TimeModel

5.0

Generated by Doxygen 1.8.14

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Module Documentation

6.1 Models

Modules

- Environment

6.1.1 Detailed Description

6.2 Environment

Modules

- Time

6.2.1 Detailed Description

6.3 Time

Files

- file class_declarations.hh

  Forward declaration of classes defined in time.hh.
- file time.hh

  JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.
- file time_converter.hh

  The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.
- file time_converter_dyn_tai.hh

  Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.
- file time_converter_dyn_tdb.hh

  Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.
- file time_converter_dyn_ude.hh

  Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.
- file time_converter_std_ude.hh

  Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.
- file time_converter_tai_gps.hh

  Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.
- file time_converter_tai_tdb.hh

  Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.
- file time_converter_tai_tt.hh

  Converts between International Atomic Time and Terrestrial Time.
- file time_converter_tai_ut1.hh

  Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.
- file time_converter_tai_utc.hh

  Converts between International Atomic Time and Coordinated Universal Time.
- file time_converter_ut1_gmst.hh

  Converts between Universal Time and Greenwich Mean Sidereal Time.
- file time_dyn.hh

  Represents the Dynamic Time in the simulation.
- file time_enum.hh

  Contains an enumeration of the formats in which time can be represented.
- file time_gmst.hh

  To represent the clock known as Greenwich Mean Sidereal Time.
- file time_gps.hh

  To represent the time associated with the Global Positioning System.
- file time_links.hh

  Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.
- file time_manager.hh

  To manage the various time representations and the converters between them throughout the simulation.
- file time_manager_init.hh

  To initialize the Time Manager.
- file time_messages.hh

  Define the class TimeMessages, the class that specifies the message IDs used in the Time model.
- file time_met.hh

A type of UDE time that allows for deliberate holds, or pauses.

- file time_standard.hh

  A class that serves as the base for all time representations that are well defined outside the simulation.
- file time_tai.hh

  Represents International Atomic Time.
- file time_tdb.hh

  Represents Barycentric Dynamic Time.
- file time_tt.hh

  Represents Terrestrial Time.
- file time_ude.hh

  Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.
- file time_ut1.hh

  Represents Universal Time.
- file time_utc.hh

  Represents Coordinated Universal Time.
- file time.cc

  JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.
- file time__add_type_update.cc

  Define JeodBaseTime::add_type_update.
- file time_converter.cc

  An abstract class that defines the basic structure of all the methods used by the converter objects.
- file time_converter_dyn_tai.cc

  Converts between International Atomic Time and Dynamic Time.
- file time_converter_dyn_tdb.cc

  Converts between Dynamic Time and Barycentric Dynamic Time.
- file time_converter_dyn_ude.cc

  Converts between Dynamic Time and a time with User-Defined-Epoch.
- file time_converter_std_ude.cc

  Define member functions for class TimeConverter_STD_UDE.
- file time_converter_tai_gps.cc

  Converts between International Atomic Time and the clock associated with the Global Positioning System.
- file time_converter_tai_tdb.cc

  Converts from International Atomic Time to Barycentric Dynamic Time.
- file time_converter_tai_tt.cc

  Converts between International Atomic Time and Terrestrial Time.
- file time_converter_tai_ut1.cc

  Converts between International Atomic Time and Universal Time.
- file time_converter_tai_utc.cc

  Converts between International Atomic Time and Coordinated Universal Time.
- file time_converter_ut1_gmst.cc

  Define member functions for class TimeConverter_UT1_GMST.
- file time_dyn.cc

  Define member functions for Dynamic Time.
- file time_gmst.cc

  Define member functions for Greenwich Mean Sidereal Time.
- file time_gps.cc

  Define member functions for the clock associated with the Global Positioning System.
- file time_manager.cc

  Define member functions for class TimeManager.
- file time_manager__initialize.cc

Define TimeManager::initialize.

- file time_manager_init.cc

  Define member functions for the Time Manager Initialization.
- file time_messages.cc

  Implement the class TimeMessages.
- file time_met.cc

  Define member functions for Mission Elapsed Time.
- file time_standard.cc

  An abstract class, this defines the basic structure of member functions for all Standard Times.
- file time_tai.cc

  Define member functions for International Atomic Time.
- file time_tdb.cc

  Define member functions Barycentric Dynamic Time.
- file time_tt.cc

  Define member functions for Terrestrial Time.
- file time_ude.cc

  Define member functions for those times with a User-Defined-Epoch.
- file time_ut1.cc

  Define member functions for Universal Time.
- file time_utc.cc

  Define member functions for Coordinated Universal Time.

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "environment/time/"

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 PATH

```
#define PATH "environment/time/"
```

Definition at line 37 of file time_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class JeodBaseTime

  JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.
- class TimeConverter

  The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.
- class TimeConverter_Dyn_TAI

  Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.
- class TimeConverter_Dyn_TDB

  Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.
- class TimeConverter_Dyn_UDE

  Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.
- class TimeConverter_STD_UDE

  Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.
- class TimeConverter_TAI_GPS

  Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.
- class TimeConverter_TAI_TDB

  Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.
- class TimeConverter_TAI_TT

  Converts between International Atomic Time and Terrestrial Time.
- class TimeConverter_TAI_UT1

  Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.
- class TimeConverter_TAI_UT1_tai_to_ut1_default_data
- class TimeConverter_TAI_UTC

  Converts between International Atomic Time and Coordinated Universal Time.
- class TimeConverter_TAI_UTC_tai_to_utc_default_data
- class TimeConverter_UT1_GMST

Converts between Universal Time and Greenwich Mean Sidereal Time.

- class TimeDyn

  Represents the Dynamic Time in the simulation.
- class TimeEnum

  Contains an enumeration of the formats in which time can be represented.
- class TimeGMST

  To represent the clock known as Greenwich Mean Sidereal Time.
- class TimeGPS

  To represent the time associated with the Global Positioning System.
- class TimeLinks
- class TimeManager

  To manage the various time representations and the converters between them throughout the simulation.
- class TimeManagerInit

  To initialize the Time Manager.
- class TimeMessages

  Specify the message IDs used in the Time model.
- class TimeMET

  A type of UDE time that allows for deliberate holds, or pauses.
- class TimeStandard

  A class that serves as the base for all time representations that are well defined outside the simulation.
- class TimeTAI

  Represents International Atomic Time.
- class TimeTDB

  Represents Barycentric Dynamic Time.
- class TimeTT

  Represents Terrestrial Time.
- class TimeUDE

  Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.
- class TimeUT1

  Represents Universal Time.
- class TimeUTC

  Represents Coordinated Universal Time.

Functions

- TimeConverter::Direction operator| (TimeConverter::Direction a, TimeConverter::Direction b)

  Bitwise or operator for combining multiple converter direction flags.

7.1.1 Detailed Description

Namespace jeod.

Construct a Time_MET.

Namespace jeod

7.1.2 Function Documentation

7.1.2.1 operator" | ()

```
TimeConverter::Direction jeod::operator| (
            TimeConverter::Direction a,
            TimeConverter::Direction b )  [inline]
```

Bitwise or operator for combining multiple converter direction flags.

Definition at line 206 of file time_converter.hh.

Chapter 8

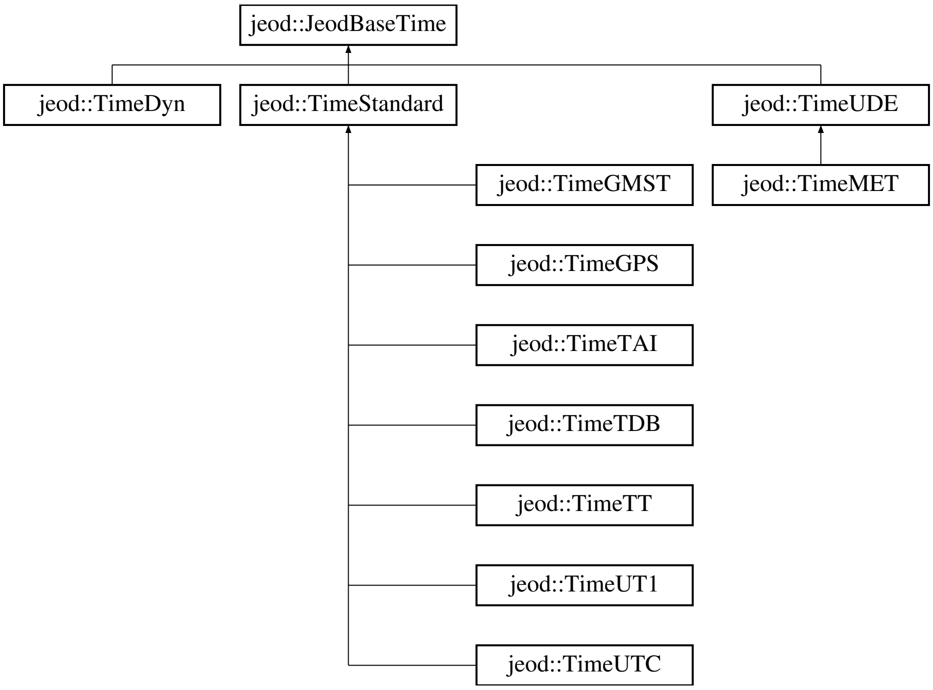
Data Structure Documentation

8.1 jeod::JeodBaseTime Class Reference

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <time.hh>
```

Inheritance diagram for jeod::JeodBaseTime:



Public Member Functions

- JeodBaseTime ()

  Construct a JeodBaseTime.
- virtual ∼JeodBaseTime ()

  Destroy a JeodBaseTime.

- virtual bool must_be_singleton (void)

  Virtual function that indicates if class must be a singleton Defaults to yes.
- virtual void set_time_by_seconds (const double new_seconds)

  Given a value of seconds, propagate to days.
- virtual void set_time_by_days (const double new_days)

  Given a value of days, propagate to seconds.
- void add_type_update (const int seeking_status, TimeManagerInit ∗tm_init)

  Recursively adds elements to the update tree.
- void set_name (std::string name_in)

  Setter for the name.
- void set_index (int idx)

  Setter for the index (force user to be carefule)
- int get_index ()

  Getter for the index.
- void override_initialized (bool init)

  Force reset the initialization status.
- bool is_initialized ()

  Read the initialization status.
- virtual void initialize_initializer_time (TimeManagerInit ∗tm_init)=0

  Initialize the time class that is used for initialization of the simulation.
- virtual void add_type_initialize (const int seeking_status, TimeManagerInit ∗tm_init)

  Default attempt to add a time-type to the initialization tree.
- virtual void initialize_from_parent (TimeManagerInit ∗tm_init)

  Default attempt to initialize a time-type from its parent.
- virtual void update (void)

  Updates each of the derived times from its parent time.

Data Fields

- double initializing_value

  Value used to define sim start time.
- int update_converter_direction

  Determines which converter function (a_to_b (+1) or b_to_a (-1)) to use.
- double seconds

  Elapsed time from epoch.
- NamedItem name

  Name of time-type.
- std::string initialize_from_name

  Name of time-type from which initial value is derived.
- std::string update_from_name

  Name of time-type from which update values are derived.
- TimeManager ∗ time_manager

  Pointer to the TimeManager.
- TimeConverter ∗ update_converter_ptr

  Pointer to the converter class needed to update the time.

Protected Member Functions

- void add_parent (JeodBaseTime &parent)

  Link the argument time as the update source for this time.

Protected Attributes

- int index

  Index-value of time-type in the registry.
- bool initialized

  Whether time has been initialized to a real time.
- double days

  Elapsed time from epoch.
- double initial_value

  Value of "seconds" at the start of the sim.
- double clock_resolution

  The resolution limit when generating clock and calendar-clock values.
- TimeLinks links

  Linkage to the hierarchy of time conversions.

Private Member Functions

- JeodBaseTime (const JeodBaseTime &)
- JeodBaseTime & operator= (const JeodBaseTime &)

Friends

- class InputProcessor
- class TimeConverter
- class TimeManagerInit
- void init_attrjeod__JeodBaseTime ()

8.1.1 Detailed Description

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

Definition at line 94 of file time.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 JeodBaseTime() [1/2]

```
jeod::JeodBaseTime::JeodBaseTime (
            void )
```

Construct a JeodBaseTime.

Definition at line 58 of file time.cc.

References clock_resolution, days, index, initial_value, initialize_from_name, initialized, initializing_value, seconds, time_manager, update_converter_direction, update_converter_ptr, and update_from_name.

8.1.2.2 ∼JeodBaseTime()

```
jeod::JeodBaseTime::∼JeodBaseTime (
            void ) [virtual]
```

Destroy a JeodBaseTime.

Definition at line 213 of file time.cc.

References links.

8.1.2.3 JeodBaseTime() [2/2]

```
jeod::JeodBaseTime::JeodBaseTime (
            const JeodBaseTime & ) [private]
```

8.1.3 Member Function Documentation

8.1.3.1 add_parent()

```
void jeod::JeodBaseTime::add_parent (
            JeodBaseTime & parent ) [protected]
```

Link the argument time as the update source for this time.

Assumptions and Limitations

- The linkage tree is currently implemented as a runtime inspection tool, and does not augment time update functionality.

Parameters

in	*parent*	the time responsible for updating this time.

Definition at line 112 of file time.cc.

References links.

Referenced by add_type_update().

8.1.3.2 add_type_initialize()

```
void jeod::JeodBaseTime::add_type_initialize (
            const int seeking_status,
            TimeManagerInit * time_manager_init )  [virtual]
```

Default attempt to add a time-type to the initialization tree.

Assumptions and Limitations

- Fails for TimeDyn, and has to be overwritten for others.

Parameters

in	*seeking_status*	status-value for auto-seek
in	*time_manager_init*	TM initializer

Reimplemented in jeod::TimeUDE, and jeod::TimeStandard.

Definition at line 91 of file time.cc.

References jeod::TimeMessages::invalid_setup_error.

8.1.3.3 add_type_update()

```
void jeod::JeodBaseTime::add_type_update (
            const int seeking_status,
            TimeManagerInit * time_manager_init )
```

Recursively adds elements to the update tree.

If the "parent" to a time-type is defined, adds the "parent" then returns to adding the "child" type. If the "parent" is not defined it searches for a suitable "parent" from the types already in the tree. If that search is successful, it adds the "child" to the tree, otherwise it returns without change.

Assumptions and Limitations

- None

Parameters

in	*seeking_status*	status-value for auto-seek.
in	*time_manager_init*	The TM initializer.

Definition at line 72 of file time__add_type_update.cc.

References add_parent(), add_type_update(), jeod::TimeManagerInit::get_conv_dir_upd(), jeod::TimeManager↩ Init::get_conv_ptr_index(), jeod::TimeManager::get_converter_ptr(), jeod::TimeManagerInit::get_status(), jeod::↩

TimeManager::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::TimeManagerInit::increment_status(), index, jeod::TimeConverter::initialize(), jeod::TimeMessages::invalid_node, jeod::TimeMessages::invalid_setup_error, jeod::TimeConverter::is_initialized(), links, jeod::TimeMessages::memory_error, name, jeod::TimeManagerInit::num_added_total, jeod::TimeManager::num_types, jeod::TimeManagerInit::set_status(), jeod::TimeManager::time_lookup(), time_manager, update_converter_direction, update_converter_ptr, and update_from_name.

Referenced by add_type_update().

8.1.3.4 get_index()

```
int jeod::JeodBaseTime::get_index ( )  [inline]
```

Getter for the index.

Definition at line 209 of file time.hh.

References index.

8.1.3.5 initialize_from_parent()

```
void jeod::JeodBaseTime::initialize_from_parent (
            TimeManagerInit * time_manager_init )  [virtual]
```

Default attempt to initialize a time-type from its parent.

Assumptions and Limitations

- Fails for TimeDyn, and has to be overwritten for others.

Parameters

in	*time_manager_init*	TM initializer

Reimplemented in jeod::TimeUDE, and jeod::TimeStandard.

Definition at line 125 of file time.cc.

References jeod::TimeMessages::invalid_setup_error.

Referenced by jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeManagerInit::initialize_time_types().

8.1.3.6 initialize_initializer_time()

```
virtual void jeod::JeodBaseTime::initialize_initializer_time (
            TimeManagerInit * tm_init ) [pure virtual]
```

Initialize the time class that is used for initialization of the simulation.

Parameters

tm_init	Time initializer.

Implemented in jeod::TimeUDE, jeod::TimeStandard, and jeod::TimeDyn.

8.1.3.7 is_initialized()

```
bool jeod::JeodBaseTime::is_initialized ( )  [inline]
```

Read the initialization status.

Definition at line 225 of file time.hh.

References initialized.

Referenced by jeod::TimeConverter_TAI_UTC::initialize(), jeod::TimeStandard::initialize_from_parent(), and jeod::TimeUDE::initialize_from_parent().

8.1.3.8 must_be_singleton()

```
bool jeod::JeodBaseTime::must_be_singleton (
            void  )  [virtual]
```

Virtual function that indicates if class must be a singleton Defaults to yes.

Returns

Boolean value

Reimplemented in jeod::TimeUDE.

Definition at line 143 of file time.cc.

8.1.3.9 operator=()

```
JeodBaseTime& jeod::JeodBaseTime::operator= (
            const JeodBaseTime &  )  [private]
```

8.1.3.10 override_initialized()

```
void jeod::JeodBaseTime::override_initialized (
             bool init )  [inline]
```

Force reset the initialization status.

Definition at line 217 of file time.hh.

References initialized.

Referenced by jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeUDE::initialize_from_parent(), and jeod::↩ TimeUDE::initialize_initializer_time().

8.1.3.11 set_index()

```
void jeod::JeodBaseTime::set_index (
             int idx )  [inline]
```

Setter for the index (force user to be carefule)

Definition at line 201 of file time.hh.

References index.

Referenced by jeod::TimeManager::register_time().

8.1.3.12 set_name()

```
void jeod::JeodBaseTime::set_name (
             std::string name_in )  [inline]
```

Setter for the name.

Definition at line 193 of file time.hh.

References name.

8.1.3.13 set_time_by_days()

```
void jeod::JeodBaseTime::set_time_by_days (
             const double new_days )  [virtual]
```

Given a value of days, propagate to seconds.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_days*	new value for days Units: day

Reimplemented in jeod::TimeUDE, jeod::TimeStandard, and jeod::TimeGPS.

Definition at line 202 of file time.cc.

References days, and seconds.

Referenced by jeod::TimeUDE::set_epoch_dyn(), jeod::TimeStandard::set_time_by_days(), and jeod::TimeUDE::set_time_by_days().

8.1.3.14 set_time_by_seconds()

```
void jeod::JeodBaseTime::set_time_by_seconds (
             const double new_seconds )  [virtual]
```

Given a value of seconds, propagate to days.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_seconds*	new value for seconds Units: s

Reimplemented in jeod::TimeUDE, jeod::TimeStandard, and jeod::TimeGPS.

Definition at line 186 of file time.cc.

References days, and seconds.

Referenced by jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::set_epoch_dyn(), jeod::TimeStandard::set_time_by_seconds(), and jeod::TimeUDE::set_time_by_seconds().

8.1.3.15 update()

```
void jeod::JeodBaseTime::update (
             void )  [virtual]
```

Updates each of the derived times from its parent time.

Assumptions and Limitations

- All but TimeDyn must have a parent; this should be defined by the user, or if not, already determined when the update_tree was built

Reimplemented in jeod::TimeDyn, and jeod::TimeMET.

Definition at line 158 of file time.cc.

References jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), jeod::Time↩ Messages::memory_error, name, update_converter_direction, and update_converter_ptr.

Referenced by jeod::TimeMET::update().

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__JeodBaseTime

```
void init_attrjeod__JeodBaseTime ( )  [friend]
```

8.1.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 96 of file time.hh.

8.1.4.3 TimeConverter

```
friend class TimeConverter  [friend]
```

Definition at line 98 of file time.hh.

8.1.4.4 TimeManagerInit

```
friend class TimeManagerInit  [friend]
```

Definition at line 99 of file time.hh.

8.1.5 Field Documentation

8.1.5.1 clock_resolution

```
double jeod::JeodBaseTime::clock_resolution [protected]
```

The resolution limit when generating clock and calendar-clock values.

Used for forcing a "tick-over" to the next minute if seconds comes within this value of 60.trick_units(s)

Definition at line 167 of file time.hh.

Referenced by jeod::TimeStandard::calculate_calendar_values(), jeod::TimeUDE::clock_update(), and JeodBaseTime().

8.1.5.2 days

```
double jeod::JeodBaseTime::days [protected]
```

Elapsed time from epoch.

trick_units(day)

Definition at line 155 of file time.hh.

Referenced by jeod::TimeStandard::convert_from_calendar(), jeod::TimeUT1::get_days(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), JeodBaseTime(), jeod::TimeStandard::seconds_of_year(), jeod::TimeUDE::set_initial_times(), jeod::TimeUDE::set_time_by_clock(), set_time_by_days(), jeod::TimeStandard::set_time_by_days(), jeod::TimeGPS::set_time_by_seconds(), set_time_by_seconds(), jeod::TimeStandard::set_time_by_seconds(), and jeod::TimeStandard::set_time_by_trunc_julian().

8.1.5.3 index

```
int jeod::JeodBaseTime::index [protected]
```

Index-value of time-type in the registry.

trick_units(–)

Definition at line 147 of file time.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), add_type_update(), get_index(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), JeodBaseTime(), set_index(), and jeod::TimeUDE::verify_epoch().

8.1.5.4 initial_value

```
double jeod::JeodBaseTime::initial_value  [protected]
```

Value of "seconds" at the start of the sim.

trick_units(s)

Definition at line 160 of file time.hh.

Referenced by jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), and JeodBaseTime().

8.1.5.5 initialize_from_name

```
std::string jeod::JeodBaseTime::initialize_from_name
```

Name of time-type from which initial value is derived.

trick_units(–)

Definition at line 127 of file time.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), JeodBaseTime(), and jeod::TimeUDE::verify_init().

8.1.5.6 initialized

```
bool jeod::JeodBaseTime::initialized  [protected]
```

Whether time has been initialized to a real time.

trick_units(–)

Definition at line 151 of file time.hh.

Referenced by jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeDyn::initialize_initializer_time(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeManagerInit::initialize_time_types(), is_initialized(), JeodBaseTime(), override_initialized(), and jeod::TimeConverter::verify_setup().

8.1.5.7 initializing_value

```
double jeod::JeodBaseTime::initializing_value
```

Value used to define sim start time.

trick_units(–)

Definition at line 107 of file time.hh.

Referenced by jeod::TimeStandard::initialize_initializer_time(), JeodBaseTime(), and jeod::TimeUDE::set_initial_↩ times().

8.1.5.8 links

```
TimeLinks jeod::JeodBaseTime::links  [protected]
```

Linkage to the hierarchy of time conversions.

Provides accessors to parent, siblings and childrentrick_units(–)

Definition at line 173 of file time.hh.

Referenced by add_parent(), add_type_update(), jeod::TimeDyn::TimeDyn(), and ∼JeodBaseTime().

8.1.5.9 name

```
NamedItem jeod::JeodBaseTime::name
```

Name of time-type.

trick_units(–)

Definition at line 122 of file time.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), add_type_↩ update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::Time↩ Converter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeStandard::initialize_from↩ _parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeU↩ DE::initialize_initializer_time(), jeod::TimeManager::register_time(), jeod::TimeManager::register_time_named(), jeod::TimeUDE::set_epoch_dyn(), jeod::TimeUDE::set_epoch_initializing_value(), jeod::TimeUDE::set_epoch_↩ std(), jeod::TimeUDE::set_epoch_ude(), jeod::TimeUDE::set_initial_times(), set_name(), jeod::TimeDyn::Time↩ Dyn(), jeod::TimeGMST::TimeGMST(), jeod::TimeGPS::TimeGPS(), jeod::TimeMET::TimeMET(), jeod::TimeT↩ AI::TimeTAI(), jeod::TimeTDB::TimeTDB(), jeod::TimeTT::TimeTT(), jeod::TimeUT1::TimeUT1(), jeod::TimeUT↩ C::TimeUTC(), update(), jeod::TimeUDE::verify_epoch(), jeod::TimeUDE::verify_init(), and jeod::TimeConverter↩ ::verify_setup().

8.1.5.10 seconds

```
double jeod::JeodBaseTime::seconds
```

Elapsed time from epoch.

trick_units(s)

Definition at line 117 of file time.hh.

Referenced by jeod::TimeUDE::clock_update(), jeod::TimeConverter_TAI_TT::convert_a_to_b(), jeod::TimeConverter_Dyn_TAI::convert_a_to_b(), jeod::TimeConverter_TAI_GPS::convert_a_to_b(), jeod::TimeConverter_Dyn_TDB::convert_a_to_b(), jeod::TimeConverter_Dyn_UDE::convert_a_to_b(), jeod::TimeConverter_STD_UDE::convert_a_to_b(), jeod::TimeConverter_TAI_TDB::convert_a_to_b(), jeod::TimeConverter_TAI_TT::convert_b_to_a(), jeod::TimeConverter_TAI_GPS::convert_b_to_a(), jeod::TimeConverter_STD_UDE::convert_b_to_a(), jeod::TimeConverter_TAI_TDB::convert_b_to_a(), jeod::TimeStandard::convert_from_calendar(), jeod::TimeManager::get_timestamp_time(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeDyn::initialize_initializer_time(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), JeodBaseTime(), jeod::TimeConverter_Dyn_UDE::reset_a_to_b_offset(), jeod::TimeConverter_STD_UDE::reset_a_to_b_offset(), jeod::TimeStandard::seconds_of_year(), jeod::TimeUDE::set_initial_times(), jeod::TimeUDE::set_time_by_clock(), set_time_by_days(), set_time_by_seconds(), jeod::TimeGPS::set_time_by_trunc_julian(), jeod::TimeStandard::set_time_by_trunc_julian(), jeod::TimeDyn::update(), and jeod::TimeDyn::update_offset().

8.1.5.11 time_manager

```
TimeManager* jeod::JeodBaseTime::time_manager
```

Pointer to the TimeManager.

trick_units(--)

Definition at line 137 of file time.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), add_type_update(), jeod::TimeStandard::calendar_update(), jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeDyn::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), JeodBaseTime(), jeod::TimeManager::register_time(), jeod::TimeStandard::seconds_of_year(), jeod::TimeDyn::update(), jeod::TimeDyn::update_offset(), jeod::TimeUDE::verify_epoch(), jeod::TimeUDE::verify_init(), jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends(), jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends(), and jeod::TimeUDE::verify_update().

8.1.5.12 update_converter_direction

```
int jeod::JeodBaseTime::update_converter_direction
```

Determines which converter function (a_to_b (+1) or b_to_a (-1)) to use.

trick_units(–)

Definition at line 112 of file time.hh.

Referenced by add_type_update(), JeodBaseTime(), and update().

8.1.5.13 update_converter_ptr

```
TimeConverter* jeod::JeodBaseTime::update_converter_ptr
```

Pointer to the converter class needed to update the time.

trick_units(–)

Definition at line 141 of file time.hh.

Referenced by add_type_update(), JeodBaseTime(), jeod::TimeMET::update(), and update().

8.1.5.14 update_from_name

```
std::string jeod::JeodBaseTime::update_from_name
```

Name of time-type from which update values are derived.

trick_units(–)

Definition at line 132 of file time.hh.

Referenced by jeod::TimeUDE::add_type_initialize(), add_type_update(), jeod::TimeUDE::convert_epoch_to↩_update(), jeod::TimeManagerInit::create_init_tree(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE↩::initialize_initializer_time(), JeodBaseTime(), and jeod::TimeUDE::verify_update().

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

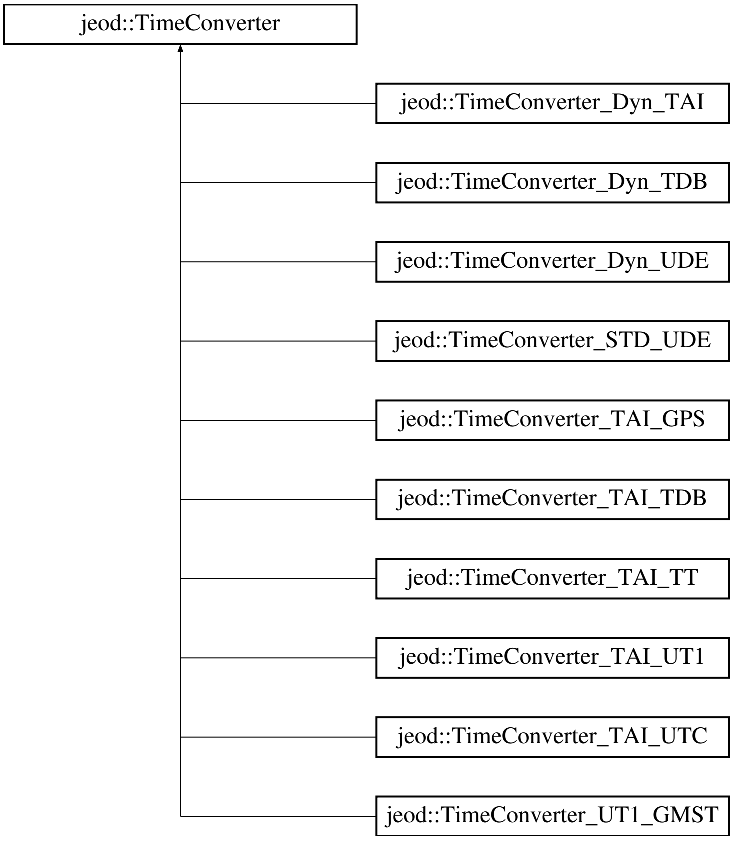
- time.hh
- time.cc
- time__add_type_update.cc

8.2 jeod::TimeConverter Class Reference

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

```
#include <time_converter.hh>
```

Inheritance diagram for jeod::TimeConverter:


jeod::TimeConverter
jeod::TimeConverter_Dyn_TAI
jeod::TimeConverter_Dyn_TDB
jeod::TimeConverter_Dyn_UDE
jeod::TimeConverter_STD_UDE
jeod::TimeConverter_TAI_GPS
jeod::TimeConverter_TAI_TDB
jeod::TimeConverter_TAI_TT
jeod::TimeConverter_TAI_UT1
jeod::TimeConverter_TAI_UTC
jeod::TimeConverter_UT1_GMST

Public Types

- enum Direction {
NO_DIRECTION = 0x0000, A_TO_B_INIT = 0x0001, B_TO_A_INIT = 0x0010, A_TO_B_UPDATE = 0x0100, B_TO_A_UPDATE = 0x1000, A_TO_B = 0x0101, B_TO_A = 0x1010, ANY_DIRECTION = 0x1111 }

  Possible conversion directions.

Public Member Functions

- virtual ∼TimeConverter ()

  Destroy a TimeConverter.
- virtual void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)=0

  Initialize the converter.
- virtual bool is_initialized (void)

  Return internal initialized status bool.
- void override_initialized (bool init)

- bool can_convert (Direction query)

  Check whether this converter is able to handle the requested conversion(s).
- virtual void convert_a_to_b (void)

  Default converter from time 'a' to time 'b'.
- virtual void convert_b_to_a (void)

  Default converter from time 'b' to time 'a'.
- virtual void reset_a_to_b_offset (void)

  Resets the offset between type a and type b mid-sim.
- virtual void verify_table_lookup_ends (void)

  This function does absolutely nothing.
- double get_a_to_b_offset (void)

  Return the offset from the parent time object to this object.

Data Fields

- std::string a_name

  name of time-type "a".
- std::string b_name

  name of time-type "b".

Protected Member Functions

- TimeConverter ()

  Construct a TimeConverter.
- void verify_setup (const JeodBaseTime ∗parent, const JeodBaseTime ∗child, const int direction)

  Verify the setup.

Protected Attributes

- bool initialized

  whether converter has been initialized.
- double a_to_b_offset

  Difference between the two time-types.
- Direction valid_directions

  Bit packed flag specifying whether how a converter can be used.

Private Member Functions

- TimeConverter (const TimeConverter &)
- TimeConverter & operator= (const TimeConverter &)

Friends

- class InputProcessor
- class JeodBaseTime
- void init_attrjeod__TimeConverter ()

8.2.1 Detailed Description

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

Definition at line 89 of file time_converter.hh.

8.2.2 Member Enumeration Documentation

8.2.2.1 Direction

```
enum jeod::TimeConverter::Direction
```

Possible conversion directions.

Enumerator

NO_DIRECTION	
A_TO_B_INIT	
B_TO_A_INIT	
A_TO_B_UPDATE	
B_TO_A_UPDATE	
A_TO_B	
B_TO_A	
ANY_DIRECTION	

Definition at line 99 of file time_converter.hh.

8.2.3 Constructor & Destructor Documentation

8.2.3.1 ∼TimeConverter()

```
jeod::TimeConverter::∼TimeConverter (
                void  )  [virtual]
```

Destroy a TimeConverter.

Definition at line 209 of file time_converter.cc.

8.2.3.2 TimeConverter() [1/2]

```
jeod::TimeConverter::TimeConverter (
            void ) [protected]
```

Construct a TimeConverter.

Definition at line 53 of file time_converter.cc.

References a_name, a_to_b_offset, b_name, initialized, NO_DIRECTION, and valid_directions.

8.2.3.3 TimeConverter() [2/2]

```
jeod::TimeConverter::TimeConverter (
            const TimeConverter & ) [private]
```

8.2.4 Member Function Documentation

8.2.4.1 can_convert()

```
bool jeod::TimeConverter::can_convert (
            Direction query )
```

Check whether this converter is able to handle the requested conversion(s).

If query is compound (e.g. CONV_ALL, CONV_A_TO_B_UPDATE|CONV_B_TO_A_UPDATE) then return true only if capable of all conversions

Returns

whether this converter can do all the conversions

Parameters

in	*query*	converter directions to check

Definition at line 124 of file time_converter.cc.

References NO_DIRECTION, and valid_directions.

8.2.4.2 convert_a_to_b()

```
void jeod::TimeConverter::convert_a_to_b (
            void ) [virtual]
```

Default converter from time 'a' to time 'b'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented in jeod::TimeConverter_TAI_UT1, jeod::TimeConverter_TAI_UTC, jeod::TimeConverter_TAI_TDB, jeod::TimeConverter_STD_UDE, jeod::TimeConverter_Dyn_UDE, jeod::TimeConverter_Dyn_TDB, jeod::TimeConverter_Dyn_TAI, jeod::TimeConverter_TAI_GPS, jeod::TimeConverter_TAI_TT, and jeod::TimeConverter_UT1_GMST.

Definition at line 150 of file time_converter.cc.

References jeod::TimeMessages::invalid_setup_error.

Referenced by jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), and jeod::JeodBaseTime::update().

8.2.4.3 convert_b_to_a()

```
void jeod::TimeConverter::convert_b_to_a (
            void ) [virtual]
```

Default converter from time 'b' to time 'a'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented in jeod::TimeConverter_TAI_UT1, jeod::TimeConverter_TAI_UTC, jeod::TimeConverter_TAI_TDB, jeod::TimeConverter_STD_UDE, jeod::TimeConverter_TAI_GPS, and jeod::TimeConverter_TAI_TT.

Definition at line 166 of file time_converter.cc.

References jeod::TimeMessages::invalid_setup_error.

Referenced by jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), and jeod::JeodBaseTime::update().

8.2.4.4 get_a_to_b_offset()

```
double jeod::TimeConverter::get_a_to_b_offset (
            void ) [inline]
```

Return the offset from the parent time object to this object.

Returns

a_to_b_offset member.

Definition at line 183 of file time_converter.hh.

References a_to_b_offset.

8.2.4.5 initialize()

```
virtual void jeod::TimeConverter::initialize (
            JeodBaseTime * parent,
            JeodBaseTime * child,
            const int direction ) [pure virtual]
```

Initialize the converter.

Parameters

in	*parent*	parent-type
in	*child*	child-type
in	*direction*	L-R, or R-L

Implemented in jeod::TimeConverter_TAI_UT1, jeod::TimeConverter_TAI_UTC, jeod::TimeConverter_TAI_TDB, jeod::TimeConverter_STD_UDE, jeod::TimeConverter_Dyn_UDE, jeod::TimeConverter_Dyn_TDB, jeod::TimeConverter_Dyn_TAI, jeod::TimeConverter_TAI_GPS, jeod::TimeConverter_TAI_TT, and jeod::TimeConverter_UT1_GMST.

Referenced by jeod::JeodBaseTime::add_type_update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeUDE::initialize_initializer_time().

8.2.4.6 is_initialized()

```
bool jeod::TimeConverter::is_initialized (
            void )  [virtual]
```

Return internal initialized status bool.

Definition at line 67 of file time_converter.cc.

References initialized.

Referenced by jeod::JeodBaseTime::add_type_update(), jeod::TimeStandard::initialize_from_parent(), and jeod::TimeUDE::initialize_from_parent().

8.2.4.7 operator=()

```
TimeConverter& jeod::TimeConverter::operator= (
            const TimeConverter &  )  [private]
```

8.2.4.8 override_initialized()

```
void jeod::TimeConverter::override_initialized (
            bool init )  [inline]
```

Definition at line 159 of file time_converter.hh.

References initialized.

Referenced by jeod::TimeUDE::convert_epoch_to_update().

8.2.4.9 reset_a_to_b_offset()

```
void jeod::TimeConverter::reset_a_to_b_offset (
            void ) [virtual]
```

Resets the offset between type a and type b mid-sim.

Reimplemented in jeod::TimeConverter_STD_UDE, and jeod::TimeConverter_Dyn_UDE.

Definition at line 180 of file time_converter.cc.

Referenced by jeod::TimeMET::update().

8.2.4.10 verify_setup()

```
void jeod::TimeConverter::verify_setup (
            const JeodBaseTime * master_ptr,
            const JeodBaseTime * sub_ptr,
            const int int_dir ) [protected]
```

Verify the setup.

Assumptions and Limitations

- None

Parameters

in	*master_ptr*	Time used to initialize the converter
in	*sub_ptr*	Other time-type associated with the converter
in	*int_dir*	+1 a=parent; -1 b=parent; 0 error

Definition at line 82 of file time_converter.cc.

References jeod::TimeMessages::initialization_error, jeod::JeodBaseTime::initialized, jeod::TimeMessages::invalid_setup_error, and jeod::JeodBaseTime::name.

Referenced by jeod::TimeConverter_TAI_TT::initialize(), jeod::TimeConverter_UT1_GMST::initialize(), jeod::TimeConverter_TAI_GPS::initialize(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeConverter_TAI_TDB::initialize(), jeod::TimeConverter_TAI_UTC::initialize(), and jeod::TimeConverter_TAI_UT1::initialize().

8.2.4.11 verify_table_lookup_ends()

```
void jeod::TimeConverter::verify_table_lookup_ends (
            void ) [virtual]
```

This function does absolutely nothing.

It is called when the simulation reverses direction (in time). If the converter uses a table lookup, this function should be replaced in that class. If the converter uses an analytic conversion, no action is needed and this (non)-function should be inherited.

Assumptions and Limitations

- None

Reimplemented in jeod::TimeConverter_TAI_UT1, and jeod::TimeConverter_TAI_UTC.

Definition at line 200 of file time_converter.cc.

8.2.5 Friends And Related Function Documentation

8.2.5.1 init_attrjeod__TimeConverter

```
void init_attrjeod__TimeConverter ( )  [friend]
```

8.2.5.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 91 of file time_converter.hh.

8.2.5.3 JeodBaseTime

```
friend class JeodBaseTime  [friend]
```

Definition at line 93 of file time_converter.hh.

8.2.6 Field Documentation

8.2.6.1 a_name

```
std::string jeod::TimeConverter::a_name
```

name of time-type "a".

trick_units(–)

Definition at line 116 of file time_converter.hh.

Referenced by jeod::TimeManager::register_converter(), TimeConverter(), jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI(), jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB(), jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE(), jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE(), jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS(), jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB(), jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT(), jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1(), jeod::TimeConverter_TAI_UTC::TimeConverter_TAI_UTC(), and jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST().

8.2.6.2 a_to_b_offset

```
double jeod::TimeConverter::a_to_b_offset  [protected]
```

Difference between the two time-types.

trick_units(–)

Definition at line 131 of file time_converter.hh.

Referenced by jeod::TimeConverter_Dyn_TAI::convert_a_to_b(), jeod::TimeConverter_TAI_GPS::convert_a_to_b(), jeod::TimeConverter_Dyn_TDB::convert_a_to_b(), jeod::TimeConverter_Dyn_UDE::convert_a_to_b(), jeod::TimeConverter_STD_UDE::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UT1::convert_a_to_b(), jeod::TimeConverter_TAI_GPS::convert_b_to_a(), jeod::TimeConverter_STD_UDE::convert_b_to_a(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), get_a_to_b_offset(), jeod::TimeConverter_TAI_TT::initialize(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_TAI_GPS::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeConverter_TAI_UTC::initialize(), jeod::TimeConverter_TAI_UT1::initialize(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), jeod::TimeConverter_Dyn_UDE::reset_a_to_b_offset(), jeod::TimeConverter_STD_UDE::reset_a_to_b_offset(), and TimeConverter().

8.2.6.3 b_name

```
std::string jeod::TimeConverter::b_name
```

name of time-type "b".

trick_units(–)

Definition at line 120 of file time_converter.hh.

Referenced by jeod::TimeManager::register_converter(), TimeConverter(), jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI(), jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB(), jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE(), jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE(), jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS(), jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB(), jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT(), jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1(), jeod::TimeConverter_TAI_UTC::TimeConverter_TAI_UTC(), and jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST().

8.2.6.4 initialized

```
bool jeod::TimeConverter::initialized  [protected]
```

whether converter has been initialized.

trick_units(–)

Definition at line 127 of file time_converter.hh.

Referenced by jeod::TimeConverter_TAI_TT::initialize(), jeod::TimeConverter_UT1_GMST::initialize(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_TAI_GPS::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeConverter_TAI_TDB::initialize(), jeod::TimeConverter_TAI_UTC::initialize(), jeod::TimeConverter_TAI_UT1::initialize(), is_initialized(), override_initialized(), and TimeConverter().

8.2.6.5 valid_directions

```
Direction jeod::TimeConverter::valid_directions  [protected]
```

Bit packed flag specifying whether how a converter can be used.

Definition at line 135 of file time_converter.hh.

Referenced by can_convert(), TimeConverter(), jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI(), jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB(), jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE(), jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE(), jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS(), jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB(), jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT(), jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1(), jeod::TimeConverter_TAI_UTC::TimeConverter_TAI_UTC(), and jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST().

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

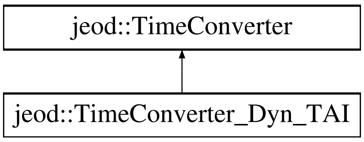
- time_converter.hh
- time_converter.cc

8.3 jeod::TimeConverter_Dyn_TAI Class Reference

Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.

```
#include <time_converter_dyn_tai.hh>
```

Inheritance diagram for jeod::TimeConverter_Dyn_TAI:



Public Member Functions

- TimeConverter_Dyn_TAI ()

  Construct a TimeConverter_Dyn_TAI.
- ∼TimeConverter_Dyn_TAI ()

  Destroy a TimeConverter_Dyn_TAI.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeDyn to TimeTAI.

Private Member Functions

- TimeConverter_Dyn_TAI (const TimeConverter_Dyn_TAI &)
- TimeConverter_Dyn_TAI & operator= (const TimeConverter_Dyn_TAI &)

Private Attributes

- TimeDyn ∗ dyn_ptr

  Converter parent time, always a TimeDyn for this converter.
- TimeTAI ∗ tai_ptr

  Converter child time, always a TimeTAI for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_Dyn_TAI ()

Additional Inherited Members

8.3.1 Detailed Description

Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.

Definition at line 88 of file time_converter_dyn_tai.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 TimeConverter_Dyn_TAI() [1/2]

```
jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI (
             void  )
```

Construct a TimeConverter_Dyn_TAI.

Definition at line 59 of file time_converter_dyn_tai.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::A_TO_B_UPDATE, jeod::TimeConverter::b_name, dyn_ptr, tai_ptr, and jeod::TimeConverter::valid_directions.

8.3.2.2 ∼TimeConverter_Dyn_TAI()

```
jeod::TimeConverter_Dyn_TAI::∼TimeConverter_Dyn_TAI (
             void  )
```

Destroy a TimeConverter_Dyn_TAI.

Definition at line 181 of file time_converter_dyn_tai.cc.

8.3.2.3 TimeConverter_Dyn_TAI() [2/2]

```
jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI (
             const TimeConverter_Dyn_TAI &  )  [private]
```

8.3.3 Member Function Documentation

8.3.3.1 convert_a_to_b()

```
void jeod::TimeConverter_Dyn_TAI::convert_a_to_b (
             void  )  [virtual]
```

Convert from TimeDyn to TimeTAI.

Reimplemented from jeod::TimeConverter.

Definition at line 165 of file time_converter_dyn_tai.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::JeodBaseTime::seconds, jeod::TimeStandard::set_time_by_seconds(), and tai_ptr.

8.3.3.2 initialize()

```
void jeod::TimeConverter_Dyn_TAI::initialize (
             JeodBaseTime ∗ parent_ptr,
             JeodBaseTime ∗ child_ptr,
             const int int_dir )  [virtual]
```

Initialize the converter.

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 77 of file time_converter_dyn_tai.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::TimeMessages::initialization_error, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::JeodBaseTime::seconds, tai_ptr, and jeod::TimeConverter::verify_setup().

8.3.3.3 operator=()

```
TimeConverter_Dyn_TAI& jeod::TimeConverter_Dyn_TAI::operator= (
             const TimeConverter_Dyn_TAI &  )  [private]
```

8.3.4 Friends And Related Function Documentation

8.3.4.1 init_attrjeod__TimeConverter_Dyn_TAI

```
void init_attrjeod__TimeConverter_Dyn_TAI ( )  [friend]
```

8.3.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 91 of file time_converter_dyn_tai.hh.

8.3.5 Field Documentation

8.3.5.1 dyn_ptr

```
TimeDyn* jeod::TimeConverter_Dyn_TAI::dyn_ptr  [private]
```

Converter parent time, always a TimeDyn for this converter.

trick_units(–)

Definition at line 97 of file time_converter_dyn_tai.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_Dyn_TAI().

8.3.5.2 tai_ptr

```
TimeTAI* jeod::TimeConverter_Dyn_TAI::tai_ptr  [private]
```

Converter child time, always a TimeTAI for this converter.

trick_units(–)

Definition at line 102 of file time_converter_dyn_tai.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_Dyn_TAI().

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

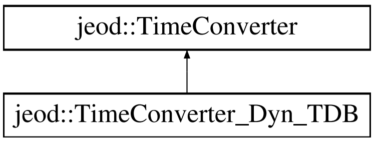
- time_converter_dyn_tai.hh
- time_converter_dyn_tai.cc

8.4 jeod::TimeConverter_Dyn_TDB Class Reference

Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

```
#include <time_converter_dyn_tdb.hh>
```

Inheritance diagram for jeod::TimeConverter_Dyn_TDB:



Public Member Functions

- TimeConverter_Dyn_TDB ()

  Construct a TimeConverter_Dyn_TDB.
- ∼TimeConverter_Dyn_TDB ()

  Destroy a TimeConverter_Dyn_TDB.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeDyn to TimeTDB.

Private Member Functions

- TimeConverter_Dyn_TDB (const TimeConverter_Dyn_TDB &)
- TimeConverter_Dyn_TDB & operator= (const TimeConverter_Dyn_TDB &)

Private Attributes

- TimeDyn ∗ dyn_ptr

  Converter parent time, always a TimeDyn for this converter.
- TimeTDB ∗ tdb_ptr

  Converter child time, always a TimeTDB for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_Dyn_TDB ()

Additional Inherited Members

8.4.1 Detailed Description

Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

Definition at line 91 of file time_converter_dyn_tdb.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 TimeConverter_Dyn_TDB() [1/2]

```
jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB (
             void   )
```

Construct a TimeConverter_Dyn_TDB.

Definition at line 59 of file time_converter_dyn_tdb.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::A_TO_B, jeod::TimeConverter::b_name, dyn_ptr, tdb_ptr, and jeod::TimeConverter::valid_directions.

8.4.2.2 ∼TimeConverter_Dyn_TDB()

```
jeod::TimeConverter_Dyn_TDB::∼TimeConverter_Dyn_TDB (
             void   )
```

Destroy a TimeConverter_Dyn_TDB.

Definition at line 151 of file time_converter_dyn_tdb.cc.

8.4.2.3 TimeConverter_Dyn_TDB() [2/2]

```
jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB (
             const TimeConverter_Dyn_TDB &   )   [private]
```

8.4.3 Member Function Documentation

8.4.3.1 convert_a_to_b()

```
void jeod::TimeConverter_Dyn_TDB::convert_a_to_b (
             void   )  [virtual]
```

Convert from TimeDyn to TimeTDB.

Reimplemented from jeod::TimeConverter.

Definition at line 139 of file time_converter_dyn_tdb.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::JeodBaseTime::seconds, jeod::TimeStandard::set_time_by_seconds(), and tdb_ptr.

8.4.3.2 initialize()

```
void jeod::TimeConverter_Dyn_TDB::initialize (
             JeodBaseTime * parent_ptr,
             JeodBaseTime * child_ptr,
             const int int_dir )  [virtual]
```

Initialize the converter.

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 77 of file time_converter_dyn_tdb.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::TimeMessages::initialization_error, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::seconds, tdb_ptr, and jeod::TimeConverter::verify_setup().

8.4.3.3 operator=()

```
TimeConverter_Dyn_TDB& jeod::TimeConverter_Dyn_TDB::operator= (
             const TimeConverter_Dyn_TDB & )  [private]
```

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__TimeConverter_Dyn_TDB

```
void init_attrjeod__TimeConverter_Dyn_TDB ( )  [friend]
```

8.4.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 93 of file time_converter_dyn_tdb.hh.

8.4.5 Field Documentation

8.4.5.1 dyn_ptr

```
TimeDyn* jeod::TimeConverter_Dyn_TDB::dyn_ptr [private]
```

Converter parent time, always a TimeDyn for this converter.

trick_units(--)

Definition at line 99 of file time_converter_dyn_tdb.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_Dyn_TDB().

8.4.5.2 tdb_ptr

```
TimeTDB* jeod::TimeConverter_Dyn_TDB::tdb_ptr [private]
```

Converter child time, always a TimeTDB for this converter.

trick_units(--)

Definition at line 104 of file time_converter_dyn_tdb.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_Dyn_TDB().

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

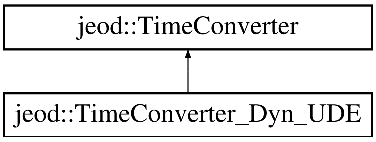
- time_converter_dyn_tdb.hh
- time_converter_dyn_tdb.cc

8.5 jeod::TimeConverter_Dyn_UDE Class Reference

Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

```
#include <time_converter_dyn_ude.hh>
```

Inheritance diagram for jeod::TimeConverter_Dyn_UDE:



Public Member Functions

- TimeConverter_Dyn_UDE ()

  Construct a TimeConverter_Dyn_UDE.
- ∼TimeConverter_Dyn_UDE ()

  Destroy a TimeConverter_Dyn_UDE.
- void reset_a_to_b_offset (void)

  Resets the value of a_to_b_offset.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeDyn to TimeUDE.

Private Member Functions

- TimeConverter_Dyn_UDE (const TimeConverter_Dyn_UDE &)
- TimeConverter_Dyn_UDE & operator= (const TimeConverter_Dyn_UDE &)

Private Attributes

- TimeDyn ∗ dyn_ptr

  Converter parent time, always a TimeDyn for this converter.
- TimeUDE ∗ ude_ptr

  Converter child time, always a TimeUDE for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_Dyn_UDE ()

Additional Inherited Members

8.5.1 Detailed Description

Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

Definition at line 91 of file time_converter_dyn_ude.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 TimeConverter_Dyn_UDE() [1/2]

```
jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE (
            void  )
```

Construct a TimeConverter_Dyn_UDE.

Definition at line 58 of file time_converter_dyn_ude.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::A_TO_B, jeod::TimeConverter::b_name, dyn_ptr, ude_ptr, and jeod::TimeConverter::valid_directions.

8.5.2.2 ∼TimeConverter_Dyn_UDE()

```
jeod::TimeConverter_Dyn_UDE::∼TimeConverter_Dyn_UDE (
            void  )
```

Destroy a TimeConverter_Dyn_UDE.

Definition at line 164 of file time_converter_dyn_ude.cc.

8.5.2.3 TimeConverter_Dyn_UDE() [2/2]

```
jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE (
            const TimeConverter_Dyn_UDE &  )  [private]
```

8.5.3 Member Function Documentation

8.5.3.1 convert_a_to_b()

```
void jeod::TimeConverter_Dyn_UDE::convert_a_to_b (
            void  )  [virtual]
```

Convert from TimeDyn to TimeUDE.

Assumptions and Limitations

- Time class UDE is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from jeod::TimeConverter.

Definition at line 140 of file time_converter_dyn_ude.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::JeodBaseTime::seconds, jeod::TimeUDE::set_↩time_by_seconds(), and ude_ptr.

8.5.3.2 initialize()

```
void jeod::TimeConverter_Dyn_UDE::initialize (
            JeodBaseTime * parent_ptr,
            JeodBaseTime * child_ptr,
            const int int_dir )  [virtual]
```

Initialize the converter.

Assumptions and Limitations

- This class converts from TimeDyn to TimeUDE *only*

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 79 of file time_converter_dyn_ude.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::TimeMessages::incomplete_setup_error, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::JeodBaseTime::seconds, ude_ptr, and jeod::TimeConverter::verify_setup().

8.5.3.3 operator=()

```
TimeConverter_Dyn_UDE& jeod::TimeConverter_Dyn_UDE::operator= (
            const TimeConverter_Dyn_UDE & )  [private]
```

8.5.3.4 reset_a_to_b_offset()

```
void jeod::TimeConverter_Dyn_UDE::reset_a_to_b_offset (
            void )  [virtual]
```

Resets the value of a_to_b_offset.

Reimplemented from jeod::TimeConverter.

Definition at line 154 of file time_converter_dyn_ude.cc.

References jeod::TimeConverter::a_to_b_offset, dyn_ptr, jeod::JeodBaseTime::seconds, and ude_ptr.

8.5.4 Friends And Related Function Documentation

8.5.4.1 init_attrjeod__TimeConverter_Dyn_UDE

```
void init_attrjeod__TimeConverter_Dyn_UDE ( )  [friend]
```

8.5.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 93 of file time_converter_dyn_ude.hh.

8.5.5 Field Documentation

8.5.5.1 dyn_ptr

```
TimeDyn* jeod::TimeConverter_Dyn_UDE::dyn_ptr  [private]
```

Converter parent time, always a TimeDyn for this converter.

trick_units(--)

Definition at line 100 of file time_converter_dyn_ude.hh.

Referenced by convert_a_to_b(), initialize(), reset_a_to_b_offset(), and TimeConverter_Dyn_UDE().

8.5.5.2 ude_ptr

```
TimeUDE* jeod::TimeConverter_Dyn_UDE::ude_ptr  [private]
```

Converter child time, always a TimeUDE for this converter.

trick_units(--)

Definition at line 105 of file time_converter_dyn_ude.hh.

Referenced by convert_a_to_b(), initialize(), reset_a_to_b_offset(), and TimeConverter_Dyn_UDE().

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

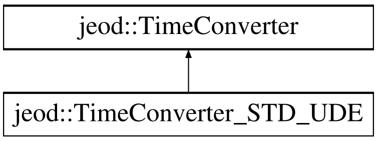
- time_converter_dyn_ude.hh
- time_converter_dyn_ude.cc

8.6 jeod::TimeConverter_STD_UDE Class Reference

Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

```
#include <time_converter_std_ude.hh>
```

Inheritance diagram for jeod::TimeConverter_STD_UDE:



Public Member Functions

- TimeConverter_STD_UDE ()

  Construct a TimeConverter_STD_UDE.
- ∼TimeConverter_STD_UDE ()

  Destroy a TimeConverter_STD_UDE.
- void reset_a_to_b_offset (void)

  Resets the value of a_to_b_offset.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeSTD to TimeUDE.
- void convert_b_to_a (void)

  Convert from TimeUDE to TimeSTD.

Private Member Functions

- TimeConverter_STD_UDE (const TimeConverter_STD_UDE &)
- TimeConverter_STD_UDE & operator= (const TimeConverter_STD_UDE &)

Private Attributes

- bool failed_null_test

  Initializing converter could be done in one of two ways.
- TimeStandard ∗ std_ptr

  Converter parent time, always a TimeSTD for this converter.
- TimeUDE ∗ ude_ptr

  Converter parent time, always a TimeUDE for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_STD_UDE ()

Additional Inherited Members

8.6.1 Detailed Description

Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

Definition at line 91 of file time_converter_std_ude.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 TimeConverter_STD_UDE() [1/2]

```
jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE (
            void  )
```

Construct a TimeConverter_STD_UDE.

Definition at line 58 of file time_converter_std_ude.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::ANY_DIRECTION, jeod::TimeConverter::b_name, failed_null_test, std_ptr, ude_ptr, and jeod::TimeConverter::valid_directions.

8.6.2.2 ∼TimeConverter_STD_UDE()

```
jeod::TimeConverter_STD_UDE::∼TimeConverter_STD_UDE (
            void  )
```

Destroy a TimeConverter_STD_UDE.

Definition at line 190 of file time_converter_std_ude.cc.

8.6.2.3 TimeConverter_STD_UDE() [2/2]

```
jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE (
            const TimeConverter_STD_UDE &  )  [private]
```

8.6.3 Member Function Documentation

8.6.3.1 convert_a_to_b()

```
void jeod::TimeConverter_STD_UDE::convert_a_to_b (
             void  )  [virtual]
```

Convert from TimeSTD to TimeUDE.

Assumptions and Limitations

- Time class UDE is based on time class STD, and counts the elapsed STD time only

Reimplemented from jeod::TimeConverter.

Definition at line 147 of file time_converter_std_ude.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::JeodBaseTime::seconds, jeod::TimeUDE::set_time_by_↩ seconds(), std_ptr, and ude_ptr.

8.6.3.2 convert_b_to_a()

```
void jeod::TimeConverter_STD_UDE::convert_b_to_a (
             void  )  [virtual]
```

Convert from TimeUDE to TimeSTD.

Assumptions and Limitations

- Time class UDE is based on time class STD, and counts the elapsed STD time only

Reimplemented from jeod::TimeConverter.

Definition at line 164 of file time_converter_std_ude.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::JeodBaseTime::seconds, jeod::TimeStandard::set_time_↩ by_seconds(), std_ptr, and ude_ptr.

8.6.3.3 initialize()

```
void jeod::TimeConverter_STD_UDE::initialize (
             JeodBaseTime * parent_ptr,
             JeodBaseTime * child_ptr,
             const int int_dir )  [virtual]
```

Initialize the converter.

Assumptions and Limitations

- This class converts from TimeDyn to TimeUDE *only*

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 80 of file time_converter_std_ude.cc.

References jeod::TimeConverter::a_to_b_offset, failed_null_test, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::JeodBaseTime::seconds, std_ptr, ude_ptr, and jeod::TimeConverter::verify_setup().

8.6.3.4 operator=()

```
TimeConverter_STD_UDE& jeod::TimeConverter_STD_UDE::operator= (
             const TimeConverter_STD_UDE &  )  [private]
```

8.6.3.5 reset_a_to_b_offset()

```
void jeod::TimeConverter_STD_UDE::reset_a_to_b_offset (
             void  )  [virtual]
```

Resets the value of a_to_b_offset.

Reimplemented from jeod::TimeConverter.

Definition at line 177 of file time_converter_std_ude.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::JeodBaseTime::seconds, std_ptr, and ude_ptr.

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__TimeConverter_STD_UDE

```
void init_attrjeod__TimeConverter_STD_UDE ( )  [friend]
```

8.6.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 93 of file time_converter_std_ude.hh.

8.6.5 Field Documentation

8.6.5.1 failed_null_test

```
bool jeod::TimeConverter_STD_UDE::failed_null_test  [private]
```

Initializing converter could be done in one of two ways.

If it fails the first time, this flag is set. If it fails a second time, it terminates.trick_units(–)

Definition at line 101 of file time_converter_std_ude.hh.

Referenced by initialize(), and TimeConverter_STD_UDE().

8.6.5.2 std_ptr

```
TimeStandard* jeod::TimeConverter_STD_UDE::std_ptr  [private]
```

Converter parent time, always a TimeSTD for this converter.

trick_units(–)

Definition at line 105 of file time_converter_std_ude.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), reset_a_to_b_offset(), and TimeConverter_STD_UDE().

8.6.5.3 ude_ptr

```
TimeUDE* jeod::TimeConverter_STD_UDE::ude_ptr  [private]
```

Converter parent time, always a TimeUDE for this converter.

trick_units(–)

Definition at line 110 of file time_converter_std_ude.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), reset_a_to_b_offset(), and TimeConverter_STD_UDE().

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

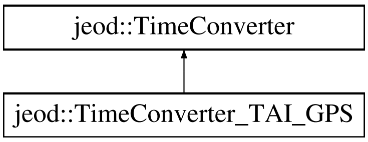
- time_converter_std_ude.hh
- time_converter_std_ude.cc

8.7 jeod::TimeConverter_TAI_GPS Class Reference

Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include <time_converter_tai_gps.hh>
```

Inheritance diagram for jeod::TimeConverter_TAI_GPS:



Public Member Functions

- TimeConverter_TAI_GPS ()

    Construct a TimeConverter_TAI_GPS.
- ∼TimeConverter_TAI_GPS ()

    Destroy a TimeConverter_TAI_GPS.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

    Initialize the converter.
- void convert_a_to_b (void)

    Convert from TimeTAI to TimeGPS.
- void convert_b_to_a (void)

    Convert from TimeGPS to TimeTAI.

Private Member Functions

- TimeConverter_TAI_GPS (const TimeConverter_TAI_GPS &)
- TimeConverter_TAI_GPS & operator= (const TimeConverter_TAI_GPS &)

Private Attributes

- TimeTAI ∗ tai_ptr

    Converter parent time, always a TimeTAI for this converter.
- TimeGPS ∗ gps_ptr

    Converter parent time, always a TimeGPS for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_TAI_GPS ()

Additional Inherited Members

8.7.1 Detailed Description

Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

Definition at line 88 of file time_converter_tai_gps.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 TimeConverter_TAI_GPS() [1/2]

```
jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS (
             void  )
```

Construct a TimeConverter_TAI_GPS.

Definition at line 59 of file time_converter_tai_gps.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::ANY_DIRECTION, jeod::TimeConverter::b_↩name, gps_ptr, tai_ptr, and jeod::TimeConverter::valid_directions.

8.7.2.2 ∼TimeConverter_TAI_GPS()

```
jeod::TimeConverter_TAI_GPS::∼TimeConverter_TAI_GPS (
             void  )
```

Destroy a TimeConverter_TAI_GPS.

Definition at line 137 of file time_converter_tai_gps.cc.

8.7.2.3 TimeConverter_TAI_GPS() [2/2]

```
jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS (
             const TimeConverter_TAI_GPS &  )  [private]
```

8.7.3 Member Function Documentation

8.7.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_GPS::convert_a_to_b (
             void ) [virtual]
```

Convert from TimeTAI to TimeGPS.

Reimplemented from jeod::TimeConverter.

Definition at line 113 of file time_converter_tai_gps.cc.

References jeod::TimeConverter::a_to_b_offset, gps_ptr, jeod::JeodBaseTime::seconds, jeod::TimeGPS::set_↩time_by_seconds(), and tai_ptr.

8.7.3.2 convert_b_to_a()

```
void jeod::TimeConverter_TAI_GPS::convert_b_to_a (
             void ) [virtual]
```

Convert from TimeGPS to TimeTAI.

Reimplemented from jeod::TimeConverter.

Definition at line 125 of file time_converter_tai_gps.cc.

References jeod::TimeConverter::a_to_b_offset, gps_ptr, jeod::JeodBaseTime::seconds, jeod::TimeStandard↩::set_time_by_seconds(), and tai_ptr.

8.7.3.3 initialize()

```
void jeod::TimeConverter_TAI_GPS::initialize (
             JeodBaseTime * parent_ptr,
             JeodBaseTime * child_ptr,
             const int int_dir ) [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 80 of file time_converter_tai_gps.cc.

References jeod::TimeConverter::a_to_b_offset, gps_ptr, jeod::TimeConverter::initialized, jeod::TimeMessages↩ ::invalid_setup_error, tai_ptr, jeod::TimeStandard::tjt_at_epoch, and jeod::TimeConverter::verify_setup().

8.7.3.4 operator=()

```
TimeConverter_TAI_GPS& jeod::TimeConverter_TAI_GPS::operator= (
             const TimeConverter_TAI_GPS &  )  [private]
```

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__TimeConverter_TAI_GPS

```
void init_attrjeod__TimeConverter_TAI_GPS ( )  [friend]
```

8.7.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 90 of file time_converter_tai_gps.hh.

8.7.5 Field Documentation

8.7.5.1 gps_ptr

```
TimeGPS* jeod::TimeConverter_TAI_GPS::gps_ptr  [private]
```

Converter parent time, always a TimeGPS for this converter.

trick_units(–)

Definition at line 102 of file time_converter_tai_gps.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_GPS().

8.7.5.2 tai_ptr

```
TimeTAI* jeod::TimeConverter_TAI_GPS::tai_ptr  [private]
```

Converter parent time, always a TimeTAI for this converter.

trick_units(–)

Definition at line 97 of file time_converter_tai_gps.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_GPS().

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

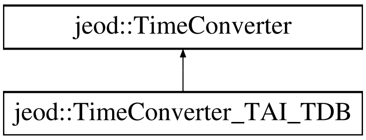
- time_converter_tai_gps.hh
- time_converter_tai_gps.cc

8.8 jeod::TimeConverter_TAI_TDB Class Reference

Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

```
#include <time_converter_tai_tdb.hh>
```

Inheritance diagram for jeod::TimeConverter_TAI_TDB:



Public Member Functions

- TimeConverter_TAI_TDB ()
- ∼TimeConverter_TAI_TDB ()
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void set_a_to_b_offset (void)
- void convert_a_to_b (void)

  Default converter from time 'a' to time 'b'.
- void convert_b_to_a (void)

  Default converter from time 'b' to time 'a'.

Private Member Functions

- TimeConverter_TAI_TDB (const TimeConverter_TAI_TDB &)
- TimeConverter_TAI_TDB & operator= (const TimeConverter_TAI_TDB &)

Private Attributes

- double TAI_to_TT_offset

    The offset from TAI to TT.
- double a_to_b_offset_epoch

    The epoch value of a_to_b_offset.
- double a_to_b_offset

    Calculated value of a_to_b_offset.
- double prev_tai_seconds

    TAI seconds from previous loop iteration.
- double prev_tdb_seconds

    TDB seconds from previous loop iteration.
- int nSteps

    Counter for number of steps in iteration.
- int nIter

    Counter for number of iterations.
- TimeTAI ∗ tai_ptr

    Converter parent time, always a TimeTAI for this converter.
- TimeTDB ∗ tdb_ptr

    Converter parent time, always a TimeTDB for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_TAI_TDB ()

Additional Inherited Members

8.8.1 Detailed Description

Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

Definition at line 90 of file time_converter_tai_tdb.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 TimeConverter_TAI_TDB() [1/2]

```
jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB (
            void  )
```

Definition at line 68 of file time_converter_tai_tdb.cc.

References jeod::TimeConverter::a_name, a_to_b_offset, a_to_b_offset_epoch, jeod::TimeConverter::ANY_DIRECTION, jeod::TimeConverter::b_name, nIter, nSteps, prev_tai_seconds, prev_tdb_seconds, tai_ptr, TAI_to_TT_offset, tdb_ptr, and jeod::TimeConverter::valid_directions.

8.8.2.2 ∼TimeConverter_TAI_TDB()

```
jeod::TimeConverter_TAI_TDB::∼TimeConverter_TAI_TDB (
            void )
```

Definition at line 194 of file time_converter_tai_tdb.cc.

8.8.2.3 TimeConverter_TAI_TDB() [2/2]

```
jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB (
            const TimeConverter_TAI_TDB & )  [private]
```

8.8.3 Member Function Documentation

8.8.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_TDB::convert_a_to_b (
            void )  [virtual]
```

Default converter from time 'a' to time 'b'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented from jeod::TimeConverter.

Definition at line 152 of file time_converter_tai_tdb.cc.

References a_to_b_offset, a_to_b_offset_epoch, jeod::JeodBaseTime::seconds, set_a_to_b_offset(), jeod::TimeStandard::set_time_by_seconds(), tai_ptr, and tdb_ptr.

8.8.3.2 convert_b_to_a()

```
void jeod::TimeConverter_TAI_TDB::convert_b_to_a (
            void )  [virtual]
```

Default converter from time 'b' to time 'a'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented from jeod::TimeConverter.

Definition at line 168 of file time_converter_tai_tdb.cc.

References a_to_b_offset, a_to_b_offset_epoch, nIter, nSteps, prev_tai_seconds, prev_tdb_seconds, jeod::JeodBaseTime::seconds, set_a_to_b_offset(), jeod::TimeStandard::set_time_by_seconds(), tai_ptr, and tdb_ptr.

8.8.3.3 initialize()

```
void jeod::TimeConverter_TAI_TDB::initialize (
            JeodBaseTime * parent,
            JeodBaseTime * child,
            const int direction )  [virtual]
```

Initialize the converter.

Parameters

in	*parent*	parent-type
in	*child*	child-type
in	*direction*	L-R, or R-L

Implements jeod::TimeConverter.

Definition at line 97 of file time_converter_tai_tdb.cc.

References a_to_b_offset_epoch, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, set_a_to_b_offset(), tai_ptr, TAI_to_TT_offset, tdb_ptr, jeod::TimeStandard::tjt_at_epoch, and jeod::TimeConverter::verify_setup().

8.8.3.4 operator=()

```
TimeConverter_TAI_TDB& jeod::TimeConverter_TAI_TDB::operator= (
             const TimeConverter_TAI_TDB &  )  [private]
```

8.8.3.5 set_a_to_b_offset()

```
void jeod::TimeConverter_TAI_TDB::set_a_to_b_offset (
             void  )
```

Definition at line 133 of file time_converter_tai_tdb.cc.

References a_to_b_offset, tai_ptr, jeod::TimeStandard::tjt_at_epoch, and jeod::TimeStandard::trunc_julian_time.

Referenced by convert_a_to_b(), convert_b_to_a(), and initialize().

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__TimeConverter_TAI_TDB

```
void init_attrjeod__TimeConverter_TAI_TDB ( )  [friend]
```

8.8.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 92 of file time_converter_tai_tdb.hh.

8.8.5 Field Documentation

8.8.5.1 a_to_b_offset

```
double jeod::TimeConverter_TAI_TDB::a_to_b_offset  [private]
```

Calculated value of a_to_b_offset.

trick_units(–)

Definition at line 107 of file time_converter_tai_tdb.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), set_a_to_b_offset(), and TimeConverter_TAI_TDB().

8.8.5.2 a_to_b_offset_epoch

```
double jeod::TimeConverter_TAI_TDB::a_to_b_offset_epoch  [private]
```

The epoch value of a_to_b_offset.

trick_units(s)

Definition at line 103 of file time_converter_tai_tdb.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_TDB().

8.8.5.3 nIter

```
int jeod::TimeConverter_TAI_TDB::nIter  [private]
```

Counter for number of iterations.

trick_units(–)

Definition at line 123 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a(), and TimeConverter_TAI_TDB().

8.8.5.4 nSteps

```
int jeod::TimeConverter_TAI_TDB::nSteps  [private]
```

Counter for number of steps in iteration.

trick_units(–)

Definition at line 119 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a(), and TimeConverter_TAI_TDB().

8.8.5.5 prev_tai_seconds

```
double jeod::TimeConverter_TAI_TDB::prev_tai_seconds  [private]
```

TAI seconds from previous loop iteration.

trick_units(s)

Definition at line 111 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a(), and TimeConverter_TAI_TDB().

8.8.5.6 prev_tdb_seconds

```
double jeod::TimeConverter_TAI_TDB::prev_tdb_seconds  [private]
```

TDB seconds from previous loop iteration.

trick_units(s)

Definition at line 115 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a(), and TimeConverter_TAI_TDB().

8.8.5.7 tai_ptr

```
TimeTAI* jeod::TimeConverter_TAI_TDB::tai_ptr  [private]
```

Converter parent time, always a TimeTAI for this converter.

trick_units(–)

Definition at line 127 of file time_converter_tai_tdb.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), set_a_to_b_offset(), and TimeConverter_TAI_TDB().

8.8.5.8 TAI_to_TT_offset

```
double jeod::TimeConverter_TAI_TDB::TAI_to_TT_offset  [private]
```

The offset from TAI to TT.

trick_units(s)

Definition at line 99 of file time_converter_tai_tdb.hh.

Referenced by initialize(), and TimeConverter_TAI_TDB().

8.8.5.9 tdb_ptr

```
TimeTDB* jeod::TimeConverter_TAI_TDB::tdb_ptr  [private]
```

Converter parent time, always a TimeTDB for this converter.

trick_units(–)

Definition at line 131 of file time_converter_tai_tdb.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_TDB().

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

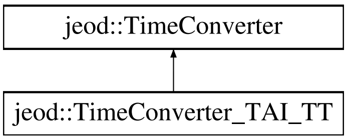
- time_converter_tai_tdb.hh
- time_converter_tai_tdb.cc

8.9 jeod::TimeConverter_TAI_TT Class Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include <time_converter_tai_tt.hh>
```

Inheritance diagram for jeod::TimeConverter_TAI_TT:



Public Member Functions

- TimeConverter_TAI_TT ()

  Construct a TimeConverter_TAI_TT.
- ∼TimeConverter_TAI_TT ()

  Destroy a TimeConverter_TAI_TT.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeTAI to TimeTT.
- void convert_b_to_a (void)

  Convert from TimeTT to TimeTAI.

Private Member Functions

- TimeConverter_TAI_TT (const TimeConverter_TAI_TT &)
- TimeConverter_TAI_TT & operator= (const TimeConverter_TAI_TT &)

Private Attributes

- TimeTAI ∗ tai_ptr

  Converter parent time, always a TimeTAI for this converter.
- TimeTT ∗ tt_ptr

  Converter parent time, always a TimeTT for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_TAI_TT ()

Additional Inherited Members

8.9.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

Definition at line 87 of file time_converter_tai_tt.hh.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 TimeConverter_TAI_TT() [1/2]

```
jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT (
            void )
```

Construct a TimeConverter_TAI_TT.

Definition at line 58 of file time_converter_tai_tt.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::ANY_DIRECTION, jeod::TimeConverter::b_name, tai_ptr, tt_ptr, and jeod::TimeConverter::valid_directions.

8.9.2.2 ∼TimeConverter_TAI_TT()

```
jeod::TimeConverter_TAI_TT::∼TimeConverter_TAI_TT (
            void )
```

Destroy a TimeConverter_TAI_TT.

Definition at line 137 of file time_converter_tai_tt.cc.

8.9.2.3 TimeConverter_TAI_TT() [2/2]

```
jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT (
            const TimeConverter_TAI_TT & ) [private]
```

8.9.3 Member Function Documentation

8.9.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_TT::convert_a_to_b (
            void ) [virtual]
```

Convert from TimeTAI to TimeTT.

Reimplemented from jeod::TimeConverter.

Definition at line 108 of file time_converter_tai_tt.cc.

References jeod::JeodBaseTime::seconds, jeod::TimeStandard::set_time_by_seconds(), tai_ptr, and tt_ptr.

8.9.3.2 convert_b_to_a()

```
void jeod::TimeConverter_TAI_TT::convert_b_to_a (
            void  )  [virtual]
```

Convert from TimeTT to TimeTAI.

Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from jeod::TimeConverter.

Definition at line 125 of file time_converter_tai_tt.cc.

References jeod::JeodBaseTime::seconds, jeod::TimeStandard::set_time_by_seconds(), tai_ptr, and tt_ptr.

8.9.3.3 initialize()

```
void jeod::TimeConverter_TAI_TT::initialize (
            JeodBaseTime * parent_ptr,
            JeodBaseTime * child_ptr,
            const int int_dir )  [virtual]
```

Initialize the converter.

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 76 of file time_converter_tai_tt.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_↩ setup_error, tai_ptr, tt_ptr, and jeod::TimeConverter::verify_setup().

8.9.3.4 operator=()

```
TimeConverter_TAI_TT& jeod::TimeConverter_TAI_TT::operator= (
            const TimeConverter_TAI_TT &  )  [private]
```

8.9.4 Friends And Related Function Documentation

8.9.4.1 init_attrjeod__TimeConverter_TAI_TT

```
void init_attrjeod__TimeConverter_TAI_TT ( )  [friend]
```

8.9.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 89 of file time_converter_tai_tt.hh.

8.9.5 Field Documentation

8.9.5.1 tai_ptr

```
TimeTAI* jeod::TimeConverter_TAI_TT::tai_ptr  [private]
```

Converter parent time, always a TimeTAI for this converter.

trick_units(–)

Definition at line 96 of file time_converter_tai_tt.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_TT().

8.9.5.2 tt_ptr

```
TimeTT* jeod::TimeConverter_TAI_TT::tt_ptr  [private]
```

Converter parent time, always a TimeTT for this converter.

trick_units(–)

Definition at line 101 of file time_converter_tai_tt.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), and TimeConverter_TAI_TT().

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

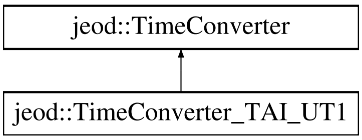
- time_converter_tai_tt.hh
- time_converter_tai_tt.cc

8.10 jeod::TimeConverter_TAI_UT1 Class Reference

Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.

```
#include <time_converter_tai_ut1.hh>
```

Inheritance diagram for jeod::TimeConverter_TAI_UT1:



Public Member Functions

- TimeConverter_TAI_UT1 ()

  Construct a TimeConverter_TAI_UT1.
- ∼TimeConverter_TAI_UT1 ()

  Destroy a TimeConverter_TAI_UT1.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeTAI to TimeUT1.
- void convert_b_to_a (void)

  Convert from TimeUT1 to TimeTAI.

Data Fields

- bool override_data_table

  "True" to enter user-specified tai-ut1 offset
- double tai_to_ut1_override_val

  User specified value (UT1 - TAI)
- int last_index

  Index of last datum in table.
- int index

  Current location in table.
- double ∗ val_vec

  Vector of values of difference between TAI-UT1.
- double ∗ when_vec

  Vector of corresponding times.

Private Member Functions

- void initialize_tai_to_ut1 (void)

  The conversion from Atomic Time (TAI) to Universal Time (UT1) involves the addition of value that is a continuous function of TAI.
- void verify_table_lookup_ends (void)

  Used when time reverses direction.
- TimeConverter_TAI_UT1 (const TimeConverter_TAI_UT1 &)
- TimeConverter_TAI_UT1 & operator= (const TimeConverter_TAI_UT1 &)

Private Attributes

- TimeTAI ∗ tai_ptr

  Converter parent time, always a TimeTAI for this converter.
- TimeUT1 ∗ ut1_ptr

  Converter parent time, always a TimeUT1 for this converter.
- double prev_when

  Time of previous calibrated datum.
- double prev_value

  Offset value of previous datum.
- double next_when

  Time of next calibrated datum.
- double next_value

  Offset value of next datum.
- double gradient

  Rate at which "value" changes wrt "when".
- bool off_table_end

  Gone past the end of the table.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_TAI_UT1 ()

Additional Inherited Members

8.10.1 Detailed Description

Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.

Definition at line 89 of file time_converter_tai_ut1.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 TimeConverter_TAI_UT1() [1/2]

```
jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1 (
            void  )
```

Construct a TimeConverter_TAI_UT1.

Definition at line 60 of file time_converter_tai_ut1.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::ANY_DIRECTION, jeod::TimeConverter::b_name, gradient, index, last_index, next_value, next_when, off_table_end, override_data_table, prev_value, prev_when, tai_ptr, tai_to_ut1_override_val, ut1_ptr, val_vec, jeod::TimeConverter::valid_directions, and when_vec.

8.10.2.2 ∼TimeConverter_TAI_UT1()

```
jeod::TimeConverter_TAI_UT1::∼TimeConverter_TAI_UT1 (
             void  )
```

Destroy a TimeConverter_TAI_UT1.

Definition at line 491 of file time_converter_tai_ut1.cc.

References val_vec, and when_vec.

8.10.2.3 TimeConverter_TAI_UT1() [2/2]

```
jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1 (
             const TimeConverter_TAI_UT1 &  )  [private]
```

8.10.3 Member Function Documentation

8.10.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_UT1::convert_a_to_b (
             void  )  [virtual]
```

Convert from TimeTAI to TimeUT1.

Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from jeod::TimeConverter.

Definition at line 258 of file time_converter_tai_ut1.cc.

References jeod::TimeConverter::a_to_b_offset, gradient, index, jeod::TimeMessages::invalid_data_error, last_↩ index, next_value, next_when, off_table_end, prev_value, prev_when, jeod::TimeStandard::set_time_by_trunc_↩ julian(), tai_ptr, jeod::TimeUT1::true_ut1, jeod::TimeStandard::trunc_julian_time, ut1_ptr, val_vec, and when_vec.

8.10.3.2 convert_b_to_a()

```
void jeod::TimeConverter_TAI_UT1::convert_b_to_a (
             void  )  [virtual]
```

Convert from TimeUT1 to TimeTAI.

Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from jeod::TimeConverter.

Definition at line 353 of file time_converter_tai_ut1.cc.

References jeod::TimeConverter::a_to_b_offset, gradient, index, jeod::TimeMessages::invalid_data_error, last_index, next_value, next_when, off_table_end, prev_value, prev_when, jeod::TimeStandard::set_time_by_trunc_julian(), tai_ptr, jeod::TimeUT1::true_ut1, jeod::TimeStandard::trunc_julian_time, ut1_ptr, val_vec, and when_vec.

8.10.3.3 initialize()

```
void jeod::TimeConverter_TAI_UT1::initialize (
             JeodBaseTime * parent_ptr,
             JeodBaseTime * child_ptr,
             const int int_dir )  [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 95 of file time_converter_tai_ut1.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeMessages::initialization_error, initialize_tai_to_ut1(), jeod::TimeConverter::initialized, next_when, prev_when, tai_ptr, jeod::TimeStandard::trunc_julian_time, ut1_ptr, and jeod::TimeConverter::verify_setup().

8.10.3.4 initialize_tai_to_ut1()

```
void jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1 (
            void ) [private]
```

The conversion from Atomic Time (TAI) to Universal Time (UT1) involves the addition of value that is a continuous function of TAI.

That value is tabulated at regular points of TAI. This function initializes that table and sets the preliminary values.

Assumptions and Limitations

- The table does not go into the future.

Definition at line 145 of file time_converter_tai_ut1.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeManager::dyn_time, gradient, index, jeod::TimeMessages::invalid_data_error, last_index, next_value, next_when, off_table_end, override_data_table, prev_value, prev_when, jeod::TimeDyn::scale_factor, tai_ptr, tai_to_ut1_override_val, jeod::JeodBaseTime::time_manager, jeod::TimeUT1::true_ut1, jeod::TimeStandard::trunc_julian_time, ut1_ptr, val_vec, and when_vec.

Referenced by initialize().

8.10.3.5 operator=()

```
TimeConverter_TAI_UT1& jeod::TimeConverter_TAI_UT1::operator= (
            const TimeConverter_TAI_UT1 & ) [private]
```

8.10.3.6 verify_table_lookup_ends()

```
void jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends (
            void ) [private], [virtual]
```

Used when time reverses direction.

Checks whether the table lookup function is using input values that are outside the scope of the table, and sets the flags appropriately

Assumptions and Limitations

- None

Reimplemented from jeod::TimeConverter.

Definition at line 449 of file time_converter_tai_ut1.cc.

References jeod::TimeManager::dyn_time, index, last_index, next_when, off_table_end, prev_when, jeod::TimeDyn::scale_factor, tai_ptr, jeod::JeodBaseTime::time_manager, jeod::TimeUT1::true_ut1, jeod::TimeStandard::trunc_julian_time, ut1_ptr, and when_vec.

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__TimeConverter_TAI_UT1

```
void init_attrjeod__TimeConverter_TAI_UT1 ( )  [friend]
```

8.10.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 91 of file time_converter_tai_ut1.hh.

8.10.5 Field Documentation

8.10.5.1 gradient

```
double jeod::TimeConverter_TAI_UT1::gradient  [private]
```

Rate at which "value" changes wrt "when".

trick_units(–)

Definition at line 155 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_ut1(), and TimeConverter_TAI_UT1().

8.10.5.2 index

```
int jeod::TimeConverter_TAI_UT1::index
```

Current location in table.

trick_units(–)

Definition at line 125 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify↩_table_lookup_ends().

8.10.5.3 last_index

```
int jeod::TimeConverter_TAI_UT1::last_index
```

Index of last datum in table.

trick_units(–)

Definition at line 121 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data↩ ::initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify_table_lookup_ends().

8.10.5.4 next_value

```
double jeod::TimeConverter_TAI_UT1::next_value  [private]
```

Offset value of next datum.

trick_units(s)

Definition at line 151 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_ut1(), and TimeConverter_TAI_UT1().

8.10.5.5 next_when

```
double jeod::TimeConverter_TAI_UT1::next_when  [private]
```

Time of next calibrated datum.

trick_units(day)

Definition at line 147 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify_table_lookup_ends().

8.10.5.6 off_table_end

```
bool jeod::TimeConverter_TAI_UT1::off_table_end  [private]
```

Gone past the end of the table.

trick_units(–)

Definition at line 159 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify↩ _table_lookup_ends().

8.10.5.7 override_data_table

```
bool jeod::TimeConverter_TAI_UT1::override_data_table
```

"True" to enter user-specified tai-ut1 offset

trick_units(--)

Definition at line 98 of file time_converter_tai_ut1.hh.

Referenced by jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize(), initialize_tai_to_ut1(), Time↩ Converter_TAI_UT1(), and jeod::TimeManagerInit::verify_converter_setup().

8.10.5.8 prev_value

```
double jeod::TimeConverter_TAI_UT1::prev_value [private]
```

Offset value of previous datum.

trick_units(s)

Definition at line 143 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_ut1(), and TimeConverter_TAI_UT1().

8.10.5.9 prev_when

```
double jeod::TimeConverter_TAI_UT1::prev_when [private]
```

Time of previous calibrated datum.

trick_units(day)

Definition at line 139 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify_table_lookup_ends().

8.10.5.10 tai_ptr

```
TimeTAI* jeod::TimeConverter_TAI_UT1::tai_ptr [private]
```

Converter parent time, always a TimeTAI for this converter.

trick_units(--)

Definition at line 105 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify_table_lookup_ends().

8.10.5.11 tai_to_ut1_override_val

```
double jeod::TimeConverter_TAI_UT1::tai_to_ut1_override_val
```

User specified value (UT1 - TAI)

trick_units(s)

Definition at line 116 of file time_converter_tai_ut1.hh.

Referenced by initialize_tai_to_ut1(), and TimeConverter_TAI_UT1().

8.10.5.12 ut1_ptr

```
TimeUT1* jeod::TimeConverter_TAI_UT1::ut1_ptr [private]
```

Converter parent time, always a TimeUT1 for this converter.

trick_units(–)

Definition at line 110 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and verify_table_lookup_ends().

8.10.5.13 val_vec

```
double* jeod::TimeConverter_TAI_UT1::val_vec
```

Vector of values of difference between TAI-UT1.

trick_units(s)

Definition at line 129 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data↩ ::initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), and ∼TimeConverter_TAI_UT1().

8.10.5.14 when_vec

```
double* jeod::TimeConverter_TAI_UT1::when_vec
```

Vector of corresponding times.

trick_units(day)

Definition at line 133 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data↩ ::initialize(), initialize_tai_to_ut1(), TimeConverter_TAI_UT1(), verify_table_lookup_ends(), and ∼TimeConverter↩ _TAI_UT1().

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

- time_converter_tai_ut1.hh
- time_converter_tai_ut1.cc

8.11 jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data Class Reference

```
#include <tai_to_ut1.hh>
```

Public Member Functions

- void initialize (TimeConverter_TAI_UT1 ∗)

8.11.1 Detailed Description

Definition at line 54 of file tai_to_ut1.hh.

8.11.2 Member Function Documentation

8.11.2.1 initialize()

```
void jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize (
            TimeConverter_TAI_UT1 ∗ TimeConverter_TAI_UT1_ptr )
```

Definition at line 41 of file tai_to_ut1.cc.

References jeod::TimeConverter_TAI_UT1::last_index, jeod::TimeConverter_TAI_UT1::override_data_table, jeod::TimeConverter_TAI_UT1::val_vec, and jeod::TimeConverter_TAI_UT1::when_vec.

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

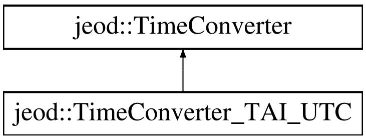
- tai_to_ut1.hh
- tai_to_ut1.cc

8.12 jeod::TimeConverter_TAI_UTC Class Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include <time_converter_tai_utc.hh>
```

Inheritance diagram for jeod::TimeConverter_TAI_UTC:



Public Member Functions

- TimeConverter_TAI_UTC ()

  Construct a TimeConverter_TAI_UTC.
- ∼TimeConverter_TAI_UTC ()

  Destroy a TimeConverter_TAI_UTC.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeTAI to TimeUTC.
- void convert_b_to_a (void)

  Convert from TimeUTC to TimeTAI.

Data Fields

- bool override_data_table

  "True" to enter user-specified tai-utc offset
- double leap_sec_override_val

  User specified value (TAI - UTC)
- int last_index

  Maximum index in the leap tables.
- int index

  Current index in the leap tables.
- int ∗ val_vec

  Tabulated values of leap_value.
- double ∗ when_vec

  Tabulated values of Julian time corresponding to changes in leap_value.

Private Member Functions

- void initialize_leap_second (void)

  The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.
- void verify_table_lookup_ends (void)

  Used when time reverses direction.
- TimeConverter_TAI_UTC (const TimeConverter_TAI_UTC &)
- TimeConverter_TAI_UTC & operator= (const TimeConverter_TAI_UTC &)

Private Attributes

- TimeTAI ∗ tai_ptr

  Converter parent time, always a TimeTAI for this converter.
- TimeUTC ∗ utc_ptr

  Converter parent time, always a TimeUTC for this converter.
- double next_when

  The next (future) UTC time of a leap second instance.
- double prev_when

  The most recent (past) UTC time of a leap second instance.
- bool off_table_end

  Flag to indicate that the current time is not covered by the leap-second tables.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_TAI_UTC ()

Additional Inherited Members

8.12.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

Definition at line 89 of file time_converter_tai_utc.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 TimeConverter_TAI_UTC() [1/2]

```
jeod::TimeConverter_TAI_UTC::TimeConverter_TAI_UTC (
            void  )
```

Construct a TimeConverter_TAI_UTC.

Definition at line 61 of file time_converter_tai_utc.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::A_TO_B, jeod::TimeConverter::b_name, jeod::TimeConverter::B_TO_A_INIT, index, last_index, leap_sec_override_val, next_when, off_table_end, override_data_table, prev_when, tai_ptr, utc_ptr, val_vec, jeod::TimeConverter::valid_directions, and when_vec.

8.12.2.2 ∼TimeConverter_TAI_UTC()

```
jeod::TimeConverter_TAI_UTC::∼TimeConverter_TAI_UTC (
            void  )
```

Destroy a TimeConverter_TAI_UTC.

Definition at line 461 of file time_converter_tai_utc.cc.

References val_vec, and when_vec.

8.12.2.3 TimeConverter_TAI_UTC() [2/2]

```
jeod::TimeConverter_TAI_UTC::TimeConverter_TAI_UTC (
            const TimeConverter_TAI_UTC &  )  [private]
```

8.12.3 Member Function Documentation

8.12.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_UTC::convert_a_to_b (
            void ) [virtual]
```

Convert from TimeTAI to TimeUTC.

Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from jeod::TimeConverter.

Definition at line 266 of file time_converter_tai_utc.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeManager::dyn_time, index, jeod::TimeMessages↩ ::invalid_data_error, last_index, next_when, off_table_end, prev_when, jeod::TimeDyn::scale_factor, jeod::↩ TimeStandard::set_time_by_trunc_julian(), tai_ptr, jeod::JeodBaseTime::time_manager, jeod::TimeUTC::true_utc, jeod::TimeStandard::trunc_julian_time, utc_ptr, val_vec, and when_vec.

8.12.3.2 convert_b_to_a()

```
void jeod::TimeConverter_TAI_UTC::convert_b_to_a (
            void ) [virtual]
```

Convert from TimeUTC to TimeTAI.

Reimplemented from jeod::TimeConverter.

Definition at line 352 of file time_converter_tai_utc.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeManager::dyn_time, index, jeod::TimeMessages↩ ::invalid_data_error, last_index, next_when, off_table_end, prev_when, jeod::TimeDyn::scale_factor, jeod::↩ TimeStandard::set_time_by_trunc_julian(), tai_ptr, jeod::JeodBaseTime::time_manager, jeod::TimeUTC::true_utc, jeod::TimeStandard::trunc_julian_time, utc_ptr, val_vec, and when_vec.

8.12.3.3 initialize()

```
void jeod::TimeConverter_TAI_UTC::initialize (
            JeodBaseTime * parent_ptr,
            JeodBaseTime * child_ptr,
            const int int_dir ) [virtual]
```

Initialize the converter.

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 90 of file time_converter_tai_utc.cc.

References jeod::TimeConverter::a_to_b_offset, index, jeod::TimeMessages::initialization_error, initialize_leap_second(), jeod::TimeConverter::initialized, jeod::JeodBaseTime::is_initialized(), tai_ptr, jeod::TimeStandard::trunc_julian_time, utc_ptr, val_vec, jeod::TimeConverter::verify_setup(), and when_vec.

8.12.3.4 initialize_leap_second()

```
void jeod::TimeConverter_TAI_UTC::initialize_leap_second (
            void ) [private]
```

The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.

The number of leap seconds at any given (historical) time is provided in a table. This function initializes that table and sets the preliminary values.

Assumptions and Limitations

- The table does not go into the future.

Definition at line 150 of file time_converter_tai_utc.cc.

References jeod::TimeConverter::a_to_b_offset, jeod::TimeManager::dyn_time, index, jeod::TimeMessages::invalid_data_error, jeod::TimeMessages::invalid_setup_error, last_index, leap_sec_override_val, next_when, off_table_end, override_data_table, prev_when, jeod::TimeDyn::scale_factor, tai_ptr, jeod::JeodBaseTime::time_manager, jeod::TimeUTC::true_utc, jeod::TimeStandard::trunc_julian_time, utc_ptr, val_vec, and when_vec.

Referenced by initialize().

8.12.3.5 operator=()

```
TimeConverter_TAI_UTC& jeod::TimeConverter_TAI_UTC::operator= (
            const TimeConverter_TAI_UTC & ) [private]
```

8.12.3.6 verify_table_lookup_ends()

```
void jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends (
            void ) [private], [virtual]
```

Used when time reverses direction.

Checks whether the table lookup function is using input values that are outside the scope of the table, and sets the flags appropriately

Reimplemented from jeod::TimeConverter.

Definition at line 420 of file time_converter_tai_utc.cc.

References jeod::TimeManager::dyn_time, index, last_index, next_when, off_table_end, prev_when, jeod::TimeDyn::scale_factor, tai_ptr, jeod::JeodBaseTime::time_manager, jeod::TimeUTC::true_utc, jeod::TimeStandard::trunc_julian_time, utc_ptr, and when_vec.

8.12.4 Friends And Related Function Documentation

8.12.4.1 init_attrjeod__TimeConverter_TAI_UTC

```
void init_attrjeod__TimeConverter_TAI_UTC ( ) [friend]
```

8.12.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 91 of file time_converter_tai_utc.hh.

8.12.5 Field Documentation

8.12.5.1 index

```
int jeod::TimeConverter_TAI_UTC::index
```

Current index in the leap tables.

trick_units(--)

Definition at line 123 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_leap_second(), TimeConverter_TAI_UTC(), and verify_table_lookup_ends().

8.12.5.2 last_index

```
int jeod::TimeConverter_TAI_UTC::last_index
```

Maximum index in the leap tables.

trick_units(–)

Definition at line 119 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data::initialize(), initialize_leap_second(), TimeConverter_TAI_UTC(), and verify_table_lookup_ends().

8.12.5.3 leap_sec_override_val

```
double jeod::TimeConverter_TAI_UTC::leap_sec_override_val
```

User specified value (TAI - UTC)

trick_units(s)

Definition at line 115 of file time_converter_tai_utc.hh.

Referenced by initialize_leap_second(), and TimeConverter_TAI_UTC().

8.12.5.4 next_when

```
double jeod::TimeConverter_TAI_UTC::next_when  [private]
```

The next (future) UTC time of a leap second instance.

trick_units(–)

Definition at line 137 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), TimeConverter_TAI_UTC(), and verify_table_lookup_ends().

8.12.5.5 off_table_end

```
bool jeod::TimeConverter_TAI_UTC::off_table_end  [private]
```

Flag to indicate that the current time is not covered by the leap-second tables.

trick_units(–)

Definition at line 147 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), TimeConverter_TAI_UTC(), and verify_table_lookup_ends().

8.12.5.6 override_data_table

```
bool jeod::TimeConverter_TAI_UTC::override_data_table
```

"True" to enter user-specified tai-utc offset

trick_units(–)

Definition at line 98 of file time_converter_tai_utc.hh.

Referenced by jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data::initialize(), initialize_leap_second(), TimeConverter_TAI_UTC(), and jeod::TimeManagerInit::verify_converter_setup().

8.12.5.7 prev_when

```
double jeod::TimeConverter_TAI_UTC::prev_when [private]
```

The most recent (past) UTC time of a leap second instance.

trick_units(–)

Definition at line 142 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), TimeConverter_TAI_UTC(), and verify_table_lookup_ends().

8.12.5.8 tai_ptr

```
TimeTAI* jeod::TimeConverter_TAI_UTC::tai_ptr [private]
```

Converter parent time, always a TimeTAI for this converter.

trick_units(–)

Definition at line 104 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_leap_second(), TimeConverter_TAI_UT↩C(), and verify_table_lookup_ends().

8.12.5.9 utc_ptr

```
TimeUTC* jeod::TimeConverter_TAI_UTC::utc_ptr [private]
```

Converter parent time, always a TimeUTC for this converter.

trick_units(–)

Definition at line 109 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_leap_second(), TimeConverter_TAI_UT↩C(), and verify_table_lookup_ends().

8.12.5.10 val_vec

```
int* jeod::TimeConverter_TAI_UTC::val_vec
```

Tabulated values of leap_value.

trick_units(s)

Definition at line 127 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data↩ ::initialize(), initialize(), initialize_leap_second(), TimeConverter_TAI_UTC(), and ∼TimeConverter_TAI_UTC().

8.12.5.11 when_vec

```
double* jeod::TimeConverter_TAI_UTC::when_vec
```

Tabulated values of Julian time corresponding to changes in leap_value.

trick_units(day)

Definition at line 132 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data↩ ::initialize(), initialize(), initialize_leap_second(), TimeConverter_TAI_UTC(), verify_table_lookup_ends(), and ∼↩ TimeConverter_TAI_UTC().

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

- time_converter_tai_utc.hh
- time_converter_tai_utc.cc

8.13 jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data Class Reference

```
#include <tai_to_utc.hh>
```

Public Member Functions

- void initialize (TimeConverter_TAI_UTC ∗)

8.13.1 Detailed Description

Definition at line 54 of file tai_to_utc.hh.

8.13.2 Member Function Documentation

8.13.2.1 initialize()

```
void jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data::initialize (
            TimeConverter_TAI_UTC * TimeConverter_TAI_UTC_ptr )
```

Definition at line 40 of file tai_to_utc.cc.

References jeod::TimeConverter_TAI_UTC::last_index, jeod::TimeConverter_TAI_UTC::override_data_table, jeod::TimeConverter_TAI_UTC::val_vec, and jeod::TimeConverter_TAI_UTC::when_vec.

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

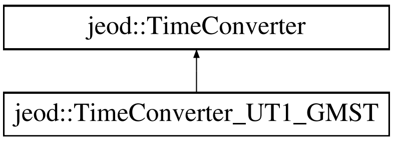
- tai_to_utc.hh
- tai_to_utc.cc

8.14 jeod::TimeConverter_UT1_GMST Class Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

```
#include <time_converter_ut1_gmst.hh>
```

Inheritance diagram for jeod::TimeConverter_UT1_GMST:



Public Member Functions

- TimeConverter_UT1_GMST ()

  Construct a TimeConverter_UT1_GMST.
- ∼TimeConverter_UT1_GMST ()

  Destroy a TimeConverter_UT1_GMST.
- void initialize (JeodBaseTime ∗parent, JeodBaseTime ∗child, const int direction)

  Initialize the converter.
- void convert_a_to_b (void)

  Convert from TimeUT1 to TimeGMST.

Private Member Functions

- TimeConverter_UT1_GMST (const TimeConverter_UT1_GMST &)
- TimeConverter_UT1_GMST & operator= (const TimeConverter_UT1_GMST &)

Private Attributes

- TimeUT1 ∗ ut1_ptr

  Converter parent time, always a TimeUT1 for this converter.
- TimeGMST ∗ gmst_ptr

  Converter parent time, always a TimeGMST for this converter.

Friends

- class InputProcessor
- void init_attrjeod__TimeConverter_UT1_GMST ()

Additional Inherited Members

8.14.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

Definition at line 87 of file time_converter_ut1_gmst.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 TimeConverter_UT1_GMST() [1/2]

```
jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST (
            void  )
```

Construct a TimeConverter_UT1_GMST.

Definition at line 58 of file time_converter_ut1_gmst.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::A_TO_B, jeod::TimeConverter::b_name, gmst_ptr, ut1_ptr, and jeod::TimeConverter::valid_directions.

8.14.2.2 ∼TimeConverter_UT1_GMST()

```
jeod::TimeConverter_UT1_GMST::∼TimeConverter_UT1_GMST (
            void  )
```

Destroy a TimeConverter_UT1_GMST.

Definition at line 148 of file time_converter_ut1_gmst.cc.

8.14.2.3 TimeConverter_UT1_GMST() [2/2]

```
jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST (
            const TimeConverter_UT1_GMST &  )  [private]
```

8.14.3 Member Function Documentation

8.14.3.1 convert_a_to_b()

```
void jeod::TimeConverter_UT1_GMST::convert_a_to_b (
            void  )  [virtual]
```

Convert from TimeUT1 to TimeGMST.

Assumptions and Limitations

- None

Reimplemented from jeod::TimeConverter.

Definition at line 111 of file time_converter_ut1_gmst.cc.

References jeod::TimeUT1::get_days(), gmst_ptr, jeod::TimeStandard::set_time_by_days(), and ut1_ptr.

8.14.3.2 initialize()

```
void jeod::TimeConverter_UT1_GMST::initialize (
            JeodBaseTime * parent_ptr,
            JeodBaseTime * child_ptr,
            const int int_dir )  [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

in	*parent_ptr*	Time used to initialize the converter
in	*child_ptr*	Other Time used to initialize the converter
in	*int_dir*	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements jeod::TimeConverter.

Definition at line 79 of file time_converter_ut1_gmst.cc.

References gmst_ptr, jeod::TimeConverter::initialized, jeod::TimeMessages::invalid_setup_error, ut1_ptr, and jeod::TimeConverter::verify_setup().

8.14.3.3 operator=()

```
TimeConverter_UT1_GMST& jeod::TimeConverter_UT1_GMST::operator= (
             const TimeConverter_UT1_GMST &  )  [private]
```

8.14.4 Friends And Related Function Documentation

8.14.4.1 init_attrjeod__TimeConverter_UT1_GMST

```
void init_attrjeod__TimeConverter_UT1_GMST ( )  [friend]
```

8.14.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 89 of file time_converter_ut1_gmst.hh.

8.14.5 Field Documentation

8.14.5.1 gmst_ptr

```
TimeGMST* jeod::TimeConverter_UT1_GMST::gmst_ptr  [private]
```

Converter parent time, always a TimeGMST for this converter.

trick_units(--)

Definition at line 101 of file time_converter_ut1_gmst.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_UT1_GMST().

8.14.5.2 ut1_ptr

```
TimeUT1* jeod::TimeConverter_UT1_GMST::ut1_ptr  [private]
```

Converter parent time, always a TimeUT1 for this converter.

trick_units(–)

Definition at line 96 of file time_converter_ut1_gmst.hh.

Referenced by convert_a_to_b(), initialize(), and TimeConverter_UT1_GMST().

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

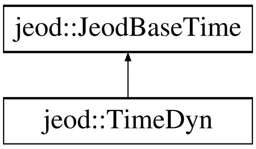
- time_converter_ut1_gmst.hh
- time_converter_ut1_gmst.cc

8.15 jeod::TimeDyn Class Reference

Represents the Dynamic Time in the simulation.

```
#include <time_dyn.hh>
```

Inheritance diagram for jeod::TimeDyn:



Public Member Functions

- TimeDyn ()

  Construct a Time_Dyn.
- ∼TimeDyn ()

  Destroy a Time_Dyn.
- bool update_offset (void)

  Changeing time direction and/or scale factor.

Data Fields

- double scale_factor

  Multiplicative difference between sim-time and dyn-time.

Private Member Functions

- void initialize_initializer_time (TimeManagerInit ∗tm_init)

  Each time type is initialized from its parent in the initialization tree, except one.
- void update (void)

  TimeDyn updates directly from simtime, and everything else from TimeDyn.
- TimeDyn (const TimeDyn &)
- TimeDyn & operator= (const TimeDyn &)

Private Attributes

- double ref_scale

  Private copy of scale_factor.
- double offset

  Extrapolated difference between sim-time and dyn-time at the sim-start (0 if there are no changes to direction or scale)

Friends

- class InputProcessor
- void init_attrjeod__TimeDyn ()

Additional Inherited Members

8.15.1 Detailed Description

Represents the Dynamic Time in the simulation.

Definition at line 86 of file time_dyn.hh.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 TimeDyn() [1/2]

```
jeod::TimeDyn::TimeDyn (
             void  )
```

Construct a Time_Dyn.

Definition at line 58 of file time_dyn.cc.

References jeod::JeodBaseTime::links, jeod::JeodBaseTime::name, offset, ref_scale, and scale_factor.

8.15.2.2 ∼TimeDyn()

```
jeod::TimeDyn::∼TimeDyn (
             void  )
```

Destroy a Time_Dyn.

Definition at line 149 of file time_dyn.cc.

8.15.2.3 TimeDyn() [2/2]

```
jeod::TimeDyn::TimeDyn (
             const TimeDyn &  )  [private]
```

8.15.3 Member Function Documentation

8.15.3.1 initialize_initializer_time()

```
void jeod::TimeDyn::initialize_initializer_time (
             TimeManagerInit * time_manager_init )  [private], [virtual]
```

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

Assumptions and Limitations

- TimeDyn cannot be used as the initializer time
- Each time representation can have its own initializer function, or can inherit the one in TimeDerived

Parameters

in	*time_manager_init*	TM initializer

Implements jeod::JeodBaseTime.

Definition at line 82 of file time_dyn.cc.

References jeod::JeodBaseTime::initialized, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime↩ ::seconds, jeod::JeodBaseTime::time_manager, and jeod::TimeManager::time_standards_exist().

8.15.3.2 operator=()

```
TimeDyn& jeod::TimeDyn::operator= (
            const TimeDyn & )  [private]
```

8.15.3.3 update()

```
void jeod::TimeDyn::update (
            void )  [private], [virtual]
```

TimeDyn updates directly from simtime, and everything else from TimeDyn.

This function does that first update from simtime

Assumptions and Limitations

- ref_scale is positive for forward-pregoressing sims, and negative for reverse-progressing sims.

Reimplemented from jeod::JeodBaseTime.

Definition at line 110 of file time_dyn.cc.

References offset, ref_scale, jeod::JeodBaseTime::seconds, jeod::TimeManager::simtime, and jeod::JeodBaseTime::time_manager.

8.15.3.4 update_offset()

```
bool jeod::TimeDyn::update_offset (
            void )
```

Changeing time direction and/or scale factor.

Returns

Void

Definition at line 124 of file time_dyn.cc.

References offset, ref_scale, scale_factor, jeod::JeodBaseTime::seconds, jeod::TimeManager::simtime, jeod::JeodBaseTime::time_manager, and jeod::TimeManager::verify_table_lookup_ends().

Referenced by jeod::TimeManager::update().

8.15.4 Friends And Related Function Documentation

8.15.4.1 init_attrjeod__TimeDyn

```
void init_attrjeod__TimeDyn ( )  [friend]
```

8.15.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 88 of file time_dyn.hh.

8.15.5 Field Documentation

8.15.5.1 offset

```
double jeod::TimeDyn::offset  [private]
```

Extrapolated difference between sim-time and dyn-time at the sim-start (0 if there are no changes to direction or scale)

trick_units(–)

Definition at line 109 of file time_dyn.hh.

Referenced by TimeDyn(), update(), and update_offset().

8.15.5.2 ref_scale

```
double jeod::TimeDyn::ref_scale  [private]
```

Private copy of scale_factor.

This value should not be changed externally; it is used for comparison purposes to identify when "scale_factor" has changed.trick_units(–)

Definition at line 104 of file time_dyn.hh.

Referenced by TimeDyn(), update(), and update_offset().

8.15.5.3 scale_factor

```
double jeod::TimeDyn::scale_factor
```

Multiplicative difference between sim-time and dyn-time.

This is the value that is changed externally.trick_units(–)

Definition at line 96 of file time_dyn.hh.

Referenced by jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b↩ _to_a(), jeod::TimeManager::get_time_scale_factor(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), TimeDyn(), update_offset(), jeod::TimeConverter_TAI_UT↩ C::verify_table_lookup_ends(), and jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends().

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

- time_dyn.hh
- time_dyn.cc

8.16 jeod::TimeEnum Class Reference

Contains an enumeration of the formats in which time can be represented.

```
#include <time_enum.hh>
```

Public Types

- enum TimeFormat {
  undefined = -1, Julian, julian, modified_julian,
  truncated_julian, calendar, clock, days_since_epoch,
  seconds_since_epoch }

  The enumeration of the formats in which time can be represented.

8.16.1 Detailed Description

Contains an enumeration of the formats in which time can be represented.

Definition at line 78 of file time_enum.hh.

8.16.2 Member Enumeration Documentation

8.16.2.1 TimeFormat

```
enum jeod::TimeEnum::TimeFormat
```

The enumeration of the formats in which time can be represented.

Enumerator

undefined	Default setting.
Julian	Full Julian representation.
julian	Full Julian representation.
modified_julian	Modified-Julian representation.
truncated_julian	Truncated-Julian representation.
calendar	Calendar (Gregorian) representation.
clock	"Calendar" representation for MET.
days_since_epoch	Days since the type's defined epoch.
seconds_since_epoch	Seconds since the type's defined epoch.

Definition at line 86 of file time_enum.hh.

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

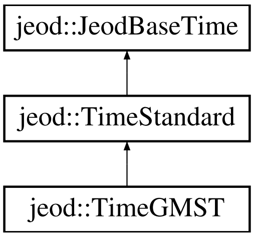
- time_enum.hh

8.17 jeod::TimeGMST Class Reference

To represent the clock known as Greenwich Mean Sidereal Time.

```
#include <time_gmst.hh>
```

Inheritance diagram for jeod::TimeGMST:



Public Member Functions

- TimeGMST ()

  Construct a Time_GMST.
- ∼TimeGMST ()

  Destroy a Time_GMST.
- void set_time_by_trunc_julian (const double nonsense)

  TJT does not function in GMST.

Private Member Functions

- void calculate_calendar_values (void)

  Protection against inheriting nonsense function.
- void set_epoch (void)

  No action.
- TimeGMST (const TimeGMST &)
- TimeGMST & operator= (const TimeGMST &)

Friends

- class InputProcessor
- void init_attrjeod__TimeGMST ()

Additional Inherited Members

8.17.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

Definition at line 82 of file time_gmst.hh.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 TimeGMST() [1/2]

```
jeod::TimeGMST::TimeGMST (
            void  )
```

Construct a Time_GMST.

Definition at line 55 of file time_gmst.cc.

References jeod::JeodBaseTime::name.

8.17.2.2 ∼TimeGMST()

```
jeod::TimeGMST::∼TimeGMST (
            void  )
```

Destroy a Time_GMST.

Definition at line 102 of file time_gmst.cc.

8.17.2.3 TimeGMST() [2/2]

```
jeod::TimeGMST::TimeGMST (
            const TimeGMST &  )  [private]
```

8.17.3 Member Function Documentation

8.17.3.1 calculate_calendar_values()

```
void jeod::TimeGMST::calculate_calendar_values (
            void  )  [private], [virtual]
```

Protection against inheriting nonsense function.

Assumptions and Limitations

- GMST does not have a conventional calendar

Reimplemented from jeod::TimeStandard.

Definition at line 68 of file time_gmst.cc.

References jeod::TimeMessages::invalid_data_error.

8.17.3.2 operator=()

```
TimeGMST& jeod::TimeGMST::operator= (
            const TimeGMST &  )  [private]
```

8.17.3.3 set_epoch()

```
void jeod::TimeGMST::set_epoch (
            void  )  [inline], [private], [virtual]
```

No action.

Function is required to make this class instantiable.

Implements jeod::TimeStandard.

Definition at line 104 of file time_gmst.hh.

8.17.3.4 set_time_by_trunc_julian()

```
void jeod::TimeGMST::set_time_by_trunc_julian (
            const double nonsense )
```

TJT does not function in GMST.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*nonsense*	Any old invalid value

Definition at line 86 of file time_gmst.cc.

References jeod::TimeMessages::invalid_data_error.

8.17.4 Friends And Related Function Documentation

8.17.4.1 init_attrjeod__TimeGMST

```
void init_attrjeod__TimeGMST ( )  [friend]
```

8.17.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_gmst.hh.

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

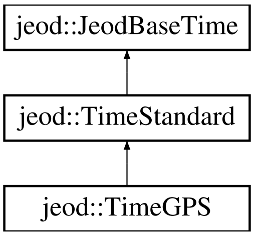
- time_gmst.hh
- time_gmst.cc

8.18 jeod::TimeGPS Class Reference

To represent the time associated with the Global Positioning System.

```
#include <time_gps.hh>
```

Inheritance diagram for jeod::TimeGPS:



Public Member Functions

- TimeGPS ()

  Construct a Time_GPS.
- ∼TimeGPS ()

  Destroy a TimeGPS.
- void set_time_by_seconds (const double new_seconds)

  Given a value of seconds, propagate to other reps.
- void set_time_by_days (const double new_seconds)

  Given a value of days, propagate to other values.
- void set_time_by_trunc_julian (const double new_tjt)

  TJT does not function in GPS.

Data Fields

- double seconds_of_day

  Seconds elapsed in last (partial) day.
- double seconds_of_week

  Seconds elapsed in last (partial) week.
- int day_of_week

  Number of whole days this week.
- int rollover_count

  Number of rollovers (1024 week blocks) since epoch.
- int week

  Number of weeks in current 1024-week block.
- int rollover_count_13_bit

  Number of rollovers (8192 week blocks) since epoch.
- int week_13_bit

  Number of weeks in current 8192-week block.

Private Member Functions

- void calculate_calendar_values (void)

  Protection against inheriting nonsense function.
- void convert_from_calendar (void)

  Protection against inheriting nonsense function.
- void set_epoch (void)

  Sets the epoch for GPS time.
- TimeGPS (const TimeGPS &)
- TimeGPS & operator= (const TimeGPS &)

Friends

- class InputProcessor
- void init_attrjeod__TimeGPS ()

Additional Inherited Members

8.18.1 Detailed Description

To represent the time associated with the Global Positioning System.

Definition at line 82 of file time_gps.hh.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 TimeGPS() [1/2]

```
jeod::TimeGPS::TimeGPS (
              void  )
```

Construct a Time_GPS.

Definition at line 55 of file time_gps.cc.

References day_of_week, jeod::JeodBaseTime::name, rollover_count, rollover_count_13_bit, seconds_of_day, seconds_of_week, set_epoch(), week, and week_13_bit.

8.18.2.2 ∼TimeGPS()

```
jeod::TimeGPS::∼TimeGPS (
              void  )
```

Destroy a TimeGPS.

Definition at line 193 of file time_gps.cc.

8.18.2.3 TimeGPS() [2/2]

```
jeod::TimeGPS::TimeGPS (
              const TimeGPS &  )  [private]
```

8.18.3 Member Function Documentation

8.18.3.1 calculate_calendar_values()

```
void jeod::TimeGPS::calculate_calendar_values (
             void  )  [private], [virtual]
```

Protection against inheriting nonsense function.

Assumptions and Limitations

- GPS does not have a conventional calendar

Reimplemented from jeod::TimeStandard.

Definition at line 110 of file time_gps.cc.

References jeod::TimeMessages::invalid_data_error.

8.18.3.2 convert_from_calendar()

```
void jeod::TimeGPS::convert_from_calendar (
             void  )  [private], [virtual]
```

Protection against inheriting nonsense function.

Assumptions and Limitations

- GPS does not have a conventional calendar

Reimplemented from jeod::TimeStandard.

Definition at line 92 of file time_gps.cc.

References jeod::TimeMessages::invalid_data_error.

8.18.3.3 operator=()

```
TimeGPS& jeod::TimeGPS::operator= (
             const TimeGPS &  )  [private]
```

8.18.3.4 set_epoch()

```
void jeod::TimeGPS::set_epoch (
            void ) [private], [virtual]
```

Sets the epoch for GPS time.

Implements jeod::TimeStandard.

Definition at line 75 of file time_gps.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeGPS().

8.18.3.5 set_time_by_days()

```
void jeod::TimeGPS::set_time_by_days (
            const double new_days ) [virtual]
```

Given a value of days, propagate to other values.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_days*	new value for days Units: day

Reimplemented from jeod::TimeStandard.

Definition at line 163 of file time_gps.cc.

References set_time_by_seconds().

8.18.3.6 set_time_by_seconds()

```
void jeod::TimeGPS::set_time_by_seconds (
            const double new_seconds ) [virtual]
```

Given a value of seconds, propagate to other reps.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_seconds*	new value for seconds Units: s

Reimplemented from jeod::TimeStandard.

Definition at line 128 of file time_gps.cc.

References day_of_week, jeod::JeodBaseTime::days, rollover_count, rollover_count_13_bit, seconds_of_day, seconds_of_week, jeod::TimeStandard::set_time_by_seconds(), week, and week_13_bit.

Referenced by jeod::TimeConverter_TAI_GPS::convert_a_to_b(), set_time_by_days(), and set_time_by_trunc_julian().

8.18.3.7 set_time_by_trunc_julian()

```
void jeod::TimeGPS::set_time_by_trunc_julian (
             const double new_tjt )
```

TJT does not function in GPS.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_tjt*	new value for Truncated Julian Time Units: day

Definition at line 179 of file time_gps.cc.

References jeod::JeodBaseTime::seconds, set_time_by_seconds(), and jeod::TimeStandard::set_time_by_trunc_julian().

8.18.4 Friends And Related Function Documentation

8.18.4.1 init_attrjeod__TimeGPS

```
void init_attrjeod__TimeGPS ( )  [friend]
```

8.18.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_gps.hh.

8.18.5 Field Documentation

8.18.5.1 day_of_week

```
int jeod::TimeGPS::day_of_week
```

Number of whole days this week.

trick_units(day)

Definition at line 100 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.2 rollover_count

```
int jeod::TimeGPS::rollover_count
```

Number of rollovers (1024 week blocks) since epoch.

trick_units(–)

Definition at line 104 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.3 rollover_count_13_bit

```
int jeod::TimeGPS::rollover_count_13_bit
```

Number of rollovers (8192 week blocks) since epoch.

trick_units(–)

Definition at line 112 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.4 seconds_of_day

```
double jeod::TimeGPS::seconds_of_day
```

Seconds elapsed in last (partial) day.

trick_units(s)

Definition at line 92 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.5 seconds_of_week

```
double jeod::TimeGPS::seconds_of_week
```

Seconds elapsed in last (partial) week.

trick_units(s)

Definition at line 96 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.6 week

```
int jeod::TimeGPS::week
```

Number of weeks in current 1024-week block.

trick_units(–)

Definition at line 108 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

8.18.5.7 week_13_bit

```
int jeod::TimeGPS::week_13_bit
```

Number of weeks in current 8192-week block.

trick_units(–)

Definition at line 116 of file time_gps.hh.

Referenced by set_time_by_seconds(), and TimeGPS().

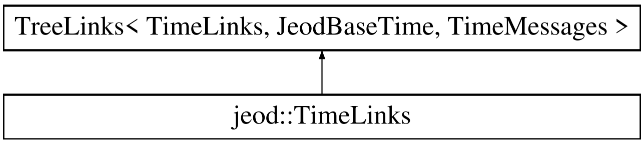
The documentation for this class was generated from the following files:

- time_gps.hh
- time_gps.cc

8.19 jeod::TimeLinks Class Reference

```
#include <time_links.hh>
```

Inheritance diagram for jeod::TimeLinks:



Public Member Functions

- TimeLinks (JeodBaseTime &time_in)
- TimeLinks ()=delete
- TimeLinks (const TimeLinks &)=delete
- void operator= (const TimeLinks &)=delete
- virtual ∼TimeLinks ()=default

  Default destructor.

Static Private Attributes

- static const unsigned int default_path_size = 8

  Default allocated number of entries in linkage container.

Friends

- class InputProcessor
- void init_attrjeod__TimeLinks ()

8.19.1 Detailed Description

Definition at line 78 of file time_links.hh.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 TimeLinks() [1/3]

```
jeod::TimeLinks::TimeLinks (
            JeodBaseTime & time_in ) [inline]
```

Definition at line 84 of file time_links.hh.

8.19.2.2 TimeLinks() [2/3]

```
jeod::TimeLinks::TimeLinks ( )  [delete]
```

8.19.2.3 TimeLinks() [3/3]

```
jeod::TimeLinks::TimeLinks (
             const TimeLinks &  )  [delete]
```

8.19.2.4 ∼TimeLinks()

```
virtual jeod::TimeLinks::∼TimeLinks ( )  [virtual], [default]
```

Default destructor.

8.19.3 Member Function Documentation

8.19.3.1 operator=()

```
void jeod::TimeLinks::operator= (
             const TimeLinks &  )  [delete]
```

8.19.4 Friends And Related Function Documentation

8.19.4.1 init_attrjeod__TimeLinks

```
void init_attrjeod__TimeLinks ( )  [friend]
```

8.19.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 80 of file time_links.hh.

8.19.5 Field Documentation

8.19.5.1 default_path_size

```
const unsigned int jeod::TimeLinks::default_path_size = 8  [static], [private]
```

Default allocated number of entries in linkage container.

trick_units(--)

Definition at line 104 of file time_links.hh.

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

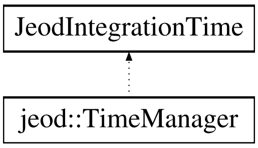
- time_links.hh

8.20 jeod::TimeManager Class Reference

To manage the various time representations and the converters between them throughout the simulation.

```
#include <time_manager.hh>
```

Inheritance diagram for jeod::TimeManager:



Public Member Functions

- TimeManager ()

  Construct a TimeManager.
- ∼TimeManager ()

  Destroy a TimeManager.
- void initialize (TimeManagerInit ∗time_manager_init)

  initializes the time manager
- int time_lookup (const std::string &name) const

  Uses a string comparison to find where in the TimeManager record a time type of a particular name is located.
- bool get_time_change_flag () const

  Returns the boolean value time_change_flag.
- JeodBaseTime ∗ get_time_ptr (const std::string &name) const

  Return a pointer to the Time object with the provided name, or NULL if no such Time object has been registered.
- JeodBaseTime ∗ get_time_ptr (const int index) const

  Return a pointer to the Time object with the provided index, or NULL if no such Time object has been registered.
- TimeConverter ∗ get_converter_ptr (const int index) const

Return a pointer to the TimeConverter object with the provided index, or NULL if no such TimeConverter object has been registered.

- bool time_standards_exist (void)

  Tests for the existence in the registry of time types that inherit from TimeStandard.
- virtual void update (double time)

  This function manages the time update process.
- void verify_table_lookup_ends (void)

  This function is called when the simulation reverses direction (in time.
- void register_time (JeodBaseTime &time_ref)

  Registers the time representation with the Time Manager.
- void register_time_named (JeodBaseTime &time_ref, const std::string &name)

  Reassigns the name to the type; this is used when there are multiple instances of a time type such as a MET or UDE.
- void register_converter (TimeConverter &converter_ref, std::string name_a="", std::string name_b="")

  Registers the time converters with the Time Manager.
- JeodIntegrationTime & get_jeod_integration_time ()

  Expose the private inheritance from JeodIntegrationTime.
- virtual double get_time_scale_factor () const

  Returns the scale factor from sim time to dynamic time.
- virtual double get_timestamp_time () const

  Returns the time used to timestamp objects, currently dynamic time seconds.

Data Fields

- double simtime

  Simulation time (sys.exec.out.time).
- TimeDyn dyn_time

  The instance of TimeDyn, the dynamic time that is used as the integration time.
- int num_types

  Size of time_types_ptrs vector.

Private Member Functions

- virtual void update_time (double time)

  Update each of the representations of time, calling the update functions for each such representation in dependency order.
- TimeManager (const TimeManager &)
- TimeManager & operator= (const TimeManager &)

Private Attributes

- bool time_change_flag

  Indicates that the dynamic scale factor changed.
- std::vector< JeodBaseTime * > time_vector

  List of pointers to time-types.
- std::vector< TimeConverter * > converter_vector

  List of pointers to time-converters.

Friends

- class InputProcessor
- class TimeManagerInit
- void init_attrjeod__TimeManager ()

8.20.1 Detailed Description

To manage the various time representations and the converters between them throughout the simulation.

Definition at line 94 of file time_manager.hh.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 TimeManager() [1/2]

```
jeod::TimeManager::TimeManager (
            void   )
```

Construct a TimeManager.

Definition at line 65 of file time_manager.cc.

8.20.2.2 ∼TimeManager()

```
jeod::TimeManager::∼TimeManager (
            void   )
```

Destroy a TimeManager.

Definition at line 505 of file time_manager.cc.

References converter_vector, and time_vector.

8.20.2.3 TimeManager() [2/2]

```
jeod::TimeManager::TimeManager (
            const TimeManager &   )   [private]
```

8.20.3 Member Function Documentation

8.20.3.1 get_converter_ptr()

```
TimeConverter * jeod::TimeManager::get_converter_ptr (
            const int index ) const
```

Return a pointer to the TimeConverter object with the provided index, or NULL if no such TimeConverter object has been registered.

Returns

TimeConverter object corresponding to index in the vector of such types.

Parameters

in	*index*	Index of object

Definition at line 88 of file time_manager.cc.

References converter_vector.

Referenced by jeod::JeodBaseTime::add_type_update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeUDE::initialize_initializer_time().

8.20.3.2 get_jeod_integration_time()

```
JeodIntegrationTime & jeod::TimeManager::get_jeod_integration_time (
            void )
```

Expose the private inheritance from JeodIntegrationTime.

Definition at line 108 of file time_manager.cc.

8.20.3.3 get_time_change_flag()

```
bool jeod::TimeManager::get_time_change_flag (
            void ) const
```

Returns the boolean value time_change_flag.

Returns

time_change_flag

Definition at line 120 of file time_manager.cc.

References time_change_flag.

8.20.3.4 get_time_ptr() [1/2]

```
JeodBaseTime * jeod::TimeManager::get_time_ptr (
            const std::string & name ) const
```

Return a pointer to the Time object with the provided name, or NULL if no such Time object has been registered.

Returns

Time object corresponding to name

Parameters

in	*name*	Name of time object

Definition at line 162 of file time_manager.cc.

References time_lookup().

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::JeodBaseTime::add_type_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeManagerInit::initialize_time_types(), and jeod::TimeUDE::verify_update().

8.20.3.5 get_time_ptr() [2/2]

```
JeodBaseTime * jeod::TimeManager::get_time_ptr (
             const int index ) const
```

Return a pointer to the Time object with the provided index, or NULL if no such Time object has been registered.

Returns

Time object corresponding to name

Parameters

in	*index*	Name of time object

Definition at line 177 of file time_manager.cc.

References time_vector.

8.20.3.6 get_time_scale_factor()

```
double jeod::TimeManager::get_time_scale_factor (
             void ) const [virtual]
```

Returns the scale factor from sim time to dynamic time.

Returns

dyn_time.scale_factor

Definition at line 133 of file time_manager.cc.

References dyn_time, and jeod::TimeDyn::scale_factor.

8.20.3.7 get_timestamp_time()

```
double jeod::TimeManager::get_timestamp_time (
             void  ) const  [virtual]
```

Returns the time used to timestamp objects, currently dynamic time seconds.

Returns

dyn_time.seconds

Definition at line 147 of file time_manager.cc.

References dyn_time, and jeod::JeodBaseTime::seconds.

8.20.3.8 initialize()

```
void jeod::TimeManager::initialize (
             TimeManagerInit * time_manager_init )
```

initializes the time manager

Parameters

in	*time_manager_init*	Initialization parameters

Definition at line 64 of file time_manager__initialize.cc.

References dyn_time, jeod::TimeManagerInit::initialize_manager(), num_types, register_time(), time_vector, and update().

8.20.3.9 operator=()

```
TimeManager& jeod::TimeManager::operator= (
             const TimeManager &  )  [private]
```

8.20.3.10 register_converter()

```
void jeod::TimeManager::register_converter (
             TimeConverter & conv_ref,
             std::string name_a = "",
             std::string name_b = "" )
```

Registers the time converters with the Time Manager.

Assumptions and Limitations

- the input values name_a and name_b will only be used if the converter-type names have not already been set. So registering a Dyn_UDE converter will ignore name_a completely because it is already set.

Parameters

in,out	*conv_ref*	ref. to converter being registered
in	*name_a*	name of type-a in the converter
in	*name_b*	name of type-b in the converter

Definition at line 266 of file time_manager.cc.

References jeod::TimeConverter::a_name, jeod::TimeConverter::b_name, converter_vector, jeod::TimeMessages::incomplete_setup_error, and jeod::TimeMessages::redundancy_error.

8.20.3.11 register_time()

```
void jeod::TimeManager::register_time (
            JeodBaseTime & time_ref )
```

Registers the time representation with the Time Manager.

Records the frequency at which the representation should be updated.

Assumptions and Limitations

- None

Parameters

in,out	*time_ref*	reference to time-type being registered

Definition at line 200 of file time_manager.cc.

References jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy_error, jeod::JeodBaseTime::set_index(), jeod::JeodBaseTime::time_manager, and time_vector.

Referenced by initialize(), and register_time_named().

8.20.3.12 register_time_named()

```
void jeod::TimeManager::register_time_named (
            JeodBaseTime & time_ref,
            const std::string & name )
```

Reassigns the name to the type; this is used when there are multiple instances of a time type such as a MET or UDE.

Registers the time representation with the Time Manager. Records the frequency at which the representation should be updated. TODO: check for duplicates

Assumptions and Limitations

- None

Parameters

in,out	*time_ref*	reference to time-type being registered
in	*name*	name of the instance being registered.

Definition at line 234 of file time_manager.cc.

References jeod::JeodBaseTime::name, and register_time().

8.20.3.13 time_lookup()

```
int jeod::TimeManager::time_lookup (
            const std::string & name ) const
```

Uses a string comparison to find where in the TimeManager record a time type of a particular name is located.

Returns the integer corresponding to the time type's index in the TimeManager.

Assumptions and Limitations

- Rarely used. If the time type address is known, it is easier to access its index "time_type.index" which returns the same result.

Returns

index value of time-type

Parameters

in	*name*	name of time-type

Definition at line 355 of file time_manager.cc.

References jeod::TimeMessages::invalid_setup_error, and time_vector.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::JeodBaseTime::add_type_update(), jeod::TimeManagerInit::create_init_tree(), get_time_ptr(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeManagerInit::populate_converter_registry(), jeod::TimeManagerInit::verify_converter_setup(), jeod::TimeUDE::verify_epoch(), jeod::TimeUDE::verify_init(), and jeod::TimeUDE::verify_update().

8.20.3.14 time_standards_exist()

```
bool jeod::TimeManager::time_standards_exist (
            void )
```

Tests for the existence in the registry of time types that inherit from TimeStandard.

Assumptions and Limitations

- None

Returns

true/false

Definition at line 330 of file time_manager.cc.

References time_vector.

Referenced by jeod::TimeDyn::initialize_initializer_time(), and jeod::TimeUDE::initialize_initializer_time().

8.20.3.15 update()

```
void jeod::TimeManager::update (
            double current_simtime )  [virtual]
```

This function manages the time update process.

It first updates each of the representations of time, calling the update functions for each time representation in dependency order. After updating the representations of time, the function then updates the dynamic time scale factor. Time change subscribers are notified if the scale factor has changed.

Note that by updating first and then checking for a change in the rate/direction of time means that these changes in rate/direction will first take affect on the next call to update_time or update.

Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the update_tree was built

Parameters

in	*current_simtime*	input time from simulation engine; it always runs forwards and allows for determination of what has and has not already been done. Units: s

Definition at line 419 of file time_manager.cc.

References dyn_time, num_types, simtime, time_change_flag, time_vector, and jeod::TimeDyn::update_offset().

Referenced by jeod::TimeStandard::calendar_update(), and initialize().

8.20.3.16 update_time()

```
void jeod::TimeManager::update_time (
            double current_simtime )  [private], [virtual]
```

Update each of the representations of time, calling the update functions for each such representation in dependency order.

Note that this function only does the first part of the task performed by TimeManager::update. It does not check for changes in the rate/direction of time.

Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the update_tree was built

Parameters

in	*current_simtime*	input time from simulation engine; it always runs forwards and allows for determination of what has and has not already been done. Units: s

Definition at line 461 of file time_manager.cc.

References num_types, simtime, and time_vector.

8.20.3.17 verify_table_lookup_ends()

```
void jeod::TimeManager::verify_table_lookup_ends (
            void  )
```

This function is called when the simulation reverses direction (in time.

It calls each time converter that uses a table lookup to check whether the current time is off the end of the table. This is important because once the off-table-end flag is set, the only reason to unset it is when time reverses direction)

Assumptions and Limitations

- None

Definition at line 492 of file time_manager.cc.

References converter_vector.

Referenced by jeod::TimeDyn::update_offset().

8.20.4 Friends And Related Function Documentation

8.20.4.1 init_attrjeod__TimeManager

```
void init_attrjeod__TimeManager ( )  [friend]
```

8.20.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 96 of file time_manager.hh.

8.20.4.3 TimeManagerInit

```
friend class TimeManagerInit  [friend]
```

Definition at line 98 of file time_manager.hh.

8.20.5 Field Documentation

8.20.5.1 converter_vector

```
std::vector<TimeConverter*> jeod::TimeManager::converter_vector  [private]
```

List of pointers to time-converters.

Definition at line 136 of file time_manager.hh.

Referenced by get_converter_ptr(), jeod::TimeManagerInit::populate_converter_registry(), register_converter(), verify_table_lookup_ends(), and ∼TimeManager().

8.20.5.2 dyn_time

```
TimeDyn jeod::TimeManager::dyn_time
```

The instance of TimeDyn, the dynamic time that is used as the integration time.

trick_units(--)

Definition at line 113 of file time_manager.hh.

Referenced by jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), get_time_scale_factor(), get_timestamp_time(), initialize(), jeod::TimeManagerInit::initialize(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), update(), jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends(), and jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends().

8.20.5.3 num_types

```
int jeod::TimeManager::num_types
```

Size of time_types_ptrs vector.

trick_units(–)

Definition at line 118 of file time_manager.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::JeodBase↩ Time::add_type_update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeManagerInit::create_init_tree(), jeod::TimeManagerInit::create_update_tree(), initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::↩ TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeManagerInit::initialize↩ _time_types(), jeod::TimeManagerInit::organize_update_list(), jeod::TimeManagerInit::populate_converter_↩ registry(), update(), update_time(), jeod::TimeManagerInit::verify_converter_setup(), and jeod::TimeManager↩ Init::verify_times_setup().

8.20.5.4 simtime

```
double jeod::TimeManager::simtime
```

Simulation time (sys.exec.out.time).

trick_units(–)

Definition at line 107 of file time_manager.hh.

Referenced by jeod::TimeStandard::calendar_update(), jeod::TimeStandard::seconds_of_year(), jeod::TimeDyn↩ ::update(), update(), jeod::TimeDyn::update_offset(), and update_time().

8.20.5.5 time_change_flag

```
bool jeod::TimeManager::time_change_flag  [private]
```

Indicates that the dynamic scale factor changed.

trick_units(–)

Definition at line 126 of file time_manager.hh.

Referenced by get_time_change_flag(), and update().

8.20.5.6 time_vector

```
std::vector<JeodBaseTime*> jeod::TimeManager::time_vector  [private]
```

List of pointers to time-types.

Definition at line 131 of file time_manager.hh.

Referenced by jeod::TimeManagerInit::create_init_tree(), jeod::TimeManagerInit::create_update_tree(), get_time_ptr(), initialize(), jeod::TimeManagerInit::initialize(), jeod::TimeManagerInit::initialize_time_types(), jeod::TimeManagerInit::organize_update_list(), jeod::TimeManagerInit::populate_converter_registry(), register_time(), time_lookup(), time_standards_exist(), update(), update_time(), jeod::TimeManagerInit::verify_times_setup(), and ∼TimeManager().

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

- time_manager.hh
- time_manager.cc
- time_manager__initialize.cc

8.21 jeod::TimeManagerInit Class Reference

To initialize the Time Manager.

```
#include <time_manager_init.hh>
```

Public Member Functions

- TimeManagerInit ()

  Construct a TimeManagerInit.
- ∼TimeManagerInit ()

  Destroy a TimeManagerInit.
- int get_conv_ptr_index (const int conv_index)

  Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the time_manager's vector of converters that corresponds to those two time-types.
- int get_conv_dir_init (const int conv_index)

  Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".
- int get_conv_dir_upd (const int conv_index)

  Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".
- int get_status (const int index)

  Returns the status of a time-type.
- void set_status (const int index, const int status_value)

  Receives an updated value for the status of a time-type.
- void increment_status (const int slave_index, const int master_index)

  Modifies the status of one time-type to be one higher than that of another type for initialization purposes.
- void initialize_manager (TimeManager *time_mgr)

  The master program behind the initialization of the time types and the time converters.
- void organize_update_list ()

  Reorganizes the update list according to initialization status.

Data Fields

- int num_added_total

    Count of the total number of time-types placed in the update tree or in the initialization tree.
- TimeEnum::TimeFormat sim_start_format

    Calendar, truncated_julian, etc.
- TimeManager * time_manager

    Pointer to the Time Manager.
- std::string initializer

    Name of the time-type used for initialization.

Protected Attributes

- int initializer_index

    Index-value of the initializer.
- int dyn_time_index

    Index-value of the type dyn-time.
- int num_added_pass

    Count of number of time-types placed in the update tree or in the initialization tree in any given pass.
- int * converter_ptrs_index

    List of the indices (in the TimeManager->time_converter_ptrs vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)
- int * init_converter_dir_table

    List of directions available for initialization for each of the converters listed in converter_class_ptrs.
- int * update_converter_dir_table

    List of directions available for run-time updates for each of the converters listed in converter_class_ptrs.
- int * status

    A running ledger of properly linked times during update tree and initialization tree construction.

Private Member Functions

- void initialize (void)

    The TimeManagerInit determines initialization and update paths for conversions between time-types.
- void verify_times_setup (void)

    A number of checks that the setup is self-consistent.
- void populate_converter_registry (void)

    The converter registry accounts for all of the converter functions that provide conversions between time types.
- void verify_converter_setup (void)

    To verify that there are no incompatibilities between specific converters.
- void initialize_time_types (void)

    Initialize each time type so that it has a starting value corresponding to dynamic_time = 0 and such that the starting values are consistent.
- void create_init_tree (void)

    Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.
- void create_update_tree (void)

    (To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - create_update_tree builds the tree, populated recursively by add_type_↩ update.
- TimeManagerInit (const TimeManagerInit &)
- TimeManagerInit & operator= (const TimeManagerInit &)

Friends

- class InputProcessor
- void init_attrjeod__TimeManagerInit ()

8.21.1 Detailed Description

To initialize the Time Manager.

Definition at line 85 of file time_manager_init.hh.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 TimeManagerInit() [1/2]

```
jeod::TimeManagerInit::TimeManagerInit (
            void   )
```

Construct a TimeManagerInit.

Definition at line 64 of file time_manager_init.cc.

References converter_ptrs_index, dyn_time_index, init_converter_dir_table, initializer, initializer_index, num_added_pass, num_added_total, sim_start_format, status, jeod::TimeEnum::undefined, and update_converter_dir_table.

8.21.2.2 ∼TimeManagerInit()

```
jeod::TimeManagerInit::∼TimeManagerInit (
            void   )
```

Destroy a TimeManagerInit.

Definition at line 811 of file time_manager_init.cc.

References converter_ptrs_index, init_converter_dir_table, status, and update_converter_dir_table.

8.21.2.3 TimeManagerInit() [2/2]

```
jeod::TimeManagerInit::TimeManagerInit (
            const TimeManagerInit &   )   [private]
```

8.21.3 Member Function Documentation

8.21.3.1 create_init_tree()

```
void jeod::TimeManagerInit::create_init_tree (
             void  )  [private]
```

Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.

Create_init_tree builds the tree, using add_type_initialize to populate the tree recursively.

Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize_from" for each time representation, except the top-level initializer type
- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

Definition at line 430 of file time_manager_init.cc.

References dyn_time_index, jeod::TimeMessages::initialization_error, initializer_index, jeod::TimeMessages::invalid_setup_error, num_added_pass, num_added_total, jeod::TimeManager::num_types, status, jeod::TimeManager::time_lookup(), time_manager, jeod::TimeManager::time_vector, and jeod::JeodBaseTime::update_from_name.

Referenced by initialize_manager().

8.21.3.2 create_update_tree()

```
void jeod::TimeManagerInit::create_update_tree (
             void  )  [private]
```

(To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - create_update_tree builds the tree, populated recursively by add_type_update.

record_update records the update paths to facilitate runtime updates)

Assumptions and Limitations

- None

Definition at line 573 of file time_manager_init.cc.

References dyn_time_index, jeod::TimeMessages::incomplete_setup_error, jeod::TimeMessages::initialization_error, num_added_pass, num_added_total, jeod::TimeManager::num_types, organize_update_list(), status, time_manager, and jeod::TimeManager::time_vector.

Referenced by initialize_manager().

8.21.3.3 get_conv_dir_init()

```
int jeod::TimeManagerInit::get_conv_dir_init (
             const int index )
```

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".

Assumptions and Limitations

- Returns 0 if no suitable converter available at initialization

Returns

Index corresponding to TimeConverter

Parameters

in	*index*	Index of object

Definition at line 712 of file time_manager_init.cc.

References init_converter_dir_table.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeUDE::initialize_initializer_time().

8.21.3.4 get_conv_dir_upd()

```
int jeod::TimeManagerInit::get_conv_dir_upd (
             const int index )
```

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".

Assumptions and Limitations

- Returns 0 if no suitable converter available at update

Returns

Index corresponding to TimeConverter

Parameters

in	*index*	Index of object

Definition at line 738 of file time_manager_init.cc.

References update_converter_dir_table.

Referenced by jeod::JeodBaseTime::add_type_update().

8.21.3.5 get_conv_ptr_index()

```
int jeod::TimeManagerInit::get_conv_ptr_index (
             const int index_in )
```

Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the time_manager's vector of converters that corresponds to those two time-types.

Returns

Index corresponding to TimeConverter

Parameters

in	*index_in*	Index of object

Definition at line 686 of file time_manager_init.cc.

References converter_ptrs_index.

Referenced by jeod::JeodBaseTime::add_type_update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeUDE::initialize_initializer_time().

8.21.3.6 get_status()

```
int jeod::TimeManagerInit::get_status (
             const int index )
```

Returns the status of a time-type.

Returns

Integer corresponding to Status

Parameters

in	*index*	Index of object

Definition at line 760 of file time_manager_init.cc.

References status.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), and jeod::JeodBaseTime::add_type_update().

8.21.3.7 increment_status()

```
void jeod::TimeManagerInit::increment_status (
            const int index_slave,
            const int index_master )
```

Modifies the status of one time-type to be one higher than that of another type for initialization purposes.

Parameters

in	*index_slave*	Index of object
in	*index_master*	Index of object

Definition at line 797 of file time_manager_init.cc.

References num_added_pass, and status.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), and jeod::JeodBaseTime::add_type_update().

8.21.3.8 initialize()

```
void jeod::TimeManagerInit::initialize (
            void ) [private]
```

The TimeManagerInit determines initialization and update paths for conversions between time-types.

This function creates and initializes the data structures necessary for these determinations

Assumptions and Limitations

- None

Definition at line 139 of file time_manager_init.cc.

References converter_ptrs_index, jeod::TimeManager::dyn_time, dyn_time_index, jeod::JeodBaseTime::index, init_converter_dir_table, jeod::JeodBaseTime::initialized, initializer, initializer_index, jeod::JeodBaseTime::seconds, status, jeod::TimeManager::time_lookup(), time_manager, jeod::TimeManager::time_vector, update_converter_dir_table, and verify_times_setup().

Referenced by initialize_manager().

8.21.3.9 initialize_manager()

```
void jeod::TimeManagerInit::initialize_manager (
            TimeManager * time_mgr )
```

The master program behind the initialization of the time types and the time converters.

Assumptions and Limitations

- None

Parameters

in,out	*time_mgr*	The time manager

Definition at line 94 of file time_manager_init.cc.

References create_init_tree(), create_update_tree(), initialize(), initialize_time_types(), populate_converter_↩ registry(), time_manager, and verify_converter_setup().

Referenced by jeod::TimeManager::initialize().

8.21.3.10 initialize_time_types()

```
void jeod::TimeManagerInit::initialize_time_types (
            void ) [private]
```

Initialize each time type so that it has a starting value corresponding to dynamic_time = 0 and such that the starting values are consistent.

Initialize_time_types repetitively calls initialize_from_parent for each time type; initialize_from_parent recursively adds types moving up the tree as necessary

Assumptions and Limitations

- An initializer time defined by the user

Definition at line 540 of file time_manager_init.cc.

References jeod::TimeManager::get_time_ptr(), jeod::JeodBaseTime::initialize_from_parent(), jeod::JeodBase↩ Time::initialized, initializer_index, jeod::TimeManager::num_types, time_manager, and jeod::TimeManager::time↩ _vector.

Referenced by initialize_manager().

8.21.3.11 operator=()

```
TimeManagerInit& jeod::TimeManagerInit::operator= (
             const TimeManagerInit &  )  [private]
```

8.21.3.12 organize_update_list()

```
void jeod::TimeManagerInit::organize_update_list ( )
```

Reorganizes the update list according to initialization status.

Definition at line 636 of file time_manager_init.cc.

References jeod::TimeManager::num_types, status, time_manager, and jeod::TimeManager::time_vector.

Referenced by create_update_tree().

8.21.3.13 populate_converter_registry()

```
void jeod::TimeManagerInit::populate_converter_registry (
             void  )  [private]
```

The converter registry accounts for all of the converter functions that provide conversions between time types.

This function populates that registry so that the existence of functional converter functions can be tested efficiently.

Assumptions and Limitations

- None

Definition at line 275 of file time_manager_init.cc.

References jeod::TimeConverter::A_TO_B_INIT, jeod::TimeConverter::A_TO_B_UPDATE, jeod::TimeConverter↩ ::B_TO_A_INIT, jeod::TimeConverter::B_TO_A_UPDATE, converter_ptrs_index, jeod::TimeManager::converter_↩ vector, init_converter_dir_table, jeod::TimeManager::num_types, jeod::TimeMessages::redundancy_error, jeod::↩ TimeManager::time_lookup(), time_manager, jeod::TimeManager::time_vector, and update_converter_dir_table.

Referenced by initialize_manager().

8.21.3.14 set_status()

```
void jeod::TimeManagerInit::set_status (
             const int index,
             const int new_status )
```

Receives an updated value for the status of a time-type.

Parameters

in	*index*	Index of object
in	*new_status*	New status value

Definition at line 781 of file time_manager_init.cc.

References status.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), and jeod::JeodBaseTime::add_type_update().

8.21.3.15 verify_converter_setup()

```
void jeod::TimeManagerInit::verify_converter_setup (
            void  )  [private]
```

To verify that there are no incompatibilities between specific converters.

Assumptions and Limitations

- The instance of TimeTAI, if it exists, has name "TAI"
- The instance of TimeUTC, if it exists, has name "UTC"
- The instance of TimeUT1, if it exists, has name "UT1"

Definition at line 367 of file time_manager_init.cc.

References converter_ptrs_index, jeod::TimeMessages::invalid_setup_error, jeod::TimeManager::num_types, jeod::TimeConverter_TAI_UTC::override_data_table, jeod::TimeConverter_TAI_UT1::override_data_table, jeod::TimeManager::time_lookup(), and time_manager.

Referenced by initialize_manager().

8.21.3.16 verify_times_setup()

```
void jeod::TimeManagerInit::verify_times_setup (
            void  )  [private]
```

A number of checks that the setup is self-consistent.

Assumptions and Limitations

- None

Definition at line 205 of file time_manager_init.cc.

References jeod::TimeMessages::incomplete_setup_error, initializer, initializer_index, jeod::TimeMessages::invalid_setup_error, jeod::TimeManager::num_types, jeod::TimeMessages::redundancy_error, time_manager, and jeod::TimeManager::time_vector.

Referenced by initialize().

8.21.4 Friends And Related Function Documentation

8.21.4.1 init_attrjeod__TimeManagerInit

```
void init_attrjeod__TimeManagerInit ( ) [friend]
```

8.21.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 87 of file time_manager_init.hh.

8.21.5 Field Documentation

8.21.5.1 converter_ptrs_index

```
int* jeod::TimeManagerInit::converter_ptrs_index [protected]
```

List of the indices (in the TimeManager->time_converter_ptrs vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)

trick_units(–)

Definition at line 132 of file time_manager_init.hh.

Referenced by get_conv_ptr_index(), initialize(), populate_converter_registry(), TimeManagerInit(), verify_converter_setup(), and ∼TimeManagerInit().

8.21.5.2 dyn_time_index

```
int jeod::TimeManagerInit::dyn_time_index [protected]
```

Index-value of the type dyn-time.

trick_units(–)

Definition at line 119 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), initialize(), and TimeManagerInit().

8.21.5.3 init_converter_dir_table

```
int* jeod::TimeManagerInit::init_converter_dir_table  [protected]
```

List of directions available for initialization for each of the converters listed in converter_class_ptrs.

trick_units(–)

Definition at line 137 of file time_manager_init.hh.

Referenced by get_conv_dir_init(), initialize(), populate_converter_registry(), TimeManagerInit(), and ∼Time←ManagerInit().

8.21.5.4 initializer

```
std::string jeod::TimeManagerInit::initializer
```

Name of the time-type used for initialization.

trick_units(–)

Definition at line 109 of file time_manager_init.hh.

Referenced by initialize(), TimeManagerInit(), and verify_times_setup().

8.21.5.5 initializer_index

```
int jeod::TimeManagerInit::initializer_index  [protected]
```

Index-value of the initializer.

trick_units(–)

Definition at line 114 of file time_manager_init.hh.

Referenced by create_init_tree(), initialize(), initialize_time_types(), TimeManagerInit(), and verify_times_setup().

8.21.5.6 num_added_pass

```
int jeod::TimeManagerInit::num_added_pass  [protected]
```

Count of number of time-types placed in the update tree or in the initialization tree in any given pass.

trick_units(–)

Definition at line 124 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), increment_status(), and TimeManagerInit().

8.21.5.7 num_added_total

```
int jeod::TimeManagerInit::num_added_total
```

Count of the total number of time-types placed in the update tree or in the initialization tree.

trick_units(–)

Definition at line 96 of file time_manager_init.hh.

Referenced by jeod::JeodBaseTime::add_type_update(), create_init_tree(), create_update_tree(), and TimeManagerInit().

8.21.5.8 sim_start_format

```
TimeEnum::TimeFormat jeod::TimeManagerInit::sim_start_format
```

Calendar, truncated_julian, etc.

trick_units(–)

Definition at line 100 of file time_manager_init.hh.

Referenced by jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), and TimeManagerInit().

8.21.5.9 status

```
int* jeod::TimeManagerInit::status  [protected]
```

A running ledger of properly linked times during update tree and initialization tree construction.

Entries correspond to times of shared indexes in time_vector (e.g. status[2] : status of time_manager->time_vector[2]) Update tree encoding: -2: undefined. Requires auto-assignment or causes error. -1: definitive error. Process will terminate. 0: uninitialized 1: THE 1st generation (root) time. dyn_time for update tree. 2: a 2nd generation time, converted from root time. n: a nth gen time, converted from (n-1)th gen time.trick_units(–)

Definition at line 156 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), get_status(), increment_status(), initialize(), organize_update_list(), set_status(), TimeManagerInit(), and ∼TimeManagerInit().

8.21.5.10 time_manager

`TimeManager* jeod::TimeManagerInit::time_manager`

Pointer to the Time Manager.

Automatically linked during init routines.trick_units(–)

Definition at line 104 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), initialize(), initialize_manager(), initialize_time_types(), organize_update_list(), populate_converter_registry(), verify_converter_setup(), and verify_times_setup().

8.21.5.11 update_converter_dir_table

`int* jeod::TimeManagerInit::update_converter_dir_table [protected]`

List of directions available for run-time updates for each of the converters listed in converter_class_ptrs.

trick_units(–)

Definition at line 142 of file time_manager_init.hh.

Referenced by get_conv_dir_upd(), initialize(), populate_converter_registry(), TimeManagerInit(), and ∼Time↩ ManagerInit().

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

- time_manager_init.hh
- time_manager_init.cc

8.22 jeod::TimeMessages Class Reference

Specify the message IDs used in the Time model.

`#include <time_messages.hh>`

Static Public Attributes

- static char const ∗ initialization_error

  Error issued when intialization fails due to some non-obvious cause.
- static char const ∗ memory_error

  Error issued when system fails because something is not where it should be, or has a value other than its assumed value.
- static char const ∗ invalid_setup_error

  Error issued when user tries to use something that doesn't exist in the model.
- static char const ∗ invalid_data_error

  Error issued when a variable is found with an illegal value.
- static char const ∗ invalid_node

  Issued when a TimeLinks node is improperly linked.
- static char const ∗ incomplete_setup_error

  Error issued when user tries to use something that doesn't exist.n the simulation This is usually a user error, brought about by not having registered something that is later needed (e.g.
- static char const ∗ redundancy_error

  Error issued when some value is multiply defined, and the code cannot determine which value to use.
- static char const ∗ duplicate_methods

  Informational only.
- static char const ∗ extension_error

  Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

Private Member Functions

- TimeMessages (void)
- TimeMessages (const TimeMessages &)
- TimeMessages & operator= (const TimeMessages &)

Friends

- class InputProcessor
- void init_attrjeod__TimeMessages ()

8.22.1 Detailed Description

Specify the message IDs used in the Time model.

Definition at line 84 of file time_messages.hh.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 TimeMessages() [1/2]

```
jeod::TimeMessages::TimeMessages (
             void  )  [private]
```

8.22.2.2 TimeMessages() [2/2]

```
jeod::TimeMessages::TimeMessages (
             const TimeMessages &  )  [private]
```

8.22.3 Member Function Documentation

8.22.3.1 operator=()

```
TimeMessages& jeod::TimeMessages::operator= (
             const TimeMessages &  )  [private]
```

8.22.4 Friends And Related Function Documentation

8.22.4.1 init_attrjeod__TimeMessages

```
void init_attrjeod__TimeMessages ( ) [friend]
```

8.22.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 87 of file time_messages.hh.

8.22.5 Field Documentation

8.22.5.1 duplicate_methods

```
char const * jeod::TimeMessages::duplicate_methods [static]
```

Initial value:

```
=
    "environment/time/" "duplicate_methods"
```

Informational only.

Issued when there are multiple equivalent methods for doing something, and one method is chosen over another. Wherever the code fills in data to accommodate, it sends this informational broadcast.trick_units(–)

Definition at line 146 of file time_messages.hh.

Referenced by jeod::TimeUDE::initialize_initializer_time().

8.22.5.2 extension_error

```
char const * jeod::TimeMessages::extension_error [static]
```

Initial value:

```
=
    "environment/time/" "extension_error"
```

Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

trick_units(–)

Definition at line 152 of file time_messages.hh.

8.22.5.3 incomplete_setup_error

```
char const * jeod::TimeMessages::incomplete_setup_error [static]
```

Initial value:

```
=
    "environment/time/"  "incomplete_setup_error"
```

Error issued when user tries to use something that doesn't exist.n the simulation This is usually a user error, brought about by not having registered something that is later needed (e.g.

not registering a TAI-UTC converter, but specifying that UTC updates from TAI). Note the distinction between invalid (typically, cannot exist) and incomplete (typically, did not define)trick_units(–)

Definition at line 132 of file time_messages.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::JeodBaseTime::add_type_update(), jeod::TimeUDE::convert_epoch_to_update(), jeod::TimeManagerInit::create_update_tree(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeManager::register_converter(), jeod::TimeUDE::set_epoch_dyn(), jeod::TimeUDE::set_epoch_std(), jeod::TimeUDE::set_epoch_ude(), jeod::TimeUDE::set_initial_times(), jeod::TimeUDE::verify_epoch(), jeod::TimeManagerInit::verify_times_setup(), and jeod::TimeUDE::verify_update().

8.22.5.4 initialization_error

```
char const * jeod::TimeMessages::initialization_error [static]
```

Initial value:

```
=
    "environment/time/"  "initialization_error"
```

Error issued when intialization fails due to some non-obvious cause.

This error is likely due to an algorithm flaw.trick_units(–)

Definition at line 97 of file time_messages.hh.

Referenced by jeod::TimeManagerInit::create_init_tree(), jeod::TimeManagerInit::create_update_tree(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_TAI_UTC::initialize(), jeod::TimeConverter_TAI_UT1::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), and jeod::TimeConverter::verify_setup().

8.22.5.5 invalid_data_error

```
char const * jeod::TimeMessages::invalid_data_error  [static]
```

Initial value:

```
=
    "environment/time/"  "invalid_data_error"
```

Error issued when a variable is found with an illegal value.

This is usually a user error, having set some value externally to some unrecognizable value.trick_units(–)

Definition at line 117 of file time_messages.hh.

Referenced by jeod::TimeGMST::calculate_calendar_values(), jeod::TimeGPS::calculate_calendar_values(), jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UT1::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), jeod::TimeGPS::convert_from_calendar(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), and jeod::TimeGMST::set_time_by_trunc_julian().

8.22.5.6 invalid_node

```
char const * jeod::TimeMessages::invalid_node  [static]
```

Initial value:

```
=
    "environment/time/"  "invalid_node"
```

Issued when a TimeLinks node is improperly linked.

trick_units(–)

Definition at line 122 of file time_messages.hh.

Referenced by jeod::JeodBaseTime::add_type_update().

8.22.5.7 invalid_setup_error

```
char const * jeod::TimeMessages::invalid_setup_error [static]
```

Initial value:

```
=
    "environment/time/"  "invalid_setup_error"
```

Error issued when user tries to use something that doesn't exist in the model.

This is usually a user error. Note the distinction between invalid and incompletetrick_units(–)

Definition at line 110 of file time_messages.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::JeodBaseTime::add_type_initialize(), jeod::JeodBaseTime::add_type_update(), jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), jeod::TimeManagerInit::create_init_tree(), jeod::TimeConverter_TAI_TT::initialize(), jeod::TimeConverter_UT1_GMST::initialize(), jeod::TimeConverter_TAI_GPS::initialize(), jeod::TimeConverter_Dyn_TAI::initialize(), jeod::TimeConverter_Dyn_TDB::initialize(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeConverter_STD_UDE::initialize(), jeod::TimeConverter_TAI_TDB::initialize(), jeod::JeodBaseTime::initialize_from_parent(), jeod::TimeDyn::initialize_initializer_time(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeUDE::initialize_initializer_time(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeUDE::set_epoch_dyn(), jeod::TimeUDE::set_epoch_initializing_value(), jeod::TimeUDE::set_epoch_std(), jeod::TimeUDE::set_epoch_times(), jeod::TimeUDE::set_epoch_ude(), jeod::TimeUDE::set_initial_times(), jeod::TimeManager::time_lookup(), jeod::TimeManagerInit::verify_converter_setup(), jeod::TimeUDE::verify_epoch(), jeod::TimeConverter::verify_setup(), jeod::TimeManagerInit::verify_times_setup(), and jeod::TimeUDE::verify_update().

8.22.5.8 memory_error

```
char const * jeod::TimeMessages::memory_error [static]
```

Initial value:

```
=
    "environment/time/"  "memory_error"
```

Error issued when system fails because something is not where it should be, or has a value other than its assumed value.

trick_units(–)

Definition at line 103 of file time_messages.hh.

Referenced by jeod::JeodBaseTime::add_type_update(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::set_initial_times(), and jeod::JeodBaseTime::update().

8.22.5.9 redundancy_error

```
char const * jeod::TimeMessages::redundancy_error [static]
```

Initial value:

```
=
    "environment/time/" "redundancy_error"
```

Error issued when some value is multiply defined, and the code cannot determine which value to use.

Usually a user-error, from attempting to use too many of the initialization options simultaneously.trick_units(–)

Definition at line 139 of file time_messages.hh.

Referenced by jeod::TimeUDE::initialize_from_parent(), jeod::TimeStandard::initialize_initializer_time(), jeod::TimeManagerInit::populate_converter_registry(), jeod::TimeManager::register_converter(), jeod::TimeManager::register_time(), jeod::TimeUDE::set_epoch_dyn(), jeod::TimeUDE::set_initial_times(), jeod::TimeUDE::verify_epoch(), jeod::TimeUDE::verify_init(), and jeod::TimeManagerInit::verify_times_setup().

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

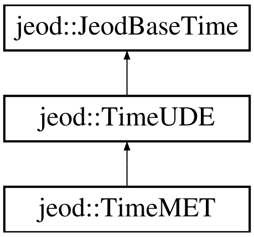
- time_messages.hh
- time_messages.cc

8.23 jeod::TimeMET Class Reference

A type of UDE time that allows for deliberate holds, or pauses.

```
#include <time_met.hh>
```

Inheritance diagram for jeod::TimeMET:



Public Member Functions

- TimeMET ()
- ∼TimeMET ()

  Destroy a Time_MET.
- void update (void)

  Updates to current time.

Data Fields

- bool hold

  Flags whether to hold time at current value.

Private Member Functions

- TimeMET (const TimeMET &)
- TimeMET & operator= (const TimeMET &)

Private Attributes

- bool previous_hold

  Previously known value of hold, used for recalculating converters.

Friends

- class InputProcessor
- void init_attrjeod__TimeMET ()

Additional Inherited Members

8.23.1 Detailed Description

A type of UDE time that allows for deliberate holds, or pauses.

Definition at line 83 of file time_met.hh.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 TimeMET() [1/2]

```
jeod::TimeMET::TimeMET (
             void )
```

Definition at line 68 of file time_met.cc.

References jeod::JeodBaseTime::name.

8.23.2.2 ∼TimeMET()

```
jeod::TimeMET::∼TimeMET (
             void  )
```

Destroy a Time_MET.

Definition at line 110 of file time_met.cc.

8.23.2.3 TimeMET() [2/2]

```
jeod::TimeMET::TimeMET (
             const TimeMET &  )  [private]
```

8.23.3 Member Function Documentation

8.23.3.1 operator=()

```
TimeMET& jeod::TimeMET::operator= (
             const TimeMET &  )  [private]
```

8.23.3.2 update()

```
void jeod::TimeMET::update (
             void  )  [virtual]
```

Updates to current time.

Reimplemented from jeod::JeodBaseTime.

Definition at line 82 of file time_met.cc.

References hold, previous_hold, jeod::TimeConverter::reset_a_to_b_offset(), jeod::JeodBaseTime::update(), and jeod::JeodBaseTime::update_converter_ptr.

8.23.4 Friends And Related Function Documentation

8.23.4.1 init_attrjeod__TimeMET

```
void init_attrjeod__TimeMET ( )  [friend]
```

8.23.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 85 of file time_met.hh.

8.23.5 Field Documentation

8.23.5.1 hold

```
bool jeod::TimeMET::hold
```

Flags whether to hold time at current value.

trick_units(–)

Definition at line 92 of file time_met.hh.

Referenced by update().

8.23.5.2 previous_hold

```
bool jeod::TimeMET::previous_hold  [private]
```

Previously known value of hold, used for recalculating converters.

trick_units(–)

Definition at line 98 of file time_met.hh.

Referenced by update().

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

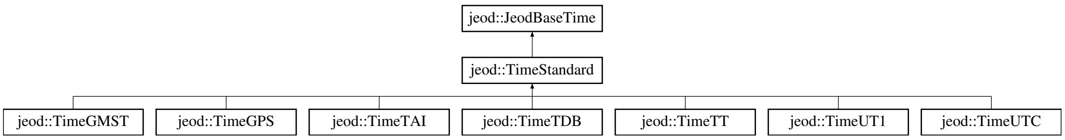
- time_met.hh
- time_met.cc

8.24 jeod::TimeStandard Class Reference

A class that serves as the base for all time representations that are well defined outside the simulation.

```
#include <time_standard.hh>
```

Inheritance diagram for jeod::TimeStandard:



Public Member Functions

- TimeStandard ()

  Construct a TimeStandard.
- virtual ∼TimeStandard ()

  Destroy a TimeStandard.
- void calendar_update (double simtime)

  Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation.
- void initialize_initializer_time (TimeManagerInit ∗tm_init)

  Each time type is initialized from its parent in the initialization tree, except one.
- void add_type_initialize (const int seeking_status, TimeManagerInit ∗tm_init)

  Recursively adds elements to the initialization tree.
- void initialize_from_parent (TimeManagerInit ∗tm_init)

  Initialize a time type from its parent on the initialization tree.
- virtual void set_time_by_seconds (const double new_seconds)

  Given a value of seconds, propagate to days and trunc_julian_time.
- virtual void set_time_by_days (const double new_days)

  Given a value of days, propagate to seconds and trunc_julian_time.
- void set_time_by_trunc_julian (const double new_tjt)

  Given a value of tjt, propagate to seconds and days.
- double julian_date_at_epoch (void)

  Returns the full Julian date at epoch, rather than the Truncated Julian Time.
- double seconds_of_year (void)

  Generate the number of seconds elapsed this year.

Data Fields

- double last_calendar_update

  The simtime when the calendar update was last run.
- int prev_julian_day

  Used for determining whether to update the date in the calendar function.
- double seconds_at_year_start

  The value of "seconds" at the start of the year in which the last seconds_of_year calculation was made.
- int year_of_last_soy

  The year in which the last seconds_of_year calculation was made.
- bool send_warning_pre_1968

This flag can be turned off by developers wanting to avoid warnings about a simulation being initialized pre-1968.

- const double tjt_mjt_offset

  Difference between Truncated Julian and Modified Julian.
- const double tjt_jd_offset

  Difference between Julian and Truncated Julian.
- double trunc_julian_time

  Truncated Julian time for this time-type.
- double julian_date

  Conventional Julian Date.
- double tjt_at_epoch

  Truncated Julian Date at epoch.
- int calendar_day

  Gregorian calendar date day number.
- int calendar_hour

  24-hour clock hour number.
- int calendar_minute

  Clock minute number.
- double calendar_second

  Clock second number.
- int calendar_year

  Gregorian calendar year.
- int calendar_month

  Gregorian calendar month.

Protected Member Functions

- virtual void convert_from_calendar (void)

  Calculate Truncated Julian date/time from Gregorian calendar date and 24-hour clock representation.
- virtual void calculate_calendar_values (void)

  Calculate Gregorian calendar date and 24-hour clock representation from Truncated Julian date/time.
- virtual void set_epoch (void)=0

  Set the epoch time.

Private Member Functions

- TimeStandard (const TimeStandard &)
- TimeStandard & operator= (const TimeStandard &)

Friends

- class InputProcessor
- class TimeUDE
- void init_attrjeod__TimeStandard ()

Additional Inherited Members

8.24.1 Detailed Description

A class that serves as the base for all time representations that are well defined outside the simulation.

Definition at line 89 of file time_standard.hh.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 TimeStandard() [1/2]

```
jeod::TimeStandard::TimeStandard (
            void  )
```

Construct a TimeStandard.

Definition at line 60 of file time_standard.cc.

8.24.2.2 ∼TimeStandard()

```
jeod::TimeStandard::∼TimeStandard (
            void  )  [virtual]
```

Destroy a TimeStandard.

Definition at line 771 of file time_standard.cc.

8.24.2.3 TimeStandard() [2/2]

```
jeod::TimeStandard::TimeStandard (
            const TimeStandard &  )  [private]
```

8.24.3 Member Function Documentation

8.24.3.1 add_type_initialize()

```
void jeod::TimeStandard::add_type_initialize (
            const int seeking_status,
            TimeManagerInit * time_manager_init )  [virtual]
```

Recursively adds elements to the initialization tree.

If the "parent" to a time-type is defined, adds the "parent" then returns to adding the "child" type. If the "parent" is not defined it searches for a suitable "parent" from the types already in the tree. If that search is successful, it adds the "child" to the tree, otherwise it returns without change.

Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize_from" for each time representation, except the top-level initializer type.
- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

Parameters

in	*seeking_status*	status-value for auto-seek
in	*time_manager_init*	The TM initializer.

Reimplemented from jeod::JeodBaseTime.

Definition at line 176 of file time_standard.cc.

References jeod::TimeManagerInit::get_conv_dir_init(), jeod::TimeManagerInit::get_status(), jeod::TimeManager↩ ::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::TimeManagerInit::increment_status(), jeod↩ ::JeodBaseTime::index, jeod::JeodBaseTime::initialize_from_name, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::TimeManager::num_types, jeod::TimeManagerInit::set_status(), jeod::Time↩ Manager::time_lookup(), and jeod::JeodBaseTime::time_manager.

8.24.3.2 calculate_calendar_values()

```
void jeod::TimeStandard::calculate_calendar_values (
            void ) [protected], [virtual]
```

Calculate Gregorian calendar date and 24-hour clock representation from Truncated Julian date/time.

Assumptions and Limitations

- Coverage is from March 1, 1600 onward.
- Produces a time in 24-hour clock format.
- Assumes that the values year, month, day, hour, minute, second, and truncated_julian_time are all present in the same class.

Reimplemented in jeod::TimeGPS, and jeod::TimeGMST.

Definition at line 295 of file time_standard.cc.

References calendar_day, calendar_hour, calendar_minute, calendar_month, calendar_second, calendar_year, jeod::JeodBaseTime::clock_resolution, prev_julian_day, and trunc_julian_time.

Referenced by calendar_update(), and seconds_of_year().

8.24.3.3 calendar_update()

```
void jeod::TimeStandard::calendar_update (
            double simtime )
```

Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation.

Makes sure that the time type on which it is called is up-to-date before doing so.

Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the update_tree was built.

Parameters

in	*simtime*	Simulation elapsed time, on the simulation clock Units: s

Definition at line 395 of file time_standard.cc.

References calculate_calendar_values(), last_calendar_update, jeod::TimeManager::simtime, jeod::JeodBase↩ Time::time_manager, and jeod::TimeManager::update().

8.24.3.4 convert_from_calendar()

```
void jeod::TimeStandard::convert_from_calendar (
            void ) [protected], [virtual]
```

Calculate Truncated Julian date/time from Gregorian calendar date and 24-hour clock representation.

Assumptions and Limitations

- Coverage s from March 1, 1600 onward.
- Assumes that time is in 24-hour clock format; 1:00:00 pm cannot be read correctly, but 13:00:00 can.
- Assumes that the values year, month, day, hour, minute, second, and truncated_julian_time are all present in the same class.

Reimplemented in jeod::TimeGPS.

Definition at line 425 of file time_standard.cc.

References calendar_day, calendar_hour, calendar_minute, calendar_month, calendar_second, calendar_year, jeod::JeodBaseTime::days, jeod::JeodBaseTime::seconds, tjt_at_epoch, and trunc_julian_time.

Referenced by initialize_initializer_time(), seconds_of_year(), and jeod::TimeUDE::set_epoch_std().

8.24.3.5 initialize_from_parent()

```
void jeod::TimeStandard::initialize_from_parent (
            TimeManagerInit * time_manager_init ) [virtual]
```

Initialize a time type from its parent on the initialization tree.

Assumptions and Limitations

- More than 1 time-type defined, otherwise this is not called.

Parameters

in	*time_manager_init*	The TM initializer.

Reimplemented from jeod::JeodBaseTime.

Definition at line 627 of file time_standard.cc.

References jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), jeod::TimeManagerInit::get_conv_dir_init(), jeod::TimeManagerInit::get_conv_ptr_index(), jeod::TimeManager::get_converter_ptr(), jeod::TimeManager::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::JeodBaseTime::index, jeod::JeodBaseTime::initial_value, jeod::TimeMessages::initialization_error, jeod::TimeConverter::initialize(), jeod::JeodBaseTime::initialize_from_name, jeod::JeodBaseTime::initialize_from_parent(), jeod::JeodBaseTime::initialized, jeod::TimeConverter::is_initialized(), jeod::JeodBaseTime::is_initialized(), jeod::TimeMessages::memory_error, jeod::JeodBaseTime::name, jeod::TimeManager::num_types, jeod::JeodBaseTime::seconds, jeod::TimeManager::time_lookup(), and jeod::JeodBaseTime::time_manager.

8.24.3.6 initialize_initializer_time()

```
void jeod::TimeStandard::initialize_initializer_time (
            TimeManagerInit * time_manager_init )  [virtual]
```

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

Assumptions and Limitations

- TimeDyn cannot be used as the initializer time.
- Each time representation can have its own initializer function, or can inherit the one in TimeStandard.

Parameters

in	*time_manager_init*	The TM initializer.

Implements jeod::JeodBaseTime.

Definition at line 484 of file time_standard.cc.

References jeod::TimeEnum::calendar, calendar_day, calendar_hour, calendar_minute, calendar_month, calendar_second, calendar_year, jeod::TimeEnum::clock, convert_from_calendar(), jeod::JeodBaseTime::days, jeod::TimeEnum::days_since_epoch, jeod::TimeMessages::incomplete_setup_error, jeod::JeodBaseTime::initial_value, jeod::JeodBaseTime::initialize_from_name, jeod::JeodBaseTime::initialized, jeod::JeodBaseTime::initializing_value, jeod::TimeMessages::invalid_data_error, jeod::TimeMessages::invalid_setup_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified_julian, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy_error, jeod::JeodBaseTime::seconds, jeod::TimeEnum::seconds_since_epoch, send_warning_pre_1968, jeod::TimeManagerInit::sim_start_format, tjt_at_epoch, trunc_julian_time, jeod::TimeEnum::truncated_julian, and jeod::TimeEnum::undefined.

8.24.3.7 julian_date_at_epoch()

```
double jeod::TimeStandard::julian_date_at_epoch (
             void  )
```

Returns the full Julian date at epoch, rather than the Truncated Julian Time.

Returns

Truncated Julian Time at the epoch of the time-type.
Units: day

Definition at line 148 of file time_standard.cc.

References tjt_at_epoch, and tjt_jd_offset.

8.24.3.8 operator=()

```
TimeStandard& jeod::TimeStandard::operator= (
             const TimeStandard &  )  [private]
```

8.24.3.9 seconds_of_year()

```
double jeod::TimeStandard::seconds_of_year (
             void  )
```

Generate the number of seconds elapsed this year.

Assumptions and Limitations

- Relies on the accuracy of the JEOD2.0 calendar.

Returns

Current second of year.

Definition at line 709 of file time_standard.cc.

References calculate_calendar_values(), calendar_day, calendar_hour, calendar_minute, calendar_month, calendar_second, calendar_year, convert_from_calendar(), jeod::JeodBaseTime::days, last_calendar_update, jeod::JeodBaseTime::seconds, seconds_at_year_start, jeod::TimeManager::simtime, jeod::JeodBaseTime::time↩_manager, trunc_julian_time, and year_of_last_soy.

8.24.3.10 set_epoch()

```
virtual void jeod::TimeStandard::set_epoch (
             void  )  [protected], [pure virtual]
```

Set the epoch time.

Implemented in jeod::TimeGPS, jeod::TimeUT1, jeod::TimeUTC, jeod::TimeGMST, jeod::TimeTDB, jeod::TimeTT, and jeod::TimeTAI.

8.24.3.11 set_time_by_days()

```
void jeod::TimeStandard::set_time_by_days (
             const double new_days )  [virtual]
```

Given a value of days, propagate to seconds and trunc_julian_time.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_days*	new value for days Units: day

Reimplemented from jeod::JeodBaseTime.

Reimplemented in jeod::TimeGPS.

Definition at line 112 of file time_standard.cc.

References jeod::JeodBaseTime::days, julian_date, jeod::JeodBaseTime::set_time_by_days(), tjt_at_epoch, tjt_jd_offset, and trunc_julian_time.

Referenced by jeod::TimeConverter_UT1_GMST::convert_a_to_b(), and jeod::TimeUDE::set_epoch_std().

8.24.3.12 set_time_by_seconds()

```
void jeod::TimeStandard::set_time_by_seconds (
             const double new_seconds )  [virtual]
```

Given a value of seconds, propagate to days and trunc_julian_time.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new_seconds*	new value for seconds Units: s

Reimplemented from jeod::JeodBaseTime.

Reimplemented in jeod::TimeGPS.

Definition at line 93 of file time_standard.cc.

References jeod::JeodBaseTime::days, julian_date, jeod::JeodBaseTime::set_time_by_seconds(), tjt_at_epoch, tjt_jd_offset, and trunc_julian_time.

Referenced by jeod::TimeConverter_TAI_TT::convert_a_to_b(), jeod::TimeConverter_Dyn_TAI::convert_a_to_b(), jeod::TimeConverter_Dyn_TDB::convert_a_to_b(), jeod::TimeConverter_TAI_TDB::convert_a_to_b(), jeod::Time↩ Converter_TAI_TT::convert_b_to_a(), jeod::TimeConverter_TAI_GPS::convert_b_to_a(), jeod::TimeConverter_↩ STD_UDE::convert_b_to_a(), jeod::TimeConverter_TAI_TDB::convert_b_to_a(), jeod::TimeUDE::set_epoch_std(), and jeod::TimeGPS::set_time_by_seconds().

8.24.3.13 set_time_by_trunc_julian()

```
void jeod::TimeStandard::set_time_by_trunc_julian (
             const double new_tjt )
```

Given a value of tjt, propagate to seconds and days.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	*new↩ _tjt*	new value for Truncated Julian Time Units: day

Definition at line 131 of file time_standard.cc.

References jeod::JeodBaseTime::days, julian_date, jeod::JeodBaseTime::seconds, tjt_at_epoch, tjt_jd_offset, and trunc_julian_time.

Referenced by jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UT1::convert_a_to↩ _b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), jeod::↩ TimeUDE::set_epoch_std(), and jeod::TimeGPS::set_time_by_trunc_julian().

8.24.4 Friends And Related Function Documentation

8.24.4.1 init_attrjeod__TimeStandard

```
void init_attrjeod__TimeStandard ( )  [friend]
```

8.24.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 91 of file time_standard.hh.

8.24.4.3 TimeUDE

```
friend class TimeUDE  [friend]
```

Definition at line 93 of file time_standard.hh.

8.24.5 Field Documentation

8.24.5.1 calendar_day

```
int jeod::TimeStandard::calendar_day
```

Gregorian calendar date day number.

trick_units(day)

Definition at line 161 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.2 calendar_hour

```
int jeod::TimeStandard::calendar_hour
```

24-hour clock hour number.

trick_units(hr)

Definition at line 166 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.3 calendar_minute

```
int jeod::TimeStandard::calendar_minute
```

Clock minute number.

trick_units(min)

Definition at line 171 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_↩ year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.4 calendar_month

```
int jeod::TimeStandard::calendar_month
```

Gregorian calendar month.

trick_units(–)

Definition at line 186 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_↩ year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.5 calendar_second

```
double jeod::TimeStandard::calendar_second
```

Clock second number.

trick_units(s)

Definition at line 176 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_↩ year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.6 calendar_year

```
int jeod::TimeStandard::calendar_year
```

Gregorian calendar year.

trick_units(–)

Definition at line 181 of file time_standard.hh.

Referenced by calculate_calendar_values(), convert_from_calendar(), initialize_initializer_time(), seconds_of_↩ year(), and jeod::TimeUDE::set_epoch_std().

8.24.5.7 julian_date

```
double jeod::TimeStandard::julian_date
```

Conventional Julian Date.

NOTE - because this value is typically so large, it has very little room for fine-detail precision. It should only ever be used as an output for the likes of terminal displays and for input to legacy code. Never use for newly developed code.trick_units(day)

Definition at line 151 of file time_standard.hh.

Referenced by set_time_by_days(), set_time_by_seconds(), and set_time_by_trunc_julian().

8.24.5.8 last_calendar_update

```
double jeod::TimeStandard::last_calendar_update
```

The simtime when the calendar update was last run.

trick_units(--)

Definition at line 101 of file time_standard.hh.

Referenced by calendar_update(), and seconds_of_year().

8.24.5.9 prev_julian_day

```
int jeod::TimeStandard::prev_julian_day
```

Used for determining whether to update the date in the calendar function.

trick_units(day)

Definition at line 107 of file time_standard.hh.

Referenced by calculate_calendar_values().

8.24.5.10 seconds_at_year_start

```
double jeod::TimeStandard::seconds_at_year_start
```

The value of "seconds" at the start of the year in which the last seconds_of_year calculation was made.

Used for seconds_of_year calculations only.trick_units(s)

Definition at line 114 of file time_standard.hh.

Referenced by seconds_of_year().

8.24.5.11 send_warning_pre_1968

```
bool jeod::TimeStandard::send_warning_pre_1968
```

This flag can be turned off by developers wanting to avoid warnings about a simulation being initialized pre-1968.

The flag defaults to true - warning will be sent.trick_units(–)

Definition at line 128 of file time_standard.hh.

Referenced by initialize_initializer_time().

8.24.5.12 tjt_at_epoch

```
double jeod::TimeStandard::tjt_at_epoch
```

Truncated Julian Date at epoch.

trick_units(day)

Definition at line 156 of file time_standard.hh.

Referenced by convert_from_calendar(), jeod::TimeConverter_TAI_GPS::initialize(), jeod::TimeConverter_TAI_TDB::initialize(), initialize_initializer_time(), julian_date_at_epoch(), jeod::TimeConverter_TAI_TDB::set_a_to_b_offset(), jeod::TimeTAI::set_epoch(), jeod::TimeTT::set_epoch(), jeod::TimeTDB::set_epoch(), jeod::TimeUTC::set_epoch(), jeod::TimeUT1::set_epoch(), jeod::TimeGPS::set_epoch(), set_time_by_days(), set_time_by_seconds(), and set_time_by_trunc_julian().

8.24.5.13 tjt_jd_offset

```
const double jeod::TimeStandard::tjt_jd_offset
```

Difference between Julian and Truncated Julian.

trick_units(day)

Definition at line 138 of file time_standard.hh.

Referenced by julian_date_at_epoch(), set_time_by_days(), set_time_by_seconds(), and set_time_by_trunc_julian().

8.24.5.14 tjt_mjt_offset

```
const double jeod::TimeStandard::tjt_mjt_offset
```

Difference between Truncated Julian and Modified Julian.

trick_units(day)

Definition at line 133 of file time_standard.hh.

8.24.5.15 trunc_julian_time

```
double jeod::TimeStandard::trunc_julian_time
```

Truncated Julian time for this time-type.

trick_units(day)

Definition at line 143 of file time_standard.hh.

Referenced by calculate_calendar_values(), jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UT1::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), convert_from_calendar(), jeod::TimeConverter_TAI_UTC::initialize(), jeod::TimeConverter_TAI_UT1::initialize(), initialize_initializer_time(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), seconds_of_year(), jeod::TimeConverter_TAI_TDB::set_a_to_b_offset(), set_time_by_days(), set_time_by_seconds(), set_time_by_trunc_julian(), jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends(), and jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends().

8.24.5.16 year_of_last_soy

```
int jeod::TimeStandard::year_of_last_soy
```

The year in which the last seconds_of_year calculation was made.

At the start of this year, seconds had value seconds_at_year_start. Used for seconds_of_year calculations only.↩
trick_units(–)

Definition at line 121 of file time_standard.hh.

Referenced by seconds_of_year().

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

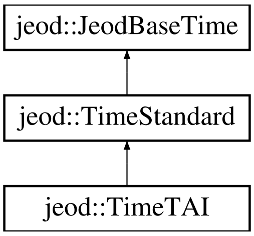
- time_standard.hh
- time_standard.cc

8.25 jeod::TimeTAI Class Reference

Represents International Atomic Time.

```
#include <time_tai.hh>
```

Inheritance diagram for jeod::TimeTAI:



Public Member Functions

- TimeTAI ()

  Construct a Time_TAI.
- ∼TimeTAI ()

  Destroy a Time_TAI.

Private Member Functions

- TimeTAI (const TimeTAI &)
- TimeTAI & operator= (const TimeTAI &)
- void set_epoch (void)

  Sets the epoch for TAI time.

Friends

- class InputProcessor
- void init_attrjeod__TimeTAI ()

Additional Inherited Members

8.25.1 Detailed Description

Represents International Atomic Time.

Definition at line 81 of file time_tai.hh.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 TimeTAI() [1/2]

```
jeod::TimeTAI::TimeTAI (
            void  )
```

Construct a Time_TAI.

Definition at line 50 of file time_tai.cc.

References jeod::JeodBaseTime::name, and set_epoch().

8.25.2.2 ∼TimeTAI()

```
jeod::TimeTAI::∼TimeTAI (
            void )
```

Destroy a Time_TAI.

Definition at line 74 of file time_tai.cc.

8.25.2.3 TimeTAI() [2/2]

```
jeod::TimeTAI::TimeTAI (
            const TimeTAI & ) [private]
```

8.25.3 Member Function Documentation

8.25.3.1 operator=()

```
TimeTAI& jeod::TimeTAI::operator= (
            const TimeTAI & ) [private]
```

8.25.3.2 set_epoch()

```
void jeod::TimeTAI::set_epoch (
            void ) [private], [virtual]
```

Sets the epoch for TAI time.

Implements jeod::TimeStandard.

Definition at line 62 of file time_tai.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeTAI().

8.25.4 Friends And Related Function Documentation

8.25.4.1 init_attrjeod__TimeTAI

```
void init_attrjeod__TimeTAI ( )  [friend]
```

8.25.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_tai.hh.

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

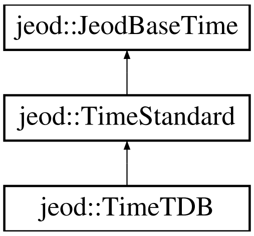
- time_tai.hh
- time_tai.cc

8.26 jeod::TimeTDB Class Reference

Represents Barycentric Dynamic Time.

```
#include <time_tdb.hh>
```

Inheritance diagram for jeod::TimeTDB:



Public Member Functions

- TimeTDB ()

  Construct a Time_TDB.
- ∼TimeTDB ()

  Destroy a Time_TDB.

Private Member Functions

- TimeTDB (const TimeTDB &)
- TimeTDB & operator= (const TimeTDB &)
- void set_epoch (void)

  Sets the epoch for TDB time.

Friends

- class InputProcessor
- void init_attrjeod__TimeTDB ()

Additional Inherited Members

8.26.1 Detailed Description

Represents Barycentric Dynamic Time.

Definition at line 82 of file time_tdb.hh.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 TimeTDB() [1/2]

```
jeod::TimeTDB::TimeTDB (
            void   )
```

Construct a Time_TDB.

Definition at line 51 of file time_tdb.cc.

References jeod::JeodBaseTime::name, and set_epoch().

8.26.2.2 ∼TimeTDB()

```
jeod::TimeTDB::∼TimeTDB (
            void   )
```

Destroy a Time_TDB.

Definition at line 75 of file time_tdb.cc.

8.26.2.3 TimeTDB() [2/2]

```
jeod::TimeTDB::TimeTDB (
            const TimeTDB &   )   [private]
```

8.26.3 Member Function Documentation

8.26.3.1 operator=()

```
TimeTDB& jeod::TimeTDB::operator= (
             const TimeTDB &  )  [private]
```

8.26.3.2 set_epoch()

```
void jeod::TimeTDB::set_epoch (
             void  )  [private], [virtual]
```

Sets the epoch for TDB time.

Implements jeod::TimeStandard.

Definition at line 63 of file time_tdb.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeTDB().

8.26.4 Friends And Related Function Documentation

8.26.4.1 init_attrjeod__TimeTDB

```
void init_attrjeod__TimeTDB ( )  [friend]
```

8.26.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_tdb.hh.

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

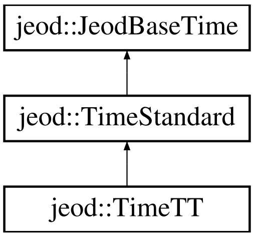
- time_tdb.hh
- time_tdb.cc

8.27 jeod::TimeTT Class Reference

Represents Terrestrial Time.

```
#include <time_tt.hh>
```

Inheritance diagram for jeod::TimeTT:



Public Member Functions

- TimeTT ()

  Construct a Time_TT.
- ∼TimeTT ()

  Destroy a Time_TT.

Private Member Functions

- TimeTT (const TimeTT &)
- TimeTT & operator= (const TimeTT &)
- void set_epoch (void)

  Sets the epoch for TT time.

Friends

- class InputProcessor
- void init_attrjeod__TimeTT ()

Additional Inherited Members

8.27.1 Detailed Description

Represents Terrestrial Time.

Definition at line 82 of file time_tt.hh.

8.27.2 Constructor & Destructor Documentation

8.27.2.1 TimeTT() [1/2]

```
jeod::TimeTT::TimeTT (
             void  )
```

Construct a Time_TT.

Definition at line 51 of file time_tt.cc.

References jeod::JeodBaseTime::name, and set_epoch().

8.27.2.2 ∼TimeTT()

```
jeod::TimeTT::∼TimeTT (
             void  )
```

Destroy a Time_TT.

Definition at line 75 of file time_tt.cc.

8.27.2.3 TimeTT() [2/2]

```
jeod::TimeTT::TimeTT (
             const TimeTT &  )  [private]
```

8.27.3 Member Function Documentation

8.27.3.1 operator=()

```
TimeTT& jeod::TimeTT::operator= (
             const TimeTT &  )  [private]
```

8.27.3.2 set_epoch()

```
void jeod::TimeTT::set_epoch (
             void  )  [private], [virtual]
```

Sets the epoch for TT time.

Implements jeod::TimeStandard.

Definition at line 63 of file time_tt.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeTT().

8.27.4 Friends And Related Function Documentation

8.27.4.1 init_attrjeod__TimeTT

```
void init_attrjeod__TimeTT ( )  [friend]
```

8.27.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_tt.hh.

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

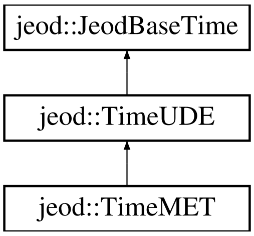
- time_tt.hh
- time_tt.cc

8.28 jeod::TimeUDE Class Reference

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

```
#include <time_ude.hh>
```

Inheritance diagram for jeod::TimeUDE:



Public Member Functions

- TimeUDE ()

  Constructor for class TimeUDE.
- virtual ∼TimeUDE ()

  Destructor for TimeUDE.
- void initialize_initializer_time (TimeManagerInit ∗tm_init)

  Each time type is initialized from its parent in the initialization tree, except one.
- void add_type_initialize (const int seeking_status, TimeManagerInit ∗tm_init)

  Adds a UDE type to the initialization tree when it is appropriate to do so.
- void initialize_from_parent (TimeManagerInit ∗tm_init)

  Initializes this time-type.
- void set_time_by_clock (void)

  sets the decimal representation of time by the clock
- void set_time_by_seconds (const double new_seconds)

  Given a seconds value, sets days and clock values.
- void set_time_by_days (const double new_days)

  Given a seconds value, sets days and clock values.
- void set_epoch_initializing_value (const double simtime, const double epoch_initializing_value)

  sets the initial epoch value

Data Fields

- int epoch_year

  Gregorian calendar year number at epoch.
- int epoch_month

  Gregorian calendar month number at epoch.
- int epoch_day

  Gregorian calendar day number at epoch.
- int epoch_hour

  24-hour clock hour number at epoch.
- int epoch_minute

  Clock minute number at epoch.
- double epoch_second

  Clock seconds value at epoch.
- int clock_day

  Whole number of days since epoch, in clock format.
- int clock_hour

  Whole number of hours since epoch, in clock format.
- int clock_minute

  Whole number of minutes since epoch, in clock format.
- double clock_second

  Number of seconds since epoch, in clock format.
- double last_clock_update

  Simtime at the last time the clock was updated.
- TimeEnum::TimeFormat epoch_format

  Format for expressing the epoch of this type (calendar, julian, etc)
- TimeEnum::TimeFormat initial_value_format

  Format for expressing the initial value of this type (calendar, julian, etc)
- std::string epoch_defined_in_name

  Name of time type in which epoch defined.

Protected Member Functions

- bool must_be_singleton ()

    Returns false in response to the question "does this time class have to be a singleton".
- void convert_epoch_to_update (JeodBaseTime ∗epoch_ptr, JeodBaseTime ∗update_ptr, TimeManagerInit ∗tm_init)

    Converts the time, as specified in the epoch time-type to the update_from time-type.
- void set_epoch_dyn (TimeDyn ∗epoch_ptr)

    Temporarily overwrites the simulation data in time type "epoch" with the epoch value.
- void set_epoch_times (JeodBaseTime ∗epoch_ptr)

    To set the times in the epoch time type coincident with the zero-point of this time-type.
- void set_epoch_ude (TimeUDE ∗epoch_ptr)

    Overwrites the data in time type "epoch" with that in this class that specifies the epoch.
- void set_epoch_std (TimeStandard ∗epoch_ptr)

    Overwrites the data in time type "epoch" with that in this class that specifies the epoch.
- void set_initial_times (void)

    Sets the initial value of this type from the myriad of initialization options.
- void clock_update ()

    converts the decimal seconds value to a clock interface
- void verify_epoch (void)

    Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.
- void verify_init (void)

    Verifies that any assignment to initialize_from is flagged as inappropriate, and tests for the presence of initializing data.
- void verify_update (void)

    Ensures that the time-type identified as "update_from" is legitimate.

Protected Attributes

- double epoch_initializing_value

    Value of epoch in appropriate format.
- bool initializing_data_present

    Whether initializing data is present.
- bool epoch_data_present

    Whether epoch data is present.
- bool epoch_value_is_set_number

    Whether there is some numerical input that could set epoch.
- bool epoch_value_is_set_calendar

    Whether there is some calendar input that could set epoch.
- bool epoch_value_is_set_clock

    Whether there is some clock input that could set epoch.
- int update_index

    The index of the time-type from which this one is updated.
- int epoch_index

    The index of the time-type in which this one's epoch is defined.

Private Member Functions

- TimeUDE (const TimeUDE &)
- TimeUDE & operator= (const TimeUDE &)

Friends

- class InputProcessor
- void init_attrjeod__TimeUDE ()

8.28.1 Detailed Description

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

Definition at line 89 of file time_ude.hh.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 TimeUDE() [1/2]

```
jeod::TimeUDE::TimeUDE (
              void  )
```

Constructor for class TimeUDE.

Assumptions and Limitations

- None

Definition at line 64 of file time_ude.cc.

8.28.2.2 ∼TimeUDE()

```
jeod::TimeUDE::∼TimeUDE (
              void  )  [virtual]
```

Destructor for TimeUDE.

Definition at line 1472 of file time_ude.cc.

8.28.2.3 TimeUDE() [2/2]

```
jeod::TimeUDE::TimeUDE (
              const TimeUDE &  )  [private]
```

8.28.3 Member Function Documentation

8.28.3.1 add_type_initialize()

```
void jeod::TimeUDE::add_type_initialize (
            const int seeking_status,
            TimeManagerInit * time_manager_init )  [virtual]
```

Adds a UDE type to the initialization tree when it is appropriate to do so.

Assumptions and Limitations

- The time type from which the UDE updates must be in the tree above the UDE.
- If the time type in which the epoch is defined is another UDE, it also must be in the tree above this UDE
- This function is only called when the UDE is NOT being used to initialize the simulation.

Parameters

in	*seeking_status*	An indicator of relative level of progression in the tree.
in	*time_manager_init*	The TM initializer.

Reimplemented from jeod::JeodBaseTime.

Definition at line 125 of file time_ude.cc.

References epoch_defined_in_name, epoch_index, jeod::TimeManagerInit::get_conv_dir_init(), jeod::TimeManagerInit::get_status(), jeod::TimeManager::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::TimeManagerInit::increment_status(), jeod::JeodBaseTime::index, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::TimeManager::num_types, jeod::TimeManagerInit::set_status(), jeod::JeodBaseTime::time_manager, jeod::JeodBaseTime::update_from_name, update_index, verify_epoch(), and verify_update().

8.28.3.2 clock_update()

```
void jeod::TimeUDE::clock_update (
            void )  [protected]
```

converts the decimal seconds value to a clock interface

Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1280 of file time_ude.cc.

References clock_day, clock_hour, clock_minute, jeod::JeodBaseTime::clock_resolution, clock_second, and jeod::JeodBaseTime::seconds.

Referenced by set_time_by_days(), and set_time_by_seconds().

8.28.3.3 convert_epoch_to_update()

```
void jeod::TimeUDE::convert_epoch_to_update (
            JeodBaseTime * epoch_ptr,
            JeodBaseTime * update_from_ptr,
            TimeManagerInit * time_manager_init ) [protected]
```

Converts the time, as specified in the epoch time-type to the update_from time-type.

This sets the update_from time at the epoch of "this", and allows for the initialization of the converter.

Assumptions and Limitations

- That there is a converter available to do this in one step
- Future work may include an extension to this routine to cover other cases.

Parameters

in	*epoch_ptr*	pointer to the epoch time-type
in	*update_from_ptr*	pointer to the time-type from which this time-type will be updated.
in	*time_manager_init*	The TM initializer.

Definition at line 252 of file time_ude.cc.

References jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), epoch_defined_in_name, epoch_index, jeod::TimeManagerInit::get_conv_dir_init(), jeod::TimeManagerInit::get_conv_ptr_index(), jeod::TimeManager::get_converter_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::TimeConverter::initialize(), jeod::JeodBaseTime::name, jeod::TimeManager::num_types, jeod::TimeConverter::override_initialized(), jeod::JeodBaseTime::override_initialized(), jeod::JeodBaseTime::time_manager, jeod::JeodBaseTime::update_from_name, and update_index.

Referenced by initialize_from_parent(), and initialize_initializer_time().

8.28.3.4 initialize_from_parent()

```
void jeod::TimeUDE::initialize_from_parent (
            TimeManagerInit * time_manager_init ) [virtual]
```

Initializes this time-type.

Assumptions and Limitations

- The subject object has a parent, a time-type with which it ticks. This has already been tested for.

Parameters

in	*time_manager_init*	The TM initializer.

Reimplemented from jeod::JeodBaseTime.

Definition at line 316 of file time_ude.cc.

References jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), convert_epoch_to_↩update(), jeod::JeodBaseTime::days, epoch_data_present, epoch_index, jeod::TimeManagerInit::get_conv_dir_↩init(), jeod::TimeManagerInit::get_conv_ptr_index(), jeod::TimeManager::get_converter_ptr(), jeod::TimeManager↩::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::JeodBaseTime::index, jeod::JeodBase↩Time::initial_value, jeod::TimeMessages::initialization_error, jeod::TimeConverter::initialize(), jeod::JeodBase↩Time::initialize_from_parent(), jeod::JeodBaseTime::initialized, initializing_data_present, jeod::TimeConverter↩::is_initialized(), jeod::JeodBaseTime::is_initialized(), jeod::JeodBaseTime::name, jeod::TimeManager::num_↩types, jeod::JeodBaseTime::override_initialized(), jeod::TimeMessages::redundancy_error, jeod::JeodBaseTime↩::seconds, set_epoch_times(), jeod::JeodBaseTime::set_time_by_seconds(), jeod::JeodBaseTime::time_manager, jeod::JeodBaseTime::update_from_name, update_index, and verify_init().

8.28.3.5 initialize_initializer_time()

```
void jeod::TimeUDE::initialize_initializer_time (
            TimeManagerInit * time_manager_init )  [virtual]
```

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

Parameters

in	*time_manager_init*	The TM initializer.

Implements jeod::JeodBaseTime.

Definition at line 511 of file time_ude.cc.

References jeod::TimeConverter::convert_a_to_b(), jeod::TimeConverter::convert_b_to_a(), convert_epoch_to↩_update(), jeod::JeodBaseTime::days, jeod::TimeMessages::duplicate_methods, epoch_data_present, epoch↩_index, jeod::TimeManagerInit::get_conv_dir_init(), jeod::TimeManagerInit::get_conv_ptr_index(), jeod::Time↩Manager::get_converter_ptr(), jeod::TimeManager::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::JeodBaseTime::index, initial_value_format, jeod::TimeConverter::initialize(), jeod::JeodBaseTime::initialized, initializing_data_present, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::Time↩Manager::num_types, jeod::JeodBaseTime::override_initialized(), jeod::JeodBaseTime::seconds, set_epoch↩_times(), jeod::TimeManagerInit::sim_start_format, jeod::JeodBaseTime::time_manager, jeod::TimeManager↩::time_standards_exist(), jeod::TimeEnum::undefined, jeod::JeodBaseTime::update_from_name, update_index, verify_epoch(), verify_init(), and verify_update().

8.28.3.6 must_be_singleton()

```
bool jeod::TimeUDE::must_be_singleton (
            void )  [protected], [virtual]
```

Returns false in response to the question "does this time class have to be a singleton".

Assumptions and Limitations

- There can be more than one UDE

Returns

false

Reimplemented from jeod::JeodBaseTime.

Definition at line 104 of file time_ude.cc.

8.28.3.7 operator=()

```
TimeUDE& jeod::TimeUDE::operator= (
             const TimeUDE & )  [private]
```

8.28.3.8 set_epoch_dyn()

```
void jeod::TimeUDE::set_epoch_dyn (
             TimeDyn * epoch_ptr )  [protected]
```

Temporarily overwrites the simulation data in time type "epoch" with the epoch value.

Assumptions and Limitations

- "Epoch" is DynTime

Parameters

in	*epoch_ptr*	pointer to the epoch time-type

Definition at line 754 of file time_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, jeod::TimeEnum::days_since_epoch, epoch_data_present, epoch_day, epoch_defined_in_name, epoch_format, epoch_hour, epoch_initializing_value, epoch_minute, epoch_month, epoch_value_is_set_number, epoch_year, jeod::TimeMessages::incomplete_setup_error, initializing_data_present, jeod::TimeMessages::invalid_setup_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified_julian, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy_error, jeod::TimeEnum::seconds_since_epoch, jeod::JeodBaseTime::set_time_by_days(), jeod::JeodBaseTime::set_time_by_seconds(), jeod::TimeEnum::truncated_julian, and jeod::TimeEnum::undefined.

Referenced by set_epoch_times().

8.28.3.9 set_epoch_initializing_value()

```
void jeod::TimeUDE::set_epoch_initializing_value (
            const double simtime,
            const double epoch )
```

sets the initial epoch value

Assumptions and Limitations

- Assumes that the number that is passed in is correctly entered with the correct units interpretation.

Parameters

in	*simtime*	Used to verify that this is at initialization
in	*epoch*	the value to be used.

Definition at line 1256 of file time_ude.cc.

References epoch_initializing_value, jeod::TimeMessages::invalid_setup_error, and jeod::JeodBaseTime::name.

8.28.3.10 set_epoch_std()

```
void jeod::TimeUDE::set_epoch_std (
            TimeStandard * epoch_ptr )  [protected]
```

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

Assumptions and Limitations

- "Epoch" is Absolute Derived Time

Parameters

in	*epoch_ptr*	pointer to the epoch time-type

Definition at line 859 of file time_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeStandard::calendar_day, jeod::TimeStandard::calendar_hour, jeod::TimeStandard::calendar_minute, jeod::TimeStandard::calendar_month, jeod::TimeStandard::calendar_second, jeod::TimeStandard::calendar_year, jeod::TimeEnum::clock, jeod::TimeStandard::convert_from_calendar(), jeod::TimeEnum::days_since_epoch, epoch_day, epoch_defined_in_name, epoch_format, epoch_hour, epoch_initializing_value, epoch_minute, epoch_month, epoch_second, epoch_value_is_set_calendar, epoch_value_is_set_number, epoch_year, jeod::TimeMessages::incomplete_setup_error, jeod::TimeMessages::invalid_setup_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified_julian, jeod::JeodBaseTime::name, jeod::TimeEnum::seconds_since_epoch, jeod::TimeStandard::set_time_by_days(), jeod::TimeStandard::set_time_by_seconds(), jeod::TimeStandard::set_time_by_trunc_julian(), jeod::TimeEnum::truncated_julian, and jeod::TimeEnum::undefined.

Referenced by set_epoch_times().

8.28.3.11 set_epoch_times()

```
void jeod::TimeUDE::set_epoch_times (
            JeodBaseTime * epoch_ptr )  [protected]
```

To set the times in the epoch time type coincident with the zero-point of this time-type.

Assumptions and Limitations

- "This" is being defined by epoch.

Parameters

in	*epoch_ptr*	pointer to the epoch time-type

Definition at line 717 of file time_ude.cc.

References jeod::TimeMessages::invalid_setup_error, set_epoch_dyn(), set_epoch_std(), and set_epoch_ude().

Referenced by initialize_from_parent(), and initialize_initializer_time().

8.28.3.12 set_epoch_ude()

```
void jeod::TimeUDE::set_epoch_ude (
            TimeUDE * epoch_ptr )  [protected]
```

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

Assumptions and Limitations

- "Epoch" is a User-Defined-Epoch Time.

Parameters

in	*epoch_ptr*	pointer to the epoch time-type

Definition at line 993 of file time_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, clock_day, clock_hour, clock_minute, clock_second, jeod::TimeEnum::days_since_epoch, epoch_day, epoch_defined_in_name, epoch_format, epoch_hour, epoch_initializing_value, epoch_minute, epoch_second, epoch_value_is_set_clock, epoch_value_is_set_number, jeod::TimeMessages::incomplete_setup_error, jeod::TimeMessages::invalid_setup_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified_julian, jeod::JeodBaseTime::name, jeod::Time↩

Enum::seconds_since_epoch, set_time_by_clock(), set_time_by_seconds(), jeod::TimeEnum::truncated_julian, and jeod::TimeEnum::undefined.

Referenced by set_epoch_times().

8.28.3.13 set_initial_times()

```
void jeod::TimeUDE::set_initial_times (
             void  )  [protected]
```

Sets the initial value of this type from the myriad of initialization options.

Assumptions and Limitations

- At least one of the following is non-zero: initializing value, clock_day, clock_hour, clock_minute, clock_↩ second, seconds, days

Definition at line 1095 of file time_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, clock_day, clock_hour, clock_minute, clock_↩ second, jeod::JeodBaseTime::days, jeod::TimeEnum::days_since_epoch, jeod::TimeMessages::incomplete_↩ setup_error, initial_value_format, initializing_data_present, jeod::JeodBaseTime::initializing_value, jeod::Time↩ Messages::invalid_setup_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeMessages::memory↩ _error, jeod::TimeEnum::modified_julian, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy_error, jeod::JeodBaseTime::seconds, jeod::TimeEnum::seconds_since_epoch, jeod::TimeEnum::truncated_julian, and jeod::TimeEnum::undefined.

Referenced by verify_init().

8.28.3.14 set_time_by_clock()

```
void jeod::TimeUDE::set_time_by_clock (
             void  )
```

sets the decimal representation of time by the clock

Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1236 of file time_ude.cc.

References clock_day, clock_hour, clock_minute, clock_second, jeod::JeodBaseTime::days, and jeod::JeodBase↩ Time::seconds.

Referenced by set_epoch_ude().

8.28.3.15 set_time_by_days()

```
void jeod::TimeUDE::set_time_by_days (
             const double new_days )  [virtual]
```

Given a seconds value, sets days and clock values.

Parameters

in	*new_days*	new value for days Units: day

Reimplemented from jeod::JeodBaseTime.

Definition at line 1203 of file time_ude.cc.

References clock_update(), and jeod::JeodBaseTime::set_time_by_days().

8.28.3.16 set_time_by_seconds()

```
void jeod::TimeUDE::set_time_by_seconds (
            const double new_seconds )  [virtual]
```

Given a seconds value, sets days and clock values.

Parameters

in	*new_seconds*	new value for seconds Units: s

Reimplemented from jeod::JeodBaseTime.

Definition at line 1217 of file time_ude.cc.

References clock_update(), and jeod::JeodBaseTime::set_time_by_seconds().

Referenced by jeod::TimeConverter_Dyn_UDE::convert_a_to_b(), jeod::TimeConverter_STD_UDE::convert_a_to_b(), and set_epoch_ude().

8.28.3.17 verify_epoch()

```
void jeod::TimeUDE::verify_epoch (
            void )  [protected]
```

Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.

Definition at line 1312 of file time_ude.cc.

References epoch_data_present, epoch_day, epoch_defined_in_name, epoch_format, epoch_hour, epoch_index, epoch_initializing_value, epoch_minute, epoch_month, epoch_second, epoch_value_is_set_calendar, epoch_value_is_set_clock, epoch_value_is_set_number, epoch_year, jeod::TimeMessages::incomplete_setup_error, jeod::JeodBaseTime::index, jeod::TimeMessages::invalid_setup_error, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy_error, jeod::TimeManager::time_lookup(), jeod::JeodBaseTime::time_manager, and jeod::TimeEnum::undefined.

Referenced by add_type_initialize(), and initialize_initializer_time().

8.28.3.18 verify_init()

```
void jeod::TimeUDE::verify_init (
            void ) [protected]
```

Verifies that any assignment to initialize_from is flagged as inappropriate, and tests for the presence of initializing data.

Definition at line 1406 of file time_ude.cc.

References jeod::JeodBaseTime::initialize_from_name, jeod::JeodBaseTime::name, jeod::TimeMessages↩ ::redundancy_error, set_initial_times(), jeod::TimeManager::time_lookup(), and jeod::JeodBaseTime::time_↩ manager.

Referenced by initialize_from_parent(), and initialize_initializer_time().

8.28.3.19 verify_update()

```
void jeod::TimeUDE::verify_update (
            void ) [protected]
```

Ensures that the time-type identified as "update_from" is legitimate.

Definition at line 1440 of file time_ude.cc.

References jeod::TimeManager::get_time_ptr(), jeod::TimeMessages::incomplete_setup_error, jeod::Time↩ Messages::invalid_setup_error, jeod::TimeManager::time_lookup(), jeod::JeodBaseTime::time_manager, jeod↩ ::JeodBaseTime::update_from_name, and update_index.

Referenced by add_type_initialize(), and initialize_initializer_time().

8.28.4 Friends And Related Function Documentation

8.28.4.1 init_attrjeod__TimeUDE

```
void init_attrjeod__TimeUDE ( ) [friend]
```

8.28.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 91 of file time_ude.hh.

8.28.5 Field Documentation

8.28.5.1 clock_day

```
int jeod::TimeUDE::clock_day
```

Whole number of days since epoch, in clock format.

trick_units(–)

Definition at line 123 of file time_ude.hh.

Referenced by clock_update(), set_epoch_ude(), set_initial_times(), and set_time_by_clock().

8.28.5.2 clock_hour

```
int jeod::TimeUDE::clock_hour
```

Whole number of hours since epoch, in clock format.

trick_units(–)

Definition at line 128 of file time_ude.hh.

Referenced by clock_update(), set_epoch_ude(), set_initial_times(), and set_time_by_clock().

8.28.5.3 clock_minute

```
int jeod::TimeUDE::clock_minute
```

Whole number of minutes since epoch, in clock format.

trick_units(–)

Definition at line 133 of file time_ude.hh.

Referenced by clock_update(), set_epoch_ude(), set_initial_times(), and set_time_by_clock().

8.28.5.4 clock_second

```
double jeod::TimeUDE::clock_second
```

Number of seconds since epoch, in clock format.

trick_units(s)

Definition at line 138 of file time_ude.hh.

Referenced by clock_update(), set_epoch_ude(), set_initial_times(), and set_time_by_clock().

8.28.5.5 epoch_data_present

```
bool jeod::TimeUDE::epoch_data_present  [protected]
```

Whether epoch data is present.

trick_units(–)

Definition at line 174 of file time_ude.hh.

Referenced by initialize_from_parent(), initialize_initializer_time(), set_epoch_dyn(), and verify_epoch().

8.28.5.6 epoch_day

```
int jeod::TimeUDE::epoch_day
```

Gregorian calendar day number at epoch.

trick_units(day)

Definition at line 106 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.7 epoch_defined_in_name

```
std::string jeod::TimeUDE::epoch_defined_in_name
```

Name of time type in which epoch defined.

trick_units(–)

Definition at line 158 of file time_ude.hh.

Referenced by add_type_initialize(), convert_epoch_to_update(), set_epoch_dyn(), set_epoch_std(), set_epoch↵_ude(), and verify_epoch().

8.28.5.8 epoch_format

```
TimeEnum::TimeFormat jeod::TimeUDE::epoch_format
```

Format for expressing the epoch of this type (calendar, julian, etc)

trick_units(–)

Definition at line 148 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.9 epoch_hour

```
int jeod::TimeUDE::epoch_hour
```

24-hour clock hour number at epoch.

trick_units(hr)

Definition at line 110 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.10 epoch_index

```
int jeod::TimeUDE::epoch_index  [protected]
```

The index of the time-type in which this one's epoch is defined.

trick_units(–)

Definition at line 199 of file time_ude.hh.

Referenced by add_type_initialize(), convert_epoch_to_update(), initialize_from_parent(), initialize_initializer_↩ time(), and verify_epoch().

8.28.5.11 epoch_initializing_value

```
double jeod::TimeUDE::epoch_initializing_value  [protected]
```

Value of epoch in appropriate format.

trick_units(–)

Definition at line 164 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_initializing_value(), set_epoch_std(), set_epoch_ude(), and verify_↩ epoch().

8.28.5.12 epoch_minute

```
int jeod::TimeUDE::epoch_minute
```

Clock minute number at epoch.

trick_units(min)

Definition at line 114 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.13 epoch_month

```
int jeod::TimeUDE::epoch_month
```

Gregorian calendar month number at epoch.

trick_units(–)

Definition at line 102 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), and verify_epoch().

8.28.5.14 epoch_second

```
double jeod::TimeUDE::epoch_second
```

Clock seconds value at epoch.

trick_units(s)

Definition at line 118 of file time_ude.hh.

Referenced by set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.15 epoch_value_is_set_calendar

```
bool jeod::TimeUDE::epoch_value_is_set_calendar  [protected]
```

Whether there is some calendar input that could set epoch.

trick_units(–)

Definition at line 184 of file time_ude.hh.

Referenced by set_epoch_std(), and verify_epoch().

8.28.5.16 epoch_value_is_set_clock

```
bool jeod::TimeUDE::epoch_value_is_set_clock  [protected]
```

Whether there is some clock input that could set epoch.

trick_units(–)

Definition at line 189 of file time_ude.hh.

Referenced by set_epoch_ude(), and verify_epoch().

8.28.5.17 epoch_value_is_set_number

```
bool jeod::TimeUDE::epoch_value_is_set_number  [protected]
```

Whether there is some numerical input that could set epoch.

trick_units(–)

Definition at line 179 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), set_epoch_ude(), and verify_epoch().

8.28.5.18 epoch_year

```
int jeod::TimeUDE::epoch_year
```

Gregorian calendar year number at epoch.

trick_units(–)

Definition at line 98 of file time_ude.hh.

Referenced by set_epoch_dyn(), set_epoch_std(), and verify_epoch().

8.28.5.19 initial_value_format

```
TimeEnum::TimeFormat jeod::TimeUDE::initial_value_format
```

Format for expressing the initial value of this type (calendar, julian, etc)

trick_units(–)

Definition at line 153 of file time_ude.hh.

Referenced by initialize_initializer_time(), and set_initial_times().

8.28.5.20 initializing_data_present

```
bool jeod::TimeUDE::initializing_data_present [protected]
```

Whether initializing data is present.

trick_units(–)

Definition at line 169 of file time_ude.hh.

Referenced by initialize_from_parent(), initialize_initializer_time(), set_epoch_dyn(), and set_initial_times().

8.28.5.21 last_clock_update

```
double jeod::TimeUDE::last_clock_update
```

Simtime at the last time the clock was updated.

trick_units(s)

Definition at line 143 of file time_ude.hh.

8.28.5.22 update_index

```
int jeod::TimeUDE::update_index [protected]
```

The index of the time-type from which this one is updated.

trick_units(–)

Definition at line 194 of file time_ude.hh.

Referenced by add_type_initialize(), convert_epoch_to_update(), initialize_from_parent(), initialize_initializer_↩ time(), and verify_update().

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

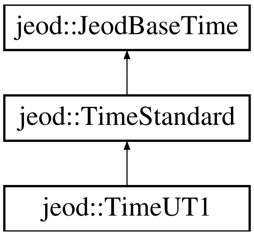
- time_ude.hh
- time_ude.cc

8.29 jeod::TimeUT1 Class Reference

Represents Universal Time.

```
#include <time_ut1.hh>
```

Inheritance diagram for jeod::TimeUT1:



Public Member Functions

- TimeUT1 ()

  Construct a Time_UT1.
- ∼TimeUT1 ()

  Destroy a Time_UT1.
- double get_days ()

  Accesses days.

Data Fields

- bool true_ut1

  "False" for comparison with older versions of JEOD

Private Member Functions

- TimeUT1 (const TimeUT1 &)
- TimeUT1 & operator= (const TimeUT1 &)
- void set_epoch (void)

  Sets the epoch for UT1 time.

Friends

- class InputProcessor
- void init_attrjeod__TimeUT1 ()

Additional Inherited Members

8.29.1 Detailed Description

Represents Universal Time.

Definition at line 82 of file time_ut1.hh.

8.29.2 Constructor & Destructor Documentation

8.29.2.1 TimeUT1() [1/2]

```
jeod::TimeUT1::TimeUT1 (
            void )
```

Construct a Time_UT1.

Definition at line 51 of file time_ut1.cc.

References jeod::JeodBaseTime::name, set_epoch(), and true_ut1.

8.29.2.2 ∼TimeUT1()

```
jeod::TimeUT1::∼TimeUT1 (
            void )
```

Destroy a Time_UT1.

Definition at line 85 of file time_ut1.cc.

8.29.2.3 TimeUT1() [2/2]

```
jeod::TimeUT1::TimeUT1 (
            const TimeUT1 & ) [private]
```

8.29.3 Member Function Documentation

8.29.3.1 get_days()

```
double jeod::TimeUT1::get_days (
            void )
```

Accesses days.

Returns

> days value
> Units: d

Definition at line 75 of file time_ut1.cc.

References jeod::JeodBaseTime::days.

Referenced by jeod::TimeConverter_UT1_GMST::convert_a_to_b().

8.29.3.2 operator=()

```
TimeUT1& jeod::TimeUT1::operator= (
             const TimeUT1 &  )  [private]
```

8.29.3.3 set_epoch()

```
void jeod::TimeUT1::set_epoch (
             void  )  [private], [virtual]
```

Sets the epoch for UT1 time.

Implements jeod::TimeStandard.

Definition at line 63 of file time_ut1.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeUT1().

8.29.4 Friends And Related Function Documentation

8.29.4.1 init_attrjeod__TimeUT1

```
void init_attrjeod__TimeUT1 ( )  [friend]
```

8.29.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file time_ut1.hh.

8.29.5 Field Documentation

8.29.5.1 true_ut1

```
bool jeod::TimeUT1::true_ut1
```

"False" for comparison with older versions of JEOD

trick_units(–)

Definition at line 91 of file time_ut1.hh.

Referenced by jeod::TimeConverter_TAI_UT1::convert_a_to_b(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1(), TimeUT1(), and jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends().

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

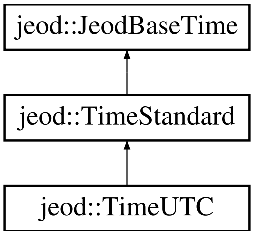
- time_ut1.hh
- time_ut1.cc

8.30 jeod::TimeUTC Class Reference

Represents Coordinated Universal Time.

```
#include <time_utc.hh>
```

Inheritance diagram for jeod::TimeUTC:



Public Member Functions

- TimeUTC ()

  Construct a Time_UTC.
- ∼TimeUTC ()

  Destroy a Time_UTC.

Data Fields

- bool true_utc

  "False" for comparison with older versions of JEOD

Private Member Functions

- TimeUTC (const TimeUTC &)
- TimeUTC & operator= (const TimeUTC &)
- void set_epoch (void)

    Sets the epoch for UTC time.

Friends

- class InputProcessor
- void init_attrjeod__TimeUTC ()

Additional Inherited Members

8.30.1 Detailed Description

Represents Coordinated Universal Time.

Definition at line 83 of file time_utc.hh.

8.30.2 Constructor & Destructor Documentation

8.30.2.1 TimeUTC() [1/2]

```
jeod::TimeUTC::TimeUTC (
             void )
```

Construct a Time_UTC.

Definition at line 51 of file time_utc.cc.

References jeod::JeodBaseTime::name, set_epoch(), and true_utc.

8.30.2.2 ∼TimeUTC()

```
jeod::TimeUTC::∼TimeUTC (
             void )
```

Destroy a Time_UTC.

Definition at line 75 of file time_utc.cc.

8.30.2.3 TimeUTC() [2/2]

```
jeod::TimeUTC::TimeUTC (
              const TimeUTC &  )  [private]
```

8.30.3 Member Function Documentation

8.30.3.1 operator=()

```
TimeUTC& jeod::TimeUTC::operator= (
              const TimeUTC &  )  [private]
```

8.30.3.2 set_epoch()

```
void jeod::TimeUTC::set_epoch (
              void  )  [private], [virtual]
```

Sets the epoch for UTC time.

Implements jeod::TimeStandard.

Definition at line 64 of file time_utc.cc.

References jeod::TimeStandard::tjt_at_epoch.

Referenced by TimeUTC().

8.30.4 Friends And Related Function Documentation

8.30.4.1 init_attrjeod__TimeUTC

```
void init_attrjeod__TimeUTC ( )  [friend]
```

8.30.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 85 of file time_utc.hh.

8.30.5 Field Documentation

8.30.5.1 true_utc

```
bool jeod::TimeUTC::true_utc
```

"False" for comparison with older versions of JEOD

trick_units(--)

Definition at line 92 of file time_utc.hh.

Referenced by jeod::TimeConverter_TAI_UTC::convert_a_to_b(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UTC::initialize_leap_second(), TimeUTC(), and jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends().

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

- time_utc.hh
- time_utc.cc

Chapter 9

File Documentation

9.1 class_declarations.hh File Reference

Forward declaration of classes defined in time.hh.

Namespaces

- jeod

  Namespace jeod.

9.1.1 Detailed Description

Forward declaration of classes defined in time.hh.

9.2 tai_to_ut1.cc File Reference

```
#include "environment/time/include/time_converter_tai_ut1.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/tai_to_ut1.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data

9.2.1 Macro Definition Documentation

9.2.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data
```

Definition at line 25 of file tai_to_ut1.cc.

9.3 tai_to_ut1.hh File Reference

Data Structures

- class jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data

Namespaces

- jeod

  Namespace jeod.

9.4 tai_to_utc.cc File Reference

```
#include "environment/time/include/time_converter_tai_utc.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/tai_to_utc.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS TimeConverter_TAI_UTC_tai_to_utc_default_data

9.4.1 Macro Definition Documentation

9.4.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS TimeConverter_TAI_UTC_tai_to_utc_default_data
```

Definition at line 23 of file tai_to_utc.cc.

9.5 tai_to_utc.hh File Reference

Data Structures

- class jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data

Namespaces

- jeod

    Namespace jeod.

9.6 time.cc File Reference

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

    Namespace jeod.

9.6.1 Detailed Description

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

9.7 time.hh File Reference

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <string>
#include <utility>
#include "utils/named_item/include/named_item.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
#include "time_links.hh"
```

Data Structures

- class jeod::JeodBaseTime

  JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

Namespaces

- jeod

  Namespace jeod.

9.7.1 Detailed Description

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

9.8 time__add_type_update.cc File Reference

Define JeodBaseTime::add_type_update.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.8.1 Detailed Description

Define JeodBaseTime::add_type_update.

This is a final method that draws in a lot of the time model functionality. Making this method a separate compilation unit enables models that only need the vtable for class Time can get that from time.o without pulling in the whole of the time model.

9.9 time_converter.cc File Reference

An abstract class that defines the basic structure of all the methods used by the converter objects.

```
#include <cstddef>
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter.hh"
#include "../include/time.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.9.1 Detailed Description

An abstract class that defines the basic structure of all the methods used by the converter objects.

9.10 time_converter.hh File Reference

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::TimeConverter

  The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

Namespaces

- jeod

  Namespace jeod.

Functions

- TimeConverter::Direction jeod::operator| (TimeConverter::Direction a, TimeConverter::Direction b)

  Bitwise or operator for combining multiple converter direction flags.

9.10.1 Detailed Description

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

9.11 time_converter_dyn_tai.cc File Reference

Converts between International Atomic Time and Dynamic Time.

```
#include <cstddef>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tai.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tai.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.11.1 Detailed Description

Converts between International Atomic Time and Dynamic Time.

9.12 time_converter_dyn_tai.hh File Reference

Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_Dyn_TAI

  Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.

Namespaces

- jeod

  Namespace jeod.

9.12.1 Detailed Description

Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time.

9.13 time_converter_dyn_tdb.cc File Reference

Converts between Dynamic Time and Barycentric Dynamic Time.

```
#include <cstddef>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tdb.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.13.1 Detailed Description

Converts between Dynamic Time and Barycentric Dynamic Time.

9.14 time_converter_dyn_tdb.hh File Reference

Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_Dyn_TDB

  Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

9.15 time_converter_dyn_ude.cc File Reference

Converts between Dynamic Time and a time with User-Defined-Epoch.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Converts between Dynamic Time and a time with User-Defined-Epoch.

9.16 time_converter_dyn_ude.hh File Reference

Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_Dyn_UDE

  Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

Namespaces

- jeod

  Namespace jeod.

9.16.1 Detailed Description

Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

There can be multiple instances of this class.

9.17 time_converter_std_ude.cc File Reference

Define member functions for class TimeConverter_STD_UDE.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_std_ude.hh"
#include "../include/time_standard.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.17.1 Detailed Description

Define member functions for class TimeConverter_STD_UDE.

9.18 time_converter_std_ude.hh File Reference

Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_STD_UDE

  Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

Namespaces

- jeod

  Namespace jeod.

9.18.1 Detailed Description

Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

There can be multiple such instances of this class.

9.19 time_converter_tai_gps.cc File Reference

Converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_gps.hh"
#include "../include/time_tai.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.19.1 Detailed Description

Converts between International Atomic Time and the clock associated with the Global Positioning System.

9.20 time_converter_tai_gps.hh File Reference

Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_TAI_GPS

  Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

Namespaces

- jeod

  Namespace jeod.

9.20.1 Detailed Description

Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

9.21 time_converter_tai_tdb.cc File Reference

Converts from International Atomic Time to Barycentric Dynamic Time.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tdb.hh"
#include "../include/time_tai.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.21.1 Detailed Description

Converts from International Atomic Time to Barycentric Dynamic Time.

9.22 time_converter_tai_tdb.hh File Reference

Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_TAI_TDB

  Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

Namespaces

- jeod

  Namespace jeod.

9.22.1 Detailed Description

Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

9.23 time_converter_tai_tt.cc File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tt.hh"
#include "../include/time_tai.hh"
#include "../include/time_tt.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

9.24 time_converter_tai_tt.hh File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_TAI_TT

  Converts between International Atomic Time and Terrestrial Time.

Namespaces

- jeod

  Namespace jeod.

9.24.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

9.25 time_converter_tai_ut1.cc File Reference

Converts between International Atomic Time and Universal Time.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_ut1.hh"
#include "../include/time_tai.hh"
#include "../include/time_ut1.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

Converts between International Atomic Time and Universal Time.

9.26 time_converter_tai_ut1.hh File Reference

Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_TAI_UT1

  Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.

Namespaces

- jeod

  Namespace jeod.

9.26.1 Detailed Description

Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time.

9.27 time_converter_tai_utc.cc File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_tai.hh"
#include "../include/time_utc.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.27.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

9.28 time_converter_tai_utc.hh File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_TAI_UTC

  Converts between International Atomic Time and Coordinated Universal Time.

Namespaces

- jeod

  Namespace jeod.

9.28.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

9.29 time_converter_ut1_gmst.cc File Reference

Define member functions for class TimeConverter_UT1_GMST.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_ut1_gmst.hh"
#include "../include/time_ut1.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.29.1 Detailed Description

Define member functions for class TimeConverter_UT1_GMST.

9.30 time_converter_ut1_gmst.hh File Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

Data Structures

- class jeod::TimeConverter_UT1_GMST

    Converts between Universal Time and Greenwich Mean Sidereal Time.

Namespaces

- jeod

    Namespace jeod.

9.30.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

9.31 time_dyn.cc File Reference

Define member functions for Dynamic Time.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time_dyn.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

    Namespace jeod.

9.31.1 Detailed Description

Define member functions for Dynamic Time.

9.32 time_dyn.hh File Reference

Represents the Dynamic Time in the simulation.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```

Data Structures

- class jeod::TimeDyn

  Represents the Dynamic Time in the simulation.

Namespaces

- jeod

  Namespace jeod.

9.32.1 Detailed Description

Represents the Dynamic Time in the simulation.

9.33 time_enum.hh File Reference

Contains an enumeration of the formats in which time can be represented.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::TimeEnum

  Contains an enumeration of the formats in which time can be represented.

Namespaces

- jeod

  Namespace jeod.

9.33.1 Detailed Description

Contains an enumeration of the formats in which time can be represented.

9.34 time_gmst.cc File Reference

Define member functions for Greenwich Mean Sidereal Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.34.1 Detailed Description

Define member functions for Greenwich Mean Sidereal Time.

9.35 time_gmst.hh File Reference

To represent the clock known as Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeGMST

  To represent the clock known as Greenwich Mean Sidereal Time.

Namespaces

- jeod

  Namespace jeod.

9.35.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

9.36 time_gps.cc File Reference

Define member functions for the clock associated with the Global Positioning System.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.36.1 Detailed Description

Define member functions for the clock associated with the Global Positioning System.

9.37 time_gps.hh File Reference

To represent the time associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeGPS

  To represent the time associated with the Global Positioning System.

Namespaces

- jeod

  Namespace jeod.

9.37.1 Detailed Description

To represent the time associated with the Global Positioning System.

9.38 time_links.hh File Reference

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

```
#include "class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/tree_links.hh"
```

Data Structures

- class jeod::TimeLinks

Namespaces

- jeod

  Namespace jeod.

9.38.1 Detailed Description

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

9.39 time_manager.cc File Reference

Define member functions for class TimeManager.

```
#include <algorithm>
#include <cstddef>
#include <cstring>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.39.1 Detailed Description

Define member functions for class TimeManager.

9.40 time_manager.hh File Reference

To manage the various time representations and the converters between them throughout the simulation.

```
#include <string>
#include <vector>
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/integration/include/jeod_integration_time.hh"
#include "time_dyn.hh"
```

Data Structures

- class jeod::TimeManager

  To manage the various time representations and the converters between them throughout the simulation.

Namespaces

- jeod

  Namespace jeod.

9.40.1 Detailed Description

To manage the various time representations and the converters between them throughout the simulation.

9.41 time_manager__initialize.cc File Reference

Define TimeManager::initialize.

```
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.41.1 Detailed Description

Define TimeManager::initialize.

This method allocates resources and invokes TimeManagerInit functionality. This method needs to be defined as a separate compilation unit.

9.42 time_manager_init.cc File Reference

Define member functions for the Time Manager Initialization.

```
#include <cstddef>
#include <typeinfo>
#include <algorithm>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_ude.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_converter_tai_ut1.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.42.1 Detailed Description

Define member functions for the Time Manager Initialization.

9.43 time_manager_init.hh File Reference

To initialize the Time Manager.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_enum.hh"
```

Data Structures

- class jeod::TimeManagerInit

  To initialize the Time Manager.

Namespaces

- jeod

  Namespace jeod.

9.43.1 Detailed Description

To initialize the Time Manager.

9.44 time_messages.cc File Reference

Implement the class TimeMessages.

```
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "environment/time/"

9.44.1 Detailed Description

Implement the class TimeMessages.

9.45 time_messages.hh File Reference

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::TimeMessages

  Specify the message IDs used in the Time model.

Namespaces

- jeod

  Namespace jeod.

9.45.1 Detailed Description

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

9.46 time_met.cc File Reference

Define member functions for Mission Elapsed Time.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_met.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.46.1 Detailed Description

Define member functions for Mission Elapsed Time.

9.47 time_met.hh File Reference

A type of UDE time that allows for deliberate holds, or pauses.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_ude.hh"
```

Data Structures

- class jeod::TimeMET

  A type of UDE time that allows for deliberate holds, or pauses.

Namespaces

- jeod

  Namespace jeod.

9.47.1 Detailed Description

A type of UDE time that allows for deliberate holds, or pauses.

9.48 time_standard.cc File Reference

An abstract class, this defines the basic structure of member functions for all Standard Times.

```
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time_standard.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.48.1 Detailed Description

An abstract class, this defines the basic structure of member functions for all Standard Times.

9.49 time_standard.hh File Reference

A class that serves as the base for all time representations that are well defined outside the simulation.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```

Data Structures

- class jeod::TimeStandard

  A class that serves as the base for all time representations that are well defined outside the simulation.

Namespaces

- jeod

  Namespace jeod.

9.49.1 Detailed Description

A class that serves as the base for all time representations that are well defined outside the simulation.

9.50 time_tai.cc File Reference

Define member functions for International Atomic Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tai.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.50.1 Detailed Description

Define member functions for International Atomic Time.

9.51 time_tai.hh File Reference

Represents International Atomic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeTAI

  Represents International Atomic Time.

Namespaces

- jeod

  Namespace jeod.

9.51.1 Detailed Description

Represents International Atomic Time.

9.52 time_tdb.cc File Reference

Define member functions Barycentric Dynamic Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tdb.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.52.1 Detailed Description

Define member functions Barycentric Dynamic Time.

9.53 time_tdb.hh File Reference

Represents Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeTDB

  Represents Barycentric Dynamic Time.

Namespaces

- jeod

  Namespace jeod.

9.53.1 Detailed Description

Represents Barycentric Dynamic Time.

9.54 time_tt.cc File Reference

Define member functions for Terrestrial Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tt.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.54.1 Detailed Description

Define member functions for Terrestrial Time.

9.55 time_tt.hh File Reference

Represents Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeTT

  Represents Terrestrial Time.

Namespaces

- jeod

  Namespace jeod.

9.55.1 Detailed Description

Represents Terrestrial Time.

9.56 time_ude.cc File Reference

Define member functions for those times with a User-Defined-Epoch.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.56.1 Detailed Description

Define member functions for those times with a User-Defined-Epoch.

9.57 time_ude.hh File Reference

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
#include "time_enum.hh"
```

Data Structures

- class jeod::TimeUDE

  Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

Namespaces

- jeod

  Namespace jeod.

9.57.1 Detailed Description

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

9.58 time_ut1.cc File Reference

Define member functions for Universal Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ut1.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.58.1 Detailed Description

Define member functions for Universal Time.

9.59 time_ut1.hh File Reference

Represents Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeUT1

  Represents Universal Time.

Namespaces

- jeod

  Namespace jeod.

9.59.1 Detailed Description

Represents Universal Time.

9.60 time_utc.cc File Reference

Define member functions for Coordinated Universal Time.

```
#include <cstddef>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_utc.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.60.1 Detailed Description

Define member functions for Coordinated Universal Time.

9.61 time_utc.hh File Reference

Represents Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

Data Structures

- class jeod::TimeUTC

  Represents Coordinated Universal Time.

Namespaces

- jeod

  Namespace jeod.

9.61.1 Detailed Description

Represents Coordinated Universal Time.

Index