1 Detailed Description

Header file for Zipcode class.

Version

0.1

Date

2023-04-10

Author

Steven Kraus

Emily Yang

Tyler Knudtson

Ashesh Nepal

4.10 zipcode.h

Go to the documentation of this file.

```

#ifndef ZIPCODE_H
#define ZIPCODE_H
#include <iostream>
#include "deltext.h"



class Zipcode
{
  public:
    char Code [10];
    char Placename [30];
    char State [5];
    char County [25];
    char Lat [10];
    char Long [10];

    Zipcode ();

    /* MODIFICATION MEMBER FUNCTIONS*/

    void Clear ();

    static int InitBuffer (DelimTextBuffer &);

    int Unpack (DelimTextBuffer &);

    int Pack (DelimTextBuffer &) const;

    /* NONMODIFICATION MEMBER FUNCTIONS*/

    void Print (std::ostream &);


    int Size();
};

#include "zipcode.cpp"
#endif
```

Index

application
    Proj3_group5.cpp, 29
    Proj4_group5.cpp, 35

Buffer
    DelimTextBuffer, 13
BufferSize
    DelimTextBuffer, 13
byteOffset
    PKIStruct, 15

Clear
    DelimTextBuffer, 7
    Zipcode, 19
Code
    Zipcode, 23
columnOrderArray
    Proj4_group5.cpp, 46
compareStates
    Proj3_group5.cpp, 30
    Proj4_group5.cpp, 36
constructStateArray
    Proj3_group5.cpp, 31
    Proj4_group5.cpp, 36
count
    DelimTextBuffer, 13
County
    Zipcode, 23

Delim
    DelimTextBuffer, 13
DelimStr
    DelimTextBuffer, 13
DelimTextBuffer, 5
    Buffer, 13
    BufferSize, 13
    Clear, 7
    count, 13
    Delim, 13
    DelimStr, 13
    DelimTextBuffer, 6
    Init, 7
    MaxBytes, 14
    NextByte, 14
    Pack, 8
    PackHeader, 8
    Print, 9
    Rbuffer, 14
    Read, 9
    ReadHeader, 10
    Unpack, 11
    UnpackHeader, 11
    Write, 12
    WriteHeader, 12
deltext.cpp, 25
deltext.h, 26
    FALSE, 27
    TRUE, 27

easternZipcode
    State, 16

FALSE
    deltext.h, 27

generateColumnOrder
    Proj4_group5.cpp, 36
generatePKI
    Proj4_group5.cpp, 37
generatePKIHeader
    Proj4_group5.cpp, 37

headerArray
    Proj4_group5.cpp, 46

Init
    DelimTextBuffer, 7
InitBuffer
    Zipcode, 19

largestLat
    State, 16
largestLong
    State, 16
Lat
    Zipcode, 23
Long
    Zipcode, 24

main
    Proj3_group5.cpp, 31
    Proj4_group5.cpp, 37
MaxBytes
    DelimTextBuffer, 14
myrandom
    Proj4_group5.cpp, 38

NextByte
    DelimTextBuffer, 14
northernZipcode
    State, 16

outputCSV
Proj4_group5.cpp, 38
outputRandomColumnCSV
Proj4_group5.cpp, 39
outputTable
Proj3_group5.cpp, 32
Proj4_group5.cpp, 41

Pack
DelimTextBuffer, 8
Zipcode, 20
PackHeader
DelimTextBuffer, 8
packHeader
Proj4_group5.cpp, 41
PKIStruct, 14
byteOffset, 15
zipcode, 15
Placename
Zipcode, 24
Print
DelimTextBuffer, 9
Zipcode, 21
Proj3_group5.cpp, 28
application, 29
compareStates, 30
constructStateArray, 31
main, 31
outputTable, 32
setZipCodes, 33
Proj4_group5.cpp, 33
application, 35
columnOrderArray, 46
compareStates, 36
constructStateArray, 36
generateColumnOrder, 36
generatePKI, 37
generatePKIHeader, 37
headerArray, 46
main, 37
myrandom, 38
outputCSV, 38
outputRandomColumnCSV, 39
outputTable, 41
packHeader, 41
readFileNoHeaderLength, 42
readFileWithHeaderLength, 43
readPKI, 44
searchPKI, 45
setZipCodes, 46

Rbuffer
DelimTextBuffer, 14
Read
DelimTextBuffer, 9
readFileNoHeaderLength
Proj4_group5.cpp, 42
readFileWithHeaderLength
Proj4_group5.cpp, 43
ReadHeader
DelimTextBuffer, 10
readPKI
Proj4_group5.cpp, 44

searchPKI
Proj4_group5.cpp, 45
setZipCodes
Proj3_group5.cpp, 33
Proj4_group5.cpp, 46
Size
Zipcode, 22
smallestLat
State, 17
smallestLong
State, 17
southernZipcode
State, 17
State, 15
easternZipcode, 16
largestLat, 16
largestLong, 16
northernZipcode, 16
smallestLat, 17
smallestLong, 17
southernZipcode, 17
State, 16
stateName, 17
westernZipcode, 17
Zipcode, 24
State.h, 47
stateName
State, 17

TRUE
deltext.h, 27

Unpack
DelimTextBuffer, 11
Zipcode, 22
UnpackHeader
DelimTextBuffer, 11

westernZipcode
State, 17
Write
DelimTextBuffer, 12
WriteHeader
DelimTextBuffer, 12

Zipcode, 18
Clear, 19
Code, 23
County, 23
InitBuffer, 19
Lat, 23
Long, 24
Pack, 20
Placename, 24

    Print, 21
    Size, 22
    State, 24
    Unpack, 22
    Zipcode, 19
zipcode
    PKIStruct, 15
zipcode.cpp, 48
zipcode.h, 50