Project-3

Generated by Doxygen 1.9.6

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

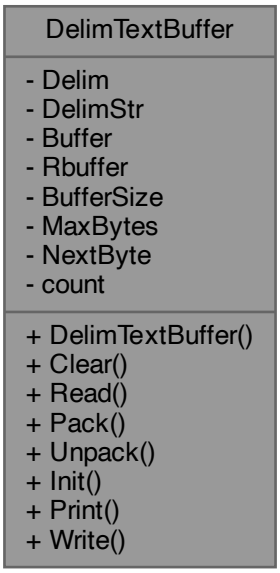
Chapter 3

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 Pack (const char *, int size=-1)
- int Unpack (char *)
- int Init (char delim, int maxBytes=10000)
- void Print (std::ostream &) const
- int Write (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

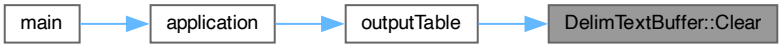
Precondition

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

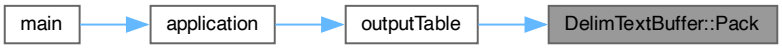
Precondition

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 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

3.1.2.5 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:

3.1.2.6 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

3.1.2.7 Write()

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

Write; Writes the entire buffer to an ostream object

Parameters

ostream	object

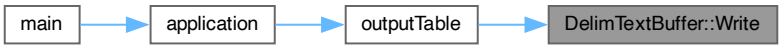
Precondition

a DelimTextBuffer must exist

Postcondition

the entire buffer is written into an ostream object with delimiters

Here is the caller graph for this function:



3.1.3 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 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]

3.2.1 Constructor & Destructor Documentation

3.2.1.1 State()

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

3.2.2 Member Data Documentation

3.2.2.1 easternZipcode

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

3.2.2.2 largestLat

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

3.2.2.3 largestLong

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

3.2.2.4 northernZipcode

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

3.2.2.5 smallestLat

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

3.2.2.6 smallestLong

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

3.2.2.7 southernZipcode

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

3.2.2.8 stateName

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

Data members

3.2.2.9 westernZipcode

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

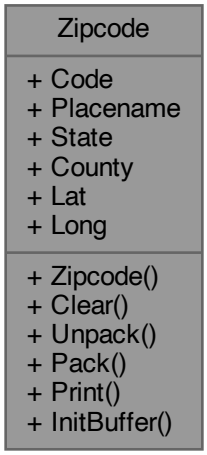
The documentation for this class was generated from the following file:

- State.h

3.3 Zipcode Class Reference

`#include <zipcode.h>`

Collaboration diagram for Zipcode:



Public Member Functions

- Zipcode ()
- void Clear ()
- int Unpack (DelimTextBuffer &)
- int Pack (DelimTextBuffer &) const
- void Print (std::ostream &)

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]

3.3.1 Constructor & Destructor Documentation

3.3.1.1 Zipcode()

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

Constructor

Postcondition

Initializes an empty Zipcode object

Here is the call graph for this function:

3.3.2 Member Function Documentation

3.3.2.1 Clear()

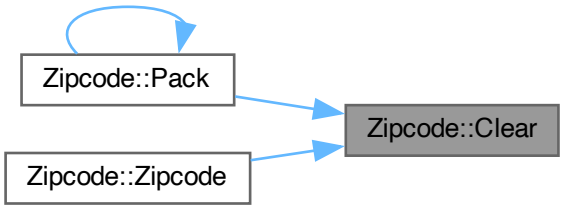
```
Zipcode::Clear ( )
```

Sets all fields to empty strings

Postcondition

All fields set to empty strings

Here is the caller graph for this function:



3.3.2.2 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

3.3.2.3 Pack()

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

Packs the Zipcode object into a DelimTextBuffer object

Parameters

The	DelimTextBuffer to pack

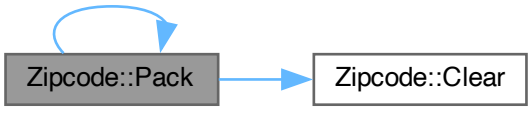
Precondition

DelimTextBuffer must exist and be initialized

Postcondition

the Zipcode object is packed into a DelimTextBuffer

Here is the call graph for this function:



Here is the caller graph for this function:

3.3.2.4 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

3.3.2.5 Unpack()

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

Unpacks DelimTextBuffer into Zipcode object

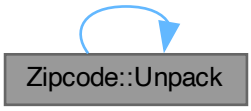
Parameters

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:

3.3.3 Member Data Documentation

3.3.3.1 Code

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

3.3.3.2 County

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

3.3.3.3 Lat

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

3.3.3.4 Long

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

3.3.3.5 Placename

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

3.3.3.6 State

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

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

- zipcode.h
- zipcode.cpp

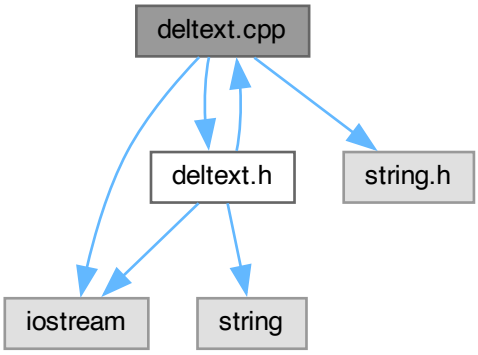
Chapter 4

File Documentation

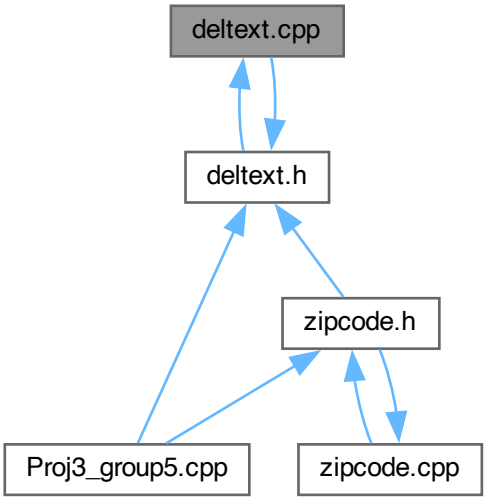
4.1 deltext.cpp File Reference

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

Include dependency graph for deltext.cpp:



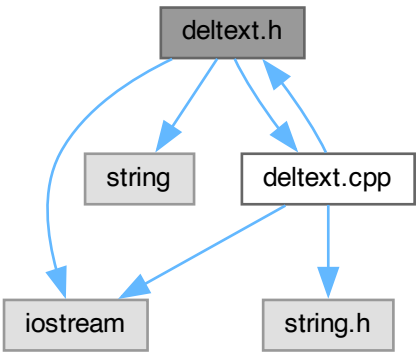
This graph shows which files directly or indirectly include this file:



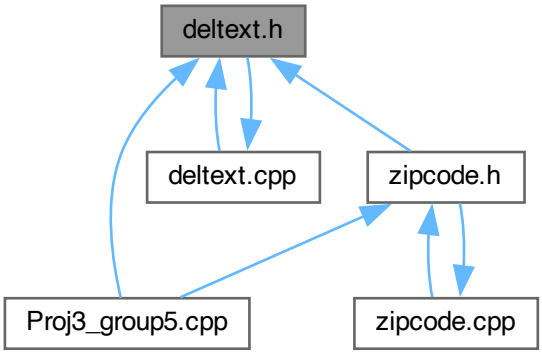
4.2 deltext.h File Reference

```
#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.

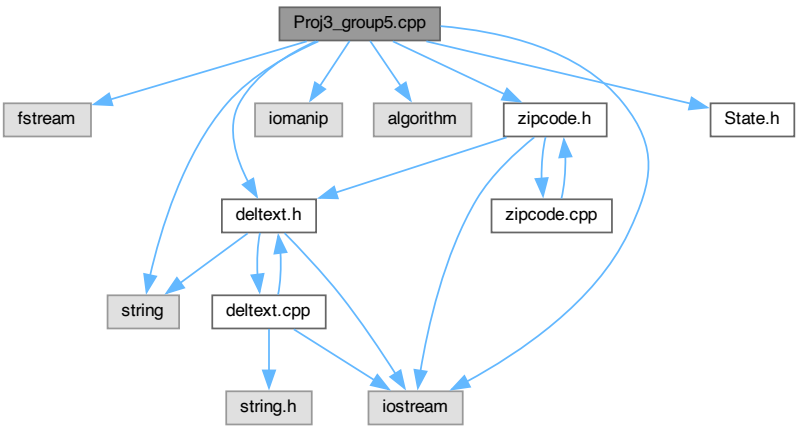
```
00001 // deltext.h
00002 // Modified example from text book
00003
00004 #ifndef DELTEXT_H
00005 #define DELTEXT_H
00006
00007 /*provides ostream and istream*/
00008 #include <iostream>
00009 /*provides use of c++ strings*/
00010 #include <string>
00011
00012 #ifndef FALSE
00013 #define FALSE (0)
00014 #define TRUE (1)
00015 #endif
00016
00017 class DelimTextBuffer
00018 // a buffer which holds delimited text fields.
00019 // Record variables can be packed into and extracted from a buffer.
00020 {  public:
00026     DelimTextBuffer (char Delim = ',', int maxBytes = 10000);
00027
00034     void Clear (); // clear fields from buffer
00035
00042     int Read (std::istream &);
00043
00051     int Pack (const char *, int size = -1);
00052
00059     int Unpack (char *);
00060
00068     int Init (char delim, int maxBytes = 10000);
00069
00078     void Print (std::ostream &) const;
00079
00086     int Write (std::ostream &) const;
00087
00088  private:
00090     char Delim;
00092     char DelimStr[3];
00094     char * Buffer;
00096     std::string Rbuffer;
00098     int BufferSize;
00100     int MaxBytes;
00102     int NextByte;
00104     int count;
00105 };
00106
00107 #include "deltext.cpp"
00108 #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 comparisions.

```
#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 comparisions.

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.

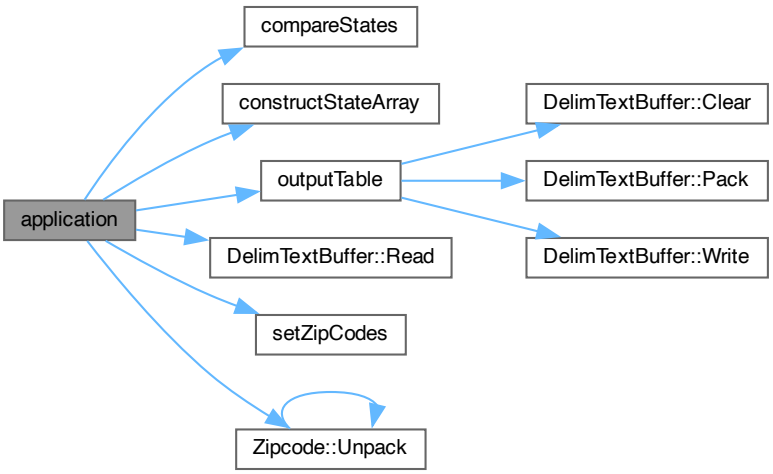
Precondition

specified InFile must be present

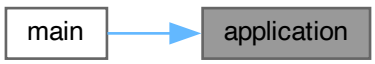
Postcondition

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()

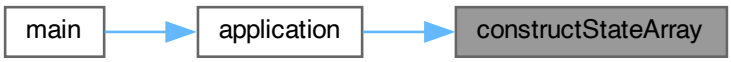
```
bool compareStates (
            State a,
            State b )
```

Here is the caller graph for this function:

4.4.2.3 constructStateArray()

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

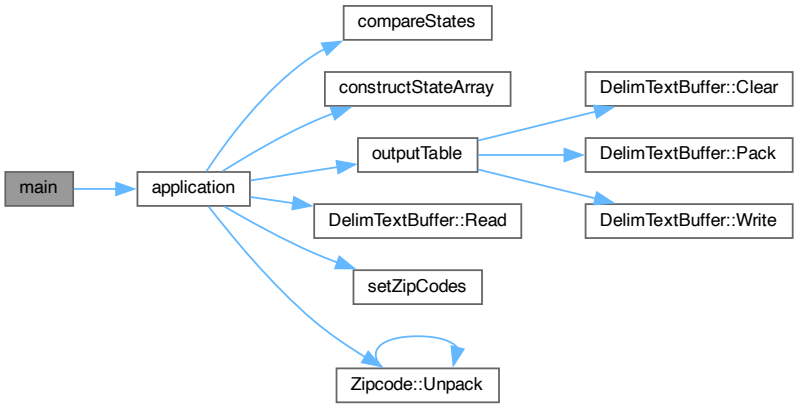
Here is the caller graph for this function:



4.4.2.4 main()

```
main ( )
```

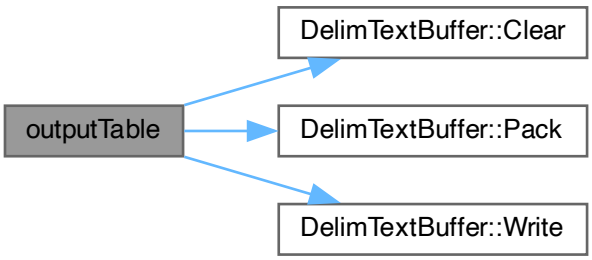
Executes the code present in application() Here is the call graph for this function:



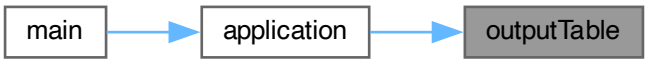
4.4.2.5 outputTable()

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

Here is the call graph for this function:



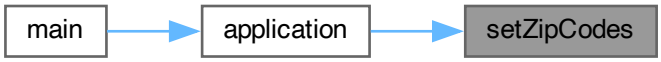
Here is the caller graph for this function:



4.4.2.6 setZipCodes()

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

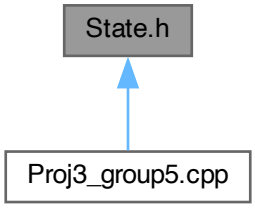
Here is the caller graph for this function:



4.5 State.h File Reference

Declaration file for State class.

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



Classes

- class State

4.5.1 Detailed Description

Declaration file for State class.

Author

Steven Kraus

Emily Yang

Tyler Knudtson

Ashesh Nepal

4.6 State.h

Go to the documentation of this file.

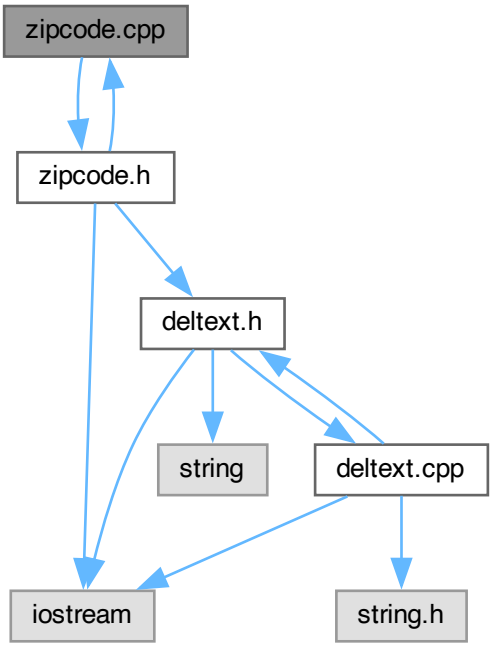
```
00001
00009 #ifndef STATE_H
00010 #define STATE_H
00011
00012 class State
00013 {
00014     //Data members set as public for easy access
00015 public:
00016     State();
00018     char stateName [5]; //State name abbriviation
00019     char easternZipcode [10]; //Easternmost Zipcode
00020     char westernZipcode [10]; // Westernmost Zipcode
00021     char northernZipcode [10]; //Northernmost Zipcode
00022     char southernZipcode [10]; //Southernmost Zipcode
00023     char largestLong [10]; //Largest longitude
00024     char smallestLong [10]; // Smallest longitude
00025     char largestLat [10]; //Largest Latitude
00026     char smallestLat [10]; // Smallest Latitude
00027 };
00028
00029
00030 State::State()
00031 {
00032     // Set each field to an empty string
00033      stateName[0]  = 0;
00034      easternZipcode[0]  = 0;
00035      westernZipcode[0]  = 0;
00036      northernZipcode[0]  = 0;
00037      southernZipcode[0]  = 0;
00038      largestLong[0]  = 0;
00039      smallestLong[0]  = 0;
00040      largestLat[0]  = 0;
00041      smallestLat[0]  = 0;
00042 }
00043 #endif
```

4.7 zipcode.cpp File Reference

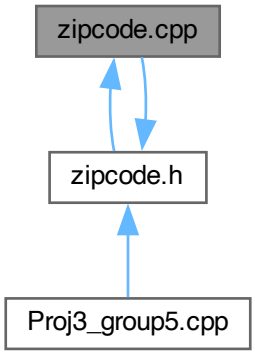
Implementation file for Zipcode class.

```
#include "zipcode.h"
```

Include dependency graph for zipcode.cpp:



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



4.7.1 Detailed Description

Implementation file for Zipcode class.

Author

Steven Kraus

Emily Yang

Tyler Knudtson

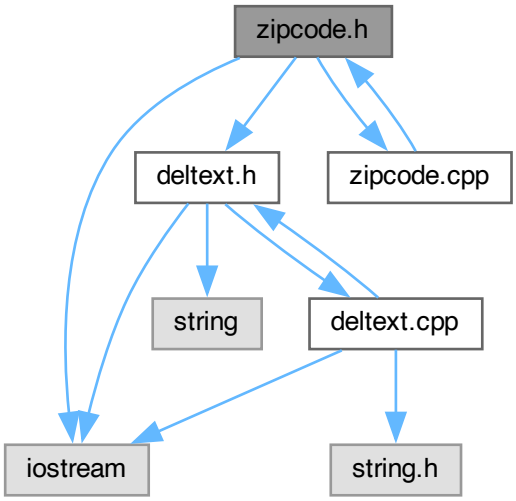
Ashesh Nepal

4.8 zipcode.h File Reference

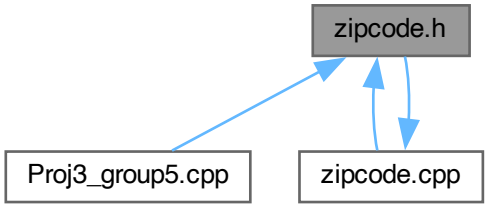
Declaration file for Zipcode class.

```
#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

4.8.1 Detailed Description

Declaration file for Zipcode class.

Author

Steven Kraus

Emily Yang

Tyler Knudtson

Ashesh Nepal

4.9 zipcode.h

Go to the documentation of this file.

```
00001
00010 #ifndef ZIPCODE_H
00011 #define ZIPCODE_H
00012 #include <iostream>
00013 #include "deltext.h"
00014
00015
00016
00024 class Zipcode
00025 {
00026   public:
00027
00028     char Code [10];
00029     char Placename [30];
00030     char State [5];
00031     char County [25];
00032     char Lat [10];
00033     char Long [10];
00034
00039     Zipcode ();
00040
00041     /* MODIFICATION MEMBER FUNCTIONS*/
00042
00047     void Clear ();
00048
00055     static int InitBuffer (DelimTextBuffer &);
00056
00062     int Unpack (DelimTextBuffer &);
00063
00070     int Pack (DelimTextBuffer &) const;
00071
00072     /* NONMODIFICATION MEMBER FUNCTIONS*/
00073
00079     void Print (std::ostream &);
00080 };
00081
00082 #include "zipcode.cpp"
00083 #endif
```

Index