

TimeModel

5.1

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	13
6.1	Models	13
6.1.1	Detailed Description	13
6.2	Environment	14
6.2.1	Detailed Description	14
6.3	Time	15
6.3.1	Detailed Description	17
6.3.2	Macro Definition Documentation	17
6.3.2.1	PATH	17

7	Namespace Documentation	19
7.1	jeod Namespace Reference	19
7.1.1	Detailed Description	20
7.1.2	Function Documentation	20
7.1.2.1	operator" ()	21
8	Data Structure Documentation	23
8.1	jeod::JeodBaseTime Class Reference	23
8.1.1	Detailed Description	25
8.1.2	Constructor & Destructor Documentation	25
8.1.2.1	JeodBaseTime() [1/2]	25
8.1.2.2	~JeodBaseTime()	26
8.1.2.3	JeodBaseTime() [2/2]	26
8.1.3	Member Function Documentation	26
8.1.3.1	add_parent()	26
8.1.3.2	add_type_initialize()	27
8.1.3.3	add_type_update()	27
8.1.3.4	get_index()	28
8.1.3.5	initialize_from_parent()	28
8.1.3.6	initialize_initializer_time()	29
8.1.3.7	is_initialized()	30
8.1.3.8	must_be_singleton()	30
8.1.3.9	operator=()	30
8.1.3.10	override_initialized()	31
8.1.3.11	set_index()	31
8.1.3.12	set_name()	31
8.1.3.13	set_time_by_days()	31
8.1.3.14	set_time_by_seconds()	32
8.1.3.15	update()	32
8.1.4	Friends And Related Function Documentation	33
8.1.4.1	init_attrjeod__JeodBaseTime	33

8.1.4.2	InputProcessor	33
8.1.4.3	TimeConverter	33
8.1.4.4	TimeManagerInit	33
8.1.5	Field Documentation	34
8.1.5.1	clock_resolution	34
8.1.5.2	days	34
8.1.5.3	index	34
8.1.5.4	initial_value	35
8.1.5.5	initialize_from_name	35
8.1.5.6	initialized	35
8.1.5.7	initializing_value	36
8.1.5.8	links	36
8.1.5.9	name	36
8.1.5.10	seconds	37
8.1.5.11	time_manager	37
8.1.5.12	update_converter_direction	38
8.1.5.13	update_converter_ptr	38
8.1.5.14	update_from_name	38
8.2	jeod::TimeConverter Class Reference	39
8.2.1	Detailed Description	40
8.2.2	Member Enumeration Documentation	41
8.2.2.1	Direction	41
8.2.3	Constructor & Destructor Documentation	41
8.2.3.1	~TimeConverter()	41
8.2.3.2	TimeConverter() [1/2]	41
8.2.3.3	TimeConverter() [2/2]	41
8.2.4	Member Function Documentation	42
8.2.4.1	can_convert()	42
8.2.4.2	convert_a_to_b()	42
8.2.4.3	convert_b_to_a()	43

8.2.4.4	get_a_to_b_offset()	43
8.2.4.5	initialize()	43
8.2.4.6	is_initialized()	44
8.2.4.7	operator=()	44
8.2.4.8	override_initialized()	44
8.2.4.9	reset_a_to_b_offset()	44
8.2.4.10	verify_setup()	45
8.2.4.11	verify_table_lookup_ends()	45
8.2.5	Friends And Related Function Documentation	46
8.2.5.1	init_attrjeod__TimeConverter	46
8.2.5.2	InputProcessor	46
8.2.5.3	JeodBaseTime	46
8.2.6	Field Documentation	46
8.2.6.1	a_name	46
8.2.6.2	a_to_b_offset	47
8.2.6.3	b_name	47
8.2.6.4	initialized	47
8.2.6.5	valid_directions	48
8.3	jeod::TimeConverter_Dyn_TAI Class Reference	48
8.3.1	Detailed Description	49
8.3.2	Constructor & Destructor Documentation	49
8.3.2.1	TimeConverter_Dyn_TAI() [1/2]	49
8.3.2.2	~TimeConverter_Dyn_TAI()	49
8.3.2.3	TimeConverter_Dyn_TAI() [2/2]	49
8.3.3	Member Function Documentation	49
8.3.3.1	convert_a_to_b()	50
8.3.3.2	initialize()	50
8.3.3.3	operator=()	50
8.3.4	Friends And Related Function Documentation	50
8.3.4.1	init_attrjeod__TimeConverter_Dyn_TAI	51

8.3.4.2	InputProcessor	51
8.3.5	Field Documentation	51
8.3.5.1	dyn_ptr	51
8.3.5.2	tai_ptr	51
8.4	jeod::TimeConverter_Dyn_TDB Class Reference	52
8.4.1	Detailed Description	52
8.4.2	Constructor & Destructor Documentation	53
8.4.2.1	TimeConverter_Dyn_TDB() [1/2]	53
8.4.2.2	~TimeConverter_Dyn_TDB()	53
8.4.2.3	TimeConverter_Dyn_TDB() [2/2]	53
8.4.3	Member Function Documentation	53
8.4.3.1	convert_a_to_b()	53
8.4.3.2	initialize()	53
8.4.3.3	operator=()	54
8.4.4	Friends And Related Function Documentation	54
8.4.4.1	init_attrjeod__TimeConverter_Dyn_TDB	54
8.4.4.2	InputProcessor	54
8.4.5	Field Documentation	54
8.4.5.1	dyn_ptr	55
8.4.5.2	tdb_ptr	55
8.5	jeod::TimeConverter_Dyn_UDE Class Reference	55
8.5.1	Detailed Description	56
8.5.2	Constructor & Destructor Documentation	56
8.5.2.1	TimeConverter_Dyn_UDE() [1/2]	56
8.5.2.2	~TimeConverter_Dyn_UDE()	57
8.5.2.3	TimeConverter_Dyn_UDE() [2/2]	57
8.5.3	Member Function Documentation	57
8.5.3.1	convert_a_to_b()	57
8.5.3.2	initialize()	57
8.5.3.3	operator=()	58

8.5.3.4	<code>reset_a_to_b_offset()</code>	58
8.5.4	Friends And Related Function Documentation	58
8.5.4.1	<code>init_attrjeod__TimeConverter_Dyn_UDE</code>	58
8.5.4.2	<code>InputProcessor</code>	58
8.5.5	Field Documentation	59
8.5.5.1	<code>dyn_ptr</code>	59
8.5.5.2	<code>ude_ptr</code>	59
8.6	<code>jeod::TimeConverter_STD_UDE</code> Class Reference	59
8.6.1	Detailed Description	60
8.6.2	Constructor & Destructor Documentation	60
8.6.2.1	<code>TimeConverter_STD_UDE()</code> [1/2]	61
8.6.2.2	<code>~TimeConverter_STD_UDE()</code>	61
8.6.2.3	<code>TimeConverter_STD_UDE()</code> [2/2]	61
8.6.3	Member Function Documentation	61
8.6.3.1	<code>convert_a_to_b()</code>	61
8.6.3.2	<code>convert_b_to_a()</code>	62
8.6.3.3	<code>initialize()</code>	62
8.6.3.4	<code>operator=()</code>	62
8.6.3.5	<code>reset_a_to_b_offset()</code>	63
8.6.4	Friends And Related Function Documentation	63
8.6.4.1	<code>init_attrjeod__TimeConverter_STD_UDE</code>	63
8.6.4.2	<code>InputProcessor</code>	63
8.6.5	Field Documentation	63
8.6.5.1	<code>failed_null_test</code>	63
8.6.5.2	<code>std_ptr</code>	64
8.6.5.3	<code>ude_ptr</code>	64
8.7	<code>jeod::TimeConverter_TAI_GPS</code> Class Reference	64
8.7.1	Detailed Description	65
8.7.2	Constructor & Destructor Documentation	65
8.7.2.1	<code>TimeConverter_TAI_GPS()</code> [1/2]	65

8.7.2.2	~TimeConverter_TAI_GPS()	66
8.7.2.3	TimeConverter_TAI_GPS() [2/2]	66
8.7.3	Member Function Documentation	66
8.7.3.1	convert_a_to_b()	66
8.7.3.2	convert_b_to_a()	66
8.7.3.3	initialize()	66
8.7.3.4	operator=()	67
8.7.4	Friends And Related Function Documentation	67
8.7.4.1	init_attrjeod__TimeConverter_TAI_GPS	67
8.7.4.2	InputProcessor	67
8.7.5	Field Documentation	67
8.7.5.1	gps_ptr	68
8.7.5.2	tai_ptr	68
8.8	jeod::TimeConverter_TAI_TDB Class Reference	68
8.8.1	Detailed Description	69
8.8.2	Constructor & Destructor Documentation	70
8.8.2.1	TimeConverter_TAI_TDB() [1/2]	70
8.8.2.2	~TimeConverter_TAI_TDB()	70
8.8.2.3	TimeConverter_TAI_TDB() [2/2]	70
8.8.3	Member Function Documentation	70
8.8.3.1	convert_a_to_b()	70
8.8.3.2	convert_b_to_a()	71
8.8.3.3	initialize()	71
8.8.3.4	operator=()	71
8.8.3.5	set_a_to_b_offset()	72
8.8.4	Friends And Related Function Documentation	72
8.8.4.1	init_attrjeod__TimeConverter_TAI_TDB	72
8.8.4.2	InputProcessor	72
8.8.5	Field Documentation	72
8.8.5.1	a_to_b_offset_epoch	72

8.8.5.2	nIter	73
8.8.5.3	nSteps	73
8.8.5.4	prev_tai_seconds	73
8.8.5.5	prev_tdb_seconds	73
8.8.5.6	tai_ptr	74
8.8.5.7	TAI_to_TT_offset	74
8.8.5.8	tdb_ptr	74
8.9	jeod::TimeConverter_TAI_TT Class Reference	75
8.9.1	Detailed Description	75
8.9.2	Constructor & Destructor Documentation	76
8.9.2.1	TimeConverter_TAI_TT() [1/2]	76
8.9.2.2	~TimeConverter_TAI_TT()	76
8.9.2.3	TimeConverter_TAI_TT() [2/2]	76
8.9.3	Member Function Documentation	76
8.9.3.1	convert_a_to_b()	76
8.9.3.2	convert_b_to_a()	77
8.9.3.3	initialize()	77
8.9.3.4	operator=()	77
8.9.4	Friends And Related Function Documentation	77
8.9.4.1	init_attrjeod__TimeConverter_TAI_TT	78
8.9.4.2	InputProcessor	78
8.9.5	Field Documentation	78
8.9.5.1	tai_ptr	78
8.9.5.2	tt_ptr	78
8.10	jeod::TimeConverter_TAI_UT1 Class Reference	79
8.10.1	Detailed Description	80
8.10.2	Constructor & Destructor Documentation	80
8.10.2.1	TimeConverter_TAI_UT1() [1/2]	80
8.10.2.2	~TimeConverter_TAI_UT1()	81
8.10.2.3	TimeConverter_TAI_UT1() [2/2]	81

8.10.3	Member Function Documentation	81
8.10.3.1	convert_a_to_b()	81
8.10.3.2	convert_b_to_a()	81
8.10.3.3	initialize()	82
8.10.3.4	initialize_tai_to_ut1()	82
8.10.3.5	operator=()	83
8.10.3.6	verify_table_lookup_ends()	83
8.10.4	Friends And Related Function Documentation	83
8.10.4.1	init_attrjeod__TimeConverter_TAI_UT1	83
8.10.4.2	InputProcessor	83
8.10.5	Field Documentation	83
8.10.5.1	gradient	84
8.10.5.2	index	84
8.10.5.3	last_index	84
8.10.5.4	next_value	84
8.10.5.5	next_when	85
8.10.5.6	off_table_end	85
8.10.5.7	override_data_table	85
8.10.5.8	prev_value	85
8.10.5.9	prev_when	86
8.10.5.10	tai_ptr	86
8.10.5.11	tai_to_ut1_override_val	86
8.10.5.12	ut1_ptr	86
8.10.5.13	val_vec	87
8.10.5.14	when_vec	87
8.11	jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data Class Reference	87
8.11.1	Detailed Description	87
8.11.2	Member Function Documentation	87
8.11.2.1	initialize()	88
8.12	jeod::TimeConverter_TAI_UTC Class Reference	88

8.12.1 Detailed Description	89
8.12.2 Constructor & Destructor Documentation	90
8.12.2.1 TimeConverter_TAI_UTC() [1/2]	90
8.12.2.2 ~TimeConverter_TAI_UTC()	90
8.12.2.3 TimeConverter_TAI_UTC() [2/2]	90
8.12.3 Member Function Documentation	90
8.12.3.1 convert_a_to_b()	90
8.12.3.2 convert_b_to_a()	91
8.12.3.3 initialize()	91
8.12.3.4 initialize_leap_second()	91
8.12.3.5 operator=()	92
8.12.3.6 verify_table_lookup_ends()	92
8.12.4 Friends And Related Function Documentation	92
8.12.4.1 init_attrjeod__TimeConverter_TAI_UTC	92
8.12.4.2 InputProcessor	93
8.12.5 Field Documentation	93
8.12.5.1 index	93
8.12.5.2 last_index	93
8.12.5.3 leap_sec_override_val	93
8.12.5.4 next_when	94
8.12.5.5 off_table_end	94
8.12.5.6 override_data_table	94
8.12.5.7 prev_when	94
8.12.5.8 tai_ptr	95
8.12.5.9 utc_ptr	95
8.12.5.10 val_vec	95
8.12.5.11 when_vec	96
8.13 jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data Class Reference	96
8.13.1 Detailed Description	96
8.13.2 Member Function Documentation	96

8.13.2.1	initialize()	96
8.14	jeod::TimeConverter_UT1_GMST Class Reference	97
8.14.1	Detailed Description	97
8.14.2	Constructor & Destructor Documentation	98
8.14.2.1	TimeConverter_UT1_GMST() [1/2]	98
8.14.2.2	~TimeConverter_UT1_GMST()	98
8.14.2.3	TimeConverter_UT1_GMST() [2/2]	98
8.14.3	Member Function Documentation	98
8.14.3.1	convert_a_to_b()	98
8.14.3.2	initialize()	98
8.14.3.3	operator=()	99
8.14.4	Friends And Related Function Documentation	99
8.14.4.1	init_attrjeod__TimeConverter_UT1_GMST	99
8.14.4.2	InputProcessor	99
8.14.5	Field Documentation	99
8.14.5.1	gmst_ptr	100
8.14.5.2	ut1_ptr	100
8.15	jeod::TimeDyn Class Reference	100
8.15.1	Detailed Description	101
8.15.2	Constructor & Destructor Documentation	101
8.15.2.1	TimeDyn() [1/2]	101
8.15.2.2	~TimeDyn()	102
8.15.2.3	TimeDyn() [2/2]	102
8.15.3	Member Function Documentation	102
8.15.3.1	initialize_initializer_time()	102
8.15.3.2	operator=()	102
8.15.3.3	update()	103
8.15.3.4	update_offset()	103
8.15.4	Friends And Related Function Documentation	103
8.15.4.1	init_attrjeod__TimeDyn	103

8.15.4.2	InputProcessor	104
8.15.5	Field Documentation	104
8.15.5.1	offset	104
8.15.5.2	ref_scale	104
8.15.5.3	scale_factor	104
8.16	jeod::TimeEnum Class Reference	105
8.16.1	Detailed Description	105
8.16.2	Member Enumeration Documentation	105
8.16.2.1	TimeFormat	105
8.17	jeod::TimeGMST Class Reference	106
8.17.1	Detailed Description	106
8.17.2	Constructor & Destructor Documentation	107
8.17.2.1	TimeGMST() [1/2]	107
8.17.2.2	~TimeGMST()	107
8.17.2.3	TimeGMST() [2/2]	107
8.17.3	Member Function Documentation	107
8.17.3.1	calculate_calendar_values()	107
8.17.3.2	operator=()	108
8.17.3.3	set_epoch()	108
8.17.3.4	set_time_by_trunc_julian()	108
8.17.4	Friends And Related Function Documentation	108
8.17.4.1	init_attrjeod__TimeGMST	108
8.17.4.2	InputProcessor	109
8.18	jeod::TimeGPS Class Reference	109
8.18.1	Detailed Description	110
8.18.2	Constructor & Destructor Documentation	110
8.18.2.1	TimeGPS() [1/2]	111
8.18.2.2	~TimeGPS()	111
8.18.2.3	TimeGPS() [2/2]	111
8.18.3	Member Function Documentation	111

8.18.3.1	calculate_calendar_values()	111
8.18.3.2	convert_from_calendar()	112
8.18.3.3	operator=()	112
8.18.3.4	set_epoch()	112
8.18.3.5	set_time_by_days()	112
8.18.3.6	set_time_by_seconds()	113
8.18.3.7	set_time_by_trunc_julian()	113
8.18.4	Friends And Related Function Documentation	114
8.18.4.1	init_attrjeod__TimeGPS	114
8.18.4.2	InputProcessor	114
8.18.5	Field Documentation	114
8.18.5.1	day_of_week	114
8.18.5.2	rollover_count	115
8.18.5.3	rollover_count_13_bit	115
8.18.5.4	seconds_of_day	115
8.18.5.5	seconds_of_week	115
8.18.5.6	week	116
8.18.5.7	week_13_bit	116
8.19	jeod::TimeLinks Class Reference	116
8.19.1	Detailed Description	117
8.19.2	Constructor & Destructor Documentation	117
8.19.2.1	TimeLinks() [1/3]	117
8.19.2.2	TimeLinks() [2/3]	117
8.19.2.3	~TimeLinks()	117
8.19.2.4	TimeLinks() [3/3]	117
8.19.3	Member Function Documentation	118
8.19.3.1	operator=()	118
8.19.4	Friends And Related Function Documentation	118
8.19.4.1	init_attrjeod__TimeLinks	118
8.19.4.2	InputProcessor	118

8.19.5	Field Documentation	118
8.19.5.1	default_path_size	118
8.20	jeod::TimeManager Class Reference	119
8.20.1	Detailed Description	120
8.20.2	Constructor & Destructor Documentation	120
8.20.2.1	TimeManager() [1/2]	120
8.20.2.2	~TimeManager()	121
8.20.2.3	TimeManager() [2/2]	121
8.20.3	Member Function Documentation	121
8.20.3.1	get_converter_ptr()	121
8.20.3.2	get_jeod_integration_time()	122
8.20.3.3	get_time_change_flag()	122
8.20.3.4	get_time_ptr() [1/2]	122
8.20.3.5	get_time_ptr() [2/2]	123
8.20.3.6	get_time_scale_factor()	123
8.20.3.7	get_timestamp_time()	123
8.20.3.8	initialize()	123
8.20.3.9	operator=()	124
8.20.3.10	register_converter()	124
8.20.3.11	register_time()	124
8.20.3.12	register_time_named()	125
8.20.3.13	time_lookup()	125
8.20.3.14	time_standards_exist()	126
8.20.3.15	update()	127
8.20.3.16	update_time()	128
8.20.3.17	verify_table_lookup_ends()	128
8.20.4	Friends And Related Function Documentation	129
8.20.4.1	init_attrjeod__TimeManager	129
8.20.4.2	InputProcessor	129
8.20.4.3	TimeManagerInit	129

8.20.5	Field Documentation	129
8.20.5.1	converter_vector	129
8.20.5.2	dyn_time	130
8.20.5.3	num_types	130
8.20.5.4	simtime	130
8.20.5.5	time_change_flag	131
8.20.5.6	time_vector	131
8.21	jeod::TimeManagerInit Class Reference	131
8.21.1	Detailed Description	133
8.21.2	Constructor & Destructor Documentation	133
8.21.2.1	TimeManagerInit() [1/2]	133
8.21.2.2	~TimeManagerInit()	133
8.21.2.3	TimeManagerInit() [2/2]	134
8.21.3	Member Function Documentation	134
8.21.3.1	create_init_tree()	134
8.21.3.2	create_update_tree()	134
8.21.3.3	get_conv_dir_init()	135
8.21.3.4	get_conv_dir_upd()	135
8.21.3.5	get_conv_ptr_index()	136
8.21.3.6	get_status()	136
8.21.3.7	increment_status()	137
8.21.3.8	initialize()	137
8.21.3.9	initialize_manager()	138
8.21.3.10	initialize_time_types()	138
8.21.3.11	operator=()	138
8.21.3.12	organize_update_list()	139
8.21.3.13	populate_converter_registry()	139
8.21.3.14	set_status()	139
8.21.3.15	verify_converter_setup()	140
8.21.3.16	verify_times_setup()	140

8.21.4	Friends And Related Function Documentation	140
8.21.4.1	init_attrjeod__TimeManagerInit	141
8.21.4.2	InputProcessor	141
8.21.5	Field Documentation	141
8.21.5.1	converter_ptrs_index	141
8.21.5.2	dyn_time_index	141
8.21.5.3	init_converter_dir_table	142
8.21.5.4	initializer	142
8.21.5.5	initializer_index	142
8.21.5.6	num_added_pass	142
8.21.5.7	num_added_total	143
8.21.5.8	sim_start_format	143
8.21.5.9	status	143
8.21.5.10	time_manager	143
8.21.5.11	update_converter_dir_table	144
8.22	jeod::TimeMessages Class Reference	144
8.22.1	Detailed Description	145
8.22.2	Constructor & Destructor Documentation	145
8.22.2.1	TimeMessages() [1/2]	145
8.22.2.2	TimeMessages() [2/2]	145
8.22.3	Member Function Documentation	145
8.22.3.1	operator=()	145
8.22.4	Friends And Related Function Documentation	145
8.22.4.1	init_attrjeod__TimeMessages	145
8.22.4.2	InputProcessor	146
8.22.5	Field Documentation	146
8.22.5.1	duplicate_methods	146
8.22.5.2	extension_error	146
8.22.5.3	incomplete_setup_error	146
8.22.5.4	initialization_error	147

8.22.5.5	invalid_data_error	147
8.22.5.6	invalid_node	147
8.22.5.7	invalid_setup_error	148
8.22.5.8	memory_error	148
8.22.5.9	redundancy_error	148
8.23	jeod::TimeMET Class Reference	149
8.23.1	Detailed Description	149
8.23.2	Constructor & Destructor Documentation	150
8.23.2.1	TimeMET() [1/2]	150
8.23.2.2	~TimeMET()	150
8.23.2.3	TimeMET() [2/2]	150
8.23.3	Member Function Documentation	150
8.23.3.1	operator=()	150
8.23.3.2	update()	150
8.23.4	Friends And Related Function Documentation	151
8.23.4.1	init_attrjeod__TimeMET	151
8.23.4.2	InputProcessor	151
8.23.5	Field Documentation	151
8.23.5.1	hold	151
8.23.5.2	previous_hold	151
8.24	jeod::TimeStandard Class Reference	152
8.24.1	Detailed Description	153
8.24.2	Constructor & Destructor Documentation	154
8.24.2.1	TimeStandard() [1/2]	154
8.24.2.2	~TimeStandard()	154
8.24.2.3	TimeStandard() [2/2]	154
8.24.3	Member Function Documentation	154
8.24.3.1	add_type_initialize()	154
8.24.3.2	calculate_calendar_values()	155
8.24.3.3	calendar_update()	155

8.24.3.4	<code>convert_from_calendar()</code>	156
8.24.3.5	<code>initialize_from_parent()</code>	156
8.24.3.6	<code>initialize_initializer_time()</code>	157
8.24.3.7	<code>julian_date_at_epoch()</code>	157
8.24.3.8	<code>operator=()</code>	158
8.24.3.9	<code>seconds_of_year()</code>	158
8.24.3.10	<code>set_epoch()</code>	158
8.24.3.11	<code>set_time_by_days()</code>	158
8.24.3.12	<code>set_time_by_seconds()</code>	159
8.24.3.13	<code>set_time_by_trunc_julian()</code>	159
8.24.4	Friends And Related Function Documentation	160
8.24.4.1	<code>init_attrjeod__TimeStandard</code>	160
8.24.4.2	<code>InputProcessor</code>	160
8.24.4.3	<code>TimeUDE</code>	160
8.24.5	Field Documentation	160
8.24.5.1	<code>calendar_day</code>	161
8.24.5.2	<code>calendar_hour</code>	161
8.24.5.3	<code>calendar_minute</code>	161
8.24.5.4	<code>calendar_month</code>	161
8.24.5.5	<code>calendar_second</code>	162
8.24.5.6	<code>calendar_year</code>	162
8.24.5.7	<code>julian_date</code>	162
8.24.5.8	<code>last_calendar_update</code>	162
8.24.5.9	<code>prev_julian_day</code>	163
8.24.5.10	<code>seconds_at_year_start</code>	163
8.24.5.11	<code>send_warning_pre_1968</code>	163
8.24.5.12	<code>tjt_at_epoch</code>	163
8.24.5.13	<code>tjt_jd_offset</code>	164
8.24.5.14	<code>tjt_mjt_offset</code>	164
8.24.5.15	<code>trunc_julian_time</code>	164

8.24.5.16 year_of_last_soy	165
8.25 jeod::TimeTAI Class Reference	165
8.25.1 Detailed Description	166
8.25.2 Constructor & Destructor Documentation	166
8.25.2.1 TimeTAI() [1/2]	166
8.25.2.2 ~TimeTAI()	166
8.25.2.3 TimeTAI() [2/2]	166
8.25.3 Member Function Documentation	166
8.25.3.1 operator=()	167
8.25.3.2 set_epoch()	167
8.25.4 Friends And Related Function Documentation	167
8.25.4.1 init_attrjeod__TimeTAI	167
8.25.4.2 InputProcessor	167
8.26 jeod::TimeTDB Class Reference	168
8.26.1 Detailed Description	168
8.26.2 Constructor & Destructor Documentation	168
8.26.2.1 TimeTDB() [1/2]	169
8.26.2.2 ~TimeTDB()	169
8.26.2.3 TimeTDB() [2/2]	169
8.26.3 Member Function Documentation	169
8.26.3.1 operator=()	169
8.26.3.2 set_epoch()	169
8.26.4 Friends And Related Function Documentation	170
8.26.4.1 init_attrjeod__TimeTDB	170
8.26.4.2 InputProcessor	170
8.27 jeod::TimeTT Class Reference	170
8.27.1 Detailed Description	171
8.27.2 Constructor & Destructor Documentation	171
8.27.2.1 TimeTT() [1/2]	171
8.27.2.2 ~TimeTT()	171

8.27.2.3	TimeTT() [2/2]	171
8.27.3	Member Function Documentation	172
8.27.3.1	operator=()	172
8.27.3.2	set_epoch()	172
8.27.4	Friends And Related Function Documentation	172
8.27.4.1	init_attrjeod__TimeTT	172
8.27.4.2	InputProcessor	172
8.28	jeod::TimeUDE Class Reference	173
8.28.1	Detailed Description	175
8.28.2	Constructor & Destructor Documentation	175
8.28.2.1	TimeUDE() [1/2]	175
8.28.2.2	~TimeUDE()	175
8.28.2.3	TimeUDE() [2/2]	176
8.28.3	Member Function Documentation	176
8.28.3.1	add_type_initialize()	176
8.28.3.2	clock_update()	176
8.28.3.3	convert_epoch_to_update()	177
8.28.3.4	initialize_from_parent()	177
8.28.3.5	initialize_initializer_time()	178
8.28.3.6	must_be_singleton()	179
8.28.3.7	operator=()	179
8.28.3.8	set_epoch_dyn()	179
8.28.3.9	set_epoch_initializing_value()	180
8.28.3.10	set_epoch_std()	180
8.28.3.11	set_epoch_times()	181
8.28.3.12	set_epoch_ude()	181
8.28.3.13	set_initial_times()	182
8.28.3.14	set_time_by_clock()	182
8.28.3.15	set_time_by_days()	182
8.28.3.16	set_time_by_seconds()	183

8.28.3.17	verify_epoch()	183
8.28.3.18	verify_init()	184
8.28.3.19	verify_update()	184
8.28.4	Friends And Related Function Documentation	184
8.28.4.1	init_attrjeod__TimeUDE	184
8.28.4.2	InputProcessor	184
8.28.5	Field Documentation	184
8.28.5.1	clock_day	185
8.28.5.2	clock_hour	185
8.28.5.3	clock_minute	185
8.28.5.4	clock_second	185
8.28.5.5	epoch_data_present	186
8.28.5.6	epoch_day	186
8.28.5.7	epoch_defined_in_name	186
8.28.5.8	epoch_format	186
8.28.5.9	epoch_hour	187
8.28.5.10	epoch_index	187
8.28.5.11	epoch_initializing_value	187
8.28.5.12	epoch_minute	187
8.28.5.13	epoch_month	188
8.28.5.14	epoch_second	188
8.28.5.15	epoch_value_is_set_calendar	188
8.28.5.16	epoch_value_is_set_clock	188
8.28.5.17	epoch_value_is_set_number	189
8.28.5.18	epoch_year	189
8.28.5.19	initial_value_format	189
8.28.5.20	initializing_data_present	189
8.28.5.21	last_clock_update	190
8.28.5.22	update_index	190
8.29	jeod::TimeUT1 Class Reference	190

8.29.1 Detailed Description	191
8.29.2 Constructor & Destructor Documentation	191
8.29.2.1 TimeUT1() [1/2]	191
8.29.2.2 ~TimeUT1()	192
8.29.2.3 TimeUT1() [2/2]	192
8.29.3 Member Function Documentation	192
8.29.3.1 get_days()	192
8.29.3.2 operator=()	192
8.29.3.3 set_epoch()	192
8.29.4 Friends And Related Function Documentation	193
8.29.4.1 init_attrjeod__TimeUT1	193
8.29.4.2 InputProcessor	193
8.29.5 Field Documentation	193
8.29.5.1 true_ut1	193
8.30 jeod::TimeUTC Class Reference	193
8.30.1 Detailed Description	194
8.30.2 Constructor & Destructor Documentation	194
8.30.2.1 TimeUTC() [1/2]	194
8.30.2.2 ~TimeUTC()	195
8.30.2.3 TimeUTC() [2/2]	195
8.30.3 Member Function Documentation	195
8.30.3.1 operator=()	195
8.30.3.2 set_epoch()	195
8.30.4 Friends And Related Function Documentation	195
8.30.4.1 init_attrjeod__TimeUTC	195
8.30.4.2 InputProcessor	196
8.30.5 Field Documentation	196
8.30.5.1 true_utc	196

9	File Documentation	197
9.1	class_declarations.hh File Reference	197
9.1.1	Detailed Description	197
9.2	tai_to_ut1.cc File Reference	197
9.2.1	Macro Definition Documentation	198
9.2.1.1	JEOD_FRIEND_CLASS	198
9.3	tai_to_ut1.hh File Reference	198
9.4	tai_to_utc.cc File Reference	198
9.4.1	Macro Definition Documentation	198
9.4.1.1	JEOD_FRIEND_CLASS	199
9.5	tai_to_utc.hh File Reference	199
9.6	time.cc File Reference	199
9.6.1	Detailed Description	199
9.7	time.hh File Reference	199
9.7.1	Detailed Description	200
9.8	time__add_type_update.cc File Reference	200
9.8.1	Detailed Description	200
9.9	time_converter.cc File Reference	201
9.9.1	Detailed Description	201
9.10	time_converter.hh File Reference	201
9.10.1	Detailed Description	202
9.11	time_converter_dyn_tai.cc File Reference	202
9.11.1	Detailed Description	202
9.12	time_converter_dyn_tai.hh File Reference	202
9.12.1	Detailed Description	203
9.13	time_converter_dyn_tdb.cc File Reference	203
9.13.1	Detailed Description	203
9.14	time_converter_dyn_tdb.hh File Reference	203
9.14.1	Detailed Description	204
9.15	time_converter_dyn_ude.cc File Reference	204

9.15.1 Detailed Description	204
9.16 time_converter_dyn_ude.hh File Reference	204
9.16.1 Detailed Description	205
9.17 time_converter_std_ude.cc File Reference	205
9.17.1 Detailed Description	205
9.18 time_converter_std_ude.hh File Reference	205
9.18.1 Detailed Description	206
9.19 time_converter_tai_gps.cc File Reference	206
9.19.1 Detailed Description	206
9.20 time_converter_tai_gps.hh File Reference	206
9.20.1 Detailed Description	207
9.21 time_converter_tai_tdb.cc File Reference	207
9.21.1 Detailed Description	207
9.22 time_converter_tai_tdb.hh File Reference	207
9.22.1 Detailed Description	208
9.23 time_converter_tai_tt.cc File Reference	208
9.23.1 Detailed Description	208
9.24 time_converter_tai_tt.hh File Reference	208
9.24.1 Detailed Description	209
9.25 time_converter_tai_ut1.cc File Reference	209
9.25.1 Detailed Description	209
9.26 time_converter_tai_ut1.hh File Reference	209
9.26.1 Detailed Description	210
9.27 time_converter_tai_utc.cc File Reference	210
9.27.1 Detailed Description	210
9.28 time_converter_tai_utc.hh File Reference	210
9.28.1 Detailed Description	211
9.29 time_converter_ut1_gmst.cc File Reference	211
9.29.1 Detailed Description	211
9.30 time_converter_ut1_gmst.hh File Reference	211

9.30.1 Detailed Description	212
9.31 time_dyn.cc File Reference	212
9.31.1 Detailed Description	212
9.32 time_dyn.hh File Reference	212
9.32.1 Detailed Description	213
9.33 time_enum.hh File Reference	213
9.33.1 Detailed Description	213
9.34 time_gmst.cc File Reference	213
9.34.1 Detailed Description	214
9.35 time_gmst.hh File Reference	214
9.35.1 Detailed Description	214
9.36 time_gps.cc File Reference	214
9.36.1 Detailed Description	215
9.37 time_gps.hh File Reference	215
9.37.1 Detailed Description	215
9.38 time_links.hh File Reference	215
9.38.1 Detailed Description	216
9.39 time_manager.cc File Reference	216
9.39.1 Detailed Description	216
9.40 time_manager.hh File Reference	216
9.40.1 Detailed Description	217
9.41 time_manager__initialize.cc File Reference	217
9.41.1 Detailed Description	217
9.42 time_manager_init.cc File Reference	217
9.42.1 Detailed Description	218
9.43 time_manager_init.hh File Reference	218
9.43.1 Detailed Description	218
9.44 time_messages.cc File Reference	218
9.44.1 Detailed Description	219
9.45 time_messages.hh File Reference	219

9.45.1 Detailed Description	219
9.46 time_met.cc File Reference	219
9.46.1 Detailed Description	220
9.47 time_met.hh File Reference	220
9.47.1 Detailed Description	220
9.48 time_standard.cc File Reference	220
9.48.1 Detailed Description	221
9.49 time_standard.hh File Reference	221
9.49.1 Detailed Description	221
9.50 time_tai.cc File Reference	221
9.50.1 Detailed Description	221
9.51 time_tai.hh File Reference	222
9.51.1 Detailed Description	222
9.52 time_tdb.cc File Reference	222
9.52.1 Detailed Description	222
9.53 time_tdb.hh File Reference	223
9.53.1 Detailed Description	223
9.54 time_tt.cc File Reference	223
9.54.1 Detailed Description	223
9.55 time_tt.hh File Reference	224
9.55.1 Detailed Description	224
9.56 time_ude.cc File Reference	224
9.56.1 Detailed Description	224
9.57 time_ude.hh File Reference	225
9.57.1 Detailed Description	225
9.58 time_ut1.cc File Reference	225
9.58.1 Detailed Description	225
9.59 time_ut1.hh File Reference	226
9.59.1 Detailed Description	226
9.60 time_utc.cc File Reference	226
9.60.1 Detailed Description	226
9.61 time_utc.hh File Reference	227
9.61.1 Detailed Description	227

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	13
Environment	14
Time	15

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	19
----------------------	--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::JeodBaseTime	23
jeod::TimeDyn	100
jeod::TimeStandard	152
jeod::TimeGMST	106
jeod::TimeGPS	109
jeod::TimeTAI	165
jeod::TimeTDB	168
jeod::TimeTT	170
jeod::TimeUT1	190
jeod::TimeUTC	193
jeod::TimeUDE	173
jeod::TimeMET	149
JeodIntegrationTime	
jeod::TimeManager	119
jeod::TimeConverter	39
jeod::TimeConverter_Dyn_TAI	48
jeod::TimeConverter_Dyn_TDB	52
jeod::TimeConverter_Dyn_UDE	55
jeod::TimeConverter_STD_UDE	59
jeod::TimeConverter_TAI_GPS	64
jeod::TimeConverter_TAI_TDB	68
jeod::TimeConverter_TAI_TT	75
jeod::TimeConverter_TAI_UT1	79
jeod::TimeConverter_TAI_UTC	88
jeod::TimeConverter_UT1_GMST	97
jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data	87
jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data	96
jeod::TimeEnum	105
jeod::TimeManagerInit	131
jeod::TimeMessages	144
TreeLinks	
jeod::TimeLinks	116

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::JeodBaseTime	
JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	23
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	39
jeod::TimeConverter_Dyn_TAI	
Define class TimeConverter_Dyn_TAI , which converts from simulation dynamic time to International Atomic Time	48
jeod::TimeConverter_Dyn_TDB	
Define class TimeConverter_Dyn_TDB , which converts from simulation dynamic time to Barycentric Dynamic Time	52
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	55
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	59
jeod::TimeConverter_TAI_GPS	
Define class TimeConverter_TAI_GPS , which converts between International Atomic Time and the clock associated with the Global Positioning System	64
jeod::TimeConverter_TAI_TDB	
Define class TimeConverter_TAI_TDB , which converts from International Atomic Time to Barycentric Dynamic Time	68
jeod::TimeConverter_TAI_TT	
Converts between International Atomic Time and Terrestrial Time	75
jeod::TimeConverter_TAI_UT1	
Define class TimeConverter_TAI_UT1 , which converts between International Atomic Time and Universal Time	79
jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data	87
jeod::TimeConverter_TAI_UTC	
Converts between International Atomic Time and Coordinated Universal Time	88
jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data	96
jeod::TimeConverter_UT1_GMST	
Converts between Universal Time and Greenwich Mean Sidereal Time	97

jeod::TimeDyn	Represents the Dynamic Time in the simulation	100
jeod::TimeEnum	Contains an enumeration of the formats in which time can be represented	105
jeod::TimeGMST	To represent the clock known as Greenwich Mean Sidereal Time	106
jeod::TimeGPS	To represent the time associated with the Global Positioning System	109
jeod::TimeLinks	116
jeod::TimeManager	To manage the various time representations and the converters between them throughout the simulation	119
jeod::TimeManagerInit	To initialize the Time Manager	131
jeod::TimeMessages	Specify the message IDs used in the Time model	144
jeod::TimeMET	A type of UDE time that allows for deliberate holds, or pauses	149
jeod::TimeStandard	A class that serves as the base for all time representations that are well defined outside the simulation	152
jeod::TimeTAI	Represents International Atomic Time	165
jeod::TimeTDB	Represents Barycentric Dynamic Time	168
jeod::TimeTT	Represents Terrestrial Time	170
jeod::TimeUDE	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition	173
jeod::TimeUT1	Represents Universal Time	190
jeod::TimeUTC	Represents Coordinated Universal Time	193

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

class_declarations.hh	Forward declaration of classes defined in time.hh	197
tai_to_ut1.cc		197
tai_to_ut1.hh		198
tai_to_utc.cc		198
tai_to_utc.hh		199
time.cc	JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	199
time.hh	JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	199
time__add_type_update.cc	Define JeodBaseTime::add_type_update	200
time_converter.cc	An abstract class that defines the basic structure of all the methods used by the converter objects	201
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	201
time_converter_dyn_tai.cc	Converts between International Atomic Time and Dynamic Time	202
time_converter_dyn_tai.hh	Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time	202
time_converter_dyn_tdb.cc	Converts between Dynamic Time and Barycentric Dynamic Time	203
time_converter_dyn_tdb.hh	Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time	203
time_converter_dyn_ude.cc	Converts between Dynamic Time and a time with User-Defined-Epoch	204
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	204
time_converter_std_ude.cc	Define member functions for class TimeConverter_STD_UDE	205

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	205
time_converter_tai_gps.cc	Converts between International Atomic Time and the clock associated with the Global Positioning System	206
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	206
time_converter_tai_tdb.cc	Converts from International Atomic Time to Barycentric Dynamic Time	207
time_converter_tai_tdb.hh	Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time	207
time_converter_tai_tt.cc	Converts between International Atomic Time and Terrestrial Time	208
time_converter_tai_tt.hh	Converts between International Atomic Time and Terrestrial Time	208
time_converter_tai_ut1.cc	Converts between International Atomic Time and Universal Time	209
time_converter_tai_ut1.hh	Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time	209
time_converter_tai_utc.cc	Converts between International Atomic Time and Coordinated Universal Time	210
time_converter_tai_utc.hh	Converts between International Atomic Time and Coordinated Universal Time	210
time_converter_ut1_gmst.cc	Define member functions for class TimeConverter_UT1_GMST	211
time_converter_ut1_gmst.hh	Converts between Universal Time and Greenwich Mean Sidereal Time	211
time_dyn.cc	Define member functions for Dynamic Time	212
time_dyn.hh	Represents the Dynamic Time in the simulation	212
time_enum.hh	Contains an enumeration of the formats in which time can be represented	213
time_gmst.cc	Define member functions for Greenwich Mean Sidereal Time	213
time_gmst.hh	To represent the clock known as Greenwich Mean Sidereal Time	214
time_gps.cc	Define member functions for the clock associated with the Global Positioning System	214
time_gps.hh	To represent the time associated with the Global Positioning System	215
time_links.hh	Define the class TimeLinks, which defines the hierarchy of JEOD time conversions	215
time_manager.cc	Define member functions for class TimeManager	216
time_manager.hh	To manage the various time representations and the converters between them throughout the simulation	216
time_manager_initialize.cc	Define TimeManager::initialize	217
time_manager_init.cc	Define member functions for the Time Manager Initialization	217
time_manager_init.hh	To initialize the Time Manager	218

time_messages.cc	Implement the class TimeMessages	218
time_messages.hh	Define the class TimeMessages, the class that specifies the message IDs used in the Time model	219
time_met.cc	Define member functions for Mission Elapsed Time	219
time_met.hh	A type of UDE time that allows for deliberate holds, or pauses	220
time_standard.cc	An abstract class, this defines the basic structure of member functions for all Standard Times .	220
time_standard.hh	A class that serves as the base for all time representations that are well defined outside the simulation	221
time_tai.cc	Define member functions for International Atomic Time	221
time_tai.hh	Represents International Atomic Time	222
time_tdb.cc	Define member functions Barycentric Dynamic Time	222
time_tdb.hh	Represents Barycentric Dynamic Time	223
time_tt.cc	Define member functions for Terrestrial Time	223
time_tt.hh	Represents Terrestrial Time	224
time_ude.cc	Define member functions for those times with a User-Defined-Epoch	224
time_ude.hh	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition	225
time_ut1.cc	Define member functions for Universal Time	225
time_ut1.hh	Represents Universal Time	226
time_utc.cc	Define member functions for Coordinated Universal Time	226
time_utc.hh	Represents Coordinated Universal Time	227

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 36 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 196 of file time_converter.hh.

Chapter 8

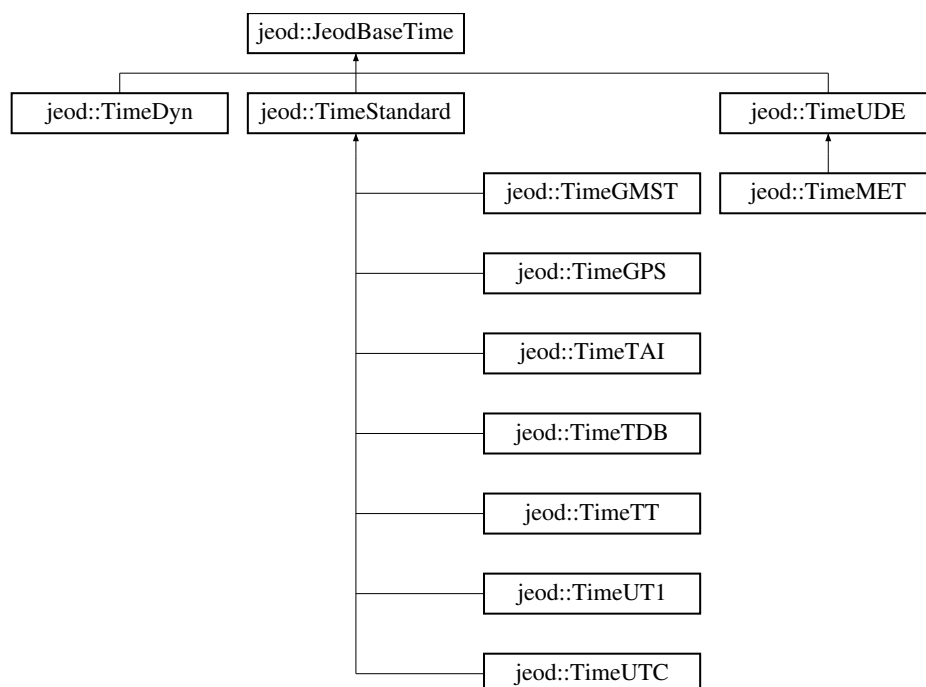
Data Structure Documentation

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](#).

- `JeodBaseTime` (const `JeodBaseTime` &)=delete
- `JeodBaseTime` & `operator=` (const `JeodBaseTime` &)=delete
- virtual bool `must_be_singleton` ()
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` ()
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](#) {0.0001}
The resolution limit when generating clock and calendar-clock values.
- [TimeLinks](#) [links](#)
Linkage to the hierarchy of time conversions.

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 93 of file time.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 [JeodBaseTime](#)() [1/2]

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

Construct a [JeodBaseTime](#).

Definition at line 57 of file time.cc.

8.1.2.2 `~JeodBaseTime()`

```
jeod::JeodBaseTime::~~JeodBaseTime ( ) [virtual]
```

Destroy a [JeodBaseTime](#).

Definition at line 182 of file time.cc.

References links.

8.1.2.3 `JeodBaseTime()` [2/2]

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

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	<i>parent</i>	the time responsible for updating this time.
----	---------------	--

Definition at line 91 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	<i>seeking_status</i>	status-value for auto-seek
in	<i>time_manager_init</i>	TM initializer

Reimplemented in [jeod::TimeUDE](#), and [jeod::TimeStandard](#).

Definition at line 70 of file time.cc.

References [jeod::TimeMessages::invalid_setup_error](#).

Referenced by [jeod::TimeStandard::add_type_initialize\(\)](#), and [jeod::TimeUDE::add_type_initialize\(\)](#).

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	<i>seeking_status</i>	status-value for auto-seek.
in	<i>time_manager_init</i>	The TM initializer.

Definition at line 70 of file time__add_type_update.cc.

References `add_parent()`, `add_type_update()`, `jeod::TimeManagerInit::get_conv_dir_upd()`, `jeod::TimeManagerInit::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 203 of file `time.hh`.

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	<i>time_manager_init</i>	TM initializer
----	--------------------------	----------------

Reimplemented in [jeod::TimeUDE](#), and [jeod::TimeStandard](#).

Definition at line 103 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

<i>tm_init</i>	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 219 of file time.hh.

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 ( ) [virtual]
```

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

Returns

Boolean value

Reimplemented in [jeod::TimeUDE](#).

Definition at line 120 of file time.cc.

8.1.3.9 operator=()

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

8.1.3.10 override_initialized()

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

Force reset the initialization status.

Definition at line 211 of file time.hh.

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 195 of file time.hh.

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 187 of file time.hh.

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	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented in [jeod::TimeUDE](#), [jeod::TimeStandard](#), and [jeod::TimeGPS](#).

Definition at line 173 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	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented in [jeod::TimeUDE](#), [jeod::TimeStandard](#), and [jeod::TimeGPS](#).

Definition at line 160 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 ( ) [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 132 of file `time.cc`.

References [jeod::TimeConverter::convert_a_to_b\(\)](#), [jeod::TimeConverter::convert_b_to_a\(\)](#), [jeod::TimeConverter::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 95 of file `time.hh`.

8.1.4.3 `TimeConverter`

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

Definition at line 95 of file `time.hh`.

8.1.4.4 `TimeManagerInit`

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

Definition at line 96 of file `time.hh`.

8.1.5 Field Documentation

8.1.5.1 clock_resolution

```
double jeod::JeodBaseTime::clock_resolution {0.0001} [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 163 of file time.hh.

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

8.1.5.2 days

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

Elapsed time from epoch.

trick_units(day)

Definition at line 151 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(), 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 143 of file time.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), add_type_update(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), 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 156 of file time.hh.

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

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 123 of file time.hh.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeStandard::initialize_initializer_time()`, 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 147 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()`, 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 103 of file time.hh.

Referenced by jeod::TimeStandard::initialize_initializer_time(), 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 children
trick_units(–)

Definition at line 169 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 118 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(), jeod::TimeDyn::TimeDyn(), jeod::↵
TimeGMST::TimeGMST(), jeod::TimeGPS::TimeGPS(), jeod::TimeMET::TimeMET(), jeod::TimeTAI::TimeTAI(),
jeod::TimeTDB::TimeTDB(), jeod::TimeTT::TimeTT(), jeod::TimeUT1::TimeUT1(), jeod::TimeUTC::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 113 of file time.hh.

Referenced by `jeod::TimeUDE::clock_update()`, `jeod::TimeConverter_TAI_TT::convert_a_to_b()`, `jeod::TimeConverter_TAI_GPS::convert_a_to_b()`, `jeod::TimeConverter_Dyn_TAI::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()`, `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 133 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()`, `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 108 of file time.hh.

Referenced by add_type_update(), 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 137 of file time.hh.

Referenced by add_type_update(), 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 128 of file time.hh.

Referenced by jeod::TimeUDE::add_type_initialize(), add_type_update(), jeod::TimeUDE::convert_epoch_to_↵
update(), jeod::TimeUDE::initialize_from_parent(), jeod::TimeUDE::initialize_initializer_time(), and jeod::TimeUD↵
E::verify_update().

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

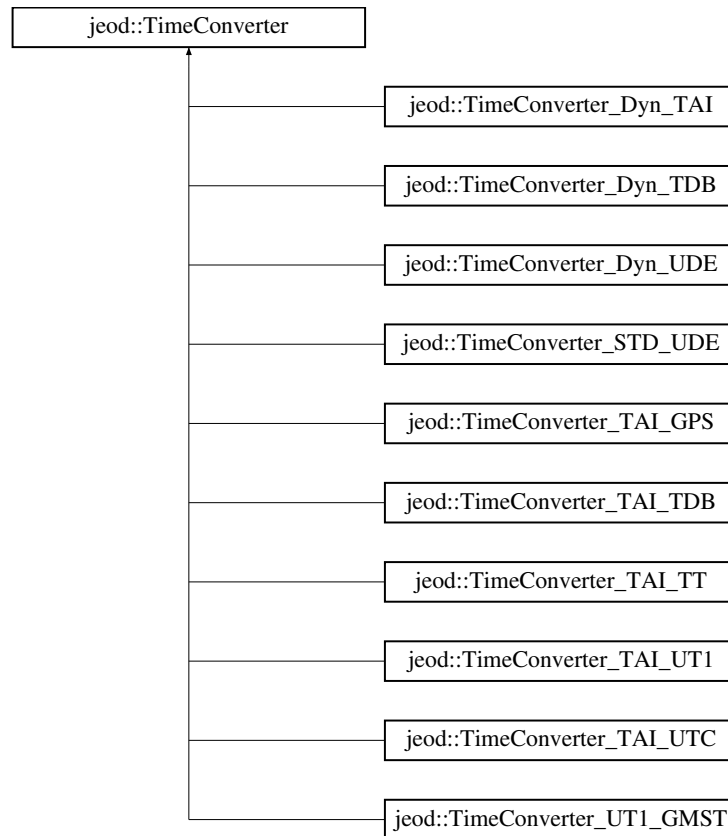
- [time.hh](#)
- [time.cc](#)
- [time__add_type_update.cc](#)

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:



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` ()=default
- `TimeConverter` (const `TimeConverter` &)=delete
- `TimeConverter` & `operator=` (const `TimeConverter` &)=delete
- virtual void `initialize` (`JeodBaseTime` *parent, `JeodBaseTime` *child, const int direction)=0
Initialize the converter.
- virtual bool `is_initialized` ()
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` ()
Default converter from time 'a' to time 'b'.
- virtual void `convert_b_to_a` ()
Default converter from time 'b' to time 'a'.
- virtual void `reset_a_to_b_offset` ()
Resets the offset between type a and type b mid-sim.
- virtual void `verify_table_lookup_ends` ()
This function does absolutely nothing.
- double `get_a_to_b_offset` ()
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` ()=default
- 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` {`NO_DIRECTION`}
Bit packed flag specifying whether how a converter can be used.

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 97 of file `time_converter.hh`.

8.2.3 Constructor & Destructor Documentation

8.2.3.1 ~TimeConverter()

```
virtual jeod::TimeConverter::~~TimeConverter ( ) [virtual], [default]
```

8.2.3.2 TimeConverter() [1/2]

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

8.2.3.3 TimeConverter() [2/2]

```
jeod::TimeConverter::TimeConverter ( ) [protected], [default]
```

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	<i>query</i>	converter directions to check
----	--------------	-------------------------------

Definition at line 111 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 ( ) [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 135 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 ( ) [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 145 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 ( ) [inline]
```

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

Returns

`a_to_b_offset` member.

Definition at line 178 of file `time_converter.hh`.

8.2.4.5 initialize()

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

Initialize the converter.

Parameters

in	<i>parent</i>	parent-type
in	<i>child</i>	child-type
in	<i>direction</i>	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 ( ) [virtual]
```

Return internal initialized status bool.

Definition at line 52 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 & ) [delete]
```

8.2.4.8 override_initialized()

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

Definition at line 154 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 ( ) [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 153 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	<i>master_ptr</i>	Time used to initialize the converter
in	<i>sub_ptr</i>	Other time-type associated with the converter
in	<i>int_dir</i>	+1 a=parent; -1 b=parent; 0 error

Definition at line 66 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 ( ) [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 166 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 91 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 113 of file time_converter.hh.

Referenced by jeod::TimeManager::register_converter(), 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 129 of file time_converter.hh.

Referenced by jeod::TimeConverter_TAI_GPS::convert_a_to_b(), jeod::TimeConverter_Dyn_TAI::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_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_TDB::convert_b_to_a(), jeod::TimeConverter_TAI_UTC::convert_b_to_a(), jeod::TimeConverter_TAI_UT1::convert_b_to_a(), jeod::TimeConverter_TAI_TT::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_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 jeod::TimeConverter_TAI_TDB::set_a_to_b_offset().

8.2.6.3 b_name

```
std::string jeod::TimeConverter::b_name {}
```

name of time-type "b".

trick_units(-)

Definition at line 118 of file time_converter.hh.

Referenced by jeod::TimeManager::register_converter(), 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 124 of file time_converter.hh.

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(), jeod::TimeConverter_TAI_UT1::initialize(), is_initialized(), and override_initialized().

8.2.6.5 valid_directions

`Direction` jeod::TimeConverter::valid_directions {NO_DIRECTION} [protected]

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

Definition at line 134 of file `time_converter.hh`.

Referenced by `can_convert()`, `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:

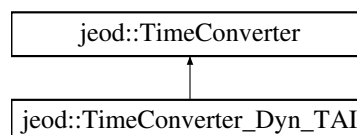
- [time_converter.hh](#)
- [time_converter.cc](#)

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 ()` override=default
- `TimeConverter_Dyn_TAI (const TimeConverter_Dyn_TAI &)=delete`
- `TimeConverter_Dyn_TAI & operator= (const TimeConverter_Dyn_TAI &)=delete`
- `void initialize (JeodBaseTime *parent, JeodBaseTime *child, const int direction)` override
Initialize the converter.
- `void convert_a_to_b ()` override
Convert from `TimeDyn` to `TimeTAI`.

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 ( )
```

Construct a [TimeConverter_Dyn_TAI](#).

Definition at line 58 of file `time_converter_dyn_tai.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B_UPDATE`, `jeod::TimeConverter::b_name`, and `jeod::TimeConverter::valid_directions`.

8.3.2.2 ~TimeConverter_Dyn_TAI()

```
jeod::TimeConverter_Dyn_TAI::~~TimeConverter_Dyn_TAI ( ) [override], [default]
```

8.3.2.3 TimeConverter_Dyn_TAI() [2/2]

```
jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI (
    const TimeConverter\_Dyn\_TAI & ) [delete]
```

8.3.3 Member Function Documentation

8.3.3.1 convert_a_to_b()

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

Convert from [TimeDyn](#) to [TimeTAI](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 175 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 ) [override], [virtual]
```

Initialize the converter.

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 71 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 & ) [delete]
```

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 90 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 95 of file `time_converter_dyn_tai.hh`.

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

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 100 of file `time_converter_dyn_tai.hh`.

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

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

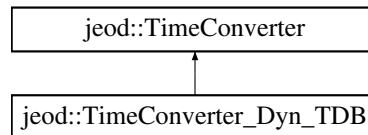
- [time_converter_dyn_tai.hh](#)
- [time_converter_dyn_tai.cc](#)

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](#) () override=default
- [TimeConverter_Dyn_TDB](#) (const [TimeConverter_Dyn_TDB](#) &)=delete
- [TimeConverter_Dyn_TDB](#) & operator= (const [TimeConverter_Dyn_TDB](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeDyn](#) to [TimeTDB](#).

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 90 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 ( )
```

Construct a [TimeConverter_Dyn_TDB](#).

Definition at line 58 of file `time_converter_dyn_tdb.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B`, `jeod::TimeConverter::b_name`, and `jeod::TimeConverter::valid_directions`.

8.4.2.2 ~TimeConverter_Dyn_TDB()

```
jeod::TimeConverter_Dyn_TDB::~~TimeConverter_Dyn_TDB ( ) [override], [default]
```

8.4.2.3 TimeConverter_Dyn_TDB() [2/2]

```
jeod::TimeConverter_Dyn_TDB::TimeConverter_Dyn_TDB (
    const TimeConverter\_Dyn\_TDB & ) [delete]
```

8.4.3 Member Function Documentation

8.4.3.1 convert_a_to_b()

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

Convert from [TimeDyn](#) to [TimeTDB](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 142 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 ) [override], [virtual]
```

Initialize the converter.

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 71 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 & ) [delete]
```

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 92 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 97 of file time_converter_dyn_tdb.hh.

Referenced by convert_a_to_b(), and initialize().

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 102 of file time_converter_dyn_tdb.hh.

Referenced by convert_a_to_b(), and initialize().

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

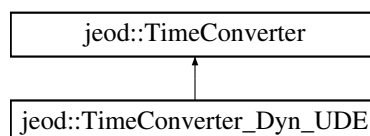
- [time_converter_dyn_tdb.hh](#)
- [time_converter_dyn_tdb.cc](#)

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](#) () override=default
- [TimeConverter_Dyn_UDE](#) (const [TimeConverter_Dyn_UDE](#) &)=delete
- [TimeConverter_Dyn_UDE](#) & operator= (const [TimeConverter_Dyn_UDE](#) &)=delete
- void [reset_a_to_b_offset](#) () override
Resets the value of a_to_b_offset.
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeDyn](#) to [TimeUDE](#).

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 90 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 ( )
```

Construct a [TimeConverter_Dyn_UDE](#).

Definition at line 57 of file [time_converter_dyn_ude.cc](#).

References [jeod::TimeConverter::a_name](#), [jeod::TimeConverter::A_TO_B](#), [jeod::TimeConverter::b_name](#), and [jeod::TimeConverter::valid_directions](#).

8.5.2.2 ~TimeConverter_Dyn_UDE()

```
jeod::TimeConverter_Dyn_UDE::~~TimeConverter_Dyn_UDE ( ) [override], [default]
```

8.5.2.3 TimeConverter_Dyn_UDE() [2/2]

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

8.5.3 Member Function Documentation

8.5.3.1 convert_a_to_b()

```
void jeod::TimeConverter_Dyn_UDE::convert_a_to_b ( ) [override], [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 145 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 ) [override], [virtual]
```

Initialize the converter.

Assumptions and Limitations

- This class converts from [TimeDyn](#) to [TimeUDE](#) *only*

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 73 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 & ) [delete]
```

8.5.3.4 reset_a_to_b_offset()

```
void jeod::TimeConverter_Dyn_UDE::reset_a_to_b_offset ( ) [override], [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 92 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 97 of file `time_converter_dyn_ude.hh`.

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

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 102 of file `time_converter_dyn_ude.hh`.

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

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

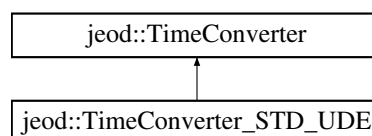
- [time_converter_dyn_ude.hh](#)
- [time_converter_dyn_ude.cc](#)

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](#) () override=default
- [TimeConverter_STD_UDE](#) (const [TimeConverter_STD_UDE](#) &)=delete
- [TimeConverter_STD_UDE](#) & [operator=](#) (const [TimeConverter_STD_UDE](#) &)=delete
- void [reset_a_to_b_offset](#) () override
Resets the value of [a_to_b_offset](#).
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeSTD](#) to [TimeUDE](#).
- void [convert_b_to_a](#) () override
Convert from [TimeUDE](#) to [TimeSTD](#).

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 90 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 ( )
```

Construct a [TimeConverter_STD_UDE](#).

Definition at line 57 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, and `jeod::TimeConverter::valid_directions`.

8.6.2.2 ~TimeConverter_STD_UDE()

```
jeod::TimeConverter_STD_UDE::~~TimeConverter_STD_UDE ( ) [override], [default]
```

8.6.2.3 TimeConverter_STD_UDE() [2/2]

```
jeod::TimeConverter_STD_UDE::TimeConverter_STD_UDE (
    const TimeConverter\_STD\_UDE & ) [delete]
```

8.6.3 Member Function Documentation

8.6.3.1 convert_a_to_b()

```
void jeod::TimeConverter_STD_UDE::convert_a_to_b ( ) [override], [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 154 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 ( ) [override], [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 166 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 ) [override], [virtual]
```

Initialize the converter.

Assumptions and Limitations

- This class converts from [TimeDyn](#) to [TimeUDE](#) *only*

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 73 of file time_converter_std_ude.cc.

References [jeod::TimeConverter::a_to_b_offset](#), [failed_null_test](#), [jeod::TimeConverter::initialized](#), [jeod::Time↵Messages::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 & ) [delete]
```

8.6.3.5 reset_a_to_b_offset()

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

Resets the value of a_to_b_offset.

Reimplemented from [jeod::TimeConverter](#).

Definition at line 174 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 92 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 98 of file time_converter_std_ude.hh.

Referenced by [initialize\(\)](#).

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 103 of file time_converter_std_ude.hh.

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

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 108 of file time_converter_std_ude.hh.

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

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

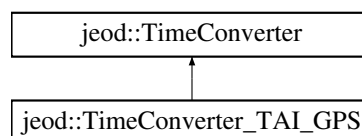
- [time_converter_std_ude.hh](#)
- [time_converter_std_ude.cc](#)

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](#) () override=default
- [TimeConverter_TAI_GPS](#) (const [TimeConverter_TAI_GPS](#) &)=delete
- [TimeConverter_TAI_GPS](#) & operator= (const [TimeConverter_TAI_GPS](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeTAI](#) to [TimeGPS](#).
- void [convert_b_to_a](#) () override
Convert from [TimeGPS](#) to [TimeTAI](#).

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 87 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 ( )
```

Construct a [TimeConverter_TAI_GPS](#).

Definition at line 58 of file [time_converter_tai_gps.cc](#).

References [jeod::TimeConverter::a_name](#), [jeod::TimeConverter::ANY_DIRECTION](#), [jeod::TimeConverter::b_name](#), and [jeod::TimeConverter::valid_directions](#).

8.7.2.2 ~TimeConverter_TAI_GPS()

```
jeod::TimeConverter_TAI_GPS::~~TimeConverter_TAI_GPS ( ) [override], [default]
```

8.7.2.3 TimeConverter_TAI_GPS() [2/2]

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

8.7.3 Member Function Documentation

8.7.3.1 convert_a_to_b()

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

Convert from [TimeTAI](#) to [TimeGPS](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 105 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 ( ) [override], [virtual]
```

Convert from [TimeGPS](#) to [TimeTAI](#).

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::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 ) [override], [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 74 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 & ) [delete]
```

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 89 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 99 of file time_converter_tai_gps.hh.

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

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 94 of file time_converter_tai_gps.hh.

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

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

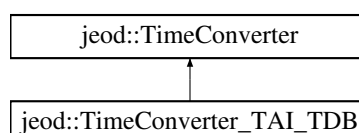
- [time_converter_tai_gps.hh](#)
- [time_converter_tai_gps.cc](#)

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](#) () override=default
- [TimeConverter_TAI_TDB](#) (const [TimeConverter_TAI_TDB](#) &)=delete
- [TimeConverter_TAI_TDB](#) & [operator=](#) (const [TimeConverter_TAI_TDB](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [set_a_to_b_offset](#) ()
- void [convert_a_to_b](#) () override
Default converter from time 'a' to time 'b'.
- void [convert_b_to_a](#) () override
Default converter from time 'b' to time 'a'.

Private Attributes

- double [TAI_to_TT_offset](#) {32.184}
The offset from TAI to TT.
- double [a_to_b_offset_epoch](#) {}
The epoch 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 89 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 ( )
```

Definition at line 67 of file `time_converter_tai_tdb.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, and `jeod::TimeConverter::valid_directions`.

8.8.2.2 ~TimeConverter_TAI_TDB()

```
jeod::TimeConverter_TAI_TDB::~~TimeConverter_TAI_TDB ( ) [override], [default]
```

8.8.2.3 TimeConverter_TAI_TDB() [2/2]

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

8.8.3 Member Function Documentation

8.8.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_TDB::convert_a_to_b ( ) [override], [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 129 of file `time_converter_tai_tdb.cc`.

References `jeod::TimeConverter::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 ( ) [override], [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 139 of file `time_converter_tai_tdb.cc`.

References [jeod::TimeConverter::a_to_b_offset](#), [a_to_b_offset_epoch](#), [nlter](#), [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 ) [override], [virtual]
```

Initialize the converter.

Parameters

in	<i>parent</i>	parent-type
in	<i>child</i>	child-type
in	<i>direction</i>	L-R, or R-L

Implements [jeod::TimeConverter](#).

Definition at line 81 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 & ) [delete]
```

8.8.3.5 set_a_to_b_offset()

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

Definition at line 114 of file time_converter_tai_tdb.cc.

References `jeod::TimeConverter::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 91 of file time_converter_tai_tdb.hh.

8.8.5 Field Documentation

8.8.5.1 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 102 of file time_converter_tai_tdb.hh.

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

8.8.5.2 nIter

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

Counter for number of iterations.

trick_units(-)

Definition at line 122 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a().

8.8.5.3 nSteps

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

Counter for number of steps in iteration.

trick_units(-)

Definition at line 117 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a().

8.8.5.4 prev_tai_seconds

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

TAI seconds from previous loop iteration.

trick_units(s)

Definition at line 107 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a().

8.8.5.5 prev_tdb_seconds

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

TDB seconds from previous loop iteration.

trick_units(s)

Definition at line 112 of file time_converter_tai_tdb.hh.

Referenced by convert_b_to_a().

8.8.5.6 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()`, and `set_a_to_b_offset()`.

8.8.5.7 TAI_to_TT_offset

```
double jeod::TimeConverter_TAI_TDB::TAI_to_TT_offset {32.184} [private]
```

The offset from TAI to TT.

This is needed because first TAI must convert to TT before applying the periodic corrections.`trick_units(s)`

Definition at line 97 of file `time_converter_tai_tdb.hh`.

Referenced by `initialize()`.

8.8.5.8 tdb_ptr

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

Converter parent time, always a [TimeTDB](#) for this converter.

trick_units(-)

Definition at line 132 of file `time_converter_tai_tdb.hh`.

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

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

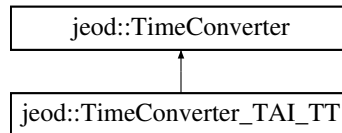
- [time_converter_tai_tdb.hh](#)
- [time_converter_tai_tdb.cc](#)

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](#) () override=default
- [TimeConverter_TAI_TT](#) (const [TimeConverter_TAI_TT](#) &)=delete
- [TimeConverter_TAI_TT](#) & operator= (const [TimeConverter_TAI_TT](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeTAI](#) to [TimeTT](#).
- void [convert_b_to_a](#) () override
Convert from [TimeTT](#) to [TimeTAI](#).

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 86 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 ( )
```

Construct a [TimeConverter_TAI_TT](#).

Definition at line 57 of file `time_converter_tai_tt.cc`.

References [jeod::TimeConverter::a_name](#), [jeod::TimeConverter::ANY_DIRECTION](#), [jeod::TimeConverter::b_name](#), and [jeod::TimeConverter::valid_directions](#).

8.9.2.2 ~TimeConverter_TAI_TT()

```
jeod::TimeConverter_TAI_TT::~~TimeConverter_TAI_TT ( ) [override], [default]
```

8.9.2.3 TimeConverter_TAI_TT() [2/2]

```
jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT (
    const TimeConverter\_TAI\_TT & ) [delete]
```

8.9.3 Member Function Documentation

8.9.3.1 convert_a_to_b()

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

Convert from [TimeTAI](#) to [TimeTT](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 101 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 ( ) [override], [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 113 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 ) [override], [virtual]
```

Initialize the converter.

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 70 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 & ) [delete]
```

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 88 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 93 of file `time_converter_tai_tt.hh`.

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

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 98 of file `time_converter_tai_tt.hh`.

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

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

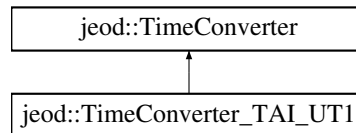
- [time_converter_tai_tt.hh](#)
- [time_converter_tai_tt.cc](#)

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](#) () override
Destroy a [TimeConverter_TAI_UT1](#).
- [TimeConverter_TAI_UT1](#) (const [TimeConverter_TAI_UT1](#) &)=delete
- [TimeConverter_TAI_UT1](#) & operator= (const [TimeConverter_TAI_UT1](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeTAI](#) to [TimeUT1](#).
- void [convert_b_to_a](#) () override
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](#) {-1}
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](#) ()
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](#) () override
Used when time reverses direction.

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 ( )
```

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](#), and [jeod::TimeConverter::valid_directions](#).

8.10.2.2 ~TimeConverter_TAI_UT1()

```
jeod::TimeConverter_TAI_UT1::~~TimeConverter_TAI_UT1 ( ) [override]
```

Destroy a [TimeConverter_TAI_UT1](#).

Definition at line 493 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 & ) [delete]
```

8.10.3 Member Function Documentation

8.10.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_UT1::convert_a_to_b ( ) [override], [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 251 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 ( ) [override], [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 350 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 ) [override], [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 76 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 ( ) [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 124 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 & ) [delete]
```

8.10.3.6 verify_table_lookup_ends()

```
void jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends ( ) [override], [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 448 of file `time_converter_tai_ut1.cc`.

References `jeod::TimeManager::dyn_time`, `index`, `last_index`, `next_when`, `off_table_end`, `prev_when`, `jeod::Time↵
Dyn::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 159 of file time_converter_tai_ut1.hh.

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

8.10.5.2 index

```
int jeod::TimeConverter_TAI_UT1::index {-1}
```

Current location in table.

trick_units(-)

Definition at line 123 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_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 118 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(), 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 154 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), and initialize_tai_to_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 149 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_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 164 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_tai_to_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 96 of file time_converter_tai_ut1.hh.

Referenced by jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize(), and initialize_tai_to_ut1().

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 144 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), and initialize_tai_to_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(), 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 102 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_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 113 of file time_converter_tai_ut1.hh.

Referenced by initialize_tai_to_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 107 of file time_converter_tai_ut1.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_tai_to_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 128 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()`, 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()`, `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 55 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 39 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:

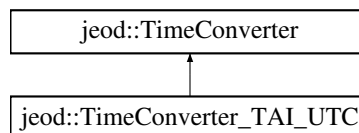
- [tai_to_ut1.hh](#)
- [tai_to_ut1.cc](#)

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 \(\)](#) override
Destroy a `TimeConverter_TAI_UTC`.
- [TimeConverter_TAI_UTC \(const TimeConverter_TAI_UTC &\)=delete](#)
- [TimeConverter_TAI_UTC & operator= \(const TimeConverter_TAI_UTC &\)=delete](#)
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b \(\)](#) override
Convert from `TimeTAI` to `TimeUTC`.
- void [convert_b_to_a \(\)](#) override
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` {-1}
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` ()
The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.
- void `verify_table_lookup_ends` () override
Used when time reverses direction.

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 88 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 ( )
```

Construct a [TimeConverter_TAI_UTC](#).

Definition at line 60 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`, and `jeod::TimeConverter::valid_directions`.

8.12.2.2 ~TimeConverter_TAI_UTC()

```
jeod::TimeConverter_TAI_UTC::~~TimeConverter_TAI_UTC ( ) [override]
```

Destroy a [TimeConverter_TAI_UTC](#).

Definition at line 467 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 & ) [delete]
```

8.12.3 Member Function Documentation

8.12.3.1 convert_a_to_b()

```
void jeod::TimeConverter_TAI_UTC::convert_a_to_b ( ) [override], [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 259 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 ( ) [override], [virtual]
```

Convert from [TimeUTC](#) to [TimeTAI](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 348 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 ) [override], [virtual]
```

Initialize the converter.

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 73 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 ( ) [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 131 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 & ) [delete]
```

8.12.3.6 verify_table_lookup_ends()

```
void jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends ( ) [override], [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 422 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 90 of file time_converter_tai_utc.hh.

8.12.5 Field Documentation

8.12.5.1 index

```
int jeod::TimeConverter_TAI_UTC::index {-1}
```

Current index in the leap tables.

trick_units(-)

Definition at line 122 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize(), initialize_leap_second(), 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 117 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(), 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 112 of file time_converter_tai_utc.hh.

Referenced by initialize_leap_second().

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 138 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), 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 150 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), 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 95 of file time_converter_tai_utc.hh.

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

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 144 of file time_converter_tai_utc.hh.

Referenced by convert_a_to_b(), convert_b_to_a(), initialize_leap_second(), 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 101 of file time_converter_tai_utc.hh.

Referenced by [convert_a_to_b\(\)](#), [convert_b_to_a\(\)](#), [initialize\(\)](#), [initialize_leap_second\(\)](#), 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 106 of file time_converter_tai_utc.hh.

Referenced by [convert_a_to_b\(\)](#), [convert_b_to_a\(\)](#), [initialize\(\)](#), [initialize_leap_second\(\)](#), 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\(\)](#), 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 133 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()`, `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 55 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 38 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:

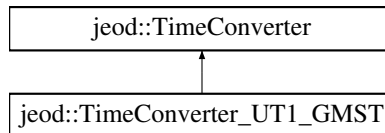
- [tai_to_utc.hh](#)
- [tai_to_utc.cc](#)

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](#) () override=default
- [TimeConverter_UT1_GMST](#) (const [TimeConverter_UT1_GMST](#) &)=delete
- [TimeConverter_UT1_GMST](#) & operator= (const [TimeConverter_UT1_GMST](#) &)=delete
- void [initialize](#) ([JeodBaseTime](#) *parent, [JeodBaseTime](#) *child, const int direction) override
Initialize the converter.
- void [convert_a_to_b](#) () override
Convert from [TimeUT1](#) to [TimeGMST](#).

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 86 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 ( )
```

Construct a [TimeConverter_UT1_GMST](#).

Definition at line 57 of file `time_converter_ut1_gmst.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B`, `jeod::TimeConverter::b_name`, and `jeod::TimeConverter::valid_directions`.

8.14.2.2 ~TimeConverter_UT1_GMST()

```
jeod::TimeConverter_UT1_GMST::~~TimeConverter_UT1_GMST ( ) [override], [default]
```

8.14.2.3 TimeConverter_UT1_GMST() [2/2]

```
jeod::TimeConverter_UT1_GMST::TimeConverter_UT1_GMST (
    const TimeConverter\_UT1\_GMST & ) [delete]
```

8.14.3 Member Function Documentation

8.14.3.1 convert_a_to_b()

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

Convert from [TimeUT1](#) to [TimeGMST](#).

Assumptions and Limitations

- None

Reimplemented from [jeod::TimeConverter](#).

Definition at line 104 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 ) [override], [virtual]
```

Initialize the converter.

Assumptions and Limitations

- None

Parameters

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

Implements [jeod::TimeConverter](#).

Definition at line 73 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 & ) [delete]
```

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 88 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 98 of file `time_converter_ut1_gmst.hh`.

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

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 93 of file `time_converter_ut1_gmst.hh`.

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

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

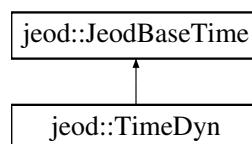
- [time_converter_ut1_gmst.hh](#)
- [time_converter_ut1_gmst.cc](#)

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](#) () override=default
- [TimeDyn](#) (const [TimeDyn](#) &)=delete
- [TimeDyn](#) & [operator=](#) (const [TimeDyn](#) &)=delete
- bool [update_offset](#) ()
Changing time direction and/or scale factor.

Data Fields

- double `scale_factor` {1.0}
Multiplicative difference between sim-time and dyn-time.

Private Member Functions

- void `initialize_initializer_time` (`TimeManagerInit *tm_init`) override
Each time type is initialized from its parent in the initialization tree, except one.
- void `update` () override
TimeDyn updates directly from simtime, and everything else from TimeDyn.

Private Attributes

- double `ref_scale` {1.0}
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 85 of file `time_dyn.hh`.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 TimeDyn() [1/2]

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

Construct a Time_Dyn.

Definition at line 57 of file `time_dyn.cc`.

References `jeod::JeodBaseTime::links`, and `jeod::JeodBaseTime::name`.

8.15.2.2 ~TimeDyn()

```
jeod::TimeDyn::~~TimeDyn ( ) [override], [default]
```

8.15.2.3 TimeDyn() [2/2]

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

8.15.3 Member Function Documentation

8.15.3.1 initialize_initializer_time()

```
void jeod::TimeDyn::initialize_initializer_time (
    TimeManagerInit * time_manager_init ) [override], [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	<i>time_manager_init</i>	TM initializer
----	--------------------------	----------------

Implements [jeod::JeodBaseTime](#).

Definition at line 75 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 & ) [delete]
```

8.15.3.3 update()

```
void jeod::TimeDyn::update ( ) [override], [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-progressing sims, and negative for reverse-progressing sims.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 100 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 ( )
```

Changing time direction and/or scale factor.

Returns

Void

Definition at line 109 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 87 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 107 of file time_dyn.hh.

Referenced by update(), and update_offset().

8.15.5.2 ref_scale

```
double jeod::TimeDyn::ref_scale {1.0} [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 101 of file time_dyn.hh.

Referenced by update(), and update_offset().

8.15.5.3 scale_factor

```
double jeod::TimeDyn::scale_factor {1.0}
```

Multiplicative difference between sim-time and dyn-time.

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

Definition at line 93 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(), update_offset(), jeod::TimeConverter_TAI_UTC::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 77 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

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

Definition at line 84 of file `time_enum.hh`.

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

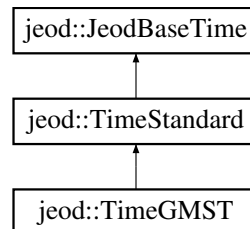
- [time_enum.hh](#)

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](#) () override=default
- [TimeGMST](#) (const [TimeGMST](#) &)=delete
- [TimeGMST](#) & [operator=](#) (const [TimeGMST](#) &)=delete
- void [set_time_by_trunc_julian](#) (const double nonsense)
TJT does not function in GMST.

Private Member Functions

- void [calculate_calendar_values](#) () override
Protection against inheriting nonsense function.
- void [set_epoch](#) () override
No action.

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 ( )
```

Construct a Time_GMST.

Definition at line 53 of file time_gmst.cc.

References [jeod::JeodBaseTime::name](#).

8.17.2.2 ~TimeGMST()

```
jeod::TimeGMST::~~TimeGMST ( ) [override], [default]
```

8.17.2.3 TimeGMST() [2/2]

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

8.17.3 Member Function Documentation

8.17.3.1 calculate_calendar_values()

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

Protection against inheriting nonsense function.

Assumptions and Limitations

- GMST does not have a conventional calendar

Reimplemented from [jeod::TimeStandard](#).

Definition at line 64 of file time_gmst.cc.

References [jeod::TimeMessages::invalid_data_error](#).

8.17.3.2 operator=()

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

8.17.3.3 set_epoch()

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

No action.

Function is required to make this class instantiable.

Implements [jeod::TimeStandard](#).

Definition at line 100 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	<i>nonsense</i>	Any old invalid value
----	-----------------	-----------------------

Definition at line 81 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 85 of file time_gmst.hh.

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

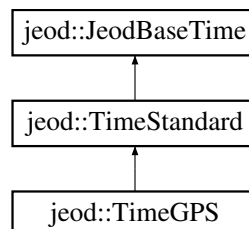
- [time_gmst.hh](#)
- [time_gmst.cc](#)

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](#) () override=default
- [TimeGPS](#) (const [TimeGPS](#) &)=delete
- [TimeGPS](#) & [operator=](#) (const [TimeGPS](#) &)=delete
- void [set_time_by_seconds](#) (const double new_seconds) override
Given a value of seconds, propagate to other reps.
- void [set_time_by_days](#) (const double new_seconds) override
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` () override
Protection against inheriting nonsense function.
- void `convert_from_calendar` () override
Protection against inheriting nonsense function.
- void `set_epoch` () override
Sets the epoch for GPS time.

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 ( )
```

Construct a Time_GPS.

Definition at line 54 of file time_gps.cc.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

8.18.2.2 ~TimeGPS()

```
jeod::TimeGPS::~~TimeGPS ( ) [override], [default]
```

8.18.2.3 TimeGPS() [2/2]

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

8.18.3 Member Function Documentation

8.18.3.1 calculate_calendar_values()

```
void jeod::TimeGPS::calculate_calendar_values ( ) [override], [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`.

8.18.3.2 convert_from_calendar()

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

Protection against inheriting nonsense function.

Assumptions and Limitations

- GPS does not have a conventional calendar

Reimplemented from [jeod::TimeStandard](#).

Definition at line 76 of file `time_gps.cc`.

References `jeod::TimeMessages::invalid_data_error`.

8.18.3.3 operator=()

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

8.18.3.4 set_epoch()

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

Sets the epoch for GPS time.

Implements [jeod::TimeStandard](#).

Definition at line 63 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 ) [override], [virtual]
```

Given a value of days, propagate to other values.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 136 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 ) [override], [virtual]
```

Given a value of seconds, propagate to other reps.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 109 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	<i>new</i> ↔ <i>_tjt</i>	new value for Truncated Julian Time Units: day
----	-----------------------------	---

Definition at line 148 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 85 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().

8.18.5.2 rollover_count

```
int jeod::TimeGPS::rollover_count {}
```

Number of rollovers (1024 week blocks) since epoch.

trick_units(-)

Definition at line 105 of file time_gps.hh.

Referenced by set_time_by_seconds().

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 115 of file time_gps.hh.

Referenced by set_time_by_seconds().

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 90 of file time_gps.hh.

Referenced by set_time_by_seconds().

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 95 of file time_gps.hh.

Referenced by set_time_by_seconds().

8.18.5.6 week

```
int jeod::TimeGPS::week {}
```

Number of weeks in current 1024-week block.

trick_units(-)

Definition at line 110 of file `time_gps.hh`.

Referenced by `set_time_by_seconds()`.

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 120 of file `time_gps.hh`.

Referenced by `set_time_by_seconds()`.

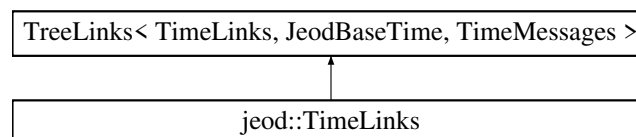
The documentation for this class was generated from the following files:

- [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](#) () override=default
- [TimeLinks](#) (const [TimeLinks](#) &)=delete
- void [operator=](#) (const [TimeLinks](#) &)=delete

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 77 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], [explicit]
```

Definition at line 81 of file `time_links.hh`.

8.19.2.2 TimeLinks() [2/3]

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

8.19.2.3 ~TimeLinks()

```
jeod::TimeLinks::~TimeLinks ( ) [override], [default]
```

8.19.2.4 TimeLinks() [3/3]

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

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 79 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 98 of file `time_links.hh`.

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

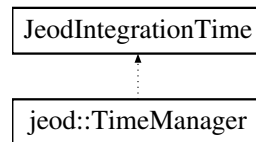
- [time_links.hh](#)

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](#) () override
Destroy a [TimeManager](#).
- [TimeManager](#) (const [TimeManager](#) &)=delete
- [TimeManager](#) & operator= (const [TimeManager](#) &)=delete
- 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](#) ()
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](#) ()
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, const std::string &name_a="", const std::string &name_b="")
Registers the time converters with the Time Manager.
- [JeodIntegrationTime](#) & [get_jeod_integration_time](#) ()
Expose the private inheritance from [JeodIntegrationTime](#).
- double [get_time_scale_factor](#) () const override
Returns the scale factor from sim time to dynamic time.
- double [get_timestamp_time](#) () const override
Returns the time used to timestamp objects, currently dynamic time seconds.

Data Fields

- double [simtime](#) {-1.0}
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

- void [update_time](#) (double time) override
Update each of the representations of time, calling the update functions for each such representation in dependency order.

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 93 of file time_manager.hh.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 TimeManager() [1/2]

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

Construct a [TimeManager](#).

Definition at line 64 of file time_manager.cc.

8.20.2.2 ~TimeManager()

```
jeod::TimeManager::~TimeManager ( ) [override]
```

Destroy a [TimeManager](#).

Definition at line 473 of file `time_manager.cc`.

References `converter_vector`, and `time_vector`.

8.20.2.3 TimeManager() [2/2]

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

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	<i>index</i>	Index of object
----	--------------	-----------------

Definition at line 79 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 ( )
```

Expose the private inheritance from JeodIntegrationTime.

Definition at line 97 of file time_manager.cc.

8.20.3.3 `get_time_change_flag()`

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

Returns the boolean value time_change_flag.

Returns

time_change_flag

Definition at line 106 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	<i>name</i>	Name of time object
----	-------------	---------------------

Definition at line 136 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	<i>index</i>	Name of time object
----	--------------	---------------------

Definition at line 149 of file time_manager.cc.

References time_vector.

8.20.3.6 `get_time_scale_factor()`

```
double jeod::TimeManager::get_time_scale_factor ( ) const [override]
```

Returns the scale factor from sim time to dynamic time.

Returns

dyn_time.scale_factor

Definition at line 115 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 ( ) const [override]
```

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

Returns

dyn_time.seconds

Definition at line 125 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	<i>time_manager_init</i>	Initialization parameters
----	--------------------------	---------------------------

Definition at line 61 of file `time_manager__initialize.cc`.

8.20.3.9 operator=()

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

8.20.3.10 register_converter()

```
void jeod::TimeManager::register_converter (
    TimeConverter & conv_ref,
    const std::string & name_a = "",
    const 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	<i>conv_ref</i>	ref. to converter being registered
in	<i>name_a</i>	name of type-a in the converter
in	<i>name_b</i>	name of type-b in the converter

Definition at line 229 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

<i>in, out</i>	<i>time_ref</i>	reference to time-type being registered
----------------	-----------------	---

Definition at line 170 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 `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.

Assumptions and Limitations

- None

Parameters

<i>in, out</i>	<i>time_ref</i>	reference to time-type being registered
<i>in</i>	<i>name</i>	name of the instance being registered.

Definition at line 200 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	<i>name</i>	name of time-type
----	-------------	-------------------

Definition at line 328 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 ( )
```

Tests for the existence in the registry of time types that inherit from [TimeStandard](#).

Assumptions and Limitations

- None

Returns

true/false

Definition at line 304 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	<i>current_simtime</i>	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 397 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()`.

8.20.3.16 `update_time()`

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

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	<i>current_simtime</i>	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 436 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 ( )
```

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 462 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 95 of file time_manager.hh.

8.20.4.3 TimeManagerInit

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

Definition at line 95 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 129 of file time_manager.hh.

Referenced by get_converter_ptr(), jeod::TimeManagerInit::populate_converter_registry(), register_converter(), jeod::TimeManagerInit::verify_converter_setup(), 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 108 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()`, `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 113 of file `time_manager.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_init_tree()`, `jeod::TimeManagerInit::create_update_tree()`, `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::TimeManagerInit::verify_times_setup()`.

8.20.5.4 simtime

```
double jeod::TimeManager::simtime {-1.0}
```

Simulation time (`sys.exec.out.time`).

```
trick_units(-)
```

Definition at line 102 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 119 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 124 of file time_manager.hh.

Referenced by jeod::TimeManagerInit::create_init_tree(), jeod::TimeManagerInit::create_update_tree(), get_time_ptr(), 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 ()`=default
- `~TimeManagerInit ()`
Destroy a `TimeManagerInit`.
- `TimeManagerInit (const TimeManagerInit &)=delete`
- `TimeManagerInit & operator= (const TimeManagerInit &)=delete`
- `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 {TimeEnum::undefined}`
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 {-1}`
Index-value of the initializer.
- `int dyn_time_index {-1}`
Index-value of the type dyn-time.
- `int num_added_pass {-1}`
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](#) ()
The [TimeManagerInit](#) determines initialization and update paths for conversions between time-types.
- void [verify_times_setup](#) ()
A number of checks that the setup is self-consistent.
- void [populate_converter_registry](#) ()
The converter registry accounts for all of the converter functions that provide conversions between time types.
- void [verify_converter_setup](#) ()
To verify that there are no incompatibilities between specific converters.
- void [initialize_time_types](#) ()
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](#) ()
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](#) ()
(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`.

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 ( ) [default]
```

8.21.2.2 ~TimeManagerInit()

```
jeod::TimeManagerInit::~~TimeManagerInit ( )
```

Destroy a [TimeManagerInit](#).

Definition at line 752 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 & ) [delete]
```

8.21.3 Member Function Documentation

8.21.3.1 create_init_tree()

```
void jeod::TimeManagerInit::create_init_tree ( ) [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 392 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, and jeod::TimeManager::time_vector.

Referenced by initialize_manager().

8.21.3.2 create_update_tree()

```
void jeod::TimeManagerInit::create_update_tree ( ) [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 530 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	<i>index</i>	Index of object
----	--------------	-----------------

Definition at line 667 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	<i>index</i>	Index of object
----	--------------	-----------------

Definition at line 692 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	<i>index</i> ↔ _in	Index of object
----	-----------------------	-----------------

Definition at line 642 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	<i>index</i>	Index of object
----	--------------	-----------------

Definition at line 712 of file `time_manager_init.cc`.

References `status`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::Jeod↔BaseTime::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	<i>index_slave</i>	Index of object
in	<i>index_master</i>	Index of object

Definition at line 743 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::Jeod↔BaseTime::add_type_update()`.

8.21.3.8 initialize()

```
void jeod::TimeManagerInit::initialize ( ) [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 105 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 66 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()`.

8.21.3.10 initialize_time_types()

```
void jeod::TimeManagerInit::initialize_time_types ( ) [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 504 of file time_manager_init.cc.

References `jeod::TimeManager::get_time_ptr()`, `jeod::JeodBaseTime::initialize_from_parent()`, `jeod::JeodBaseTime::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 & ) [delete]
```

8.21.3.12 organize_update_list()

```
void jeod::TimeManagerInit::organize_update_list ( )
```

Reorganizes the update list according to initialization status.

Definition at line 598 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 ( ) [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 244 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	<i>index</i>	Index of object
in	<i>new_status</i>	New status value

Definition at line 732 of file time_manager_init.cc.

References status.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::Jeod↔BaseTime::add_type_update()`.

8.21.3.15 `verify_converter_setup()`

```
void jeod::TimeManagerInit::verify_converter_setup ( ) [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 340 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `jeod::TimeManager::converter_vector`, `jeod::TimeMessages::invalid_setup_↔error`, `jeod::TimeManager::num_types`, `jeod::TimeManager::time_lookup()`, and `time_manager`.

Referenced by `initialize_manager()`.

8.21.3.16 `verify_times_setup()`

```
void jeod::TimeManagerInit::verify_times_setup ( ) [private]
```

A number of checks that the setup is self-consistent.

Assumptions and Limitations

- None

Definition at line 164 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 134 of file time_manager_init.hh.

Referenced by get_conv_ptr_index(), initialize(), populate_converter_registry(), verify_converter_setup(), and ~TimeManagerInit().

8.21.5.2 dyn_time_index

```
int jeod::TimeManagerInit::dyn_time_index {-1} [protected]
```

Index-value of the type dyn-time.

trick_units(-)

Definition at line 118 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), and initialize().

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 140 of file time_manager_init.hh.

Referenced by get_conv_dir_init(), initialize(), populate_converter_registry(), and ~TimeManagerInit().

8.21.5.4 initializer

```
std::string jeod::TimeManagerInit::initializer {""}
```

Name of the time-type used for initialization.

trick_units(-)

Definition at line 108 of file time_manager_init.hh.

Referenced by initialize(), and verify_times_setup().

8.21.5.5 initializer_index

```
int jeod::TimeManagerInit::initializer_index {-1} [protected]
```

Index-value of the initializer.

trick_units(-)

Definition at line 113 of file time_manager_init.hh.

Referenced by create_init_tree(), initialize(), initialize_time_types(), and verify_times_setup().

8.21.5.6 num_added_pass

```
int jeod::TimeManagerInit::num_added_pass {-1} [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(), and increment_status().

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 93 of file time_manager_init.hh.

Referenced by jeod::JeodBaseTime::add_type_update(), create_init_tree(), and create_update_tree().

8.21.5.8 sim_start_format

```
TimeEnum::TimeFormat jeod::TimeManagerInit::sim_start_format {TimeEnum::undefined}
```

Calendar, truncated_julian, etc.

trick_units(—)

Definition at line 98 of file time_manager_init.hh.

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

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 161 of file time_manager_init.hh.

Referenced by create_init_tree(), create_update_tree(), get_status(), increment_status(), initialize(), organize_update_list(), set_status(), 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 103 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 146 of file time_manager_init.hh.

Referenced by get_conv_dir_upd(), initialize(), populate_converter_registry(), and ~TimeManagerInit().

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>
```

Public Member Functions

- [TimeMessages](#) ()=delete
- [TimeMessages](#) (const [TimeMessages](#) &)=delete
- [TimeMessages](#) & operator= (const [TimeMessages](#) &)=delete

Static Public Attributes

- static const char * [initialization_error](#) = "environment/time/" "initialization_error"
Error issued when initialization fails due to some non-obvious cause.
- static const char * [memory_error](#) = "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.
- static const char * [invalid_setup_error](#) = "environment/time/" "invalid_setup_error"
Error issued when user tries to use something that doesn't exist in the model.
- static const char * [invalid_data_error](#) = "environment/time/" "invalid_data_error"
Error issued when a variable is found with an illegal value.
- static const char * [invalid_node](#) = "environment/time/" "invalid_node"
Issued when a [TimeLinks](#) node is improperly linked.
- static const char * [incomplete_setup_error](#) = "environment/time/" "incomplete_setup_error"
Error issued when user tries to use something that doesn't exist in the simulation. This is usually a user error, brought about by not having registered something that is later needed (e.g.
- static const char * [redundancy_error](#) = "environment/time/" "redundancy_error"
Error issued when some value is multiply defined, and the code cannot determine which value to use.
- static const char * [duplicate_methods](#) = "environment/time/" "duplicate_methods"
Informational only.
- static const char * [extension_error](#) = "environment/time/" "extension_error"
Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__TimeMessages](#) ()

8.22.1 Detailed Description

Specify the message IDs used in the Time model.

Definition at line 82 of file `time_messages.hh`.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 TimeMessages() [1/2]

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

8.22.2.2 TimeMessages() [2/2]

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

8.22.3 Member Function Documentation

8.22.3.1 operator=()

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

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 84 of file time_messages.hh.

8.22.5 Field Documentation

8.22.5.1 duplicate_methods

```
const char * jeod::TimeMessages::duplicate_methods = "environment/time/" "duplicate_methods"
[static]
```

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 140 of file time_messages.hh.

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

8.22.5.2 extension_error

```
const char * jeod::TimeMessages::extension_error = "environment/time/" "extension_error" [static]
```

Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

trick_units(-)

Definition at line 146 of file time_messages.hh.

8.22.5.3 incomplete_setup_error

```
const char * jeod::TimeMessages::incomplete_setup_error = "environment/time/" "incomplete_↵
setup_error" [static]
```

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 126 of file time_messages.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_update↵
_tree(), jeod::TimeConverter_Dyn_UDE::initialize(), jeod::TimeStandard::initialize_from_parent(), jeod::TimeUD↵
E::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

```
const char * jeod::TimeMessages::initialization_error = "environment/time/" "initialization_↵
error" [static]
```

Error issued when initialization fails due to some non-obvious cause.

This error is likely due to an algorithm flaw.`trick_units(-)`

Definition at line 91 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::TimeU↵
DE::initialize_from_parent()`, and `jeod::TimeConverter::verify_setup()`.

8.22.5.5 invalid_data_error

```
const char * jeod::TimeMessages::invalid_data_error = "environment/time/" "invalid_data_error"
[static]
```

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 111 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::TimeG↵
PS::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

```
const char * jeod::TimeMessages::invalid_node = "environment/time/" "invalid_node" [static]
```

Issued when a [TimeLinks](#) node is improperly linked.

`trick_units(-)`

Definition at line 116 of file `time_messages.hh`.

Referenced by `jeod::JeodBaseTime::add_type_update()`.

8.22.5.7 invalid_setup_error

```
const char * jeod::TimeMessages::invalid_setup_error = "environment/time/" "invalid_setup_↵
error" [static]
```

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 incompletetrack_units(–)

Definition at line 104 of file time_messages.hh.

Referenced by jeod::TimeStandard::add_type_initialize(), jeod::TimeUDE::add_type_initialize(), jeod::JeodBase↵
Time::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_U↵
DE::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

```
const char * jeod::TimeMessages::memory_error = "environment/time/" "memory_error" [static]
```

Error issued when system fails because something is not where it should be, or has a value other than its assumed value.

track_units(–)

Definition at line 97 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

```
const char * jeod::TimeMessages::redundancy_error = "environment/time/" "redundancy_error"
[static]
```

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.track_units(–)

Definition at line 133 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:

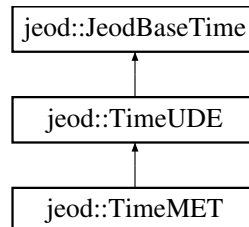
- [time_messages.hh](#)
- [time_messages.cc](#)

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](#) () override=default
 - [TimeMET](#) (const [TimeMET](#) &)=delete
 - [TimeMET](#) & [operator=](#) (const [TimeMET](#) &)=delete
 - void [update](#) () override
- Updates to current time.*

Data Fields

- bool [hold](#) {}
- Flags whether to hold time at current value.*

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 ( )
```

Definition at line 66 of file time_met.cc.

References [jeod::JeodBaseTime::name](#).

8.23.2.2 ~TimeMET()

```
jeod::TimeMET::~~TimeMET ( ) [override], [default]
```

8.23.2.3 TimeMET() [2/2]

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

8.23.3 Member Function Documentation

8.23.3.1 operator=()

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

8.23.3.2 update()

```
void jeod::TimeMET::update ( ) [override], [virtual]
```

Updates to current time.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 74 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 86 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 91 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 97 of file time_met.hh.

Referenced by update().

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

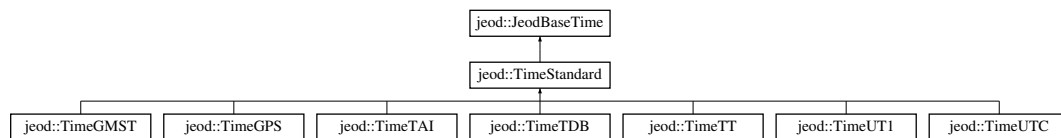
- [time_met.hh](#)
- [time_met.cc](#)

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()`=default
- `~TimeStandard()` override=default
- `TimeStandard(const TimeStandard &)=delete`
- `TimeStandard & operator=(const TimeStandard &)=delete`
- `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) override`
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) override`
Recursively adds elements to the initialization tree.
- `void initialize_from_parent(TimeManagerInit *tm_init) override`
Initialize a time type from its parent on the initialization tree.
- `void set_time_by_seconds(const double new_seconds) override`
Given a value of seconds, propagate to days and trunc_julian_time.
- `void set_time_by_days(const double new_days) override`
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()`
Returns the full Julian date at epoch, rather than the Truncated Julian Time.
- `double seconds_of_year()`
Generate the number of seconds elapsed this year.

Data Fields

- `double last_calendar_update {-100000.0}`
The simtime when the calendar update was last run.
- `int prev_julian_day {-1000000000}`
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 {-1000000}`
The year in which the last seconds_of_year calculation was made.
- `bool send_warning_pre_1968 {true}`

This flag can be turned off by developers wanting to avoid warnings about a simulation being initialized pre-1968.

- const double [tjt_mjt_offset](#) {40000.0}

Difference between Truncated Julian and Modified Julian.

- const double [tjt_jd_offset](#) {2440000.5}

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](#) ()

Calculate Truncated Julian date/time from Gregorian calendar date and 24-hour clock representation.

- virtual void [calculate_calendar_values](#) ()

Calculate Gregorian calendar date and 24-hour clock representation from Truncated Julian date/time.

- virtual void [set_epoch](#) ()=0

Set the epoch time.

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 88 of file `time_standard.hh`.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 TimeStandard() [1/2]

```
jeod::TimeStandard::TimeStandard ( ) [default]
```

8.24.2.2 ~TimeStandard()

```
jeod::TimeStandard::~~TimeStandard ( ) [override], [default]
```

8.24.2.3 TimeStandard() [2/2]

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

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 ) [override], [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	<i>seeking_status</i>	status-value for auto-seek
in	<i>time_manager_init</i>	The TM initializer.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 128 of file `time_standard.cc`.

References `jeod::JeodBaseTime::add_type_initialize()`, `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::TimeManager::time_lookup()`, and `jeod::JeodBaseTime::time_manager`.

8.24.3.2 calculate_calendar_values()

```
void jeod::TimeStandard::calculate_calendar_values ( ) [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 255 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	<i>simtime</i>	Simulation elapsed time, on the simulation clock Units: s
----	----------------	--

Definition at line 343 of file `time_standard.cc`.

References `calculate_calendar_values()`, `last_calendar_update`, `jeod::TimeManager::simtime`, `jeod::JeodBaseTime::time_manager`, and `jeod::TimeManager::update()`.

8.24.3.4 `convert_from_calendar()`

```
void jeod::TimeStandard::convert_from_calendar ( ) [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 370 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 ) [override], [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	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 593 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 ) [override], [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	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 422 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 ( )
```

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 105 of file time_standard.cc.

References [tjt_at_epoch](#), and [tjt_jd_offset](#).

8.24.3.8 operator=()

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

8.24.3.9 seconds_of_year()

```
double jeod::TimeStandard::seconds_of_year ( )
```

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 679 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 ( ) [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 ) [override], [virtual]
```

Given a value of days, propagate to seconds and [trunc_julian_time](#).

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 77 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 ) [override], [virtual]
```

Given a value of seconds, propagate to days and `trunc_julian_time`.

Assumptions and Limitations

- 86400 seconds = 1 day

Parameters

in	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 63 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	<i>new</i> ↔ <i>_tjt</i>	new value for Truncated Julian Time Units: day
----	-----------------------------	---

Definition at line 92 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 90 of file time_standard.hh.

8.24.4.3 TimeUDE

```
friend class TimeUDE [friend]
```

Definition at line 90 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 157 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 162 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 167 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 182 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 172 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 177 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 147 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 {-100000.0}
```

The simtime when the calendar update was last run.

trick_units(-)

Definition at line 97 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 {-1000000000}
```

Used for determining whether to update the date in the calendar function.

trick_units(day)

Definition at line 103 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 110 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 {true}
```

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 124 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 152 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 {2440000.5}
```

Difference between Julian and Truncated Julian.

`trick_units(day)`

Definition at line 134 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 {40000.0}
```

Difference between Truncated Julian and Modified Julian.

`trick_units(day)`

Definition at line 129 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 139 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 {-1000000}
```

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 117 of file time_standard.hh.

Referenced by seconds_of_year().

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

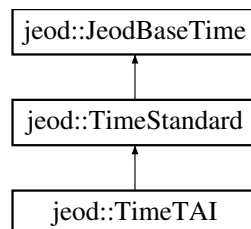
- [time_standard.hh](#)
- [time_standard.cc](#)

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](#) () override=default
- [TimeTAI](#) (const [TimeTAI](#) &)=delete
- [TimeTAI](#) & [operator=](#) (const [TimeTAI](#) &)=delete

Private Member Functions

- void [set_epoch](#) () override
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 ( )
```

Construct a `Time_TAI`.

Definition at line 49 of file `time_tai.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

8.25.2.2 ~TimeTAI()

```
jeod::TimeTAI::~~TimeTAI ( ) [override], [default]
```

8.25.2.3 TimeTAI() [2/2]

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

8.25.3 Member Function Documentation

8.25.3.1 operator=()

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

8.25.3.2 set_epoch()

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

Sets the epoch for TAI time.

Implements [jeod::TimeStandard](#).

Definition at line 58 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:

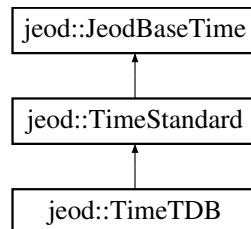
- [time_tai.hh](#)
- [time_tai.cc](#)

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](#) () override=default
- [TimeTDB](#) (const [TimeTDB](#) &)=delete
- [TimeTDB](#) & [operator=](#) (const [TimeTDB](#) &)=delete

Private Member Functions

- void [set_epoch](#) () override
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 ( )
```

Construct a Time_TDB.

Definition at line 49 of file time_tdb.cc.

References [jeod::JeodBaseTime::name](#), and [set_epoch\(\)](#).

8.26.2.2 ~TimeTDB()

```
jeod::TimeTDB::~~TimeTDB ( ) [override], [default]
```

8.26.2.3 TimeTDB() [2/2]

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

8.26.3 Member Function Documentation**8.26.3.1 operator=()**

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

8.26.3.2 set_epoch()

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

Sets the epoch for TDB time.

Implements [jeod::TimeStandard](#).

Definition at line 58 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 85 of file time_tdb.hh.

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

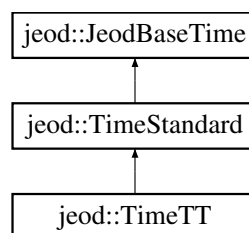
- [time_tdb.hh](#)
- [time_tdb.cc](#)

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](#) () override=default
- [TimeTT](#) (const [TimeTT](#) &)=delete
- [TimeTT](#) & [operator=](#) (const [TimeTT](#) &)=delete

Private Member Functions

- void [set_epoch](#) () override
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 ( )
```

Construct a `Time_TT`.

Definition at line 49 of file `time_tt.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

8.27.2.2 ~TimeTT()

```
jeod::TimeTT::~~TimeTT ( ) [override], [default]
```

8.27.2.3 TimeTT() [2/2]

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

8.27.3 Member Function Documentation

8.27.3.1 operator=()

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

8.27.3.2 set_epoch()

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

Sets the epoch for TT time.

Implements [jeod::TimeStandard](#).

Definition at line 58 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 85 of file `time_tt.hh`.

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

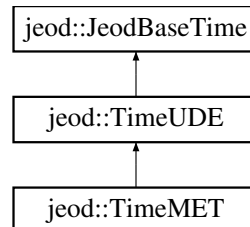
- [time_tt.hh](#)
- [time_tt.cc](#)

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](#) ()=default
- [~TimeUDE](#) () override=default
- [TimeUDE](#) (const [TimeUDE](#) &)=delete
- [TimeUDE](#) & [operator=](#) (const [TimeUDE](#) &)=delete
- void [initialize_initializer_time](#) ([TimeManagerInit](#) *tm_init) override
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) override
Adds a UDE type to the initialization tree when it is appropriate to do so.
- void [initialize_from_parent](#) ([TimeManagerInit](#) *tm_init) override
Initializes this time-type.
- void [set_time_by_clock](#) ()
sets the decimal representation of time by the clock
- void [set_time_by_seconds](#) (const double new_seconds) override
Given a seconds value, sets days and clock values.
- void [set_time_by_days](#) (const double new_days) override
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` {`TimeEnum::undefined`}
Format for expressing the epoch of this type (calendar, julian, etc)
- `TimeEnum::TimeFormat initial_value_format` {`TimeEnum::undefined`}
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` () override
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` ()
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` ()
Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.
- void `verify_init` ()
Verifies that any assignment to initialize_from is flagged as inappropriate, and tests for the presence of initializing data.
- void `verify_update` ()
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.

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 ( ) [default]
```

8.28.2.2 ~TimeUDE()

```
jeod::TimeUDE::~~TimeUDE ( ) [override], [default]
```

8.28.2.3 TimeUDE() [2/2]

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

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 ) [override], [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	<i>seeking_status</i>	An indicator of relative level of progression in the tree.
in	<i>time_manager_init</i>	The TM initializer.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 84 of file `time_ude.cc`.

References [jeod::JeodBaseTime::add_type_initialize\(\)](#), [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 ( ) [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 1353 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	<i>epoch_ptr</i>	pointer to the epoch time-type
in	<i>update_from_ptr</i>	pointer to the time-type from which this time-type will be updated.
in	<i>time_manager_init</i>	The TM initializer.

Definition at line 216 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 ) [override], [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	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 278 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 ) [override], [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	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 488 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 ( ) [override], [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 65 of file time_ude.cc.

8.28.3.7 operator=()

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

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	<i>epoch_ptr</i>	pointer to the epoch time-type
----	------------------	--------------------------------

Definition at line 759 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	<i>simtime</i>	Used to verify that this is at initialization
in	<i>epoch</i>	the value to be used.

Definition at line 1328 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	<i>epoch_ptr</i>	pointer to the epoch time-type
----	------------------	--------------------------------

Definition at line 888 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	<i>epoch_ptr</i>	pointer to the epoch time-type
----	------------------	--------------------------------

Definition at line 722 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	<i>epoch_ptr</i>	pointer to the epoch time-type
----	------------------	--------------------------------

Definition at line 1046 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::Time_↵
Enum::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 ( ) [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 1169 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 ( )
```

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 1313 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 ) [override], [virtual]
```

Given a seconds value, sets days and clock values.

Parameters

in	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 1291 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 ) [override], [virtual]
```

Given a seconds value, sets days and clock values.

Parameters

in	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 1301 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 ( ) [protected]
```

Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.

Definition at line 1384 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 ( ) [protected]
```

Verifies that any assignment to `initialize_from` is flagged as inappropriate, and tests for the presence of initializing data.

Definition at line 1491 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 ( ) [protected]
```

Ensures that the time-type identified as "update_from" is legitimate.

Definition at line 1528 of file `time_ude.cc`.

References `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeMessages::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 125 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 130 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 135 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 140 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 177 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 105 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 161 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 {TimeEnum::undefined}
```

Format for expressing the epoch of this type (calendar, julian, etc)

trick_units(-)

Definition at line 150 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 202 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 167 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 115 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 100 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 120 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 187 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 192 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 182 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 95 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 {TimeEnum::undefined}
```

Format for expressing the initial value of this type (calendar, julian, etc)

trick_units(—)

Definition at line 156 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 172 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 145 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 197 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:

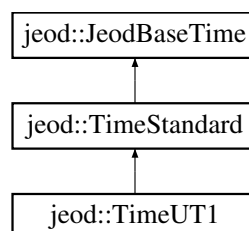
- [time_ude.hh](#)
- [time_ude.cc](#)

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](#) () override=default
- [TimeUT1](#) (const [TimeUT1](#) &)=delete
- [TimeUT1](#) & [operator=](#) (const [TimeUT1](#) &)=delete
- double [get_days](#) ()
Accesses days.

Data Fields

- bool [true_ut1](#) {true}
"False" for comparison with older versions of JEOD

Private Member Functions

- void [set_epoch](#) () override
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 ( )
```

Construct a Time_UT1.

Definition at line 49 of file `time_ut1.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

8.29.2.2 ~TimeUT1()

```
jeod::TimeUT1::~~TimeUT1 ( ) [override], [default]
```

8.29.2.3 TimeUT1() [2/2]

```
jeod::TimeUT1::TimeUT1 (
    const TimeUT1 & ) [delete]
```

8.29.3 Member Function Documentation

8.29.3.1 get_days()

```
double jeod::TimeUT1::get_days ( )
```

Accesses days.

Returns

days value
Units: d

Definition at line 67 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 & ) [delete]
```

8.29.3.3 set_epoch()

```
void jeod::TimeUT1::set_epoch ( ) [override], [private], [virtual]
```

Sets the epoch for UT1 time.

Implements [jeod::TimeStandard](#).

Definition at line 58 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 {true}
```

"False" for comparison with older versions of JEOD

trick_units(-)

Definition at line 89 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()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

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

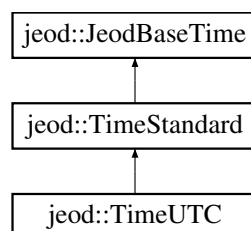
- [time_ut1.hh](#)
- [time_ut1.cc](#)

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](#) () override=default
- [TimeUTC](#) (const [TimeUTC](#) &)=delete
- [TimeUTC](#) & [operator=](#) (const [TimeUTC](#) &)=delete

Data Fields

- bool [true_utc](#) {true}
"False" for comparison with older versions of JEOD

Private Member Functions

- void [set_epoch](#) () override
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 ( )
```

Construct a Time_UTC.

Definition at line 49 of file `time_utc.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

8.30.2.2 ~TimeUTC()

```
jeod::TimeUTC::~~TimeUTC ( ) [override], [default]
```

8.30.2.3 TimeUTC() [2/2]

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

8.30.3 Member Function Documentation

8.30.3.1 operator=()

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

8.30.3.2 set_epoch()

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

Sets the epoch for UTC time.

Implements [jeod::TimeStandard](#).

Definition at line 58 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 86 of file time_utc.hh.

8.30.5 Field Documentation

8.30.5.1 true_utc

```
bool jeod::TimeUTC::true_utc {true}
```

"False" for comparison with older versions of JEOD

trick_units(-)

Definition at line 91 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(), 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 26 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 24 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 <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.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 <cstdlib>
#include <stdlib.h>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time.hh"
#include "../include/time_converter.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 <cmath>
#include <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_dyn_tai.hh"
#include "../include/time_dyn.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.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 <cmath>
#include <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_dyn_tdb.hh"
#include "../include/time_dyn.hh"
#include "../include/time_messages.hh"
#include "../include/time_tdb.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_dyn_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_messages.hh"
#include "../include/time_ude.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 <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_std_ude.hh"
#include "../include/time_messages.hh"
#include "../include/time_standard.hh"
#include "../include/time_ude.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_tai_gps.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_tai_tdb.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.hh"
#include "../include/time_tdb.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_tai_tt.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.hh"
#include "../include/time_tt.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_tai_ut1.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.hh"
#include "../include/time_ut1.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/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
#include "../include/time_tai.hh"
#include "../include/time_utc.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter_ut1_gmst.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
#include "../include/time_ut1.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 <cstdint>
#include "utils/math/include/numerical.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_dyn.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
#include "../include/time_standard.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 <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.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/ref_frames/include/tree_links.hh"
#include "utils/sim_interface/include/jeod_class.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 <cstdlib>
#include <cstring>
#include "utils/math/include/numerical.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.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"
#include "../include/time_standard.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/integration/include/jeod_integration_time.hh"
#include "utils/sim_interface/include/jeod_class.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/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.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 <algorithm>
#include <cstdlib>
#include <typeinfo>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time_converter_tai_utl.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
#include "../include/time_ude.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.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/math/include/numerical.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
#include "../include/time_standard.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.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/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time_converter.hh"
#include "../include/time_dyn.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
#include "../include/time_standard.hh"
#include "../include/time_ude.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 "time.hh"
#include "time_enum.hh"
#include "utils/sim_interface/include/jeod_class.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 <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.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 <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.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

- ~JeodBaseTime
 - jeod::JeodBaseTime, [25](#)
- ~TimeConverter
 - jeod::TimeConverter, [41](#)
- ~TimeConverter_Dyn_TAI
 - jeod::TimeConverter_Dyn_TAI, [49](#)
- ~TimeConverter_Dyn_TDB
 - jeod::TimeConverter_Dyn_TDB, [53](#)
- ~TimeConverter_Dyn_UDE
 - jeod::TimeConverter_Dyn_UDE, [56](#)
- ~TimeConverter_STD_UDE
 - jeod::TimeConverter_STD_UDE, [61](#)
- ~TimeConverter_TAI_GPS
 - jeod::TimeConverter_TAI_GPS, [65](#)
- ~TimeConverter_TAI_TT
 - jeod::TimeConverter_TAI_TT, [76](#)
- ~TimeConverter_TAI_TDB
 - jeod::TimeConverter_TAI_TDB, [70](#)
- ~TimeConverter_TAI_UT1
 - jeod::TimeConverter_TAI_UT1, [80](#)
- ~TimeConverter_TAI_UTC
 - jeod::TimeConverter_TAI_UTC, [90](#)
- ~TimeConverter_UT1_GMST
 - jeod::TimeConverter_UT1_GMST, [98](#)
- ~TimeDyn
 - jeod::TimeDyn, [101](#)
- ~TimeGMST
 - jeod::TimeGMST, [107](#)
- ~TimeGPS
 - jeod::TimeGPS, [111](#)
- ~TimeLinks
 - jeod::TimeLinks, [117](#)
- ~TimeMET
 - jeod::TimeMET, [150](#)
- ~TimeManager
 - jeod::TimeManager, [120](#)
- ~TimeManagerInit
 - jeod::TimeManagerInit, [133](#)
- ~TimeStandard
 - jeod::TimeStandard, [154](#)
- ~TimeTT
 - jeod::TimeTT, [171](#)
- ~TimeTAI
 - jeod::TimeTAI, [166](#)
- ~TimeTDB
 - jeod::TimeTDB, [169](#)
- ~TimeUDE
 - jeod::TimeUDE, [175](#)
- ~TimeUT1
 - jeod::TimeUT1, [191](#)
- ~TimeUTC
 - jeod::TimeUTC, [194](#)
- a_name
 - jeod::TimeConverter, [46](#)
- a_to_b_offset
 - jeod::TimeConverter, [46](#)
- a_to_b_offset_epoch
 - jeod::TimeConverter_TAI_TDB, [72](#)
- add_parent
 - jeod::JeodBaseTime, [26](#)
- add_type_initialize
 - jeod::JeodBaseTime, [26](#)
 - jeod::TimeStandard, [154](#)
 - jeod::TimeUDE, [176](#)
- add_type_update
 - jeod::JeodBaseTime, [27](#)
- b_name
 - jeod::TimeConverter, [47](#)
- calculate_calendar_values
 - jeod::TimeGMST, [107](#)
 - jeod::TimeGPS, [111](#)
 - jeod::TimeStandard, [155](#)
- calendar_day
 - jeod::TimeStandard, [160](#)
- calendar_hour
 - jeod::TimeStandard, [161](#)
- calendar_minute
 - jeod::TimeStandard, [161](#)
- calendar_month
 - jeod::TimeStandard, [161](#)
- calendar_second
 - jeod::TimeStandard, [161](#)
- calendar_update
 - jeod::TimeStandard, [155](#)
- calendar_year
 - jeod::TimeStandard, [162](#)
- can_convert
 - jeod::TimeConverter, [42](#)
- class_declarations.hh, [197](#)
- clock_day
 - jeod::TimeUDE, [184](#)
- clock_hour
 - jeod::TimeUDE, [185](#)
- clock_minute
 - jeod::TimeUDE, [185](#)
- clock_resolution

- jeod::JeodBaseTime, 34
- clock_second
 - jeod::TimeUDE, 185
- clock_update
 - jeod::TimeUDE, 176
- convert_a_to_b
 - jeod::TimeConverter, 42
 - jeod::TimeConverter_Dyn_TAI, 49
 - jeod::TimeConverter_Dyn_TDB, 53
 - jeod::TimeConverter_Dyn_UDE, 57
 - jeod::TimeConverter_STD_UDE, 61
 - jeod::TimeConverter_TAI_GPS, 66
 - jeod::TimeConverter_TAI_TDB, 70
 - jeod::TimeConverter_TAI_TT, 76
 - jeod::TimeConverter_TAI_UT1, 81
 - jeod::TimeConverter_TAI_UTC, 90
 - jeod::TimeConverter_UT1_GMST, 98
- convert_b_to_a
 - jeod::TimeConverter, 42
 - jeod::TimeConverter_STD_UDE, 61
 - jeod::TimeConverter_TAI_GPS, 66
 - jeod::TimeConverter_TAI_TDB, 70
 - jeod::TimeConverter_TAI_TT, 76
 - jeod::TimeConverter_TAI_UT1, 81
 - jeod::TimeConverter_TAI_UTC, 90
- convert_epoch_to_update
 - jeod::TimeUDE, 177
- convert_from_calendar
 - jeod::TimeGPS, 111
 - jeod::TimeStandard, 156
- converter_ptrs_index
 - jeod::TimeManagerInit, 141
- converter_vector
 - jeod::TimeManager, 129
- create_init_tree
 - jeod::TimeManagerInit, 134
- create_update_tree
 - jeod::TimeManagerInit, 134
- day_of_week
 - jeod::TimeGPS, 114
- days
 - jeod::JeodBaseTime, 34
- default_path_size
 - jeod::TimeLinks, 118
- Direction
 - jeod::TimeConverter, 41
- duplicate_methods
 - jeod::TimeMessages, 146
- dyn_ptr
 - jeod::TimeConverter_Dyn_TAI, 51
 - jeod::TimeConverter_Dyn_TDB, 54
 - jeod::TimeConverter_Dyn_UDE, 59
- dyn_time
 - jeod::TimeManager, 129
- dyn_time_index
 - jeod::TimeManagerInit, 141
- Environment, 14
- epoch_data_present
 - jeod::TimeUDE, 185
- epoch_day
 - jeod::TimeUDE, 186
- epoch_defined_in_name
 - jeod::TimeUDE, 186
- epoch_format
 - jeod::TimeUDE, 186
- epoch_hour
 - jeod::TimeUDE, 186
- epoch_index
 - jeod::TimeUDE, 187
- epoch_initializing_value
 - jeod::TimeUDE, 187
- epoch_minute
 - jeod::TimeUDE, 187
- epoch_month
 - jeod::TimeUDE, 187
- epoch_second
 - jeod::TimeUDE, 188
- epoch_value_is_set_calendar
 - jeod::TimeUDE, 188
- epoch_value_is_set_clock
 - jeod::TimeUDE, 188
- epoch_value_is_set_number
 - jeod::TimeUDE, 188
- epoch_year
 - jeod::TimeUDE, 189
- extension_error
 - jeod::TimeMessages, 146
- failed_null_test
 - jeod::TimeConverter_STD_UDE, 63
- get_a_to_b_offset
 - jeod::TimeConverter, 43
- get_conv_dir_init
 - jeod::TimeManagerInit, 134
- get_conv_dir_upd
 - jeod::TimeManagerInit, 135
- get_conv_ptr_index
 - jeod::TimeManagerInit, 136
- get_converter_ptr
 - jeod::TimeManager, 121
- get_days
 - jeod::TimeUT1, 192
- get_index
 - jeod::JeodBaseTime, 28
- get_jeod_integration_time
 - jeod::TimeManager, 121
- get_status
 - jeod::TimeManagerInit, 136
- get_time_change_flag
 - jeod::TimeManager, 122
- get_time_ptr
 - jeod::TimeManager, 122
- get_time_scale_factor
 - jeod::TimeManager, 123
- get_timestamp_time

- jeod::TimeManager, 123
- gmst_ptr
 - jeod::TimeConverter_UT1_GMST, 99
- gps_ptr
 - jeod::TimeConverter_TAI_GPS, 67
- gradient
 - jeod::TimeConverter_TAI_UT1, 83
- hold
 - jeod::TimeMET, 151
- incomplete_setup_error
 - jeod::TimeMessages, 146
- increment_status
 - jeod::TimeManagerInit, 137
- index
 - jeod::JeodBaseTime, 34
 - jeod::TimeConverter_TAI_UT1, 84
 - jeod::TimeConverter_TAI_UTC, 93
- init_attrjeod__JeodBaseTime
 - jeod::JeodBaseTime, 33
- init_attrjeod__TimeConverter
 - jeod::TimeConverter, 46
- init_attrjeod__TimeConverter_Dyn_TAI
 - jeod::TimeConverter_Dyn_TAI, 50
- init_attrjeod__TimeConverter_Dyn_TDB
 - jeod::TimeConverter_Dyn_TDB, 54
- init_attrjeod__TimeConverter_Dyn_UDE
 - jeod::TimeConverter_Dyn_UDE, 58
- init_attrjeod__TimeConverter_STD_UDE
 - jeod::TimeConverter_STD_UDE, 63
- init_attrjeod__TimeConverter_TAI_GPS
 - jeod::TimeConverter_TAI_GPS, 67
- init_attrjeod__TimeConverter_TAI_TDB
 - jeod::TimeConverter_TAI_TDB, 72
- init_attrjeod__TimeConverter_TAI_TT
 - jeod::TimeConverter_TAI_TT, 77
- init_attrjeod__TimeConverter_TAI_UT1
 - jeod::TimeConverter_TAI_UT1, 83
- init_attrjeod__TimeConverter_TAI_UTC
 - jeod::TimeConverter_TAI_UTC, 92
- init_attrjeod__TimeConverter_UT1_GMST
 - jeod::TimeConverter_UT1_GMST, 99
- init_attrjeod__TimeDyn
 - jeod::TimeDyn, 103
- init_attrjeod__TimeGMST
 - jeod::TimeGMST, 108
- init_attrjeod__TimeGPS
 - jeod::TimeGPS, 114
- init_attrjeod__TimeLinks
 - jeod::TimeLinks, 118
- init_attrjeod__TimeMET
 - jeod::TimeMET, 151
- init_attrjeod__TimeManager
 - jeod::TimeManager, 129
- init_attrjeod__TimeManagerInit
 - jeod::TimeManagerInit, 140
- init_attrjeod__TimeMessages
 - jeod::TimeMessages, 145
- init_attrjeod__TimeStandard
 - jeod::TimeStandard, 160
- init_attrjeod__TimeTAI
 - jeod::TimeTAI, 167
- init_attrjeod__TimeTDB
 - jeod::TimeTDB, 170
- init_attrjeod__TimeTT
 - jeod::TimeTT, 172
- init_attrjeod__TimeUDE
 - jeod::TimeUDE, 184
- init_attrjeod__TimeUT1
 - jeod::TimeUT1, 193
- init_attrjeod__TimeUTC
 - jeod::TimeUTC, 195
- init_converter_dir_table
 - jeod::TimeManagerInit, 141
- initial_value
 - jeod::JeodBaseTime, 34
- initial_value_format
 - jeod::TimeUDE, 189
- initialization_error
 - jeod::TimeMessages, 146
- initialize
 - jeod::TimeConverter, 43
 - jeod::TimeConverter_Dyn_TAI, 50
 - jeod::TimeConverter_Dyn_TDB, 53
 - jeod::TimeConverter_Dyn_UDE, 57
 - jeod::TimeConverter_STD_UDE, 62
 - jeod::TimeConverter_TAI_GPS, 66
 - jeod::TimeConverter_TAI_TDB, 71
 - jeod::TimeConverter_TAI_TT, 77
 - jeod::TimeConverter_TAI_UT1, 81
 - jeod::TimeConverter_TAI_UT1_tai_to_ut1 ↔
 - default_data, 87
 - jeod::TimeConverter_TAI_UTC_tai_to_utc ↔
 - default_data, 96
 - jeod::TimeConverter_TAI_UTC, 91
 - jeod::TimeConverter_UT1_GMST, 98
 - jeod::TimeManager, 123
 - jeod::TimeManagerInit, 137
- initialize_from_name
 - jeod::JeodBaseTime, 35
- initialize_from_parent
 - jeod::JeodBaseTime, 28
 - jeod::TimeStandard, 156
 - jeod::TimeUDE, 177
- initialize_initializer_time
 - jeod::JeodBaseTime, 28
 - jeod::TimeDyn, 102
 - jeod::TimeStandard, 157
 - jeod::TimeUDE, 178
- initialize_leap_second
 - jeod::TimeConverter_TAI_UTC, 91
- initialize_manager
 - jeod::TimeManagerInit, 137
- initialize_tai_to_ut1
 - jeod::TimeConverter_TAI_UT1, 82
- initialize_time_types

- jeod::TimeManagerInit, 138
- initialized
 - jeod::JeodBaseTime, 35
 - jeod::TimeConverter, 47
- initializer
 - jeod::TimeManagerInit, 142
- initializer_index
 - jeod::TimeManagerInit, 142
- initializing_data_present
 - jeod::TimeUDE, 189
- initializing_value
 - jeod::JeodBaseTime, 35
- InputProcessor
 - jeod::JeodBaseTime, 33
 - jeod::TimeConverter, 46
 - jeod::TimeConverter_Dyn_TAI, 51
 - jeod::TimeConverter_Dyn_TDB, 54
 - jeod::TimeConverter_Dyn_UDE, 58
 - jeod::TimeConverter_STD_UDE, 63
 - jeod::TimeConverter_TAI_GPS, 67
 - jeod::TimeConverter_TAI_TDB, 72
 - jeod::TimeConverter_TAI_TT, 78
 - jeod::TimeConverter_TAI_UT1, 83
 - jeod::TimeConverter_TAI_UTC, 92
 - jeod::TimeConverter_UT1_GMST, 99
 - jeod::TimeDyn, 103
 - jeod::TimeGMST, 108
 - jeod::TimeGPS, 114
 - jeod::TimeLinks, 118
 - jeod::TimeMET, 151
 - jeod::TimeManager, 129
 - jeod::TimeManagerInit, 141
 - jeod::TimeMessages, 145
 - jeod::TimeStandard, 160
 - jeod::TimeTAI, 167
 - jeod::TimeTDB, 170
 - jeod::TimeTT, 172
 - jeod::TimeUDE, 184
 - jeod::TimeUT1, 193
 - jeod::TimeUTC, 195
- invalid_data_error
 - jeod::TimeMessages, 147
- invalid_node
 - jeod::TimeMessages, 147
- invalid_setup_error
 - jeod::TimeMessages, 147
- is_initialized
 - jeod::JeodBaseTime, 30
 - jeod::TimeConverter, 44
- JEOD_FRIEND_CLASS
 - tai_to_ut1.cc, 198
 - tai_to_utc.cc, 198
- jeod, 19
 - operator|, 20
- jeod::JeodBaseTime, 23
 - ~JeodBaseTime, 25
 - add_parent, 26
 - add_type_initialize, 26
 - add_type_update, 27
 - clock_resolution, 34
 - days, 34
 - get_index, 28
 - index, 34
 - init_attrjeod__JeodBaseTime, 33
 - initial_value, 34
 - initialize_from_name, 35
 - initialize_from_parent, 28
 - initialize_initializer_time, 28
 - initialized, 35
 - initializing_value, 35
 - InputProcessor, 33
 - is_initialized, 30
 - JeodBaseTime, 25, 26
 - links, 36
 - must_be_singleton, 30
 - name, 36
 - operator=, 30
 - override_initialized, 30
 - seconds, 36
 - set_index, 31
 - set_name, 31
 - set_time_by_days, 31
 - set_time_by_seconds, 32
 - time_manager, 37
 - TimeConverter, 33
 - TimeManagerInit, 33
 - update, 32
 - update_converter_direction, 37
 - update_converter_ptr, 38
 - update_from_name, 38
- jeod::TimeConverter, 39
 - ~TimeConverter, 41
 - a_name, 46
 - a_to_b_offset, 46
 - b_name, 47
 - can_convert, 42
 - convert_a_to_b, 42
 - convert_b_to_a, 42
 - Direction, 41
 - get_a_to_b_offset, 43
 - init_attrjeod__TimeConverter, 46
 - initialize, 43
 - initialized, 47
 - InputProcessor, 46
 - is_initialized, 44
 - JeodBaseTime, 46
 - operator=, 44
 - override_initialized, 44
 - reset_a_to_b_offset, 44
 - TimeConverter, 41
 - valid_directions, 47
 - verify_setup, 44
 - verify_table_lookup_ends, 45
- jeod::TimeConverter_Dyn_TAI, 48
 - ~TimeConverter_Dyn_TAI, 49
 - convert_a_to_b, 49

- dyn_ptr, 51
- init_attrjeod__TimeConverter_Dyn_TAI, 50
- initialize, 50
- InputProcessor, 51
- operator=, 50
- tai_ptr, 51
- TimeConverter_Dyn_TAI, 49
- jeod::TimeConverter_Dyn_TDB, 52
 - ~TimeConverter_Dyn_TDB, 53
 - convert_a_to_b, 53
 - dyn_ptr, 54
 - init_attrjeod__TimeConverter_Dyn_TDB, 54
 - initialize, 53
 - InputProcessor, 54
 - operator=, 54
 - tdb_ptr, 55
 - TimeConverter_Dyn_TDB, 53
- jeod::TimeConverter_Dyn_UDE, 55
 - ~TimeConverter_Dyn_UDE, 56
 - convert_a_to_b, 57
 - dyn_ptr, 59
 - init_attrjeod__TimeConverter_Dyn_UDE, 58
 - initialize, 57
 - InputProcessor, 58
 - operator=, 58
 - reset_a_to_b_offset, 58
 - TimeConverter_Dyn_UDE, 56, 57
 - ude_ptr, 59
- jeod::TimeConverter_STD_UDE, 59
 - ~TimeConverter_STD_UDE, 61
 - convert_a_to_b, 61
 - convert_b_to_a, 61
 - failed_null_test, 63
 - init_attrjeod__TimeConverter_STD_UDE, 63
 - initialize, 62
 - InputProcessor, 63
 - operator=, 62
 - reset_a_to_b_offset, 62
 - std_ptr, 63
 - TimeConverter_STD_UDE, 60, 61
 - ude_ptr, 64
- jeod::TimeConverter_TAI_GPS, 64
 - ~TimeConverter_TAI_GPS, 65
 - convert_a_to_b, 66
 - convert_b_to_a, 66
 - gps_ptr, 67
 - init_attrjeod__TimeConverter_TAI_GPS, 67
 - initialize, 66
 - InputProcessor, 67
 - operator=, 67
 - tai_ptr, 68
 - TimeConverter_TAI_GPS, 65, 66
- jeod::TimeConverter_TAI_TDB, 68
 - ~TimeConverter_TAI_TDB, 70
 - a_to_b_offset_epoch, 72
 - convert_a_to_b, 70
 - convert_b_to_a, 70
 - init_attrjeod__TimeConverter_TAI_TDB, 72
 - initialize, 71
 - InputProcessor, 72
 - nIter, 72
 - nSteps, 73
 - operator=, 71
 - prev_tai_seconds, 73
 - prev_tdb_seconds, 73
 - set_a_to_b_offset, 71
 - TAI_to_TT_offset, 74
 - tai_ptr, 73
 - tdb_ptr, 74
 - TimeConverter_TAI_TDB, 70
- jeod::TimeConverter_TAI_TT, 75
 - ~TimeConverter_TAI_TT, 76
 - convert_a_to_b, 76
 - convert_b_to_a, 76
 - init_attrjeod__TimeConverter_TAI_TT, 77
 - initialize, 77
 - InputProcessor, 78
 - operator=, 77
 - tai_ptr, 78
 - TimeConverter_TAI_TT, 76
 - tt_ptr, 78
- jeod::TimeConverter_TAI_UT1, 79
 - ~TimeConverter_TAI_UT1, 80
 - convert_a_to_b, 81
 - convert_b_to_a, 81
 - gradient, 83
 - index, 84
 - init_attrjeod__TimeConverter_TAI_UT1, 83
 - initialize, 81
 - initialize_tai_to_ut1, 82
 - InputProcessor, 83
 - last_index, 84
 - next_value, 84
 - next_when, 84
 - off_table_end, 85
 - operator=, 82
 - override_data_table, 85
 - prev_value, 85
 - prev_when, 85
 - tai_ptr, 86
 - tai_to_ut1_override_val, 86
 - TimeConverter_TAI_UT1, 80, 81
 - ut1_ptr, 86
 - val_vec, 86
 - verify_table_lookup_ends, 83
 - when_vec, 87
- jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_↔
 - data, 87
 - initialize, 87
- jeod::TimeConverter_TAI_UTC_tai_to_utc_default_↔
 - data, 96
 - initialize, 96
- jeod::TimeConverter_TAI_UTC, 88
 - ~TimeConverter_TAI_UTC, 90
 - convert_a_to_b, 90
 - convert_b_to_a, 90

- index, 93
- init_attrjeod__TimeConverter_TAI_UTC, 92
- initialize, 91
- initialize_leap_second, 91
- InputProcessor, 92
- last_index, 93
- leap_sec_override_val, 93
- next_when, 93
- off_table_end, 94
- operator=, 92
- override_data_table, 94
- prev_when, 94
- tai_ptr, 94
- TimeConverter_TAI_UTC, 90
- utc_ptr, 95
- val_vec, 95
- verify_table_lookup_ends, 92
- when_vec, 95
- jeod::TimeConverter_UT1_GMST, 97
 - ~TimeConverter_UT1_GMST, 98
 - convert_a_to_b, 98
 - gmst_ptr, 99
 - init_attrjeod__TimeConverter_UT1_GMST, 99
 - initialize, 98
 - InputProcessor, 99
 - operator=, 99
 - TimeConverter_UT1_GMST, 98
 - ut1_ptr, 100
- jeod::TimeDyn, 100
 - ~TimeDyn, 101
 - init_attrjeod__TimeDyn, 103
 - initialize_initializer_time, 102
 - InputProcessor, 103
 - offset, 104
 - operator=, 102
 - ref_scale, 104
 - scale_factor, 104
 - TimeDyn, 101, 102
 - update, 102
 - update_offset, 103
- jeod::TimeEnum, 105
 - TimeFormat, 105
- jeod::TimeGMST, 106
 - ~TimeGMST, 107
 - calculate_calendar_values, 107
 - init_attrjeod__TimeGMST, 108
 - InputProcessor, 108
 - operator=, 107
 - set_epoch, 108
 - set_time_by_trunc_julian, 108
 - TimeGMST, 107
- jeod::TimeGPS, 109
 - ~TimeGPS, 111
 - calculate_calendar_values, 111
 - convert_from_calendar, 111
 - day_of_week, 114
 - init_attrjeod__TimeGPS, 114
 - InputProcessor, 114
 - operator=, 112
 - rollover_count, 114
 - rollover_count_13_bit, 115
 - seconds_of_day, 115
 - seconds_of_week, 115
 - set_epoch, 112
 - set_time_by_days, 112
 - set_time_by_seconds, 113
 - set_time_by_trunc_julian, 113
 - TimeGPS, 110, 111
 - week, 115
 - week_13_bit, 116
- jeod::TimeLinks, 116
 - ~TimeLinks, 117
 - default_path_size, 118
 - init_attrjeod__TimeLinks, 118
 - InputProcessor, 118
 - operator=, 118
 - TimeLinks, 117
- jeod::TimeMET, 149
 - ~TimeMET, 150
 - hold, 151
 - init_attrjeod__TimeMET, 151
 - InputProcessor, 151
 - operator=, 150
 - previous_hold, 151
 - TimeMET, 150
 - update, 150
- jeod::TimeManager, 119
 - ~TimeManager, 120
 - converter_vector, 129
 - dyn_time, 129
 - get_converter_ptr, 121
 - get_jeod_integration_time, 121
 - get_time_change_flag, 122
 - get_time_ptr, 122
 - get_time_scale_factor, 123
 - get_timestamp_time, 123
 - init_attrjeod__TimeManager, 129
 - initialize, 123
 - InputProcessor, 129
 - num_types, 130
 - operator=, 124
 - register_converter, 124
 - register_time, 124
 - register_time_named, 125
 - simtime, 130
 - time_change_flag, 130
 - time_lookup, 125
 - time_standards_exist, 126
 - time_vector, 131
 - TimeManager, 120, 121
 - TimeManagerInit, 129
 - update, 126
 - update_time, 128
 - verify_table_lookup_ends, 128
- jeod::TimeManagerInit, 131
 - ~TimeManagerInit, 133

- converter_ptr_index, 141
- create_init_tree, 134
- create_update_tree, 134
- dyn_time_index, 141
- get_conv_dir_init, 134
- get_conv_dir_upd, 135
- get_conv_ptr_index, 136
- get_status, 136
- increment_status, 137
- init_attrjeod__TimeManagerInit, 140
- init_converter_dir_table, 141
- initialize, 137
- initialize_manager, 137
- initialize_time_types, 138
- initializer, 142
- initializer_index, 142
- InputProcessor, 141
- num_added_pass, 142
- num_added_total, 142
- operator=, 138
- organize_update_list, 138
- populate_converter_registry, 139
- set_status, 139
- sim_start_format, 143
- status, 143
- time_manager, 143
- TimeManagerInit, 133
- update_converter_dir_table, 143
- verify_converter_setup, 140
- verify_times_setup, 140
- jeod::TimeMessages, 144
 - duplicate_methods, 146
 - extension_error, 146
 - incomplete_setup_error, 146
 - init_attrjeod__TimeMessages, 145
 - initialization_error, 146
 - InputProcessor, 145
 - invalid_data_error, 147
 - invalid_node, 147
 - invalid_setup_error, 147
 - memory_error, 148
 - operator=, 145
 - redundancy_error, 148
 - TimeMessages, 145
- jeod::TimeStandard, 152
 - ~TimeStandard, 154
 - add_type_initialize, 154
 - calculate_calendar_values, 155
 - calendar_day, 160
 - calendar_hour, 161
 - calendar_minute, 161
 - calendar_month, 161
 - calendar_second, 161
 - calendar_update, 155
 - calendar_year, 162
 - convert_from_calendar, 156
 - init_attrjeod__TimeStandard, 160
 - initialize_from_parent, 156
 - initialize_initializer_time, 157
 - InputProcessor, 160
 - julian_date, 162
 - julian_date_at_epoch, 157
 - last_calendar_update, 162
 - operator=, 158
 - prev_julian_day, 162
 - seconds_at_year_start, 163
 - seconds_of_year, 158
 - send_warning_pre_1968, 163
 - set_epoch, 158
 - set_time_by_days, 158
 - set_time_by_seconds, 159
 - set_time_by_trunc_julian, 159
 - TimeStandard, 154
 - TimeUDE, 160
 - tjt_at_epoch, 163
 - tjt_jd_offset, 163
 - tjt_mjt_offset, 164
 - trunc_julian_time, 164
 - year_of_last_soy, 164
- jeod::TimeTAI, 165
 - ~TimeTAI, 166
 - init_attrjeod__TimeTAI, 167
 - InputProcessor, 167
 - operator=, 166
 - set_epoch, 167
 - TimeTAI, 166
- jeod::TimeTDB, 168
 - ~TimeTDB, 169
 - init_attrjeod__TimeTDB, 170
 - InputProcessor, 170
 - operator=, 169
 - set_epoch, 169
 - TimeTDB, 168, 169
- jeod::TimeTT, 170
 - ~TimeTT, 171
 - init_attrjeod__TimeTT, 172
 - InputProcessor, 172
 - operator=, 172
 - set_epoch, 172
 - TimeTT, 171
- jeod::TimeUDE, 173
 - ~TimeUDE, 175
 - add_type_initialize, 176
 - clock_day, 184
 - clock_hour, 185
 - clock_minute, 185
 - clock_second, 185
 - clock_update, 176
 - convert_epoch_to_update, 177
 - epoch_data_present, 185
 - epoch_day, 186
 - epoch_defined_in_name, 186
 - epoch_format, 186
 - epoch_hour, 186
 - epoch_index, 187
 - epoch_initializing_value, 187

- epoch_minute, [187](#)
- epoch_month, [187](#)
- epoch_second, [188](#)
- epoch_value_is_set_calendar, [188](#)
- epoch_value_is_set_clock, [188](#)
- epoch_value_is_set_number, [188](#)
- epoch_year, [189](#)
- init_attrjeod__TimeUDE, [184](#)
- initial_value_format, [189](#)
- initialize_from_parent, [177](#)
- initialize_initializer_time, [178](#)
- initializing_data_present, [189](#)
- InputProcessor, [184](#)
- last_clock_update, [189](#)
- must_be_singleton, [178](#)
- operator=, [179](#)
- set_epoch_dyn, [179](#)
- set_epoch_initializing_value, [180](#)
- set_epoch_std, [180](#)
- set_epoch_times, [181](#)
- set_epoch_ude, [181](#)
- set_initial_times, [182](#)
- set_time_by_clock, [182](#)
- set_time_by_days, [182](#)
- set_time_by_seconds, [183](#)
- TimeUDE, [175](#)
- update_index, [190](#)
- verify_epoch, [183](#)
- verify_init, [183](#)
- verify_update, [184](#)
- jeod::TimeUT1, [190](#)
 - ~TimeUT1, [191](#)
 - get_days, [192](#)
 - init_attrjeod__TimeUT1, [193](#)
 - InputProcessor, [193](#)
 - operator=, [192](#)
 - set_epoch, [192](#)
 - TimeUT1, [191](#), [192](#)
 - true_ut1, [193](#)
- jeod::TimeUTC, [193](#)
 - ~TimeUTC, [194](#)
 - init_attrjeod__TimeUTC, [195](#)
 - InputProcessor, [195](#)
 - operator=, [195](#)
 - set_epoch, [195](#)
 - TimeUTC, [194](#), [195](#)
 - true_utc, [196](#)
- JeodBaseTime
 - jeod::JeodBaseTime, [25](#), [26](#)
 - jeod::TimeConverter, [46](#)
- julian_date
 - jeod::TimeStandard, [162](#)
- julian_date_at_epoch
 - jeod::TimeStandard, [157](#)
- last_calendar_update
 - jeod::TimeStandard, [162](#)
- last_clock_update
 - jeod::TimeUDE, [189](#)
- last_index
 - jeod::TimeConverter_TAI_UT1, [84](#)
 - jeod::TimeConverter_TAI_UTC, [93](#)
- leap_sec_override_val
 - jeod::TimeConverter_TAI_UTC, [93](#)
- links
 - jeod::JeodBaseTime, [36](#)
- memory_error
 - jeod::TimeMessages, [148](#)
- Models, [13](#)
- must_be_singleton
 - jeod::JeodBaseTime, [30](#)
 - jeod::TimeUDE, [178](#)
- nIter
 - jeod::TimeConverter_TAI_TDB, [72](#)
- nSteps
 - jeod::TimeConverter_TAI_TDB, [73](#)
- name
 - jeod::JeodBaseTime, [36](#)
- next_value
 - jeod::TimeConverter_TAI_UT1, [84](#)
- next_when
 - jeod::TimeConverter_TAI_UT1, [84](#)
 - jeod::TimeConverter_TAI_UTC, [93](#)
- num_added_pass
 - jeod::TimeManagerInit, [142](#)
- num_added_total
 - jeod::TimeManagerInit, [142](#)
- num_types
 - jeod::TimeManager, [130](#)
- off_table_end
 - jeod::TimeConverter_TAI_UT1, [85](#)
 - jeod::TimeConverter_TAI_UTC, [94](#)
- offset
 - jeod::TimeDyn, [104](#)
- operator=
 - jeod::JeodBaseTime, [30](#)
 - jeod::TimeConverter, [44](#)
 - jeod::TimeConverter_Dyn_TAI, [50](#)
 - jeod::TimeConverter_Dyn_TDB, [54](#)
 - jeod::TimeConverter_Dyn_UDE, [58](#)
 - jeod::TimeConverter_STD_UDE, [62](#)
 - jeod::TimeConverter_TAI_GPS, [67](#)
 - jeod::TimeConverter_TAI_TDB, [71](#)
 - jeod::TimeConverter_TAI_TT, [77](#)
 - jeod::TimeConverter_TAI_UT1, [82](#)
 - jeod::TimeConverter_TAI_UTC, [92](#)
 - jeod::TimeConverter_UT1_GMST, [99](#)
 - jeod::TimeDyn, [102](#)
 - jeod::TimeGMST, [107](#)
 - jeod::TimeGPS, [112](#)
 - jeod::TimeLinks, [118](#)
 - jeod::TimeMET, [150](#)
 - jeod::TimeManager, [124](#)
 - jeod::TimeManagerInit, [138](#)
 - jeod::TimeMessages, [145](#)

- jeod::TimeStandard, 158
- jeod::TimeTAI, 166
- jeod::TimeTDB, 169
- jeod::TimeTT, 172
- jeod::TimeUDE, 179
- jeod::TimeUT1, 192
- jeod::TimeUTC, 195
- operator |
 - jeod, 20
- organize_update_list
 - jeod::TimeManagerInit, 138
- override_data_table
 - jeod::TimeConverter_TAI_UT1, 85
 - jeod::TimeConverter_TAI_UTC, 94
- override_initialized
 - jeod::JeodBaseTime, 30
 - jeod::TimeConverter, 44
- PATH
 - Time, 17
- populate_converter_registry
 - jeod::TimeManagerInit, 139
- prev_julian_day
 - jeod::TimeStandard, 162
- prev_tai_seconds
 - jeod::TimeConverter_TAI_TDB, 73
- prev_tdb_seconds
 - jeod::TimeConverter_TAI_TDB, 73
- prev_value
 - jeod::TimeConverter_TAI_UT1, 85
- prev_when
 - jeod::TimeConverter_TAI_UT1, 85
 - jeod::TimeConverter_TAI_UTC, 94
- previous_hold
 - jeod::TimeMET, 151
- redundancy_error
 - jeod::TimeMessages, 148
- ref_scale
 - jeod::TimeDyn, 104
- register_converter
 - jeod::TimeManager, 124
- register_time
 - jeod::TimeManager, 124
- register_time_named
 - jeod::TimeManager, 125
- reset_a_to_b_offset
 - jeod::TimeConverter, 44
 - jeod::TimeConverter_Dyn_UDE, 58
 - jeod::TimeConverter_STD_UDE, 62
- rollover_count
 - jeod::TimeGPS, 114
- rollover_count_13_bit
 - jeod::TimeGPS, 115
- scale_factor
 - jeod::TimeDyn, 104
- seconds
 - jeod::JeodBaseTime, 36
- seconds_at_year_start
 - jeod::TimeStandard, 163
- seconds_of_day
 - jeod::TimeGPS, 115
- seconds_of_week
 - jeod::TimeGPS, 115
- seconds_of_year
 - jeod::TimeStandard, 158
- send_warning_pre_1968
 - jeod::TimeStandard, 163
- set_a_to_b_offset
 - jeod::TimeConverter_TAI_TDB, 71
- set_epoch
 - jeod::TimeGMST, 108
 - jeod::TimeGPS, 112
 - jeod::TimeStandard, 158
 - jeod::TimeTAI, 167
 - jeod::TimeTDB, 169
 - jeod::TimeTT, 172
 - jeod::TimeUT1, 192
 - jeod::TimeUTC, 195
- set_epoch_dyn
 - jeod::TimeUDE, 179
- set_epoch_initializing_value
 - jeod::TimeUDE, 180
- set_epoch_std
 - jeod::TimeUDE, 180
- set_epoch_times
 - jeod::TimeUDE, 181
- set_epoch_ude
 - jeod::TimeUDE, 181
- set_index
 - jeod::JeodBaseTime, 31
- set_initial_times
 - jeod::TimeUDE, 182
- set_name
 - jeod::JeodBaseTime, 31
- set_status
 - jeod::TimeManagerInit, 139
- set_time_by_clock
 - jeod::TimeUDE, 182
- set_time_by_days
 - jeod::JeodBaseTime, 31
 - jeod::TimeGPS, 112
 - jeod::TimeStandard, 158
 - jeod::TimeUDE, 182
- set_time_by_seconds
 - jeod::JeodBaseTime, 32
 - jeod::TimeGPS, 113
 - jeod::TimeStandard, 159
 - jeod::TimeUDE, 183
- set_time_by_trunc_julian
 - jeod::TimeGMST, 108
 - jeod::TimeGPS, 113
 - jeod::TimeStandard, 159
- sim_start_format
 - jeod::TimeManagerInit, 143
- simtime

- jeod::TimeManager, 130
- status
 - jeod::TimeManagerInit, 143
- std_ptr
 - jeod::TimeConverter_STD_UDE, 63
- TAI_to_TT_offset
 - jeod::TimeConverter_TAI_TDB, 74
- tai_ptr
 - jeod::TimeConverter_Dyn_TAI, 51
 - jeod::TimeConverter_TAI_GPS, 68
 - jeod::TimeConverter_TAI_TDB, 73
 - jeod::TimeConverter_TAI_TT, 78
 - jeod::TimeConverter_TAI_UT1, 86
 - jeod::TimeConverter_TAI_UTC, 94
- tai_to_ut1.cc, 197
 - JEOD_FRIEND_CLASS, 198
- tai_to_ut1.hh, 198
- tai_to_ut1_override_val
 - jeod::TimeConverter_TAI_UT1, 86
- tai_to_utc.cc, 198
 - JEOD_FRIEND_CLASS, 198
- tai_to_utc.hh, 199
- tdb_ptr
 - jeod::TimeConverter_Dyn_TDB, 55
 - jeod::TimeConverter_TAI_TDB, 74
- Time, 15
 - PATH, 17
- time.cc, 199
- time.hh, 199
- time__add_type_update.cc, 200
- time_change_flag
 - jeod::TimeManager, 130
- time_converter.cc, 201
- time_converter.hh, 201
- time_converter_dyn_tai.cc, 202
- time_converter_dyn_tai.hh, 202
- time_converter_dyn_tdb.cc, 203
- time_converter_dyn_tdb.hh, 203
- time_converter_dyn_ude.cc, 204
- time_converter_dyn_ude.hh, 204
- time_converter_std_ude.cc, 205
- time_converter_std_ude.hh, 205
- time_converter_tai_gps.cc, 206
- time_converter_tai_gps.hh, 206
- time_converter_tai_tdb.cc, 207
- time_converter_tai_tdb.hh, 207
- time_converter_tai_tt.cc, 208
- time_converter_tai_tt.hh, 208
- time_converter_tai_ut1.cc, 209
- time_converter_tai_ut1.hh, 209
- time_converter_tai_utc.cc, 210
- time_converter_tai_utc.hh, 210
- time_converter_ut1_gmst.cc, 211
- time_converter_ut1_gmst.hh, 211
- time_dyn.cc, 212
- time_dyn.hh, 212
- time_enum.hh, 213
- time_gmst.cc, 213
- time_gmst.hh, 214
- time_gps.cc, 214
- time_gps.hh, 215
- time_links.hh, 215
- time_lookup
 - jeod::TimeManager, 125
- time_manager
 - jeod::JeodBaseTime, 37
 - jeod::TimeManagerInit, 143
- time_manager.cc, 216
- time_manager.hh, 216
- time_manager__initialize.cc, 217
- time_manager_init.cc, 217
- time_manager_init.hh, 218
- time_messages.cc, 218
- time_messages.hh, 219
- time_met.cc, 219
- time_met.hh, 220
- time_standard.cc, 220
- time_standard.hh, 221
- time_standards_exist
 - jeod::TimeManager, 126
- time_tai.cc, 221
- time_tai.hh, 222
- time_tdb.cc, 222
- time_tdb.hh, 223
- time_tt.cc, 223
- time_tt.hh, 224
- time_ude.cc, 224
- time_ude.hh, 225
- time_ut1.cc, 225
- time_ut1.hh, 226
- time_utc.cc, 226
- time_utc.hh, 227
- time_vector
 - jeod::TimeManager, 131
- TimeConverter
 - jeod::JeodBaseTime, 33
 - jeod::TimeConverter, 41
- TimeConverter_Dyn_TAI
 - jeod::TimeConverter_Dyn_TAI, 49
- TimeConverter_Dyn_TDB
 - jeod::TimeConverter_Dyn_TDB, 53
- TimeConverter_Dyn_UDE
 - jeod::TimeConverter_Dyn_UDE, 56, 57
- TimeConverter_STD_UDE
 - jeod::TimeConverter_STD_UDE, 60, 61
- TimeConverter_TAI_GPS
 - jeod::TimeConverter_TAI_GPS, 65, 66
- TimeConverter_TAI_TDB
 - jeod::TimeConverter_TAI_TDB, 70
- TimeConverter_TAI_TT
 - jeod::TimeConverter_TAI_TT, 76
- TimeConverter_TAI_UT1
 - jeod::TimeConverter_TAI_UT1, 80, 81
- TimeConverter_TAI_UTC
 - jeod::TimeConverter_TAI_UTC, 90
- TimeConverter_UT1_GMST

- jeod::TimeConverter_UT1_GMST, 98
- TimeDyn
 - jeod::TimeDyn, 101, 102
- TimeFormat
 - jeod::TimeEnum, 105
- TimeGMST
 - jeod::TimeGMST, 107
- TimeGPS
 - jeod::TimeGPS, 110, 111
- TimeLinks
 - jeod::TimeLinks, 117
- TimeMET
 - jeod::TimeMET, 150
- TimeManager
 - jeod::TimeManager, 120, 121
- TimeManagerInit
 - jeod::JeodBaseTime, 33
 - jeod::TimeManager, 129
 - jeod::TimeManagerInit, 133
- TimeMessages
 - jeod::TimeMessages, 145
- TimeStandard
 - jeod::TimeStandard, 154
- TimeTAI
 - jeod::TimeTAI, 166
- TimeTDB
 - jeod::TimeTDB, 168, 169
- TimeTT
 - jeod::TimeTT, 171
- TimeUDE
 - jeod::TimeStandard, 160
 - jeod::TimeUDE, 175
- TimeUT1
 - jeod::TimeUT1, 191, 192
- TimeUTC
 - jeod::TimeUTC, 194, 195
- tjt_at_epoch
 - jeod::TimeStandard, 163
- tjt_jd_offset
 - jeod::TimeStandard, 163
- tjt_mjt_offset
 - jeod::TimeStandard, 164
- true_ut1
 - jeod::TimeUT1, 193
- true_utc
 - jeod::TimeUTC, 196
- trunc_julian_time
 - jeod::TimeStandard, 164
- tt_ptr
 - jeod::TimeConverter_TAI_TT, 78
- ude_ptr
 - jeod::TimeConverter_Dyn_UDE, 59
 - jeod::TimeConverter_STD_UDE, 64
- update
 - jeod::JeodBaseTime, 32
 - jeod::TimeDyn, 102
 - jeod::TimeMET, 150
 - jeod::TimeManager, 126
- update_converter_dir_table
 - jeod::TimeManagerInit, 143
- update_converter_direction
 - jeod::JeodBaseTime, 37
- update_converter_ptr
 - jeod::JeodBaseTime, 38
- update_from_name
 - jeod::JeodBaseTime, 38
- update_index
 - jeod::TimeUDE, 190
- update_offset
 - jeod::TimeDyn, 103
- update_time
 - jeod::TimeManager, 128
- ut1_ptr
 - jeod::TimeConverter_TAI_UT1, 86
 - jeod::TimeConverter_UT1_GMST, 100
- utc_ptr
 - jeod::TimeConverter_TAI_UTC, 95
- val_vec
 - jeod::TimeConverter_TAI_UT1, 86
 - jeod::TimeConverter_TAI_UTC, 95
- valid_directions
 - jeod::TimeConverter, 47
- verify_converter_setup
 - jeod::TimeManagerInit, 140
- verify_epoch
 - jeod::TimeUDE, 183
- verify_init
 - jeod::TimeUDE, 183
- verify_setup
 - jeod::TimeConverter, 44
- verify_table_lookup_ends
 - jeod::TimeConverter, 45
 - jeod::TimeConverter_TAI_UT1, 83
 - jeod::TimeConverter_TAI_UTC, 92
 - jeod::TimeManager, 128
- verify_times_setup
 - jeod::TimeManagerInit, 140
- verify_update
 - jeod::TimeUDE, 184
- week
 - jeod::TimeGPS, 115
- week_13_bit
 - jeod::TimeGPS, 116
- when_vec
 - jeod::TimeConverter_TAI_UT1, 87
 - jeod::TimeConverter_TAI_UTC, 95
- year_of_last_soy
 - jeod::TimeStandard, 164