

# 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	ephemerisGE Class Reference . . . . .	5
3.2	ephemerisR Class Reference . . . . .	6
3.3	inout Class Reference . . . . .	6
3.4	int_pair Class Reference . . . . .	7
3.5	internalTime Class Reference . . . . .	7
3.5.1	Detailed Description . . . . .	8
3.5.2	Constructor & Destructor Documentation . . . . .	8
3.5.2.1	internalTime(int Y, int M, int D, int h, int m, int s) . . . . .	8
3.5.2.2	internalTime() . . . . .	8
3.5.3	Member Function Documentation . . . . .	9
3.5.3.1	parse(std::string strtime) . . . . .	9
3.5.3.2	toUNIXTime() . . . . .	9
3.5.4	Member Data Documentation . . . . .	9
3.5.4.1	year . . . . .	9
3.6	navigation Class Reference . . . . .	9
3.6.1	Detailed Description . . . . .	10
3.6.2	Constructor & Destructor Documentation . . . . .	10

3.6.2.1	navigation(std::vector< std::string > fnames)	10
3.6.2.2	navigation()	11
3.6.3	Member Function Documentation	11
3.6.3.1	applyRotations(float &Lk, float &ik, float &uk, float &rk, triple &pos)	11
3.6.3.2	eccAnomaly(float M, float e)	11
3.6.3.3	getPositionGE(ephemerisGE &initial, int t, triple &pos)	11
3.6.3.4	getPositionR(ephemerisR &initialConditions, int h, triple &pos)	12
3.6.3.5	read()	12
3.7	ObsData Class Reference	13
3.7.1	Detailed Description	15
3.7.2	Constructor & Destructor Documentation	15
3.7.2.1	ObsData(std::vector< std::string > fvec, std::string sysString)	15
3.7.3	Member Function Documentation	16
3.7.3.1	lagrangeInterpolation(float *target, float *s, float *e, int deg)	16
3.7.3.2	pre_process(int minArcLen, int intrpollIntrvl, int deg)	16
3.7.3.3	read()	17
3.7.3.4	setArcStartEnd()	17
3.7.3.5	setSysFlags(std::string sysString)	17
3.7.4	Member Data Documentation	18
3.7.4.1	arcs	18
3.7.4.2	arcs2	18
3.7.4.3	arcs3	18
3.7.4.4	BDU_ucTEC	18
3.7.4.5	GAL_ucTEC	18
3.7.4.6	GLO_ucTEC	18
3.7.4.7	GPS_ucTEC	18
3.7.4.8	hasTOFO	19
3.7.4.9	numArcs	19
3.7.4.10	S	19
3.7.4.11	S_arcnum	19
3.7.4.12	S_prn	19
3.8	ptr_pair Class Reference	19
3.8.1	Detailed Description	20
3.8.2	Constructor & Destructor Documentation	20
3.8.2.1	ptr_pair()	20
3.8.2.2	ptr_pair(float *s, float *e)	20
3.9	triple Class Reference	20
3.9.1	Detailed Description	21
3.9.2	Member Function Documentation	21
3.9.2.1	dump(std::ostream &s)	21

<b>4 File Documentation</b>	<b>23</b>
4.1 /home/mowais/GTEC/src/internalTime.hpp File Reference . . . . .	23
4.1.1 Detailed Description . . . . .	24
4.2 /home/mowais/GTEC/src/navigation.hpp File Reference . . . . .	24
4.2.1 Detailed Description . . . . .	24
4.3 /home/mowais/GTEC/src/ObsData.hpp File Reference . . . . .	25
4.3.1 Detailed Description . . . . .	25
4.4 /home/mowais/GTEC/src/ptr_pair.hpp File Reference . . . . .	25
4.4.1 Detailed Description . . . . .	26
4.5 /home/mowais/GTEC/src/triple.hpp File Reference . . . . .	26
4.5.1 Detailed Description . . . . .	27
<b>Index</b>	<b>29</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">ephemerisGE</a>	5
<a href="#">ephemerisR</a>	6
<a href="#">inout</a>	6
<a href="#">int_pair</a>	7
<a href="#">internalTime</a>	7
<a href="#">navigation</a>	9
<a href="#">ObsData</a>	13
<a href="#">ptr_pair</a>	19
<a href="#">triple</a>	20





## Chapter 2

# File Index

### 2.1 File List

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

/home/mowais/GTEC/src/ <b>constants.hpp</b>	??
/home/mowais/GTEC/src/ <b>ephemerisGE.hpp</b>	??
/home/mowais/GTEC/src/ <b>ephemerisR.hpp</b>	??
/home/mowais/GTEC/src/ <b>inout.hpp</b>	??
/home/mowais/GTEC/src/ <b>int_pair.hpp</b>	??
/home/mowais/GTEC/src/ <a href="#">internalTime.hpp</a>	
Class defining internal time format	23
/home/mowais/GTEC/src/ <a href="#">navigation.hpp</a>	
This class navigation data	24
/home/mowais/GTEC/src/ <a href="#">ObsData.hpp</a>	
Class defining observation data	25
/home/mowais/GTEC/src/ <a href="#">ptr_pair.hpp</a>	
Class defining pointer pairs	25
/home/mowais/GTEC/src/ <a href="#">triple.hpp</a>	
This class defines a 3-D Coordinate	26



## Chapter 3

# Class Documentation

### 3.1 ephemerisGE Class Reference

#### Public Attributes

- int **Toc**
- int **Toe**
- int **week**
- float **Ahalf**
- float **e**
- float **M0**
- float **w**
- float **i0**
- float **Omega0**
- float **deltan**
- float **idot**
- float **Omegadot**
- float **Cuc**
- float **Cus**
- float **Crc**
- float **Crs**
- float **Cic**
- float **Cis**

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

- /home/mowais/GTEC/src/ephemerisGE.hpp

## 3.2 ephemerisR Class Reference

### Public Attributes

- int **tb**
- float **px**
- float **py**
- float **pz**
- float **vx**
- float **vy**
- float **vz**
- float **xdd**
- float **ydd**
- float **zdd**

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

- /home/mowais/GTEC/src/ephemerisR.hpp

## 3.3 inout Class Reference

### Public Member Functions

- void **process\_Inputs** (int ac, char \*args[ ])
- void **dump** (std::ostream &s)

### Public Attributes

- int **nobsfiles**
- int **nnavfiles**
- bool **systemGPS**
- bool **systemGlonass**
- bool **systemGalileo**
- bool **systemBeidou**
- bool **systemQZSS**
- int **numDays**
- std::string **inputDirectory**
- std::string **satSys**
- std::string **marker**
- int **samplingTime**
- int **firstDayOfYear**
- int **year**
- int **minArcLen**
- int **interpIntrvl**
- int **deg**
- std::vector< std::string > **obsfiles**
- std::vector< std::string > **navfiles**

### Private Member Functions

- void **checkInputFiles** ()

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

- /home/mowais/GTEC/src/inout.hpp
- /home/mowais/GTEC/src/inout.cpp

## 3.4 int\_pair Class Reference

### Public Member Functions

- **int\_pair** (int, int)

### Public Attributes

- int **start**
- int **end**

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

- /home/mowais/GTEC/src/int\_pair.hpp
- /home/mowais/GTEC/src/int\_pair.cpp

## 3.5 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)
- **internalTime** ()

## 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.*

### 3.5.1 Detailed Description

#### Author

Muhammad Owais

#### Date

04/12/16

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 `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

#### 3.5.2.2 `internalTime::internalTime ( )`

Default Constructor.

### 3.5.3 Member Function Documentation

#### 3.5.3.1 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.
----------------	---------------------------

#### 3.5.3.2 void internalTime::toUNIXTime ( )

Member Function, providing UNIX time.

Member function, converting stored time to UNIX time.

### 3.5.4 Member Data Documentation

#### 3.5.4.1 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](#)

## 3.6 navigation Class Reference

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

### Public Member Functions

- void [read](#) ()  
*Member function read.*
- [navigation](#) (std::vector< std::string > frames)  
*Constructor with Input files.*
- void [getPositionR](#) ([ephemerisR](#) &initialConditions, int h, [triple](#) &pos)  
*Function to compute GLONASS satellite positions.*
- void [getPositionGE](#) ([ephemerisGE](#) &initial, int t, [triple](#) &pos)  
*Function to compute GPS/Galileo/BeiDou satellite positions.*

## Public Attributes

- `std::vector< std::string > fileNamees`  
*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 ()`
- `float eccAnomaly (float M, float e)`  
*Function to compute eccentricity anomaly *Ek*.*
- `void applyRotations (float &Lk, float &ik, float &uk, float &rk, triple &pos)`  
*This Function apply rotations around *uk*, *ik* and *Lk*.*

### 3.6.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 `navigation::navigation ( std::vector< std::string > fnames )`

Constructor with Input files.

Constructs navigation object by reading input navigation files defined by *fnames*.

#### Parameters

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



### 3.6.2.2 navigation::navigation ( ) [private]

Default Constructor. Hidden, cannot be used.

## 3.6.3 Member Function Documentation

### 3.6.3.1 void navigation::applyRotations ( float & *Lk*, float & *ik*, float & *uk*, float & *rk*, triple & *pos* ) [private]

This Function apply rotations around *uk*, *ik* and *Lk*.

This Function apply rotations around *uk*, *ik* and *Lk*, Rotation ==  $\begin{bmatrix} X_k \\ rk \\ Y_k \\ 0 \end{bmatrix} = R3(-Lk)R1(-ik)R3(-uk) \begin{bmatrix} 0 \\ 0 \\ Z_k \\ 0 \end{bmatrix}$  Where *R1* and *R3* are the rotation matrices defined at: [http://www.navipedia.net/index.php/Transformation\\_between\\_Terrestrial\\_Frames](http://www.navipedia.net/index.php/Transformation_between_Terrestrial_Frames) By Hernández-Pajares, Technical University of Catalonia, Spain.

#### Parameters

<i>Lk</i>	Longitude of the ascending node LAMBD <sub>Ak</sub> .
<i>ik</i>	Inclination of the orbital plane.
<i>uk</i>	Argument of latitude.
<i>rk</i>	Radial distance <i>rk</i> .
<i>pos</i>	triple object returned with computed coordinates.

### 3.6.3.2 float navigation::eccAnomaly ( float *M*, float *e* ) [private]

Function to compute eccentricity anomaly *E<sub>k</sub>*.

This Function computes eccentricity anomaly *E<sub>k</sub>* by Solving (iteratively) the Kepler equation for the eccentricity anomaly, using Newton–Raphson method, Equation  $\rightarrow M_k = E_k - (e * \sin(E_k))$

#### Parameters

<i>M</i>	mean anomaly for reference time <i>t<sub>k</sub></i> .
<i>e</i>	eccentricity.

### 3.6.3.3 void navigation::getPositionGE ( ephemerisGE & *initial*, int *t*, triple & *pos* )

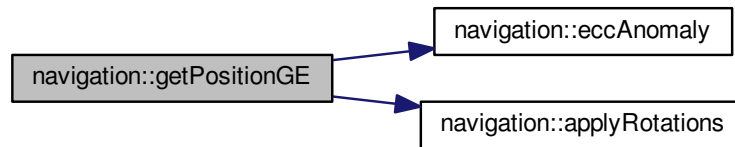
Function to compute GPS/Galileo/BeiDou satellite positions.

This function calculates GPS/Galileo/BeiDou satellite coordinates given an [ephemerisGE](#) object and time for which coordinates are required.

#### Parameters

<i>initial</i>	<a href="#">ephemerisGE</a> object containing initial Keplerian elements.
<i>t</i>	Integer time for which coordinates are to be computed.
<i>pos</i>	triple object returned with computed coordinates.

Here is the call graph for this function:



#### 3.6.3.4 void navigation::getPositionR ( *ephemerisR* & *initialConditions*, int *h*, *triple* & *pos* )

Function to compute GLONASS satellite positions.

This function calculates GLONASS satellite coordinates given an [ephemerisR](#) object, and a step size.

##### Parameters

<i>initialConditions</i>	<a href="#">ephemerisR</a> object containing initial conditions.
<i>h</i>	Integer step size for next coordinate.
<i>pos</i>	triple object returned with computed coordinates.

#### 3.6.3.5 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:



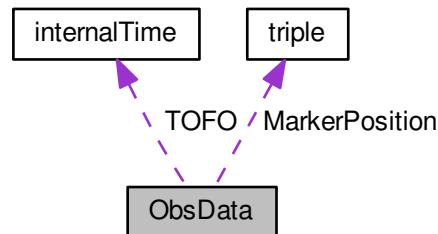
The documentation for this class was generated from the following files:

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

## 3.7 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.*

### 3.7.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 [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:



### 3.7.3 Member Function Documentation

#### 3.7.3.1 `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.

#### 3.7.3.2 `void ObsData::pre_process ( int minArcLen, int intrpolIntrvl, int deg )`

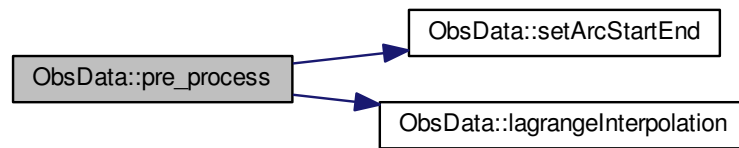
Function to perform preprocessing.

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:

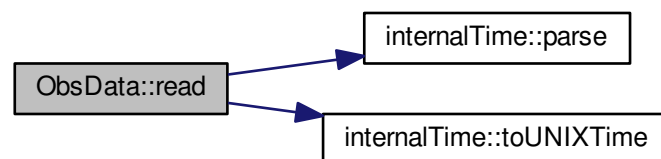


### 3.7.3.3 void ObsData::read ( )

Member function read.

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

Here is the call graph for this function:



### 3.7.3.4 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.

### 3.7.3.5 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.
------------------	--

### 3.7.4 Member Data Documentation

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

Initial non-zero arc pinters.

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

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

Arc pointers without zeros.

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

#### 3.7.4.3 `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.

#### 3.7.4.4 `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](#).

#### 3.7.4.5 `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](#).

#### 3.7.4.6 `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](#).

#### 3.7.4.7 `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](#).



**3.7.4.8 bool ObsData::hasTOFO**

Time of first observation flag.

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

**3.7.4.9 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](#).

**3.7.4.10 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.

**3.7.4.11 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](#).

**3.7.4.12 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](#)

**3.8 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.*

### 3.8.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 `ptr_pair::ptr_pair ( )`

Default constructor.

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

#### 3.8.2.2 `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`

## 3.9 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.*

### 3.9.1 Detailed Description

#### Author

Muhammad Owais

#### Date

05/12/16

### 3.9.2 Member Function Documentation

#### 3.9.2.1 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

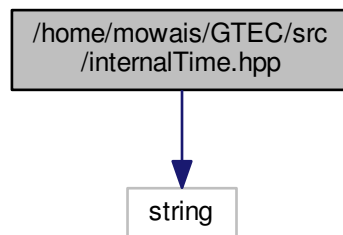
# File Documentation

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

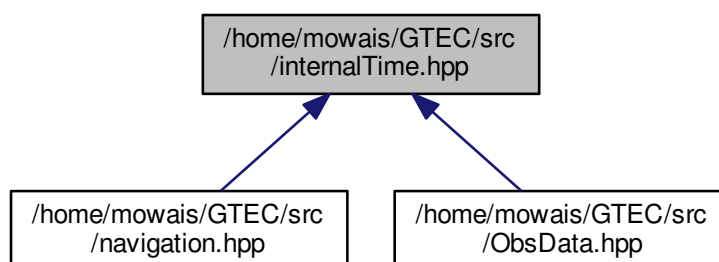
Class defining internal time format.

```
#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

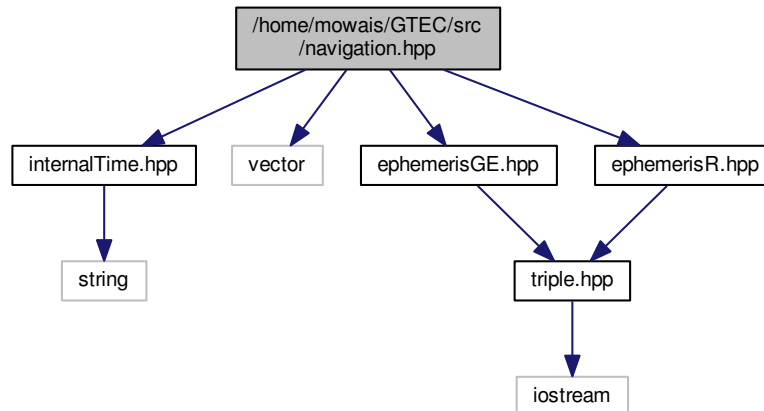
Class defining internal time format.

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

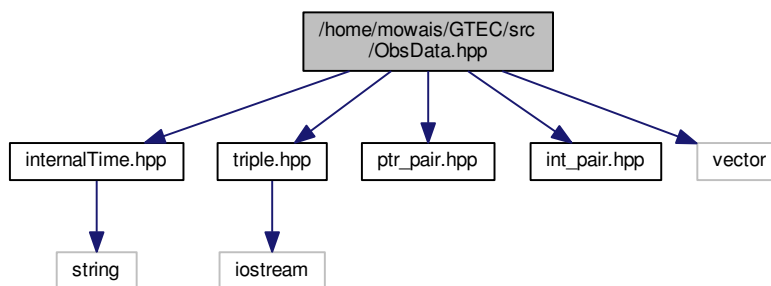
This class navigation data.

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

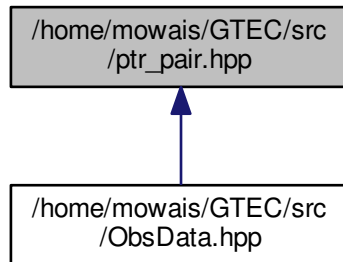
Class defining observation data.

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.

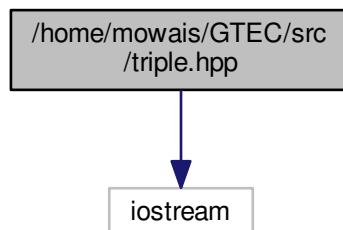
This Class Defines pointer pairs objects used in preprocessing to define arcs. Each arc could be defined as a [ptr\\_pair](#) object having a start pointer (pointer to first value in arc) and an end pointer (pointer to last value in arc).

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

This class defines a 3-D Coordinate.

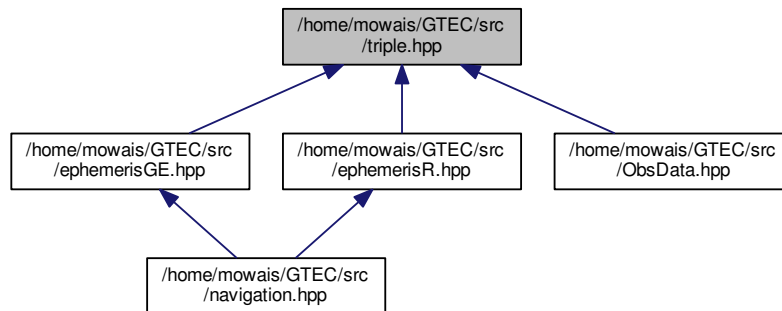
```
#include <iostream>
```

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, [25](#)  
/home/mowais/GTEC/src/internalTime.hpp, [23](#)  
/home/mowais/GTEC/src/navigation.hpp, [24](#)  
/home/mowais/GTEC/src/ptr\_pair.hpp, [25](#)  
/home/mowais/GTEC/src/triple.hpp, [26](#)

applyRotations  
    navigation, [11](#)

arcs  
    ObsData, [18](#)

arcs2  
    ObsData, [18](#)

arcs3  
    ObsData, [18](#)

BDU\_ucTEC  
    ObsData, [18](#)

dump  
    triple, [21](#)

eccAnomaly  
    navigation, [11](#)

ephemerisGE, [5](#)

ephemerisR, [6](#)

GAL\_ucTEC  
    ObsData, [18](#)

GLO\_ucTEC  
    ObsData, [18](#)

GPS\_ucTEC  
    ObsData, [18](#)

getPositionGE  
    navigation, [11](#)

getPositionR  
    navigation, [12](#)

hasTOFO  
    ObsData, [18](#)

inout, [6](#)

int\_pair, [7](#)

internalTime, [7](#)  
    internalTime, [8](#)  
    parse, [9](#)  
    toUNIXTime, [9](#)  
    year, [9](#)

lagrangeInterpolation  
    ObsData, [16](#)

navigation, [9](#)

applyRotations, [11](#)

eccAnomaly, [11](#)

getPositionGE, [11](#)

getPositionR, [12](#)

navigation, [10](#)

read, [12](#)

numArcs  
    ObsData, [19](#)

ObsData, [13](#)  
    arcs, [18](#)  
    arcs2, [18](#)  
    arcs3, [18](#)  
    BDU\_ucTEC, [18](#)  
    GAL\_ucTEC, [18](#)  
    GLO\_ucTEC, [18](#)  
    GPS\_ucTEC, [18](#)  
    hasTOFO, [18](#)  
    lagrangeInterpolation, [16](#)  
    numArcs, [19](#)  
    ObsData, [15](#)  
    pre\_process, [16](#)  
    read, [17](#)  
    S, [19](#)  
    S\_arcnum, [19](#)  
    S\_prn, [19](#)  
    setArcStartEnd, [17](#)  
    setSysFlags, [17](#)

parse  
    internalTime, [9](#)

pre\_process  
    ObsData, [16](#)

ptr\_pair, [19](#)  
    ptr\_pair, [20](#)

read  
    navigation, [12](#)  
    ObsData, [17](#)

S  
    ObsData, [19](#)

S\_arcnum  
    ObsData, [19](#)

S\_prn  
    ObsData, [19](#)

setArcStartEnd  
    ObsData, [17](#)

setSysFlags  
    ObsData, [17](#)

toUNIXTime  
    [internalTime](#), [9](#)  
triple, [20](#)  
    [dump](#), [21](#)  
  
year  
    [internalTime](#), [9](#)