

# GTEC

version 0.1

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	internalTime Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	6
3.1.2	Constructor & Destructor Documentation . . . . .	6
3.1.2.1	internalTime(int Y, int M, int D, int h, int m, int s) . . . . .	6
3.1.2.2	internalTime() . . . . .	6
3.1.3	Member Function Documentation . . . . .	6
3.1.3.1	parse(std::string strtime) . . . . .	6
3.1.3.2	toUNIXTime() . . . . .	7
3.1.4	Member Data Documentation . . . . .	7
3.1.4.1	year . . . . .	7
3.2	navigation Class Reference . . . . .	7
3.2.1	Detailed Description . . . . .	8
3.2.2	Constructor & Destructor Documentation . . . . .	8
3.2.2.1	navigation(std::vector< std::string > fnames) . . . . .	8
3.2.2.2	navigation() . . . . .	8
3.2.3	Member Function Documentation . . . . .	8
3.2.3.1	read() . . . . .	8

3.3	ObsData Class Reference . . . . .	9
3.3.1	Detailed Description . . . . .	11
3.3.2	Constructor & Destructor Documentation . . . . .	11
3.3.2.1	ObsData(std::vector< std::string > fvec, std::string sysString) . . . . .	11
3.3.3	Member Function Documentation . . . . .	12
3.3.3.1	lagrangeInterpolation(float *target, float *s, float *e, int deg) . . . . .	12
3.3.3.2	pre_process(int minArcLen, int intrpollIntrvl, int deg) . . . . .	12
3.3.3.3	read() . . . . .	13
3.3.3.4	setArcStartEnd() . . . . .	13
3.3.3.5	setSysFlags(std::string sysString) . . . . .	13
3.3.4	Member Data Documentation . . . . .	14
3.3.4.1	arcs . . . . .	14
3.3.4.2	arcs2 . . . . .	14
3.3.4.3	arcs3 . . . . .	14
3.3.4.4	BDU_ucTEC . . . . .	14
3.3.4.5	GAL_ucTEC . . . . .	14
3.3.4.6	GLO_ucTEC . . . . .	14
3.3.4.7	GPS_ucTEC . . . . .	14
3.3.4.8	hasTOFO . . . . .	15
3.3.4.9	numArcs . . . . .	15
3.3.4.10	S . . . . .	15
3.3.4.11	S_arcnum . . . . .	15
3.3.4.12	S_prn . . . . .	15
3.4	ptr_pair Class Reference . . . . .	15
3.4.1	Detailed Description . . . . .	16
3.4.2	Constructor & Destructor Documentation . . . . .	16
3.4.2.1	ptr_pair() . . . . .	16
3.4.2.2	ptr_pair(float *s, float *e) . . . . .	16
3.5	triple Class Reference . . . . .	16
3.5.1	Detailed Description . . . . .	17
3.5.2	Member Function Documentation . . . . .	17
3.5.2.1	dump(std::ostream &s) . . . . .	17

<b>4 File Documentation</b>	<b>19</b>
4.1 /home/mowais/GTEC/src/internalTime.hpp File Reference . . . . .	19
4.1.1 Detailed Description . . . . .	20
4.2 /home/mowais/GTEC/src/navigation.hpp File Reference . . . . .	20
4.2.1 Detailed Description . . . . .	20
4.3 /home/mowais/GTEC/src/ObsData.hpp File Reference . . . . .	21
4.3.1 Detailed Description . . . . .	21
4.4 /home/mowais/GTEC/src/ptr_pair.hpp File Reference . . . . .	21
4.4.1 Detailed Description . . . . .	22
4.5 /home/mowais/GTEC/src/triple.hpp File Reference . . . . .	22
4.5.1 Detailed Description . . . . .	23
<b>Index</b>	<b>25</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">internalTime</a>	5
<a href="#">navigation</a>	7
<a href="#">ObsData</a>	9
<a href="#">ptr_pair</a>	15
<a href="#">triple</a>	16





## Chapter 2

# File Index

### 2.1 File List

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

/home/mowais/GTEC/src/ <a href="#">internalTime.hpp</a>	
Class defining internal time format . . . . .	19
/home/mowais/GTEC/src/ <a href="#">navigation.hpp</a>	
This class navigation data . . . . .	20
/home/mowais/GTEC/src/ <a href="#">ObsData.hpp</a>	
Class defining observation data . . . . .	21
/home/mowais/GTEC/src/ <a href="#">ptr_pair.hpp</a>	
Class defining pointer pairs . . . . .	21
/home/mowais/GTEC/src/ <a href="#">triple.hpp</a>	
This class defines a 3-D Coordinate . . . . .	22



## Chapter 3

# Class Documentation

### 3.1 internalTime Class Reference

```
#include <internalTime.hpp>
```

#### Public Member Functions

- `internalTime` (int Y, int M, int D, int h, int m, int s)  
*Constructor with explicit values.*
- void `parse` (std::string strtime)  
*Member function parse.*
- void `parse` (std::string, std::string &)
- void `toUNIXTime` ()  
*Member Function, providing UNIX time.*
- void `toUNIXTime` (int)

#### Public Attributes

- int `year`
- int `month`  
*Stores Month as Integer.*
- int `day`  
*Stores day as Integer.*
- int `hour`  
*Stores hour as Integer.*
- int `minute`  
*Stores minute as Integer.*
- int `second`  
*Stores second as Integer.*
- int `UNIX`  
*Stores Converted UNIX Time as Integer.*

#### Private Member Functions

- `internalTime` ()

### 3.1.1 Detailed Description

#### Author

Muhammad Owais

#### Date

04/12/16

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 `internalTime::internalTime ( int Y, int M, int D, int h, int m, int s )`

Constructor with explicit values.

Constructs `internalTime` object explicitly taking date/time values as parameters. Requires 6 integers (YY↔YY,MM,DD,hh,mm,ss).

#### Parameters

<i>Y</i>	year(YYYY), given as integer
<i>M</i>	Month(MM), given as integer
<i>D</i>	Day(DD), given as integer
<i>h</i>	Hour(hh), given as integer
<i>m</i>	Minute(mm), given as integer
<i>s</i>	Second(ss), given as integer

#### 3.1.2.2 `internalTime::internalTime ( )` `[private]`

Default Constructor. Hidden, cannot be used.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 `void internalTime::parse ( std::string strtime )`

Member function parse.

Member function parse sets internal values by parsing a given string representing date/time values.

#### Parameters

<i>strtime</i>	string representing time.
----------------	---------------------------

## 3.1.3.2 void internalTime::toUNIXTime ( )

Member Function, providing UNIX time.

Member function, converting stored time to UNIX time.

## 3.1.4 Member Data Documentation

## 3.1.4.1 int internalTime::year

Stores Year as Integer

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

- /home/mowais/GTEC/src/[internalTime.hpp](#)
- /home/mowais/GTEC/src/internalTime.cpp

## 3.2 navigation Class Reference

```
#include <navigation.hpp>
```

## Public Member Functions

- void [read](#) ()  
*Member function read.*
- [navigation](#) (std::vector< std::string > fnames)  
*Constructor with Input files.*

## Public Attributes

- std::vector< std::string > [fileNames](#)  
*list of file names to read from*
- float [version](#)  
*Stores RINEX version.*
- int [leapSeconds](#)  
*Stores leapSeconds from Navigation files.*
- std::vector< std::vector< ephemerisGE > > [ephemeris\\_G](#)  
*Vector to store objects of type ephemerisGE for GPS.*
- std::vector< std::vector< ephemerisGE > > [ephemeris\\_E](#)  
*Vector to store objects of type ephemerisGE for Galileo.*
- std::vector< std::vector< ephemerisR > > [ephemeris\\_R](#)  
*Vector to store objects of type ephemerisR for GLONASS.*
- std::vector< std::vector< ephemerisGE > > [ephemeris\\_C](#)  
*Vector to store objects of type ephemerisGE for BeiDou.*

## Private Member Functions

- [navigation](#) ()

### 3.2.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 `navigation::navigation ( std::vector< std::string > frames )`

Constructor with Input files.

Constructs navigation object by reading input navigation files defined by frames.

#### Parameters

<i>frames</i>	Vector of Navigation file names.
---------------	----------------------------------

#### 3.2.2.2 `navigation::navigation ( )` [private]

Default Constructor. Hidden, cannot be used.

### 3.2.3 Member Function Documentation

#### 3.2.3.1 `void navigation::read ( )`

Member function read.

Member function read parses input navigation files and constructs internal navigation structure.

Here is the call graph for this function:



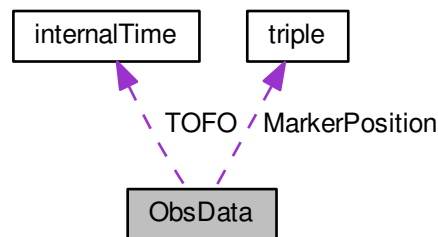
The documentation for this class was generated from the following files:

- [/home/mowais/GTEC/src/navigation.hpp](#)
- [/home/mowais/GTEC/src/navigation.cpp](#)

### 3.3 ObsData Class Reference

```
#include <ObsData.hpp>
```

Collaboration diagram for ObsData:



#### Public Member Functions

- void [read](#) ()  
*Member function read.*
- [ObsData](#) (std::vector< std::string > fvec, std::string sysString)  
*Constructor with Input files, and system string.*
- int **dumpArc** (char, int)
- int **dumpArcByTime** (char, int)
- int **dumpArcBinary** (char, int)
- int **dumpSizes** ()
- void [pre\\_process](#) (int minArcLen, int intrpollIntrvl, int deg)  
*Function to perform preprocessing.*
- void **dumpNonZeroArcs** ()
- int **dumpArcBinaryPtrsAll** ()
- int **dumpArcValuePtrsAll** ()

## Public Attributes

- `std::vector< std::string > fnames`  
*list of file names to read from*
- `triple MarkerPosition`  
*triple Object to store receiver-station position*
- `float version`  
*Stores RINEX version of observation files.*
- `int interval`  
*Interval between observations in data file.*
- `bool hasGPS`  
*Flag to indicate whether Data file contains GPS Data.*
- `bool hasGLO`  
*Flag to indicate whether Data file contains GLONASS Data.*
- `bool hasGAL`  
*Flag to indicate whether Data file contains Galileo Data.*
- `bool hasBEI`  
*Flag to indicate whether Data file contains BeiDou Data.*
- `bool readGPS`  
*Flag to indicate whether to process GPS Data.*
- `bool readGLO`  
*Flag to indicate whether to process GLONASS Data.*
- `bool readGAL`  
*Flag to indicate whether to process Galileo Data.*
- `bool readBEI`  
*Flag to indicate whether to process BeiDou Data.*
- `bool hasTOFO`  
*Time of first observation flag.*
- `std::string TOFO_system`  
*Time system of first observation from observation Header.*
- `internalTime TOFO`  
*internalTime Object to store Time of first observation*
- `std::vector< int > timeline_main`  
*Integer vector to store epochs in UNIX time.*
- `std::vector< std::vector< float > > GPS_ucTEC`  
*Vectors to store raw non-calibrated TEC for GPS Satellites.*
- `std::vector< std::vector< float > > GLO_ucTEC`  
*Vectors to store raw non-calibrated TEC for GLONASS Satellites.*
- `std::vector< std::vector< float > > GAL_ucTEC`  
*Vectors to store raw non-calibrated TEC for Galileo Satellites.*
- `std::vector< std::vector< float > > BDU_ucTEC`  
*Vectors to store raw non-calibrated TEC for BeiDou Satellites.*
- `std::vector< float > S`  
*Stores matrix S (non-calibrated TEC).*
- `std::vector< int > S_arcnum`  
*Stores arc numbers for S.*
- `std::vector< int > S_prn`  
*Stores Satellite IDs for S.*
- `int size_of_S`  
*Indicates size of S.*
- `std::vector< int_pair > intse`



- int [numArcs](#)  
*Indicates total number of arcs.*
- std::vector< int > **prnid**
- int **GPS\_Mark** [32]
- int **GLO\_Mark** [24]
- int **GAL\_Mark** [30]
- int **BDU\_Mark** [34]
- int **NonZero\_Mark** [120]
- int **numNonZeroArcs**
- std::vector< [ptr\\_pair](#) > [arcs](#)  
*Initial non-zero arc pinters.*
- std::vector< [ptr\\_pair](#) > [arcs2](#)  
*Arc pointers without zeros.*
- std::vector< [ptr\\_pair](#) > [arcs3](#)  
*Arc pointers without gaps.*

### Private Member Functions

- [ObsData](#) ()  
*default hidden Constructor*
- void [setSysFlags](#) (std::string sysString)  
*Sets system flags.*
- int **pad\_zero** (int)
- void **resetMark** ()
- int **pad\_zero** ()
- void **markNonZeroArcs** (int, int)
- void **getnumNonZeroArcs** ()
- void [setArcStartEnd](#) ()  
*Sets Arc pointers using [ptr\\_pair](#) objects.*
- int [lagrangeInterpolation](#) (float \*target, float \*s, float \*e, int deg)  
*Function to perform lagrange interpolation.*

### 3.3.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 [ObsData::ObsData](#) ( std::vector< std::string > *fvec*, std::string *sysString* )

Constructor with Input files, and system string.

Constructs observation object by seting input observation file name vector [frames](#) given file names and setting system flags given system string.

## Parameters

<i>fvec</i>	Vector of observation file names.
<i>sysString</i>	string (any combination of 'G','R','E','C') defining constellations being processed.

Here is the call graph for this function:



### 3.3.3 Member Function Documentation

#### 3.3.3.1 int ObsData::lagrangeInterpolation ( float \* *target*, float \* *s*, float \* *e*, int *deg* ) [private]

Function to perform lagrange interpolation.

This function performs lagrange Interpolation needed in preprocessing phase, given a required degree for interpolation.

## Parameters

<i>target</i>	pointer to the value being interpolated.
<i>s</i>	start pointer of the arc.
<i>e</i>	end pointer of the arc.
<i>deg</i>	degree of Interpolation.

#### 3.3.3.2 void ObsData::pre\_process ( int *minArcLen*, int *intrpolIntrvl*, int *deg* )

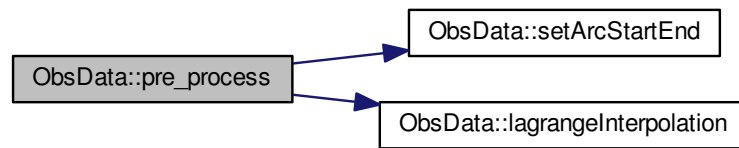
Function to perform preprocessing.

This function performs preprocessing by filling gaps using lagrange interpolation and removing phase jumps using quartiles and Inter Quartile Range.

## Parameters

<i>minArcLen</i>	minimum data duration(Seconds) to consider an arc valid.
<i>intrpolIntrvl</i>	Maximum gap duration (Seconds) to interpolate.
<i>deg</i>	Degree of Interpolation, passed to <a href="#">lagrangeInterpolation</a> .

Here is the call graph for this function:

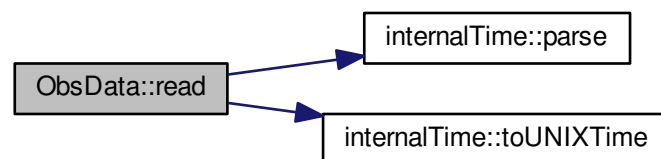


#### 3.3.3.3 void ObsData::read ( )

Member function read.

Member function read parses input observation files and constructs internal observation structure.

Here is the call graph for this function:



#### 3.3.3.4 void ObsData::setArcStartEnd ( ) [private]

Sets Arc pointers using [ptr\\_pair](#) objects.

This function sets Arc pointers to start/end pairs using [ptr\\_pair](#), which serve as input arcs to preprocessing phase.

#### 3.3.3.5 void ObsData::setSysFlags ( std::string sysString ) [private]

Sets system flags.

This function sets system flags given sysString, to indicate which constellations are processed.

##### Parameters

<i>sysString</i>	string (any combination of 'G','R','E','C') indicating constellations being processed.
------------------	--

### 3.3.4 Member Data Documentation

#### 3.3.4.1 `std::vector<ptr_pair> ObsData::arcs`

Initial non-zero arc pinters.

[ptr\\_pair](#) Object containing Initial non-zero arcs, without preprocessing being applied.

#### 3.3.4.2 `std::vector<ptr_pair> ObsData::arcs2`

Arc pointers without zeros.

[ptr\\_pair](#) Object containing arcs, without leading and trailing zeros.

#### 3.3.4.3 `std::vector<ptr_pair> ObsData::arcs3`

Arc pointers without gaps.

[ptr\\_pair](#) Object containing arcs, with gaps removed by [lagrangeInterpolation](#) and phase jumps removed. These are the processed Arcs.

#### 3.3.4.4 `std::vector< std::vector<float> > ObsData::BDU_ucTEC`

Vectors to store raw non-calibrated TEC for BeiDou Satellites.

This is a Vector of float-vectors, where first index is the Satellite prn-id and the second index is the raw non-calibrated TEC for BeiDou satellites corresponding to the epoch index in [timeline\\_main](#).

#### 3.3.4.5 `std::vector< std::vector<float> > ObsData::GAL_ucTEC`

Vectors to store raw non-calibrated TEC for Galileo Satellites.

This is a Vector of float-vectors, where first index is the Satellite prn-id and the second index is the raw non-calibrated TEC for Galileo satellites corresponding to the epoch index in [timeline\\_main](#).

#### 3.3.4.6 `std::vector< std::vector<float> > ObsData::GLO_ucTEC`

Vectors to store raw non-calibrated TEC for GLONASS Satellites.

This is a Vector of float-vectors, where first index is the Satellite prn-id and the second index is the raw non-calibrated TEC for GLONASS satellites corresponding to the epoch index in [timeline\\_main](#).

#### 3.3.4.7 `std::vector< std::vector<float> > ObsData::GPS_ucTEC`

Vectors to store raw non-calibrated TEC for GPS Satellites.

This is a Vector of float-vectors, where first index is the Satellite prn-id and the second index is the raw non-calibrated TEC for GPS satellites corresponding to the epoch index in [timeline\\_main](#).

**3.3.4.8 bool ObsData::hasTOFO**

Time of first observation flag.

Flag to indicate whether Time of first observation was present in observation Header.

**3.3.4.9 int ObsData::numArcs**

Indicates total number of arcs.

Indicates total number of arcs formed. Arc numbers are defined by pre\_processing phase using [pre\\_process](#).

**3.3.4.10 std::vector<float> ObsData::S**

Stores matrix S (non-calibrated TEC).

This vector stores all computed non-calibrated TEC values, arranged by epochs. This is the input vector given to the system solver.

**3.3.4.11 std::vector<int> ObsData::S\_arcnum**

Stores arc numbers for [S](#).

This vector stores for each element in [S](#) , a corresponding value indicating the its arc number. Arc numbers are defined by pre\_processing phase using [pre\\_process](#).

**3.3.4.12 std::vector<int> ObsData::S\_prn**

Stores Satellite IDs for [S](#).

This vector stores for each element in [S](#) , a corresponding value indicating the its Satellite ID.

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

- [/home/mowais/GTEC/src/ObsData.hpp](#)
- [/home/mowais/GTEC/src/ObsData.cpp](#)

**3.4 ptr\_pair Class Reference**

```
#include <ptr_pair.hpp>
```

**Public Member Functions**

- [ptr\\_pair](#) ()  
*Default constructor.*
- [ptr\\_pair](#) (float \*s, float \*e)  
*Custom constructor.*

## Public Attributes

- float \* [start](#)  
*Start pointer.*
- float \* [end](#)  
*End pointer.*

### 3.4.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 `ptr_pair::ptr_pair ( )`

Default constructor.

Default constructor, creates [ptr\\_pair](#) object with NULLL start and end pointers.

#### 3.4.2.2 `ptr_pair::ptr_pair ( float * s, float * e )`

Custom constructor.

Constructor, creates [ptr\\_pair](#) object with start and end pointers set to given pointers.

#### Parameters

<code>s</code>	Input start pointer for new <a href="#">ptr_pair</a> object.
<code>e</code>	Input end pointer for new <a href="#">ptr_pair</a> object.

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

- `/home/mowais/GTEC/src/ptr\_pair.hpp`
- `/home/mowais/GTEC/src/ptr\_pair.cpp`

## 3.5 `triple` Class Reference

```
#include <triple.hpp>
```

## Public Member Functions

- void [dump](#) (std::ostream &s)  
*Member function dump.*

## Public Attributes

- double [X](#)  
*Stores X Coordinate.*
- double [Y](#)  
*Stores Y Coordinate.*
- double [Z](#)  
*Stores Z Coordinate.*

### 3.5.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.5.2 Member Function Documentation

#### 3.5.2.1 void triple::dump ( std::ostream & s )

Member function dump.

Member function dump output coordinates into a given output stream.

#### Parameters

s	output stream
---	---------------

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

- /home/mowais/GTEC/src/[triple.hpp](#)
- /home/mowais/GTEC/src/[triple.cpp](#)





## Chapter 4

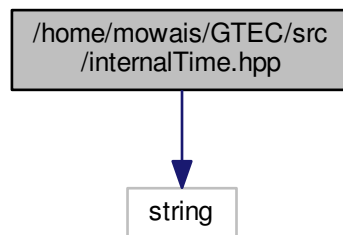
# File Documentation

### 4.1 /home/mowais/GTEC/src/internalTime.hpp File Reference

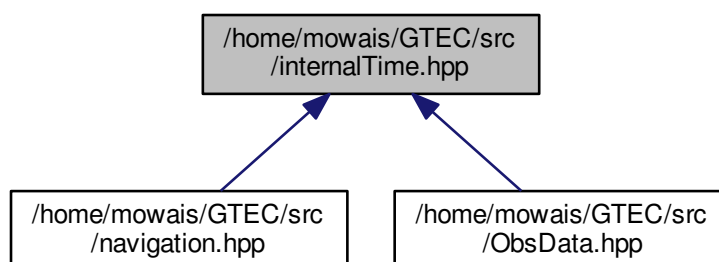
Class defining internal time format.

```
#include <string>
```

Include dependency graph for internalTime.hpp:



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



## Classes

- class [internalTime](#)

### 4.1.1 Detailed Description

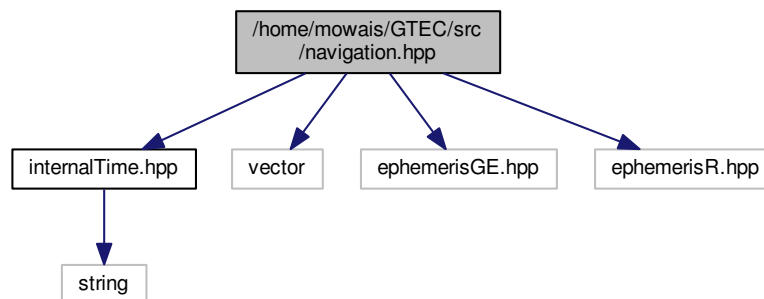
Class defining internal time format.

This Class Defines Internal time which is based on Unix Time. It stores the normal Date/Time as (Year,Month,Day,Hour,Minute,Second), while also providing equivalent UNIX Time. An instance of this class could be generated by explicitly providing normal Date/Time values or by providing a string which would be parse to store time in both formats.

## 4.2 /home/mowais/GTEC/src/navigation.hpp File Reference

This class navigation data.

```
#include "internalTime.hpp"
#include <vector>
#include "ephemerisGE.hpp"
#include "ephemerisR.hpp"
Include dependency graph for navigation.hpp:
```



## Classes

- class [navigation](#)

### 4.2.1 Detailed Description

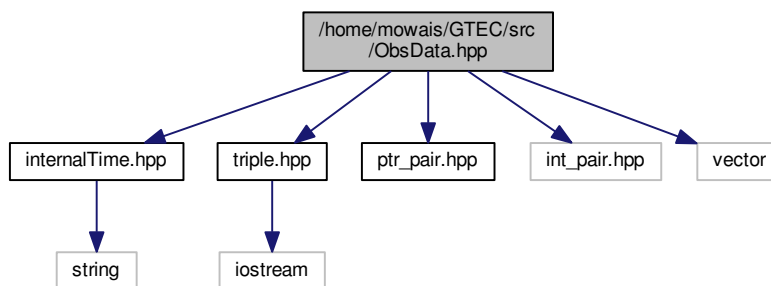
This class navigation data.

This class defines navigation data, stored after reading RINEX navigation files, for different constellations.

### 4.3 /home/mowais/GTEC/src/ObsData.hpp File Reference

Class defining observation data.

```
#include "internalTime.hpp"
#include "triple.hpp"
#include "ptr_pair.hpp"
#include "int_pair.hpp"
#include <vector>
Include dependency graph for ObsData.hpp:
```



#### Classes

- class [ObsData](#)

#### 4.3.1 Detailed Description

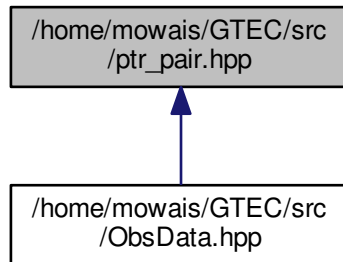
Class defining observation data.

This Class Defines observation data handling, including reading from observation files and storing in internal data structure, the raw non-calibrated TEC from phase observables. This class also includes preprocessing routines being applied to internal data structure, and allot of dump routines for debugging and plotting arc states.

### 4.4 /home/mowais/GTEC/src/ptr\_pair.hpp File Reference

Class defining pointer pairs.

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



## Classes

- class [ptr\\_pair](#)

### 4.4.1 Detailed Description

Class defining pointer pairs.

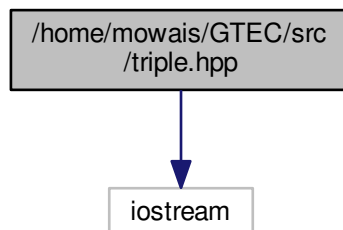
This Class Defines pointer pairs objects used in preprocessing to define arcs. Each arc could be defined as a [ptr\\_pair](#) object having a start pointer (pointer to first value in arc) and an end pointer (pointer to last value in arc).

## 4.5 /home/mowais/GTEC/src/triple.hpp File Reference

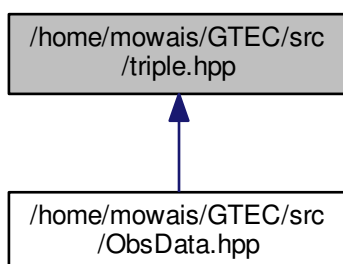
This class defines a 3-D Coordinate.

```
#include <iostream>
```

Include dependency graph for triple.hpp:



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



## Classes

- class [triple](#)

### 4.5.1 Detailed Description

This class defines a 3-D Coordinate.



# Index

/home/mowais/GTEC/src/ObsData.hpp, [21](#)  
/home/mowais/GTEC/src/internalTime.hpp, [19](#)  
/home/mowais/GTEC/src/navigation.hpp, [20](#)  
/home/mowais/GTEC/src/ptr\_pair.hpp, [21](#)  
/home/mowais/GTEC/src/triple.hpp, [22](#)

arcs  
    ObsData, [14](#)  
arcs2  
    ObsData, [14](#)  
arcs3  
    ObsData, [14](#)

BDU\_ucTEC  
    ObsData, [14](#)

dump  
    triple, [17](#)

GAL\_ucTEC  
    ObsData, [14](#)  
GLO\_ucTEC  
    ObsData, [14](#)  
GPS\_ucTEC  
    ObsData, [14](#)

hasTOFO  
    ObsData, [14](#)

internalTime, [5](#)  
    internalTime, [6](#)  
    parse, [6](#)  
    toUNIXTime, [6](#)  
    year, [7](#)

lagrangeInterpolation  
    ObsData, [12](#)

navigation, [7](#)  
    navigation, [8](#)  
    read, [8](#)

numArcs  
    ObsData, [15](#)

ObsData, [9](#)  
    arcs, [14](#)  
    arcs2, [14](#)  
    arcs3, [14](#)  
    BDU\_ucTEC, [14](#)  
    GAL\_ucTEC, [14](#)  
    GLO\_ucTEC, [14](#)

GPS\_ucTEC, [14](#)  
hasTOFO, [14](#)  
lagrangeInterpolation, [12](#)  
numArcs, [15](#)  
ObsData, [11](#)  
pre\_process, [12](#)  
read, [13](#)  
S, [15](#)  
S\_arcnum, [15](#)  
S\_prn, [15](#)  
setArcStartEnd, [13](#)  
setSysFlags, [13](#)

parse  
    internalTime, [6](#)  
pre\_process  
    ObsData, [12](#)  
ptr\_pair, [15](#)  
    ptr\_pair, [16](#)

read  
    navigation, [8](#)  
    ObsData, [13](#)

S  
    ObsData, [15](#)  
S\_arcnum  
    ObsData, [15](#)  
S\_prn  
    ObsData, [15](#)  
setArcStartEnd  
    ObsData, [13](#)  
setSysFlags  
    ObsData, [13](#)

toUNIXTime  
    internalTime, [6](#)  
triple, [16](#)  
    dump, [17](#)

year  
    internalTime, [7](#)