

TimeModel

5.0

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>13</b>
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

<b>7</b>	<b>Namespace Documentation</b>	<b>19</b>
7.1	jeod Namespace Reference . . . . .	19
7.1.1	Detailed Description . . . . .	20
7.1.2	Function Documentation . . . . .	20
7.1.2.1	operator"   () . . . . .	21
<b>8</b>	<b>Data Structure Documentation</b>	<b>23</b>
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	<a href="#">InputProcessor</a>	33
8.1.4.3	<a href="#">TimeConverter</a>	33
8.1.4.4	<a href="#">TimeManagerInit</a>	33
8.1.5	<a href="#">Field Documentation</a>	34
8.1.5.1	<a href="#">clock_resolution</a>	34
8.1.5.2	<a href="#">days</a>	34
8.1.5.3	<a href="#">index</a>	34
8.1.5.4	<a href="#">initial_value</a>	35
8.1.5.5	<a href="#">initialize_from_name</a>	35
8.1.5.6	<a href="#">initialized</a>	35
8.1.5.7	<a href="#">initializing_value</a>	36
8.1.5.8	<a href="#">links</a>	36
8.1.5.9	<a href="#">name</a>	36
8.1.5.10	<a href="#">seconds</a>	37
8.1.5.11	<a href="#">time_manager</a>	37
8.1.5.12	<a href="#">update_converter_direction</a>	38
8.1.5.13	<a href="#">update_converter_ptr</a>	38
8.1.5.14	<a href="#">update_from_name</a>	38
8.2	<a href="#">jeod::TimeConverter Class Reference</a>	39
8.2.1	<a href="#">Detailed Description</a>	41
8.2.2	<a href="#">Member Enumeration Documentation</a>	41
8.2.2.1	<a href="#">Direction</a>	41
8.2.3	<a href="#">Constructor &amp; Destructor Documentation</a>	41
8.2.3.1	<a href="#">~TimeConverter()</a>	41
8.2.3.2	<a href="#">TimeConverter() [1/2]</a>	42
8.2.3.3	<a href="#">TimeConverter() [2/2]</a>	42
8.2.4	<a href="#">Member Function Documentation</a>	42
8.2.4.1	<a href="#">can_convert()</a>	42
8.2.4.2	<a href="#">convert_a_to_b()</a>	42
8.2.4.3	<a href="#">convert_b_to_a()</a>	43

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

8.3.4.2	<a href="#">InputProcessor</a>	51
8.3.5	<a href="#">Field Documentation</a>	51
8.3.5.1	<a href="#">dyn_ptr</a>	52
8.3.5.2	<a href="#">tai_ptr</a>	52
8.4	<a href="#">jeod::TimeConverter_Dyn_TDB Class Reference</a>	52
8.4.1	<a href="#">Detailed Description</a>	53
8.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	53
8.4.2.1	<a href="#">TimeConverter_Dyn_TDB() [1/2]</a>	54
8.4.2.2	<a href="#">~TimeConverter_Dyn_TDB()</a>	54
8.4.2.3	<a href="#">TimeConverter_Dyn_TDB() [2/2]</a>	54
8.4.3	<a href="#">Member Function Documentation</a>	54
8.4.3.1	<a href="#">convert_a_to_b()</a>	54
8.4.3.2	<a href="#">initialize()</a>	54
8.4.3.3	<a href="#">operator=()</a>	55
8.4.4	<a href="#">Friends And Related Function Documentation</a>	55
8.4.4.1	<a href="#">init_attrjeod__TimeConverter_Dyn_TDB</a>	55
8.4.4.2	<a href="#">InputProcessor</a>	55
8.4.5	<a href="#">Field Documentation</a>	55
8.4.5.1	<a href="#">dyn_ptr</a>	56
8.4.5.2	<a href="#">tdb_ptr</a>	56
8.5	<a href="#">jeod::TimeConverter_Dyn_UDE Class Reference</a>	56
8.5.1	<a href="#">Detailed Description</a>	57
8.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	57
8.5.2.1	<a href="#">TimeConverter_Dyn_UDE() [1/2]</a>	58
8.5.2.2	<a href="#">~TimeConverter_Dyn_UDE()</a>	58
8.5.2.3	<a href="#">TimeConverter_Dyn_UDE() [2/2]</a>	58
8.5.3	<a href="#">Member Function Documentation</a>	58
8.5.3.1	<a href="#">convert_a_to_b()</a>	58
8.5.3.2	<a href="#">initialize()</a>	59
8.5.3.3	<a href="#">operator=()</a>	59

8.5.3.4	<code>reset_a_to_b_offset()</code>	59
8.5.4	Friends And Related Function Documentation	60
8.5.4.1	<code>init_attrjeod__TimeConverter_Dyn_UDE</code>	60
8.5.4.2	<code>InputProcessor</code>	60
8.5.5	Field Documentation	60
8.5.5.1	<code>dyn_ptr</code>	60
8.5.5.2	<code>ude_ptr</code>	60
8.6	<code>jeod::TimeConverter_STD_UDE</code> Class Reference	61
8.6.1	Detailed Description	62
8.6.2	Constructor & Destructor Documentation	62
8.6.2.1	<code>TimeConverter_STD_UDE()</code> [1/2]	62
8.6.2.2	<code>~TimeConverter_STD_UDE()</code>	62
8.6.2.3	<code>TimeConverter_STD_UDE()</code> [2/2]	62
8.6.3	Member Function Documentation	62
8.6.3.1	<code>convert_a_to_b()</code>	63
8.6.3.2	<code>convert_b_to_a()</code>	63
8.6.3.3	<code>initialize()</code>	63
8.6.3.4	<code>operator=()</code>	64
8.6.3.5	<code>reset_a_to_b_offset()</code>	64
8.6.4	Friends And Related Function Documentation	64
8.6.4.1	<code>init_attrjeod__TimeConverter_STD_UDE</code>	64
8.6.4.2	<code>InputProcessor</code>	65
8.6.5	Field Documentation	65
8.6.5.1	<code>failed_null_test</code>	65
8.6.5.2	<code>std_ptr</code>	65
8.6.5.3	<code>ude_ptr</code>	65
8.7	<code>jeod::TimeConverter_TAI_GPS</code> Class Reference	66
8.7.1	Detailed Description	67
8.7.2	Constructor & Destructor Documentation	67
8.7.2.1	<code>TimeConverter_TAI_GPS()</code> [1/2]	67



8.7.2.2	<a href="#">~TimeConverter_TAI_GPS()</a>	67
8.7.2.3	<a href="#">TimeConverter_TAI_GPS()</a> [2/2]	67
8.7.3	Member Function Documentation	67
8.7.3.1	<a href="#">convert_a_to_b()</a>	68
8.7.3.2	<a href="#">convert_b_to_a()</a>	68
8.7.3.3	<a href="#">initialize()</a>	68
8.7.3.4	<a href="#">operator=()</a>	69
8.7.4	Friends And Related Function Documentation	69
8.7.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_GPS</a>	69
8.7.4.2	<a href="#">InputProcessor</a>	69
8.7.5	Field Documentation	69
8.7.5.1	<a href="#">gps_ptr</a>	69
8.7.5.2	<a href="#">tai_ptr</a>	70
8.8	<a href="#">jeod::TimeConverter_TAI_TDB Class Reference</a>	70
8.8.1	Detailed Description	71
8.8.2	Constructor & Destructor Documentation	71
8.8.2.1	<a href="#">TimeConverter_TAI_TDB()</a> [1/2]	71
8.8.2.2	<a href="#">~TimeConverter_TAI_TDB()</a>	72
8.8.2.3	<a href="#">TimeConverter_TAI_TDB()</a> [2/2]	72
8.8.3	Member Function Documentation	72
8.8.3.1	<a href="#">convert_a_to_b()</a>	72
8.8.3.2	<a href="#">convert_b_to_a()</a>	72
8.8.3.3	<a href="#">initialize()</a>	72
8.8.3.4	<a href="#">operator=()</a>	73
8.8.3.5	<a href="#">set_a_to_b_offset()</a>	73
8.8.4	Friends And Related Function Documentation	73
8.8.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_TDB</a>	73
8.8.4.2	<a href="#">InputProcessor</a>	73
8.8.5	Field Documentation	74
8.8.5.1	<a href="#">a_to_b_offset</a>	74

8.8.5.2	<a href="#">a_to_b_offset_epoch</a>	74
8.8.5.3	<a href="#">nIter</a>	74
8.8.5.4	<a href="#">nSteps</a>	75
8.8.5.5	<a href="#">prev_tai_seconds</a>	75
8.8.5.6	<a href="#">prev_tdb_seconds</a>	75
8.8.5.7	<a href="#">tai_ptr</a>	75
8.8.5.8	<a href="#">TAI_to_TT_offset</a>	76
8.8.5.9	<a href="#">tdb_ptr</a>	76
8.9	<a href="#">jeod::TimeConverter_TAI_TT Class Reference</a>	76
8.9.1	<a href="#">Detailed Description</a>	77
8.9.2	<a href="#">Constructor &amp; Destructor Documentation</a>	77
8.9.2.1	<a href="#">TimeConverter_TAI_TT() [1/2]</a>	78
8.9.2.2	<a href="#">~TimeConverter_TAI_TT()</a>	78
8.9.2.3	<a href="#">TimeConverter_TAI_TT() [2/2]</a>	78
8.9.3	<a href="#">Member Function Documentation</a>	78
8.9.3.1	<a href="#">convert_a_to_b()</a>	78
8.9.3.2	<a href="#">convert_b_to_a()</a>	79
8.9.3.3	<a href="#">initialize()</a>	79
8.9.3.4	<a href="#">operator=()</a>	79
8.9.4	<a href="#">Friends And Related Function Documentation</a>	79
8.9.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_TT</a>	80
8.9.4.2	<a href="#">InputProcessor</a>	80
8.9.5	<a href="#">Field Documentation</a>	80
8.9.5.1	<a href="#">tai_ptr</a>	80
8.9.5.2	<a href="#">tt_ptr</a>	80
8.10	<a href="#">jeod::TimeConverter_TAI_UT1 Class Reference</a>	81
8.10.1	<a href="#">Detailed Description</a>	82
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	82
8.10.2.1	<a href="#">TimeConverter_TAI_UT1() [1/2]</a>	82
8.10.2.2	<a href="#">~TimeConverter_TAI_UT1()</a>	83

8.10.2.3	<a href="#">TimeConverter_TAI_UT1() [2/2]</a>	83
8.10.3	<a href="#">Member Function Documentation</a>	83
8.10.3.1	<a href="#">convert_a_to_b()</a>	83
8.10.3.2	<a href="#">convert_b_to_a()</a>	84
8.10.3.3	<a href="#">initialize()</a>	84
8.10.3.4	<a href="#">initialize_tai_to_ut1()</a>	85
8.10.3.5	<a href="#">operator=()</a>	85
8.10.3.6	<a href="#">verify_table_lookup_ends()</a>	85
8.10.4	<a href="#">Friends And Related Function Documentation</a>	86
8.10.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_UT1</a>	86
8.10.4.2	<a href="#">InputProcessor</a>	86
8.10.5	<a href="#">Field Documentation</a>	86
8.10.5.1	<a href="#">gradient</a>	86
8.10.5.2	<a href="#">index</a>	86
8.10.5.3	<a href="#">last_index</a>	87
8.10.5.4	<a href="#">next_value</a>	87
8.10.5.5	<a href="#">next_when</a>	87
8.10.5.6	<a href="#">off_table_end</a>	87
8.10.5.7	<a href="#">override_data_table</a>	88
8.10.5.8	<a href="#">prev_value</a>	88
8.10.5.9	<a href="#">prev_when</a>	88
8.10.5.10	<a href="#">tai_ptr</a>	88
8.10.5.11	<a href="#">tai_to_ut1_override_val</a>	89
8.10.5.12	<a href="#">ut1_ptr</a>	89
8.10.5.13	<a href="#">val_vec</a>	89
8.10.5.14	<a href="#">when_vec</a>	89
8.11	<a href="#">jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data Class Reference</a>	90
8.11.1	<a href="#">Detailed Description</a>	90
8.11.2	<a href="#">Member Function Documentation</a>	90
8.11.2.1	<a href="#">initialize()</a>	90

8.12	<a href="#">jeod::TimeConverter_TAI_UTC Class Reference</a>	90
8.12.1	<a href="#">Detailed Description</a>	92
8.12.2	<a href="#">Constructor &amp; Destructor Documentation</a>	92
8.12.2.1	<a href="#">TimeConverter_TAI_UTC() [1/2]</a>	92
8.12.2.2	<a href="#">~TimeConverter_TAI_UTC()</a>	92
8.12.2.3	<a href="#">TimeConverter_TAI_UTC() [2/2]</a>	92
8.12.3	<a href="#">Member Function Documentation</a>	93
8.12.3.1	<a href="#">convert_a_to_b()</a>	93
8.12.3.2	<a href="#">convert_b_to_a()</a>	93
8.12.3.3	<a href="#">initialize()</a>	93
8.12.3.4	<a href="#">initialize_leap_second()</a>	94
8.12.3.5	<a href="#">operator=()</a>	94
8.12.3.6	<a href="#">verify_table_lookup_ends()</a>	95
8.12.4	<a href="#">Friends And Related Function Documentation</a>	95
8.12.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_UTC</a>	95
8.12.4.2	<a href="#">InputProcessor</a>	95
8.12.5	<a href="#">Field Documentation</a>	95
8.12.5.1	<a href="#">index</a>	95
8.12.5.2	<a href="#">last_index</a>	96
8.12.5.3	<a href="#">leap_sec_override_val</a>	96
8.12.5.4	<a href="#">next_when</a>	96
8.12.5.5	<a href="#">off_table_end</a>	96
8.12.5.6	<a href="#">override_data_table</a>	97
8.12.5.7	<a href="#">prev_when</a>	97
8.12.5.8	<a href="#">tai_ptr</a>	97
8.12.5.9	<a href="#">utc_ptr</a>	97
8.12.5.10	<a href="#">val_vec</a>	98
8.12.5.11	<a href="#">when_vec</a>	98
8.13	<a href="#">jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data Class Reference</a>	98
8.13.1	<a href="#">Detailed Description</a>	98

8.13.2	Member Function Documentation	99
8.13.2.1	initialize()	99
8.14	jeod::TimeConverter_UT1_GMST Class Reference	99
8.14.1	Detailed Description	100
8.14.2	Constructor & Destructor Documentation	100
8.14.2.1	TimeConverter_UT1_GMST() [1/2]	100
8.14.2.2	~TimeConverter_UT1_GMST()	100
8.14.2.3	TimeConverter_UT1_GMST() [2/2]	101
8.14.3	Member Function Documentation	101
8.14.3.1	convert_a_to_b()	101
8.14.3.2	initialize()	101
8.14.3.3	operator=()	102
8.14.4	Friends And Related Function Documentation	102
8.14.4.1	init_attrjeod__TimeConverter_UT1_GMST	102
8.14.4.2	InputProcessor	102
8.14.5	Field Documentation	102
8.14.5.1	gmst_ptr	102
8.14.5.2	ut1_ptr	103
8.15	jeod::TimeDyn Class Reference	103
8.15.1	Detailed Description	104
8.15.2	Constructor & Destructor Documentation	104
8.15.2.1	TimeDyn() [1/2]	104
8.15.2.2	~TimeDyn()	105
8.15.2.3	TimeDyn() [2/2]	105
8.15.3	Member Function Documentation	105
8.15.3.1	initialize_initializer_time()	105
8.15.3.2	operator=()	106
8.15.3.3	update()	106
8.15.3.4	update_offset()	106
8.15.4	Friends And Related Function Documentation	106

8.15.4.1	<a href="#">init_attrjeod__TimeDyn</a>	107
8.15.4.2	<a href="#">InputProcessor</a>	107
8.15.5	<a href="#">Field Documentation</a>	107
8.15.5.1	<a href="#">offset</a>	107
8.15.5.2	<a href="#">ref_scale</a>	107
8.15.5.3	<a href="#">scale_factor</a>	108
8.16	<a href="#">jeod::TimeEnum Class Reference</a>	108
8.16.1	<a href="#">Detailed Description</a>	108
8.16.2	<a href="#">Member Enumeration Documentation</a>	108
8.16.2.1	<a href="#">TimeFormat</a>	108
8.17	<a href="#">jeod::TimeGMST Class Reference</a>	109
8.17.1	<a href="#">Detailed Description</a>	110
8.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	110
8.17.2.1	<a href="#">TimeGMST() [1/2]</a>	110
8.17.2.2	<a href="#">~TimeGMST()</a>	110
8.17.2.3	<a href="#">TimeGMST() [2/2]</a>	110
8.17.3	<a href="#">Member Function Documentation</a>	111
8.17.3.1	<a href="#">calculate_calendar_values()</a>	111
8.17.3.2	<a href="#">operator=()</a>	111
8.17.3.3	<a href="#">set_epoch()</a>	111
8.17.3.4	<a href="#">set_time_by_trunc_julian()</a>	111
8.17.4	<a href="#">Friends And Related Function Documentation</a>	112
8.17.4.1	<a href="#">init_attrjeod__TimeGMST</a>	112
8.17.4.2	<a href="#">InputProcessor</a>	112
8.18	<a href="#">jeod::TimeGPS Class Reference</a>	112
8.18.1	<a href="#">Detailed Description</a>	114
8.18.2	<a href="#">Constructor &amp; Destructor Documentation</a>	114
8.18.2.1	<a href="#">TimeGPS() [1/2]</a>	114
8.18.2.2	<a href="#">~TimeGPS()</a>	114
8.18.2.3	<a href="#">TimeGPS() [2/2]</a>	114

8.18.3	Member Function Documentation	114
8.18.3.1	calculate_calendar_values()	115
8.18.3.2	convert_from_calendar()	115
8.18.3.3	operator=()	115
8.18.3.4	set_epoch()	116
8.18.3.5	set_time_by_days()	116
8.18.3.6	set_time_by_seconds()	116
8.18.3.7	set_time_by_trunc_julian()	117
8.18.4	Friends And Related Function Documentation	117
8.18.4.1	init_attrjeod__TimeGPS	117
8.18.4.2	InputProcessor	118
8.18.5	Field Documentation	118
8.18.5.1	day_of_week	118
8.18.5.2	rollover_count	118
8.18.5.3	rollover_count_13_bit	118
8.18.5.4	seconds_of_day	119
8.18.5.5	seconds_of_week	119
8.18.5.6	week	119
8.18.5.7	week_13_bit	119
8.19	jeod::TimeLinks Class Reference	120
8.19.1	Detailed Description	120
8.19.2	Constructor & Destructor Documentation	120
8.19.2.1	TimeLinks() [1/3]	120
8.19.2.2	TimeLinks() [2/3]	121
8.19.2.3	TimeLinks() [3/3]	121
8.19.2.4	~TimeLinks()	121
8.19.3	Member Function Documentation	121
8.19.3.1	operator=()	121
8.19.4	Friends And Related Function Documentation	121
8.19.4.1	init_attrjeod__TimeLinks	121

8.19.4.2	InputProcessor	121
8.19.5	Field Documentation	122
8.19.5.1	default_path_size	122
8.20	jeod::TimeManager Class Reference	122
8.20.1	Detailed Description	124
8.20.2	Constructor & Destructor Documentation	124
8.20.2.1	TimeManager() [1/2]	124
8.20.2.2	~TimeManager()	124
8.20.2.3	TimeManager() [2/2]	124
8.20.3	Member Function Documentation	124
8.20.3.1	get_converter_ptr()	124
8.20.3.2	get_jeod_integration_time()	125
8.20.3.3	get_time_change_flag()	125
8.20.3.4	get_time_ptr() [1/2]	125
8.20.3.5	get_time_ptr() [2/2]	126
8.20.3.6	get_time_scale_factor()	126
8.20.3.7	get_timestamp_time()	127
8.20.3.8	initialize()	127
8.20.3.9	operator=()	127
8.20.3.10	register_converter()	127
8.20.3.11	register_time()	128
8.20.3.12	register_time_named()	128
8.20.3.13	time_lookup()	129
8.20.3.14	time_standards_exist()	129
8.20.3.15	update()	130
8.20.3.16	update_time()	131
8.20.3.17	verify_table_lookup_ends()	131
8.20.4	Friends And Related Function Documentation	131
8.20.4.1	init_attrjeod__TimeManager	132
8.20.4.2	InputProcessor	132



8.20.4.3	<a href="#">TimeManagerInit</a>	132
8.20.5	<a href="#">Field Documentation</a>	132
8.20.5.1	<a href="#">converter_vector</a>	132
8.20.5.2	<a href="#">dyn_time</a>	132
8.20.5.3	<a href="#">num_types</a>	133
8.20.5.4	<a href="#">simtime</a>	133
8.20.5.5	<a href="#">time_change_flag</a>	133
8.20.5.6	<a href="#">time_vector</a>	134
8.21	<a href="#">jeod::TimeManagerInit Class Reference</a>	134
8.21.1	<a href="#">Detailed Description</a>	136
8.21.2	<a href="#">Constructor &amp; Destructor Documentation</a>	136
8.21.2.1	<a href="#">TimeManagerInit() [1/2]</a>	136
8.21.2.2	<a href="#">~TimeManagerInit()</a>	136
8.21.2.3	<a href="#">TimeManagerInit() [2/2]</a>	136
8.21.3	<a href="#">Member Function Documentation</a>	137
8.21.3.1	<a href="#">create_init_tree()</a>	137
8.21.3.2	<a href="#">create_update_tree()</a>	137
8.21.3.3	<a href="#">get_conv_dir_init()</a>	138
8.21.3.4	<a href="#">get_conv_dir_upd()</a>	138
8.21.3.5	<a href="#">get_conv_ptr_index()</a>	139
8.21.3.6	<a href="#">get_status()</a>	139
8.21.3.7	<a href="#">increment_status()</a>	140
8.21.3.8	<a href="#">initialize()</a>	140
8.21.3.9	<a href="#">initialize_manager()</a>	141
8.21.3.10	<a href="#">initialize_time_types()</a>	141
8.21.3.11	<a href="#">operator=()</a>	142
8.21.3.12	<a href="#">organize_update_list()</a>	142
8.21.3.13	<a href="#">populate_converter_registry()</a>	142
8.21.3.14	<a href="#">set_status()</a>	142
8.21.3.15	<a href="#">verify_converter_setup()</a>	143

8.21.3.16	verify_times_setup()	143
8.21.4	Friends And Related Function Documentation	144
8.21.4.1	init_attrjeod__TimeManagerInit	144
8.21.4.2	InputProcessor	144
8.21.5	Field Documentation	144
8.21.5.1	converter_ptrs_index	144
8.21.5.2	dyn_time_index	144
8.21.5.3	init_converter_dir_table	145
8.21.5.4	initializer	145
8.21.5.5	initializer_index	145
8.21.5.6	num_added_pass	145
8.21.5.7	num_added_total	146
8.21.5.8	sim_start_format	146
8.21.5.9	status	146
8.21.5.10	time_manager	147
8.21.5.11	update_converter_dir_table	147
8.22	jeod::TimeMessages Class Reference	147
8.22.1	Detailed Description	148
8.22.2	Constructor & Destructor Documentation	148
8.22.2.1	TimeMessages() [1/2]	148
8.22.2.2	TimeMessages() [2/2]	148
8.22.3	Member Function Documentation	148
8.22.3.1	operator=()	148
8.22.4	Friends And Related Function Documentation	148
8.22.4.1	init_attrjeod__TimeMessages	149
8.22.4.2	InputProcessor	149
8.22.5	Field Documentation	149
8.22.5.1	duplicate_methods	149
8.22.5.2	extension_error	149
8.22.5.3	incomplete_setup_error	150

8.22.5.4	<a href="#">initialization_error</a>	150
8.22.5.5	<a href="#">invalid_data_error</a>	151
8.22.5.6	<a href="#">invalid_node</a>	151
8.22.5.7	<a href="#">invalid_setup_error</a>	152
8.22.5.8	<a href="#">memory_error</a>	152
8.22.5.9	<a href="#">redundancy_error</a>	153
8.23	<a href="#">jeod::TimeMET Class Reference</a>	153
8.23.1	<a href="#">Detailed Description</a>	154
8.23.2	<a href="#">Constructor &amp; Destructor Documentation</a>	154
8.23.2.1	<a href="#">TimeMET() [1/2]</a>	154
8.23.2.2	<a href="#">~TimeMET()</a>	155
8.23.2.3	<a href="#">TimeMET() [2/2]</a>	155
8.23.3	<a href="#">Member Function Documentation</a>	155
8.23.3.1	<a href="#">operator=()</a>	155
8.23.3.2	<a href="#">update()</a>	155
8.23.4	<a href="#">Friends And Related Function Documentation</a>	155
8.23.4.1	<a href="#">init_attrjeod__TimeMET</a>	156
8.23.4.2	<a href="#">InputProcessor</a>	156
8.23.5	<a href="#">Field Documentation</a>	156
8.23.5.1	<a href="#">hold</a>	156
8.23.5.2	<a href="#">previous_hold</a>	156
8.24	<a href="#">jeod::TimeStandard Class Reference</a>	157
8.24.1	<a href="#">Detailed Description</a>	158
8.24.2	<a href="#">Constructor &amp; Destructor Documentation</a>	159
8.24.2.1	<a href="#">TimeStandard() [1/2]</a>	159
8.24.2.2	<a href="#">~TimeStandard()</a>	159
8.24.2.3	<a href="#">TimeStandard() [2/2]</a>	159
8.24.3	<a href="#">Member Function Documentation</a>	159
8.24.3.1	<a href="#">add_type_initialize()</a>	159
8.24.3.2	<a href="#">calculate_calendar_values()</a>	160

8.24.3.3	<code>calendar_update()</code>	160
8.24.3.4	<code>convert_from_calendar()</code>	161
8.24.3.5	<code>initialize_from_parent()</code>	161
8.24.3.6	<code>initialize_initializer_time()</code>	162
8.24.3.7	<code>julian_date_at_epoch()</code>	163
8.24.3.8	<code>operator=()</code>	163
8.24.3.9	<code>seconds_of_year()</code>	163
8.24.3.10	<code>set_epoch()</code>	164
8.24.3.11	<code>set_time_by_days()</code>	164
8.24.3.12	<code>set_time_by_seconds()</code>	164
8.24.3.13	<code>set_time_by_trunc_julian()</code>	165
8.24.4	Friends And Related Function Documentation	165
8.24.4.1	<code>init_attrjeod__TimeStandard</code>	166
8.24.4.2	<code>InputProcessor</code>	166
8.24.4.3	<code>TimeUDE</code>	166
8.24.5	Field Documentation	166
8.24.5.1	<code>calendar_day</code>	166
8.24.5.2	<code>calendar_hour</code>	166
8.24.5.3	<code>calendar_minute</code>	167
8.24.5.4	<code>calendar_month</code>	167
8.24.5.5	<code>calendar_second</code>	167
8.24.5.6	<code>calendar_year</code>	167
8.24.5.7	<code>julian_date</code>	168
8.24.5.8	<code>last_calendar_update</code>	168
8.24.5.9	<code>prev_julian_day</code>	168
8.24.5.10	<code>seconds_at_year_start</code>	168
8.24.5.11	<code>send_warning_pre_1968</code>	169
8.24.5.12	<code>tjt_at_epoch</code>	169
8.24.5.13	<code>tjt_jd_offset</code>	169
8.24.5.14	<code>tjt_mjt_offset</code>	169

8.24.5.15 trunc_julian_time . . . . .	170
8.24.5.16 year_of_last_soy . . . . .	170
8.25 jeod::TimeTAI Class Reference . . . . .	170
8.25.1 Detailed Description . . . . .	171
8.25.2 Constructor & Destructor Documentation . . . . .	171
8.25.2.1 TimeTAI() [1/2] . . . . .	171
8.25.2.2 ~TimeTAI() . . . . .	172
8.25.2.3 TimeTAI() [2/2] . . . . .	172
8.25.3 Member Function Documentation . . . . .	172
8.25.3.1 operator=() . . . . .	172
8.25.3.2 set_epoch() . . . . .	172
8.25.4 Friends And Related Function Documentation . . . . .	172
8.25.4.1 init_attrjeod__TimeTAI . . . . .	173
8.25.4.2 InputProcessor . . . . .	173
8.26 jeod::TimeTDB Class Reference . . . . .	173
8.26.1 Detailed Description . . . . .	174
8.26.2 Constructor & Destructor Documentation . . . . .	174
8.26.2.1 TimeTDB() [1/2] . . . . .	174
8.26.2.2 ~TimeTDB() . . . . .	174
8.26.2.3 TimeTDB() [2/2] . . . . .	174
8.26.3 Member Function Documentation . . . . .	175
8.26.3.1 operator=() . . . . .	175
8.26.3.2 set_epoch() . . . . .	175
8.26.4 Friends And Related Function Documentation . . . . .	175
8.26.4.1 init_attrjeod__TimeTDB . . . . .	175
8.26.4.2 InputProcessor . . . . .	175
8.27 jeod::TimeTT Class Reference . . . . .	176
8.27.1 Detailed Description . . . . .	176
8.27.2 Constructor & Destructor Documentation . . . . .	176
8.27.2.1 TimeTT() [1/2] . . . . .	177

8.27.2.2	<a href="#">~TimeTT()</a>	177
8.27.2.3	<a href="#">TimeTT()</a> [2/2]	177
8.27.3	<a href="#">Member Function Documentation</a>	177
8.27.3.1	<a href="#">operator=()</a>	177
8.27.3.2	<a href="#">set_epoch()</a>	177
8.27.4	<a href="#">Friends And Related Function Documentation</a>	178
8.27.4.1	<a href="#">init_attrjeod__TimeTT</a>	178
8.27.4.2	<a href="#">InputProcessor</a>	178
8.28	<a href="#">jeod::TimeUDE Class Reference</a>	178
8.28.1	<a href="#">Detailed Description</a>	181
8.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	181
8.28.2.1	<a href="#">TimeUDE()</a> [1/2]	181
8.28.2.2	<a href="#">~TimeUDE()</a>	181
8.28.2.3	<a href="#">TimeUDE()</a> [2/2]	181
8.28.3	<a href="#">Member Function Documentation</a>	182
8.28.3.1	<a href="#">add_type_initialize()</a>	182
8.28.3.2	<a href="#">clock_update()</a>	182
8.28.3.3	<a href="#">convert_epoch_to_update()</a>	183
8.28.3.4	<a href="#">initialize_from_parent()</a>	183
8.28.3.5	<a href="#">initialize_initializer_time()</a>	184
8.28.3.6	<a href="#">must_be_singleton()</a>	184
8.28.3.7	<a href="#">operator=()</a>	185
8.28.3.8	<a href="#">set_epoch_dyn()</a>	185
8.28.3.9	<a href="#">set_epoch_initializing_value()</a>	186
8.28.3.10	<a href="#">set_epoch_std()</a>	186
8.28.3.11	<a href="#">set_epoch_times()</a>	187
8.28.3.12	<a href="#">set_epoch_ude()</a>	187
8.28.3.13	<a href="#">set_initial_times()</a>	188
8.28.3.14	<a href="#">set_time_by_clock()</a>	188
8.28.3.15	<a href="#">set_time_by_days()</a>	188

8.28.3.16	set_time_by_seconds()	189
8.28.3.17	verify_epoch()	189
8.28.3.18	verify_init()	190
8.28.3.19	verify_update()	190
8.28.4	Friends And Related Function Documentation	190
8.28.4.1	init_attrjeod__TimeUDE	190
8.28.4.2	InputProcessor	190
8.28.5	Field Documentation	191
8.28.5.1	clock_day	191
8.28.5.2	clock_hour	191
8.28.5.3	clock_minute	191
8.28.5.4	clock_second	192
8.28.5.5	epoch_data_present	192
8.28.5.6	epoch_day	192
8.28.5.7	epoch_defined_in_name	192
8.28.5.8	epoch_format	193
8.28.5.9	epoch_hour	193
8.28.5.10	epoch_index	193
8.28.5.11	epoch_initializing_value	193
8.28.5.12	epoch_minute	194
8.28.5.13	epoch_month	194
8.28.5.14	epoch_second	194
8.28.5.15	epoch_value_is_set_calendar	194
8.28.5.16	epoch_value_is_set_clock	195
8.28.5.17	epoch_value_is_set_number	195
8.28.5.18	epoch_year	195
8.28.5.19	initial_value_format	195
8.28.5.20	initializing_data_present	196
8.28.5.21	last_clock_update	196
8.28.5.22	update_index	196

8.29 jeod::TimeUT1 Class Reference	197
8.29.1 Detailed Description	197
8.29.2 Constructor & Destructor Documentation	198
8.29.2.1 TimeUT1() [1/2]	198
8.29.2.2 ~TimeUT1()	198
8.29.2.3 TimeUT1() [2/2]	198
8.29.3 Member Function Documentation	198
8.29.3.1 get_days()	198
8.29.3.2 operator=()	199
8.29.3.3 set_epoch()	199
8.29.4 Friends And Related Function Documentation	199
8.29.4.1 init_attrjeod__TimeUT1	199
8.29.4.2 InputProcessor	199
8.29.5 Field Documentation	199
8.29.5.1 true_ut1	200
8.30 jeod::TimeUTC Class Reference	200
8.30.1 Detailed Description	201
8.30.2 Constructor & Destructor Documentation	201
8.30.2.1 TimeUTC() [1/2]	201
8.30.2.2 ~TimeUTC()	201
8.30.2.3 TimeUTC() [2/2]	202
8.30.3 Member Function Documentation	202
8.30.3.1 operator=()	202
8.30.3.2 set_epoch()	202
8.30.4 Friends And Related Function Documentation	202
8.30.4.1 init_attrjeod__TimeUTC	202
8.30.4.2 InputProcessor	202
8.30.5 Field Documentation	203
8.30.5.1 true_utc	203



<b>9 File Documentation</b>	<b>205</b>
9.1 class_declarations.hh File Reference	205
9.1.1 Detailed Description	205
9.2 tai_to_ut1.cc File Reference	205
9.2.1 Macro Definition Documentation	206
9.2.1.1 JEOD_FRIEND_CLASS	206
9.3 tai_to_ut1.hh File Reference	206
9.4 tai_to_utc.cc File Reference	206
9.4.1 Macro Definition Documentation	206
9.4.1.1 JEOD_FRIEND_CLASS	207
9.5 tai_to_utc.hh File Reference	207
9.6 time.cc File Reference	207
9.6.1 Detailed Description	207
9.7 time.hh File Reference	207
9.7.1 Detailed Description	208
9.8 time__add_type_update.cc File Reference	208
9.8.1 Detailed Description	208
9.9 time_converter.cc File Reference	209
9.9.1 Detailed Description	209
9.10 time_converter.hh File Reference	209
9.10.1 Detailed Description	210
9.11 time_converter_dyn_tai.cc File Reference	210
9.11.1 Detailed Description	210
9.12 time_converter_dyn_tai.hh File Reference	210
9.12.1 Detailed Description	211
9.13 time_converter_dyn_tdb.cc File Reference	211
9.13.1 Detailed Description	211
9.14 time_converter_dyn_tdb.hh File Reference	211
9.14.1 Detailed Description	212
9.15 time_converter_dyn_ude.cc File Reference	212

9.15.1 Detailed Description . . . . .	212
9.16 time_converter_dyn_ude.hh File Reference . . . . .	212
9.16.1 Detailed Description . . . . .	213
9.17 time_converter_std_ude.cc File Reference . . . . .	213
9.17.1 Detailed Description . . . . .	213
9.18 time_converter_std_ude.hh File Reference . . . . .	213
9.18.1 Detailed Description . . . . .	214
9.19 time_converter_tai_gps.cc File Reference . . . . .	214
9.19.1 Detailed Description . . . . .	214
9.20 time_converter_tai_gps.hh File Reference . . . . .	214
9.20.1 Detailed Description . . . . .	215
9.21 time_converter_tai_tdb.cc File Reference . . . . .	215
9.21.1 Detailed Description . . . . .	215
9.22 time_converter_tai_tdb.hh File Reference . . . . .	215
9.22.1 Detailed Description . . . . .	216
9.23 time_converter_tai_tt.cc File Reference . . . . .	216
9.23.1 Detailed Description . . . . .	216
9.24 time_converter_tai_tt.hh File Reference . . . . .	216
9.24.1 Detailed Description . . . . .	217
9.25 time_converter_tai_ut1.cc File Reference . . . . .	217
9.25.1 Detailed Description . . . . .	217
9.26 time_converter_tai_ut1.hh File Reference . . . . .	217
9.26.1 Detailed Description . . . . .	218
9.27 time_converter_tai_utc.cc File Reference . . . . .	218
9.27.1 Detailed Description . . . . .	218
9.28 time_converter_tai_utc.hh File Reference . . . . .	218
9.28.1 Detailed Description . . . . .	219
9.29 time_converter_ut1_gmst.cc File Reference . . . . .	219
9.29.1 Detailed Description . . . . .	219
9.30 time_converter_ut1_gmst.hh File Reference . . . . .	219

9.30.1 Detailed Description . . . . .	220
9.31 time_dyn.cc File Reference . . . . .	220
9.31.1 Detailed Description . . . . .	220
9.32 time_dyn.hh File Reference . . . . .	220
9.32.1 Detailed Description . . . . .	221
9.33 time_enum.hh File Reference . . . . .	221
9.33.1 Detailed Description . . . . .	221
9.34 time_gmst.cc File Reference . . . . .	221
9.34.1 Detailed Description . . . . .	222
9.35 time_gmst.hh File Reference . . . . .	222
9.35.1 Detailed Description . . . . .	222
9.36 time_gps.cc File Reference . . . . .	222
9.36.1 Detailed Description . . . . .	223
9.37 time_gps.hh File Reference . . . . .	223
9.37.1 Detailed Description . . . . .	223
9.38 time_links.hh File Reference . . . . .	223
9.38.1 Detailed Description . . . . .	224
9.39 time_manager.cc File Reference . . . . .	224
9.39.1 Detailed Description . . . . .	224
9.40 time_manager.hh File Reference . . . . .	224
9.40.1 Detailed Description . . . . .	225
9.41 time_manager__initialize.cc File Reference . . . . .	225
9.41.1 Detailed Description . . . . .	225
9.42 time_manager_init.cc File Reference . . . . .	225
9.42.1 Detailed Description . . . . .	226
9.43 time_manager_init.hh File Reference . . . . .	226
9.43.1 Detailed Description . . . . .	226
9.44 time_messages.cc File Reference . . . . .	226
9.44.1 Detailed Description . . . . .	227
9.45 time_messages.hh File Reference . . . . .	227

9.45.1 Detailed Description . . . . .	227
9.46 time_met.cc File Reference . . . . .	227
9.46.1 Detailed Description . . . . .	228
9.47 time_met.hh File Reference . . . . .	228
9.47.1 Detailed Description . . . . .	228
9.48 time_standard.cc File Reference . . . . .	228
9.48.1 Detailed Description . . . . .	229
9.49 time_standard.hh File Reference . . . . .	229
9.49.1 Detailed Description . . . . .	229
9.50 time_tai.cc File Reference . . . . .	229
9.50.1 Detailed Description . . . . .	229
9.51 time_tai.hh File Reference . . . . .	230
9.51.1 Detailed Description . . . . .	230
9.52 time_tdb.cc File Reference . . . . .	230
9.52.1 Detailed Description . . . . .	230
9.53 time_tdb.hh File Reference . . . . .	231
9.53.1 Detailed Description . . . . .	231
9.54 time_tt.cc File Reference . . . . .	231
9.54.1 Detailed Description . . . . .	231
9.55 time_tt.hh File Reference . . . . .	232
9.55.1 Detailed Description . . . . .	232
9.56 time_ude.cc File Reference . . . . .	232
9.56.1 Detailed Description . . . . .	232
9.57 time_ude.hh File Reference . . . . .	233
9.57.1 Detailed Description . . . . .	233
9.58 time_ut1.cc File Reference . . . . .	233
9.58.1 Detailed Description . . . . .	233
9.59 time_ut1.hh File Reference . . . . .	234
9.59.1 Detailed Description . . . . .	234
9.60 time_utc.cc File Reference . . . . .	234
9.60.1 Detailed Description . . . . .	234
9.61 time_utc.hh File Reference . . . . .	235
9.61.1 Detailed Description . . . . .	235

# 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:

<a href="#">jeod</a>	Namespace jeod . . . . .	<a href="#">19</a>
----------------------	--------------------------	--------------------





## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::JeodBaseTime . . . . .	23
jeod::TimeDyn . . . . .	103
jeod::TimeStandard . . . . .	157
jeod::TimeGMST . . . . .	109
jeod::TimeGPS . . . . .	112
jeod::TimeTAI . . . . .	170
jeod::TimeTDB . . . . .	173
jeod::TimeTT . . . . .	176
jeod::TimeUT1 . . . . .	197
jeod::TimeUTC . . . . .	200
jeod::TimeUDE . . . . .	178
jeod::TimeMET . . . . .	153
JeodIntegrationTime	
jeod::TimeManager . . . . .	122
jeod::TimeConverter . . . . .	39
jeod::TimeConverter_Dyn_TAI . . . . .	48
jeod::TimeConverter_Dyn_TDB . . . . .	52
jeod::TimeConverter_Dyn_UDE . . . . .	56
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 . . . . .	99
jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data . . . . .	90
jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data . . . . .	98
jeod::TimeEnum . . . . .	108
jeod::TimeManagerInit . . . . .	134
jeod::TimeMessages . . . . .	147
TreeLinks	
jeod::TimeLinks . . . . .	120



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::JeodBaseTime</a>	
<a href="#">JeodBaseTime</a> is an abstract class, containing the basic structure of all clocks that run in JEOD	23
<a href="#">jeod::TimeConverter</a>	
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
<a href="#">jeod::TimeConverter_Dyn_TAI</a>	
Define class <a href="#">TimeConverter_Dyn_TAI</a> , which converts from simulation dynamic time to International Atomic Time	48
<a href="#">jeod::TimeConverter_Dyn_TDB</a>	
Define class <a href="#">TimeConverter_Dyn_TDB</a> , which converts from simulation dynamic time to Barycentric Dynamic Time	52
<a href="#">jeod::TimeConverter_Dyn_UDE</a>	
Define class <a href="#">TimeConverter_Dyn_UDE</a> , which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time	56
<a href="#">jeod::TimeConverter_STD_UDE</a>	
Define class <a href="#">TimeConverter_STD_UDE</a> , which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time	61
<a href="#">jeod::TimeConverter_TAI_GPS</a>	
Define class <a href="#">TimeConverter_TAI_GPS</a> , which converts between International Atomic Time and the clock associated with the Global Positioning System	66
<a href="#">jeod::TimeConverter_TAI_TDB</a>	
Define class <a href="#">TimeConverter_TAI_TDB</a> , which converts from International Atomic Time to Barycentric Dynamic Time	70
<a href="#">jeod::TimeConverter_TAI_TT</a>	
Converts between International Atomic Time and Terrestrial Time	76
<a href="#">jeod::TimeConverter_TAI_UT1</a>	
Define class <a href="#">TimeConverter_TAI_UT1</a> , which converts between International Atomic Time and Universal Time	81
<a href="#">jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data</a>	90
<a href="#">jeod::TimeConverter_TAI_UTC</a>	
Converts between International Atomic Time and Coordinated Universal Time	90
<a href="#">jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data</a>	98
<a href="#">jeod::TimeConverter_UT1_GMST</a>	
Converts between Universal Time and Greenwich Mean Sidereal Time	99

<a href="#">jeod::TimeDyn</a>	Represents the Dynamic Time in the simulation . . . . .	103
<a href="#">jeod::TimeEnum</a>	Contains an enumeration of the formats in which time can be represented . . . . .	108
<a href="#">jeod::TimeGMST</a>	To represent the clock known as Greenwich Mean Sidereal Time . . . . .	109
<a href="#">jeod::TimeGPS</a>	To represent the time associated with the Global Positioning System . . . . .	112
<a href="#">jeod::TimeLinks</a>	. . . . .	120
<a href="#">jeod::TimeManager</a>	To manage the various time representations and the converters between them throughout the simulation . . . . .	122
<a href="#">jeod::TimeManagerInit</a>	To initialize the Time Manager . . . . .	134
<a href="#">jeod::TimeMessages</a>	Specify the message IDs used in the Time model . . . . .	147
<a href="#">jeod::TimeMET</a>	A type of UDE time that allows for deliberate holds, or pauses . . . . .	153
<a href="#">jeod::TimeStandard</a>	A class that serves as the base for all time representations that are well defined outside the simulation . . . . .	157
<a href="#">jeod::TimeTAI</a>	Represents International Atomic Time . . . . .	170
<a href="#">jeod::TimeTDB</a>	Represents Barycentric Dynamic Time . . . . .	173
<a href="#">jeod::TimeTT</a>	Represents Terrestrial Time . . . . .	176
<a href="#">jeod::TimeUDE</a>	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition . . . . .	178
<a href="#">jeod::TimeUT1</a>	Represents Universal Time . . . . .	197
<a href="#">jeod::TimeUTC</a>	Represents Coordinated Universal Time . . . . .	200

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">class_declarations.hh</a>	Forward declaration of classes defined in <a href="#">time.hh</a>	205
<a href="#">tai_to_ut1.cc</a>		205
<a href="#">tai_to_ut1.hh</a>		206
<a href="#">tai_to_utc.cc</a>		206
<a href="#">tai_to_utc.hh</a>		207
<a href="#">time.cc</a>	JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	207
<a href="#">time.hh</a>	JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	207
<a href="#">time__add_type_update.cc</a>	Define JeodBaseTime::add_type_update	208
<a href="#">time_converter.cc</a>	An abstract class that defines the basic structure of all the methods used by the converter objects	209
<a href="#">time_converter.hh</a>	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	209
<a href="#">time_converter_dyn_tai.cc</a>	Converts between International Atomic Time and Dynamic Time	210
<a href="#">time_converter_dyn_tai.hh</a>	Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time	210
<a href="#">time_converter_dyn_tdb.cc</a>	Converts between Dynamic Time and Barycentric Dynamic Time	211
<a href="#">time_converter_dyn_tdb.hh</a>	Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time	211
<a href="#">time_converter_dyn_ude.cc</a>	Converts between Dynamic Time and a time with User-Defined-Epoch	212
<a href="#">time_converter_dyn_ude.hh</a>	Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time	212
<a href="#">time_converter_std_ude.cc</a>	Define member functions for class TimeConverter_STD_UDE	213

<a href="#">time_converter_std_ude.hh</a>	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 . . . . .	213
<a href="#">time_converter_tai_gps.cc</a>	Converts between International Atomic Time and the clock associated with the Global Positioning System . . . . .	214
<a href="#">time_converter_tai_gps.hh</a>	Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System . . . . .	214
<a href="#">time_converter_tai_tdb.cc</a>	Converts from International Atomic Time to Barycentric Dynamic Time . . . . .	215
<a href="#">time_converter_tai_tdb.hh</a>	Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time . . . . .	215
<a href="#">time_converter_tai_tt.cc</a>	Converts between International Atomic Time and Terrestrial Time . . . . .	216
<a href="#">time_converter_tai_tt.hh</a>	Converts between International Atomic Time and Terrestrial Time . . . . .	216
<a href="#">time_converter_tai_ut1.cc</a>	Converts between International Atomic Time and Universal Time . . . . .	217
<a href="#">time_converter_tai_ut1.hh</a>	Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time . . . . .	217
<a href="#">time_converter_tai_utc.cc</a>	Converts between International Atomic Time and Coordinated Universal Time . . . . .	218
<a href="#">time_converter_tai_utc.hh</a>	Converts between International Atomic Time and Coordinated Universal Time . . . . .	218
<a href="#">time_converter_ut1_gmst.cc</a>	Define member functions for class TimeConverter_UT1_GMST . . . . .	219
<a href="#">time_converter_ut1_gmst.hh</a>	Converts between Universal Time and Greenwich Mean Sidereal Time . . . . .	219
<a href="#">time_dyn.cc</a>	Define member functions for Dynamic Time . . . . .	220
<a href="#">time_dyn.hh</a>	Represents the Dynamic Time in the simulation . . . . .	220
<a href="#">time_enum.hh</a>	Contains an enumeration of the formats in which time can be represented . . . . .	221
<a href="#">time_gmst.cc</a>	Define member functions for Greenwich Mean Sidereal Time . . . . .	221
<a href="#">time_gmst.hh</a>	To represent the clock known as Greenwich Mean Sidereal Time . . . . .	222
<a href="#">time_gps.cc</a>	Define member functions for the clock associated with the Global Positioning System . . . . .	222
<a href="#">time_gps.hh</a>	To represent the time associated with the Global Positioning System . . . . .	223
<a href="#">time_links.hh</a>	Define the class TimeLinks, which defines the hierarchy of JEOD time conversions . . . . .	223
<a href="#">time_manager.cc</a>	Define member functions for class TimeManager . . . . .	224
<a href="#">time_manager.hh</a>	To manage the various time representations and the converters between them throughout the simulation . . . . .	224
<a href="#">time_manager_initialize.cc</a>	Define TimeManager::initialize . . . . .	225
<a href="#">time_manager_init.cc</a>	Define member functions for the Time Manager Initialization . . . . .	225
<a href="#">time_manager_init.hh</a>	To initialize the Time Manager . . . . .	226

<a href="#">time_messages.cc</a>	Implement the class TimeMessages . . . . .	226
<a href="#">time_messages.hh</a>	Define the class TimeMessages, the class that specifies the message IDs used in the Time model . . . . .	227
<a href="#">time_met.cc</a>	Define member functions for Mission Elapsed Time . . . . .	227
<a href="#">time_met.hh</a>	A type of UDE time that allows for deliberate holds, or pauses . . . . .	228
<a href="#">time_standard.cc</a>	An abstract class, this defines the basic structure of member functions for all Standard Times . . . . .	228
<a href="#">time_standard.hh</a>	A class that serves as the base for all time representations that are well defined outside the simulation . . . . .	229
<a href="#">time_tai.cc</a>	Define member functions for International Atomic Time . . . . .	229
<a href="#">time_tai.hh</a>	Represents International Atomic Time . . . . .	230
<a href="#">time_tdb.cc</a>	Define member functions Barycentric Dynamic Time . . . . .	230
<a href="#">time_tdb.hh</a>	Represents Barycentric Dynamic Time . . . . .	231
<a href="#">time_tt.cc</a>	Define member functions for Terrestrial Time . . . . .	231
<a href="#">time_tt.hh</a>	Represents Terrestrial Time . . . . .	232
<a href="#">time_ude.cc</a>	Define member functions for those times with a User-Defined-Epoch . . . . .	232
<a href="#">time_ude.hh</a>	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition . . . . .	233
<a href="#">time_ut1.cc</a>	Define member functions for Universal Time . . . . .	233
<a href="#">time_ut1.hh</a>	Represents Universal Time . . . . .	234
<a href="#">time_utc.cc</a>	Define member functions for Coordinated Universal Time . . . . .	234
<a href="#">time_utc.hh</a>	Represents Coordinated Universal Time . . . . .	235





## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Environment](#)

#### 6.1.1 Detailed Description

## 6.2 Environment

### Modules

- [Time](#)

#### 6.2.1 Detailed Description

## 6.3 Time

### Files

- file [class\\_declarations.hh](#)  
*Forward declaration of classes defined in [time.hh](#).*
- file [time.hh](#)  
*JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*
- file [time\\_converter.hh](#)  
*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*
- file [time\\_converter\\_dyn\\_tai.hh](#)  
*Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.*
- file [time\\_converter\\_dyn\\_tdb.hh](#)  
*Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.*
- file [time\\_converter\\_dyn\\_ude.hh](#)  
*Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.*
- file [time\\_converter\\_std\\_ude.hh](#)  
*Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.*
- file [time\\_converter\\_tai\\_gps.hh](#)  
*Define class TimeConverter\_TAI\_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.*
- file [time\\_converter\\_tai\\_tdb.hh](#)  
*Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.*
- file [time\\_converter\\_tai\\_tt.hh](#)  
*Converts between International Atomic Time and Terrestrial Time.*
- file [time\\_converter\\_tai\\_ut1.hh](#)  
*Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.*
- file [time\\_converter\\_tai\\_utc.hh](#)  
*Converts between International Atomic Time and Coordinated Universal Time.*
- file [time\\_converter\\_ut1\\_gmst.hh](#)  
*Converts between Universal Time and Greenwich Mean Sidereal Time.*
- file [time\\_dyn.hh](#)  
*Represents the Dynamic Time in the simulation.*
- file [time\\_enum.hh](#)  
*Contains an enumeration of the formats in which time can be represented.*
- file [time\\_gmst.hh](#)  
*To represent the clock known as Greenwich Mean Sidereal Time.*
- file [time\\_gps.hh](#)  
*To represent the time associated with the Global Positioning System.*
- file [time\\_links.hh](#)  
*Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.*
- file [time\\_manager.hh](#)  
*To manage the various time representations and the converters between them throughout the simulation.*
- file [time\\_manager\\_init.hh](#)  
*To initialize the Time Manager.*
- file [time\\_messages.hh](#)  
*Define the class TimeMessages, the class that specifies the message IDs used in the Time model.*
- file [time\\_met.hh](#)

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

  - file [time\\_standard.hh](#)
- A class that serves as the base for all time representations that are well defined outside the simulation.*

  - file [time\\_tai.hh](#)
- Represents International Atomic Time.*

  - file [time\\_tdb.hh](#)
- Represents Barycentric Dynamic Time.*

  - file [time\\_tt.hh](#)
- Represents Terrestrial Time.*

  - file [time\\_ude.hh](#)
- Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.*

  - file [time\\_ut1.hh](#)
- Represents Universal Time.*

  - file [time\\_utc.hh](#)
- Represents Coordinated Universal Time.*

  - file [time.cc](#)
- JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*

  - file [time\\_\\_add\\_type\\_update.cc](#)
- Define JeodBaseTime::add\_type\_update.*

  - file [time\\_converter.cc](#)
- An abstract class that defines the basic structure of all the methods used by the converter objects.*

  - file [time\\_converter\\_dyn\\_tai.cc](#)
- Converts between International Atomic Time and Dynamic Time.*

  - file [time\\_converter\\_dyn\\_tdb.cc](#)
- Converts between Dynamic Time and Barycentric Dynamic Time.*

  - file [time\\_converter\\_dyn\\_ude.cc](#)
- Converts between Dynamic Time and a time with User-Defined-Epoch.*

  - file [time\\_converter\\_std\\_ude.cc](#)
- Define member functions for class TimeConverter\_STD\_UDE.*

  - file [time\\_converter\\_tai\\_gps.cc](#)
- Converts between International Atomic Time and the clock associated with the Global Positioning System.*

  - file [time\\_converter\\_tai\\_tdb.cc](#)
- Converts from International Atomic Time to Barycentric Dynamic Time.*

  - file [time\\_converter\\_tai\\_tt.cc](#)
- Converts between International Atomic Time and Terrestrial Time.*

  - file [time\\_converter\\_tai\\_ut1.cc](#)
- Converts between International Atomic Time and Universal Time.*

  - file [time\\_converter\\_tai\\_utc.cc](#)
- Converts between International Atomic Time and Coordinated Universal Time.*

  - file [time\\_converter\\_ut1\\_gmst.cc](#)
- Define member functions for class TimeConverter\_UT1\_GMST.*

  - file [time\\_dyn.cc](#)
- Define member functions for Dynamic Time.*

  - file [time\\_gmst.cc](#)
- Define member functions for Greenwich Mean Sidereal Time.*

  - file [time\\_gps.cc](#)
- Define member functions for the clock associated with the Global Positioning System.*

  - file [time\\_manager.cc](#)
- Define member functions for class TimeManager.*

  - file [time\\_manager\\_\\_initialize.cc](#)

- Define TimeManager::initialize.*
- file [time\\_manager\\_init.cc](#)  
*Define member functions for the Time Manager Initialization.*
- file [time\\_messages.cc](#)  
*Implement the class TimeMessages.*
- file [time\\_met.cc](#)  
*Define member functions for Mission Elapsed Time.*
- file [time\\_standard.cc](#)  
*An abstract class, this defines the basic structure of member functions for all Standard Times.*
- file [time\\_tai.cc](#)  
*Define member functions for International Atomic Time.*
- file [time\\_tdb.cc](#)  
*Define member functions Barycentric Dynamic Time.*
- file [time\\_tt.cc](#)  
*Define member functions for Terrestrial Time.*
- file [time\\_ude.cc](#)  
*Define member functions for those times with a User-Defined-Epoch.*
- file [time\\_ut1.cc](#)  
*Define member functions for Universal Time.*
- file [time\\_utc.cc](#)  
*Define member functions for Coordinated Universal Time.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

## Macros

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

### 6.3.1 Detailed Description

### 6.3.2 Macro Definition Documentation

#### 6.3.2.1 PATH

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

Definition at line 37 of file `time_messages.cc`.



## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [JeodBaseTime](#)  
*JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*
- class [TimeConverter](#)  
*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*
- class [TimeConverter\\_Dyn\\_TAI](#)  
*Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.*
- class [TimeConverter\\_Dyn\\_TDB](#)  
*Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.*
- class [TimeConverter\\_Dyn\\_UDE](#)  
*Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.*
- class [TimeConverter\\_STD\\_UDE](#)  
*Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.*
- class [TimeConverter\\_TAI\\_GPS](#)  
*Define class TimeConverter\_TAI\_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.*
- class [TimeConverter\\_TAI\\_TDB](#)  
*Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.*
- class [TimeConverter\\_TAI\\_TT](#)  
*Converts between International Atomic Time and Terrestrial Time.*
- class [TimeConverter\\_TAI\\_UT1](#)  
*Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.*
- class [TimeConverter\\_TAI\\_UT1\\_tai\\_to\\_ut1\\_default\\_data](#)
- class [TimeConverter\\_TAI\\_UTC](#)  
*Converts between International Atomic Time and Coordinated Universal Time.*
- class [TimeConverter\\_TAI\\_UTC\\_tai\\_to\\_utc\\_default\\_data](#)
- class [TimeConverter\\_UT1\\_GMST](#)

- Converts between Universal Time and Greenwich Mean Sidereal Time.*

  - class [TimeDyn](#)

*Represents the Dynamic Time in the simulation.*
  - class [TimeEnum](#)

*Contains an enumeration of the formats in which time can be represented.*
  - class [TimeGMST](#)

*To represent the clock known as Greenwich Mean Sidereal Time.*
  - class [TimeGPS](#)

*To represent the time associated with the Global Positioning System.*
  - class [TimeLinks](#)
  - class [TimeManager](#)

*To manage the various time representations and the converters between them throughout the simulation.*
  - class [TimeManagerInit](#)

*To initialize the Time Manager.*
  - class [TimeMessages](#)

*Specify the message IDs used in the Time model.*
  - class [TimeMET](#)

*A type of UDE time that allows for deliberate holds, or pauses.*
  - class [TimeStandard](#)

*A class that serves as the base for all time representations that are well defined outside the simulation.*
  - class [TimeTAI](#)

*Represents International Atomic Time.*
  - class [TimeTDB](#)

*Represents Barycentric Dynamic Time.*
  - class [TimeTT](#)

*Represents Terrestrial Time.*
  - class [TimeUDE](#)

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

*Represents Universal Time.*
  - class [TimeUTC](#)

*Represents Coordinated Universal Time.*

## Functions

- [TimeConverter::Direction operator|](#) ([TimeConverter::Direction](#) a, [TimeConverter::Direction](#) b)

*Bitwise or operator for combining multiple converter direction flags.*

### 7.1.1 Detailed Description

Namespace `jeod`.

Construct a `Time_MET`.

Namespace `jeod`

### 7.1.2 Function Documentation



### 7.1.2.1 operator" | ()

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

Bitwise or operator for combining multiple converter direction flags.

Definition at line 206 of file time\_converter.hh.



## Chapter 8

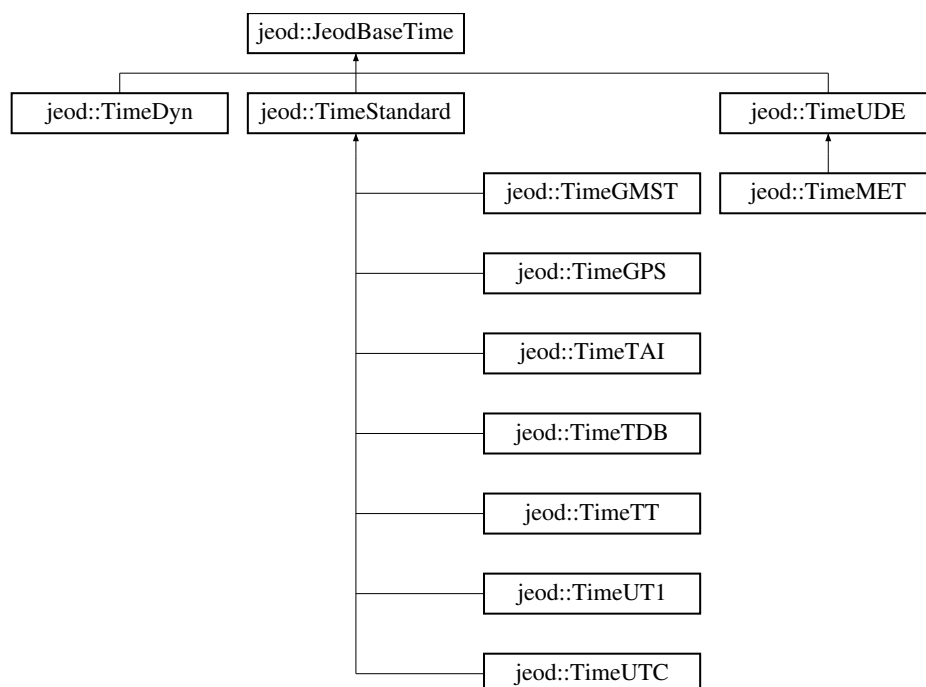
# Data Structure Documentation

### 8.1 jeod::JeodBaseTime Class Reference

[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.

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

Inheritance diagram for jeod::JeodBaseTime:



#### Public Member Functions

- [JeodBaseTime](#) ()  
*Construct a [JeodBaseTime](#).*
- virtual [~JeodBaseTime](#) ()  
*Destroy a [JeodBaseTime](#).*

- virtual bool `must_be_singleton` (void)  
*Virtual function that indicates if class must be a singleton Defaults to yes.*
- virtual void `set_time_by_seconds` (const double new\_seconds)  
*Given a value of seconds, propagate to days.*
- virtual void `set_time_by_days` (const double new\_days)  
*Given a value of days, propagate to seconds.*
- void `add_type_update` (const int seeking\_status, `TimeManagerInit` \*tm\_init)  
*Recursively adds elements to the update tree.*
- void `set_name` (std::string name\_in)  
*Setter for the name.*
- void `set_index` (int idx)  
*Setter for the index (force user to be carefule)*
- int `get_index` ()  
*Getter for the index.*
- void `override_initialized` (bool init)  
*Force reset the initialization status.*
- bool `is_initialized` ()  
*Read the initialization status.*
- virtual void `initialize_initializer_time` (`TimeManagerInit` \*tm\_init)=0  
*Initialize the time class that is used for initialization of the simulation.*
- virtual void `add_type_initialize` (const int seeking\_status, `TimeManagerInit` \*tm\_init)  
*Default attempt to add a time-type to the initialization tree.*
- virtual void `initialize_from_parent` (`TimeManagerInit` \*tm\_init)  
*Default attempt to initialize a time-type from its parent.*
- virtual void `update` (void)  
*Updates each of the derived times from its parent time.*

## Data Fields

- double `initializing_value`  
*Value used to define sim start time.*
- int `update_converter_direction`  
*Determines which converter function (a\_to\_b (+1) or b\_to\_a (-1)) to use.*
- double `seconds`  
*Elapsed time from epoch.*
- `NamedItem` `name`  
*Name of time-type.*
- std::string `initialize_from_name`  
*Name of time-type from which initial value is derived.*
- std::string `update_from_name`  
*Name of time-type from which update values are derived.*
- `TimeManager` \* `time_manager`  
*Pointer to the `TimeManager`.*
- `TimeConverter` \* `update_converter_ptr`  
*Pointer to the converter class needed to update the time.*

## Protected Member Functions

- void `add_parent` (`JeodBaseTime` &parent)  
*Link the argument time as the update source for this time.*

## Protected Attributes

- int [index](#)  
*Index-value of time-type in the registry.*
- bool [initialized](#)  
*Whether time has been initialized to a real time.*
- double [days](#)  
*Elapsed time from epoch.*
- double [initial\\_value](#)  
*Value of "seconds" at the start of the sim.*
- double [clock\\_resolution](#)  
*The resolution limit when generating clock and calendar-clock values.*
- [TimeLinks](#) [links](#)  
*Linkage to the hierarchy of time conversions.*

## Private Member Functions

- [JeodBaseTime](#) (const [JeodBaseTime](#) &)
- [JeodBaseTime](#) & [operator=](#) (const [JeodBaseTime](#) &)

## Friends

- class [InputProcessor](#)
- class [TimeConverter](#)
- class [TimeManagerInit](#)
- void [init\\_attrjeod\\_\\_JeodBaseTime](#) ()

### 8.1.1 Detailed Description

[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.

Definition at line 94 of file time.hh.

### 8.1.2 Constructor & Destructor Documentation

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

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

Construct a [JeodBaseTime](#).

Definition at line 58 of file time.cc.

References [clock\\_resolution](#), [days](#), [index](#), [initial\\_value](#), [initialize\\_from\\_name](#), [initialized](#), [initializing\\_value](#), [seconds](#), [time\\_manager](#), [update\\_converter\\_direction](#), [update\\_converter\\_ptr](#), and [update\\_from\\_name](#).

### 8.1.2.2 `~JeodBaseTime()`

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

Destroy a [JeodBaseTime](#).

Definition at line 213 of file time.cc.

References links.

### 8.1.2.3 `JeodBaseTime()` [2/2]

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

## 8.1.3 Member Function Documentation

### 8.1.3.1 `add_parent()`

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

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

#### Assumptions and Limitations

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

#### Parameters

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

Definition at line 112 of file time.cc.

References links.

Referenced by `add_type_update()`.

## 8.1.3.2 add\_type\_initialize()

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

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

## Assumptions and Limitations

- Fails for [TimeDyn](#), and has to be overwritten for others.

## Parameters

in	<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 91 of file time.cc.

References [jeod::TimeMessages::invalid\\_setup\\_error](#).

## 8.1.3.3 add\_type\_update()

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

Recursively adds elements to the update tree.

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

## Assumptions and Limitations

- None

## Parameters

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

Definition at line 72 of file time\_\_add\_type\_update.cc.

References [add\\_parent\(\)](#), [add\\_type\\_update\(\)](#), [jeod::TimeManagerInit::get\\_conv\\_dir\\_upd\(\)](#), [jeod::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 209 of file time.hh.

References index.

#### 8.1.3.5 initialize\_from\_parent()

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

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

#### Assumptions and Limitations

- Fails for [TimeDyn](#), and has to be overwritten for others.

#### Parameters

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

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

Definition at line 125 of file time.cc.

References jeod::TimeMessages::invalid\_setup\_error.

Referenced by jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), and jeod::↵  
 TimeManagerInit::initialize\_time\_types().



#### 8.1.3.6 initialize\_initializer\_time()

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

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

**Parameters**

<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 225 of file time.hh.

References initialized.

Referenced by [jeod::TimeConverter\\_TAI\\_UTC::initialize\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), and [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#).

**8.1.3.8 must\_be\_singleton()**

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

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

**Returns**

Boolean value

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

Definition at line 143 of file time.cc.

**8.1.3.9 operator=()**

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

#### 8.1.3.10 override\_initialized()

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

Force reset the initialization status.

Definition at line 217 of file time.hh.

References initialized.

Referenced by jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeUDE::initialize\_from\_parent(), and jeod::TimeUDE::initialize\_initializer\_time().

#### 8.1.3.11 set\_index()

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

Setter for the index (force user to be carefule)

Definition at line 201 of file time.hh.

References index.

Referenced by jeod::TimeManager::register\_time().

#### 8.1.3.12 set\_name()

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

Setter for the name.

Definition at line 193 of file time.hh.

References name.

#### 8.1.3.13 set\_time\_by\_days()

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

Given a value of days, propagate to seconds.

#### Assumptions and Limitations

- 86400 seconds = 1 day

**Parameters**

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

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

Definition at line 202 of file time.cc.

References days, and seconds.

Referenced by [jeod::TimeUDE::set\\_epoch\\_dyn\(\)](#), [jeod::TimeStandard::set\\_time\\_by\\_days\(\)](#), and [jeod::TimeUDE::set\\_time\\_by\\_days\(\)](#).

**8.1.3.14 set\_time\_by\_seconds()**

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

Given a value of seconds, propagate to days.

**Assumptions and Limitations**

- 86400 seconds = 1 day

**Parameters**

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

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

Definition at line 186 of file time.cc.

References days, and seconds.

Referenced by [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::set\\_epoch\\_dyn\(\)](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), and [jeod::TimeUDE::set\\_time\\_by\\_seconds\(\)](#).

**8.1.3.15 update()**

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

Updates each of the derived times from its parent time.

### Assumptions and Limitations

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

Reimplemented in [jeod::TimeDyn](#), and [jeod::TimeMET](#).

Definition at line 158 of file `time.cc`.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [jeod::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 96 of file `time.hh`.

### 8.1.4.3 `TimeConverter`

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

Definition at line 98 of file `time.hh`.

### 8.1.4.4 `TimeManagerInit`

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

Definition at line 99 of file `time.hh`.

## 8.1.5 Field Documentation

### 8.1.5.1 clock\_resolution

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

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

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

Definition at line 167 of file time.hh.

Referenced by jeod::TimeStandard::calculate\_calendar\_values(), jeod::TimeUDE::clock\_update(), and JeodBaseTime().

### 8.1.5.2 days

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

Elapsed time from epoch.

trick\_units(day)

Definition at line 155 of file time.hh.

Referenced by jeod::TimeStandard::convert\_from\_calendar(), jeod::TimeUT1::get\_days(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), jeod::TimeStandard::seconds\_of\_year(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeUDE::set\_time\_by\_clock(), set\_time\_by\_days(), jeod::TimeStandard::set\_time\_by\_days(), jeod::TimeGPS::set\_time\_by\_seconds(), set\_time\_by\_seconds(), jeod::TimeStandard::set\_time\_by\_seconds(), and jeod::TimeStandard::set\_time\_by\_trunc\_julian().

### 8.1.5.3 index

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

Index-value of time-type in the registry.

trick\_units(-)

Definition at line 147 of file time.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), add\_type\_update(), get\_index(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), set\_index(), and jeod::TimeUDE::verify\_epoch().

## 8.1.5.4 initial\_value

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

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

trick\_units(s)

Definition at line 160 of file time.hh.

Referenced by jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), and JeodBaseTime().

## 8.1.5.5 initialize\_from\_name

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

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

trick\_units(-)

Definition at line 127 of file time.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), JeodBaseTime(), and jeod::TimeUDE::verify\_init().

## 8.1.5.6 initialized

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

Whether time has been initialized to a real time.

trick\_units(-)

Definition at line 151 of file time.hh.

Referenced by jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeDyn::initialize\_initializer\_time(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeManagerInit::initialize\_time\_types(), is\_initialized(), JeodBaseTime(), override\_initialized(), and jeod::TimeConverter::verify\_setup().

### 8.1.5.7 initializing\_value

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

Value used to define sim start time.

trick\_units(-)

Definition at line 107 of file time.hh.

Referenced by jeod::TimeStandard::initialize\_initializer\_time(), JeodBaseTime(), and jeod::TimeUDE::set\_initial\_↵times().

### 8.1.5.8 links

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

Linkage to the hierarchy of time conversions.

Provides accessors to parent, siblings and childrentrick\_units(-)

Definition at line 173 of file time.hh.

Referenced by add\_parent(), add\_type\_update(), jeod::TimeDyn::TimeDyn(), and ~JeodBaseTime().

### 8.1.5.9 name

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

Name of time-type.

trick\_units(-)

Definition at line 122 of file time.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), add\_type\_↵update(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::Time↵Converter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeStandard::initialize\_from↵\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeU↵DE::initialize\_initializer\_time(), jeod::TimeManager::register\_time(), jeod::TimeManager::register\_time\_named(), jeod::TimeUDE::set\_epoch\_dyn(), jeod::TimeUDE::set\_epoch\_initializing\_value(), jeod::TimeUDE::set\_epoch\_↵std(), jeod::TimeUDE::set\_epoch\_uode(), jeod::TimeUDE::set\_initial\_times(), set\_name(), jeod::TimeDyn::Time↵Dyn(), jeod::TimeGMST::TimeGMST(), jeod::TimeGPS::TimeGPS(), jeod::TimeMET::TimeMET(), jeod::TimeT↵AI::TimeTAI(), jeod::TimeTDB::TimeTDB(), jeod::TimeTT::TimeTT(), jeod::TimeUT1::TimeUT1(), jeod::TimeUT↵C::TimeUTC(), update(), jeod::TimeUDE::verify\_epoch(), jeod::TimeUDE::verify\_init(), and jeod::TimeConverter↵::verify\_setup().



## 8.1.5.10 seconds

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

Elapsed time from epoch.

trick\_units(s)

Definition at line 117 of file time.hh.

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

## 8.1.5.11 time\_manager

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

Pointer to the [TimeManager](#).

trick\_units(-)

Definition at line 137 of file time.hh.

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

#### 8.1.5.12 `update_converter_direction`

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

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

`trick_units(-)`

Definition at line 112 of file `time.hh`.

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

#### 8.1.5.13 `update_converter_ptr`

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

Pointer to the converter class needed to update the time.

`trick_units(-)`

Definition at line 141 of file `time.hh`.

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

#### 8.1.5.14 `update_from_name`

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

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

`trick_units(-)`

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

Referenced by `jeod::TimeUDE::add_type_initialize()`, `add_type_update()`, `jeod::TimeUDE::convert_epoch_to_↵  
_update()`, `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeUDE↵  
::initialize_initializer_time()`, `JeodBaseTime()`, and `jeod::TimeUDE::verify_update()`.

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

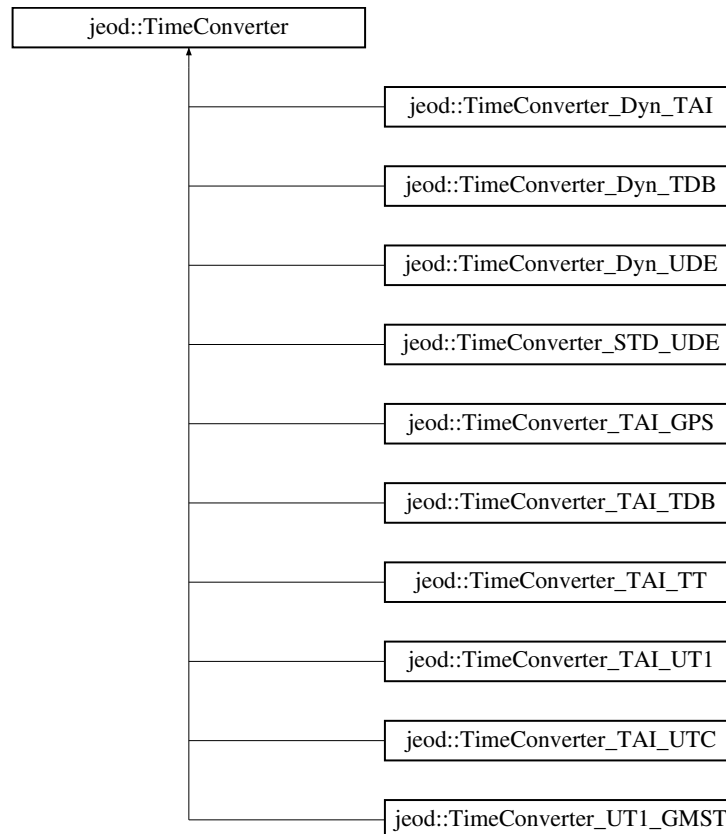
- [time.hh](#)
- [time.cc](#)
- [time\\_\\_add\\_type\\_update.cc](#)

## 8.2 jeod::TimeConverter Class Reference

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

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

Inheritance diagram for jeod::TimeConverter:



### Public Types

- enum [Direction](#) {  
[NO\\_DIRECTION](#) = 0x0000, [A\\_TO\\_B\\_INIT](#) = 0x0001, [B\\_TO\\_A\\_INIT](#) = 0x0010, [A\\_TO\\_B\\_UPDATE](#) = 0x0100,  
[B\\_TO\\_A\\_UPDATE](#) = 0x1000, [A\\_TO\\_B](#) = 0x0101, [B\\_TO\\_A](#) = 0x1010, [ANY\\_DIRECTION](#) = 0x1111 }

*Possible conversion directions.*

### Public Member Functions

- virtual [~TimeConverter](#) ()  
*Destroy a [TimeConverter](#).*
- virtual void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)=0  
*Initialize the converter.*
- virtual bool [is\\_initialized](#) (void)  
*Return internal initialized status bool.*
- void [override\\_initialized](#) (bool init)

- bool `can_convert` (`Direction` query)  
*Check whether this converter is able to handle the requested conversion(s).*
- virtual void `convert_a_to_b` (void)  
*Default converter from time 'a' to time 'b'.*
- virtual void `convert_b_to_a` (void)  
*Default converter from time 'b' to time 'a'.*
- virtual void `reset_a_to_b_offset` (void)  
*Resets the offset between type a and type b mid-sim.*
- virtual void `verify_table_lookup_ends` (void)  
*This function does absolutely nothing.*
- double `get_a_to_b_offset` (void)  
*Return the offset from the parent time object to this object.*

## Data Fields

- std::string `a_name`  
*name of time-type "a".*
- std::string `b_name`  
*name of time-type "b".*

## Protected Member Functions

- `TimeConverter` ()  
*Construct a `TimeConverter`.*
- void `verify_setup` (const `JeodBaseTime` \*parent, const `JeodBaseTime` \*child, const int direction)  
*Verify the setup.*

## Protected Attributes

- bool `initialized`  
*whether converter has been initialized.*
- double `a_to_b_offset`  
*Difference between the two time-types.*
- `Direction` `valid_directions`  
*Bit packed flag specifying whether how a converter can be used.*

## Private Member Functions

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

## Friends

- class `InputProcessor`
- class `JeodBaseTime`
- void `init_attrjeod__TimeConverter` ()

### 8.2.1 Detailed Description

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

Definition at line 89 of file `time_converter.hh`.

### 8.2.2 Member Enumeration Documentation

#### 8.2.2.1 Direction

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

Possible conversion directions.

Enumerator

NO_DIRECTION	
A_TO_B_INIT	
B_TO_A_INIT	
A_TO_B_UPDATE	
B_TO_A_UPDATE	
A_TO_B	
B_TO_A	
ANY_DIRECTION	

Definition at line 99 of file `time_converter.hh`.

### 8.2.3 Constructor & Destructor Documentation

#### 8.2.3.1 ~TimeConverter()

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

Destroy a [TimeConverter](#).

Definition at line 209 of file `time_converter.cc`.

**8.2.3.2 TimeConverter()** [1/2]

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

Construct a [TimeConverter](#).

Definition at line 53 of file time\_converter.cc.

References [a\\_name](#), [a\\_to\\_b\\_offset](#), [b\\_name](#), [initialized](#), [NO\\_DIRECTION](#), and [valid\\_directions](#).

**8.2.3.3 TimeConverter()** [2/2]

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

**8.2.4 Member Function Documentation****8.2.4.1 can\_convert()**

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

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

If query is compound (e.g. CONV\_ALL, CONV\_A\_TO\_B\_UPDATE|CONV\_B\_TO\_A\_UPDATE) then return true only if capable of all conversions

**Returns**

whether this converter can do all the conversions

**Parameters**

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

Definition at line 124 of file time\_converter.cc.

References [NO\\_DIRECTION](#), and [valid\\_directions](#).

**8.2.4.2 convert\_a\_to\_b()**

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

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

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

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), [jeod::TimeConverter\\_TAI\\_UTC](#), [jeod::TimeConverter\\_TAI\\_TDB](#), [jeod::TimeConverter\\_STD\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_TDB](#), [jeod::TimeConverter\\_Dyn\\_TAI](#), [jeod::TimeConverter\\_TAI\\_GPS](#), [jeod::TimeConverter\\_TAI\\_TT](#), and [jeod::TimeConverter\\_UT1\\_GMST](#).

Definition at line 150 of file `time_converter.cc`.

References [jeod::TimeMessages::invalid\\_setup\\_error](#).

Referenced by [jeod::TimeUDE::convert\\_epoch\\_to\\_update\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::initialize\\_initializer\\_time\(\)](#), and [jeod::JeodBaseTime::update\(\)](#).

#### 8.2.4.3 convert\_b\_to\_a()

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

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

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

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), [jeod::TimeConverter\\_TAI\\_UTC](#), [jeod::TimeConverter\\_TAI\\_TDB](#), [jeod::TimeConverter\\_STD\\_UDE](#), [jeod::TimeConverter\\_TAI\\_GPS](#), and [jeod::TimeConverter\\_TAI\\_TT](#).

Definition at line 166 of file `time_converter.cc`.

References [jeod::TimeMessages::invalid\\_setup\\_error](#).

Referenced by [jeod::TimeUDE::convert\\_epoch\\_to\\_update\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::initialize\\_initializer\\_time\(\)](#), and [jeod::JeodBaseTime::update\(\)](#).

#### 8.2.4.4 get\_a\_to\_b\_offset()

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

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

**Returns**

`a_to_b_offset` member.

Definition at line 183 of file `time_converter.hh`.

References [a\\_to\\_b\\_offset](#).

#### 8.2.4.5 initialize()

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

Initialize the converter.

## Parameters

in	<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 (
    void ) [virtual]
```

Return internal initialized status bool.

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

References initialized.

Referenced by [jeod::JeodBaseTime::add\\_type\\_update\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), and [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#).

8.2.4.7 `operator=()`

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

8.2.4.8 `override_initialized()`

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

Definition at line 159 of file `time_converter.hh`.

References initialized.

Referenced by [jeod::TimeUDE::convert\\_epoch\\_to\\_update\(\)](#).



## 8.2.4.9 reset\_a\_to\_b\_offset()

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

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

Reimplemented in [jeod::TimeConverter\\_STD\\_UDE](#), and [jeod::TimeConverter\\_Dyn\\_UDE](#).

Definition at line 180 of file `time_converter.cc`.

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

## 8.2.4.10 verify\_setup()

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

Verify the setup.

## Assumptions and Limitations

- None

## Parameters

in	<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 82 of file `time_converter.cc`.

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

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

## 8.2.4.11 verify\_table\_lookup\_ends()

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

This function does absolutely nothing.

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

#### Assumptions and Limitations

- None

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), and [jeod::TimeConverter\\_TAI\\_UTC](#).

Definition at line 200 of file `time_converter.cc`.

## 8.2.5 Friends And Related Function Documentation

### 8.2.5.1 `init_attrjeod__TimeConverter`

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

### 8.2.5.2 `InputProcessor`

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

Definition at line 91 of file `time_converter.hh`.

### 8.2.5.3 `JeodBaseTime`

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

Definition at line 93 of file `time_converter.hh`.

## 8.2.6 Field Documentation

### 8.2.6.1 a\_name

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

name of time-type "a".

trick\_units(—)

Definition at line 116 of file time\_converter.hh.

Referenced by jeod::TimeManager::register\_converter(), TimeConverter(), jeod::TimeConverter\_Dyn\_TAI::TimeConverter\_Dyn\_TAI(), jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB(), jeod::TimeConverter\_Dyn\_UDE::TimeConverter\_Dyn\_UDE(), jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE(), jeod::TimeConverter\_TAI\_GPS::TimeConverter\_TAI\_GPS(), jeod::TimeConverter\_TAI\_TDB::TimeConverter\_TAI\_TDB(), jeod::TimeConverter\_TAI\_TT::TimeConverter\_TAI\_TT(), jeod::TimeConverter\_TAI\_UT1::TimeConverter\_TAI\_UT1(), jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC(), and jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST().

### 8.2.6.2 a\_to\_b\_offset

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

Difference between the two time-types.

trick\_units(—)

Definition at line 131 of file time\_converter.hh.

Referenced by jeod::TimeConverter\_Dyn\_TAI::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_GPS::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_TDB::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_STD\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UTC::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UT1::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_GPS::convert\_b\_to\_a(), jeod::TimeConverter\_STD\_UDE::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UTC::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1::convert\_b\_to\_a(), get\_a\_to\_b\_offset(), jeod::TimeConverter\_TAI\_TT::initialize(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::TimeConverter\_TAI\_GPS::initialize(), jeod::TimeConverter\_Dyn\_TDB::initialize(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeConverter\_TAI\_UTC::initialize(), jeod::TimeConverter\_TAI\_UT1::initialize(), jeod::TimeConverter\_TAI\_UTC::initialize\_leap\_second(), jeod::TimeConverter\_TAI\_UT1::initialize\_tai\_to\_ut1(), jeod::TimeConverter\_Dyn\_UDE::reset\_a\_to\_b\_offset(), jeod::TimeConverter\_STD\_UDE::reset\_a\_to\_b\_offset(), and TimeConverter().

### 8.2.6.3 b\_name

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

name of time-type "b".

trick\_units(—)

Definition at line 120 of file time\_converter.hh.

Referenced by jeod::TimeManager::register\_converter(), TimeConverter(), jeod::TimeConverter\_Dyn\_TAI::TimeConverter\_Dyn\_TAI(), jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB(), jeod::TimeConverter\_Dyn\_UDE::TimeConverter\_Dyn\_UDE(), jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE(), jeod::TimeConverter\_TAI\_GPS::TimeConverter\_TAI\_GPS(), jeod::TimeConverter\_TAI\_TDB::TimeConverter\_TAI\_TDB(), jeod::TimeConverter\_TAI\_TT::TimeConverter\_TAI\_TT(), jeod::TimeConverter\_TAI\_UT1::TimeConverter\_TAI\_UT1(), jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC(), and jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST().

#### 8.2.6.4 initialized

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

whether converter has been initialized.

trick\_units(-)

Definition at line 127 of file time\_converter.hh.

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

#### 8.2.6.5 valid\_directions

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

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

Definition at line 135 of file time\_converter.hh.

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

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

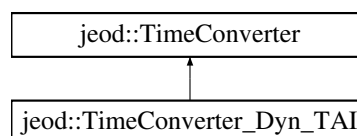
- [time\\_converter.hh](#)
- [time\\_converter.cc](#)

## 8.3 jeod::TimeConverter\_Dyn\_TAI Class Reference

Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.

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

Inheritance diagram for `jeod::TimeConverter_Dyn_TAI`:



## Public Member Functions

- [TimeConverter\\_Dyn\\_TAI](#) ()  
*Construct a [TimeConverter\\_Dyn\\_TAI](#).*
- [~TimeConverter\\_Dyn\\_TAI](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_TAI](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeTAI](#).*

## Private Member Functions

- [TimeConverter\\_Dyn\\_TAI](#) (const [TimeConverter\\_Dyn\\_TAI](#) &)
- [TimeConverter\\_Dyn\\_TAI](#) & operator= (const [TimeConverter\\_Dyn\\_TAI](#) &)

## Private Attributes

- [TimeDyn](#) \* dyn\_ptr  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeTAI](#) \* tai\_ptr  
*Converter child time, always a [TimeTAI](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_TAI](#) ()

## Additional Inherited Members

### 8.3.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.

Definition at line 88 of file `time_converter_dyn_tai.hh`.

### 8.3.2 Constructor & Destructor Documentation

**8.3.2.1 TimeConverter\_Dyn\_TAI()** [1/2]

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

Construct a [TimeConverter\\_Dyn\\_TAI](#).

Definition at line 59 of file `time_converter_dyn_tai.cc`.

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

**8.3.2.2 ~TimeConverter\_Dyn\_TAI()**

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

Destroy a [TimeConverter\\_Dyn\\_TAI](#).

Definition at line 181 of file `time_converter_dyn_tai.cc`.

**8.3.2.3 TimeConverter\_Dyn\_TAI()** [2/2]

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

**8.3.3 Member Function Documentation****8.3.3.1 convert\_a\_to\_b()**

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

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

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

Definition at line 165 of file `time_converter_dyn_tai.cc`.

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

**8.3.3.2 initialize()**

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

Initialize the converter.

## Parameters

in	<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 77 of file `time_converter_dyn_tai.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [dyn\\_ptr](#), [jeod::TimeMessages::initialization\\_error](#), [jeod::TimeConverter::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::name](#), [jeod::JeodBaseTime::seconds](#), [tai\\_ptr](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).

### 8.3.3.3 operator=()

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

## 8.3.4 Friends And Related Function Documentation

### 8.3.4.1 init\_attrjeod\_\_TimeConverter\_Dyn\_TAI

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

### 8.3.4.2 InputProcessor

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

Definition at line 91 of file `time_converter_dyn_tai.hh`.

## 8.3.5 Field Documentation

### 8.3.5.1 dyn\_ptr

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

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

trick\_units(-)

Definition at line 97 of file time\_converter\_dyn\_tai.hh.

Referenced by convert\_a\_to\_b(), initialize(), and TimeConverter\_Dyn\_TAI().

### 8.3.5.2 tai\_ptr

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

Converter child time, always a [TimeTAI](#) for this converter.

trick\_units(-)

Definition at line 102 of file time\_converter\_dyn\_tai.hh.

Referenced by convert\_a\_to\_b(), initialize(), and TimeConverter\_Dyn\_TAI().

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

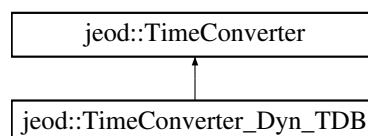
- [time\\_converter\\_dyn\\_tai.hh](#)
- [time\\_converter\\_dyn\\_tai.cc](#)

## 8.4 jeod::TimeConverter\_Dyn\_TDB Class Reference

Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.

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

Inheritance diagram for jeod::TimeConverter\_Dyn\_TDB:





## Public Member Functions

- [TimeConverter\\_Dyn\\_TDB](#) ()  
*Construct a [TimeConverter\\_Dyn\\_TDB](#).*
- [~TimeConverter\\_Dyn\\_TDB](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_TDB](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeTDB](#).*

## Private Member Functions

- [TimeConverter\\_Dyn\\_TDB](#) (const [TimeConverter\\_Dyn\\_TDB](#) &)
- [TimeConverter\\_Dyn\\_TDB](#) & [operator=](#) (const [TimeConverter\\_Dyn\\_TDB](#) &)

## Private Attributes

- [TimeDyn](#) \* [dyn\\_ptr](#)  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeTDB](#) \* [tdb\\_ptr](#)  
*Converter child time, always a [TimeTDB](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_TDB](#) ()

## Additional Inherited Members

### 8.4.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.

Definition at line 91 of file `time_converter_dyn_tdb.hh`.

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 TimeConverter\_Dyn\_TDB() [1/2]

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

Construct a [TimeConverter\\_Dyn\\_TDB](#).

Definition at line 59 of file `time_converter_dyn_tdb.cc`.

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

#### 8.4.2.2 ~TimeConverter\_Dyn\_TDB()

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

Destroy a [TimeConverter\\_Dyn\\_TDB](#).

Definition at line 151 of file `time_converter_dyn_tdb.cc`.

#### 8.4.2.3 TimeConverter\_Dyn\_TDB() [2/2]

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

### 8.4.3 Member Function Documentation

#### 8.4.3.1 convert\_a\_to\_b()

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

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

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

Definition at line 139 of file `time_converter_dyn_tdb.cc`.

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

#### 8.4.3.2 initialize()

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

Initialize the converter.

## Parameters

in	<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 77 of file `time_converter_dyn_tdb.cc`.

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

#### 8.4.3.3 operator=()

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

### 8.4.4 Friends And Related Function Documentation

#### 8.4.4.1 init\_attrjeod\_\_TimeConverter\_Dyn\_TDB

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

#### 8.4.4.2 InputProcessor

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

Definition at line 93 of file `time_converter_dyn_tdb.hh`.

### 8.4.5 Field Documentation

#### 8.4.5.1 dyn\_ptr

`TimeDyn*` jeod::TimeConverter\_Dyn\_TDB::dyn\_ptr [private]

Converter parent time, always a `TimeDyn` for this converter.

trick\_units(-)

Definition at line 99 of file time\_converter\_dyn\_tdb.hh.

Referenced by convert\_a\_to\_b(), initialize(), and TimeConverter\_Dyn\_TDB().

#### 8.4.5.2 tdb\_ptr

`TimeTDB*` jeod::TimeConverter\_Dyn\_TDB::tdb\_ptr [private]

Converter child time, always a `TimeTDB` for this converter.

trick\_units(-)

Definition at line 104 of file time\_converter\_dyn\_tdb.hh.

Referenced by convert\_a\_to\_b(), initialize(), and TimeConverter\_Dyn\_TDB().

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

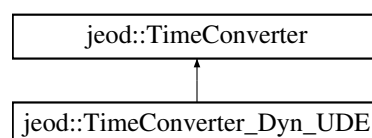
- [time\\_converter\\_dyn\\_tdb.hh](#)
- [time\\_converter\\_dyn\\_tdb.cc](#)

## 8.5 jeod::TimeConverter\_Dyn\_UDE Class Reference

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

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

Inheritance diagram for jeod::TimeConverter\_Dyn\_UDE:



## Public Member Functions

- [TimeConverter\\_Dyn\\_UDE](#) ()  
*Construct a [TimeConverter\\_Dyn\\_UDE](#).*
- [~TimeConverter\\_Dyn\\_UDE](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_UDE](#).*
- void [reset\\_a\\_to\\_b\\_offset](#) (void)  
*Resets the value of [a\\_to\\_b\\_offset](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeUDE](#).*

## Private Member Functions

- [TimeConverter\\_Dyn\\_UDE](#) (const [TimeConverter\\_Dyn\\_UDE](#) &)
- [TimeConverter\\_Dyn\\_UDE](#) & operator= (const [TimeConverter\\_Dyn\\_UDE](#) &)

## Private Attributes

- [TimeDyn](#) \* [dyn\\_ptr](#)  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeUDE](#) \* [ude\\_ptr](#)  
*Converter child time, always a [TimeUDE](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_UDE](#) ()

## Additional Inherited Members

### 8.5.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_UDE](#), which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

Definition at line 91 of file [time\\_converter\\_dyn\\_ude.hh](#).

### 8.5.2 Constructor & Destructor Documentation

### 8.5.2.1 TimeConverter\_Dyn\_UDE() [1/2]

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

Construct a [TimeConverter\\_Dyn\\_UDE](#).

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

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

### 8.5.2.2 ~TimeConverter\_Dyn\_UDE()

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

Destroy a [TimeConverter\\_Dyn\\_UDE](#).

Definition at line 164 of file `time_converter_dyn_ude.cc`.

### 8.5.2.3 TimeConverter\_Dyn\_UDE() [2/2]

```
jeod::TimeConverter_Dyn_UDE::TimeConverter_Dyn_UDE (
    const TimeConverter\_Dyn\_UDE & ) [private]
```

## 8.5.3 Member Function Documentation

### 8.5.3.1 convert\_a\_to\_b()

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

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

#### Assumptions and Limitations

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

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

Definition at line 140 of file `time_converter_dyn_ude.cc`.

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

### 8.5.3.2 initialize()

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

Initialize the converter.

#### Assumptions and Limitations

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

#### Parameters

in	<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 79 of file `time_converter_dyn_ude.cc`.

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

### 8.5.3.3 operator=()

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

### 8.5.3.4 reset\_a\_to\_b\_offset()

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

Resets the value of `a_to_b_offset`.

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

Definition at line 154 of file `time_converter_dyn_ude.cc`.

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

## 8.5.4 Friends And Related Function Documentation

### 8.5.4.1 init\_attrjeod\_\_TimeConverter\_Dyn\_UDE

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

### 8.5.4.2 InputProcessor

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

Definition at line 93 of file time\_converter\_dyn\_ude.hh.

## 8.5.5 Field Documentation

### 8.5.5.1 dyn\_ptr

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

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

trick\_units(-)

Definition at line 100 of file time\_converter\_dyn\_ude.hh.

Referenced by convert\_a\_to\_b(), initialize(), reset\_a\_to\_b\_offset(), and TimeConverter\_Dyn\_UDE().

### 8.5.5.2 ude\_ptr

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

Converter child time, always a [TimeUDE](#) for this converter.

trick\_units(-)

Definition at line 105 of file time\_converter\_dyn\_ude.hh.

Referenced by convert\_a\_to\_b(), initialize(), reset\_a\_to\_b\_offset(), and TimeConverter\_Dyn\_UDE().

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

- [time\\_converter\\_dyn\\_ude.hh](#)
- [time\\_converter\\_dyn\\_ude.cc](#)

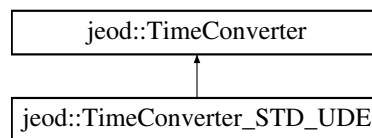


## 8.6 jeod::TimeConverter\_STD\_UDE Class Reference

Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

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

Inheritance diagram for jeod::TimeConverter\_STD\_UDE:



### Public Member Functions

- [TimeConverter\\_STD\\_UDE](#) ()  
*Construct a [TimeConverter\\_STD\\_UDE](#).*
- [~TimeConverter\\_STD\\_UDE](#) ()  
*Destroy a [TimeConverter\\_STD\\_UDE](#).*
- void [reset\\_a\\_to\\_b\\_offset](#) (void)  
*Resets the value of [a\\_to\\_b\\_offset](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeSTD](#) to [TimeUDE](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeUDE](#) to [TimeSTD](#).*

### Private Member Functions

- [TimeConverter\\_STD\\_UDE](#) (const [TimeConverter\\_STD\\_UDE](#) &)
- [TimeConverter\\_STD\\_UDE](#) & [operator=](#) (const [TimeConverter\\_STD\\_UDE](#) &)

### Private Attributes

- bool [failed\\_null\\_test](#)  
*Initializing converter could be done in one of two ways.*
- [TimeStandard](#) \* [std\\_ptr](#)  
*Converter parent time, always a [TimeSTD](#) for this converter.*
- [TimeUDE](#) \* [ude\\_ptr](#)  
*Converter parent time, always a [TimeUDE](#) for this converter.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_STD\\_UDE](#) ()

## Additional Inherited Members

### 8.6.1 Detailed Description

Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

Definition at line 91 of file `time_converter_std_ude.hh`.

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 TimeConverter\_STD\_UDE() [1/2]

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

Construct a [TimeConverter\\_STD\\_UDE](#).

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

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

#### 8.6.2.2 ~TimeConverter\_STD\_UDE()

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

Destroy a [TimeConverter\\_STD\\_UDE](#).

Definition at line 190 of file `time_converter_std_ude.cc`.

#### 8.6.2.3 TimeConverter\_STD\_UDE() [2/2]

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

### 8.6.3 Member Function Documentation

## 8.6.3.1 convert\_a\_to\_b()

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

Convert from TimeSTD to [TimeUDE](#).

## Assumptions and Limitations

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

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

Definition at line 147 of file time\_converter\_std\_ude.cc.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeUDE::set\\_time\\_by\\_↵seconds\(\)](#), [std\\_ptr](#), and [ude\\_ptr](#).

## 8.6.3.2 convert\_b\_to\_a()

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

Convert from [TimeUDE](#) to TimeSTD.

## Assumptions and Limitations

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

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

Definition at line 164 of file time\_converter\_std\_ude.cc.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeStandard::set\\_time\\_↵by\\_seconds\(\)](#), [std\\_ptr](#), and [ude\\_ptr](#).

## 8.6.3.3 initialize()

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

Initialize the converter.

## Assumptions and Limitations

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

**Parameters**

in	<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 80 of file `time_converter_std_ude.cc`.

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

**8.6.3.4 operator=()**

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

**8.6.3.5 reset\_a\_to\_b\_offset()**

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

Resets the value of `a_to_b_offset`.

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

Definition at line 177 of file `time_converter_std_ude.cc`.

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

**8.6.4 Friends And Related Function Documentation****8.6.4.1 init\_attrjeod\_\_TimeConverter\_STD\_UDE**

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

#### 8.6.4.2 InputProcessor

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

Definition at line 93 of file `time_converter_std_ude.hh`.

### 8.6.5 Field Documentation

#### 8.6.5.1 failed\_null\_test

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

Initializing converter could be done in one of two ways.

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

Definition at line 101 of file `time_converter_std_ude.hh`.

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

#### 8.6.5.2 std\_ptr

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

Converter parent time, always a `TimeSTD` for this converter.

`trick_units(-)`

Definition at line 105 of file `time_converter_std_ude.hh`.

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

#### 8.6.5.3 ude\_ptr

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

Converter parent time, always a `TimeUDE` for this converter.

`trick_units(-)`

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

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

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

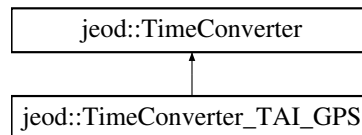
- [time\\_converter\\_std\\_ude.hh](#)
- [time\\_converter\\_std\\_ude.cc](#)

## 8.7 jeod::TimeConverter\_TAI\_GPS Class Reference

Define class [TimeConverter\\_TAI\\_GPS](#), which converts between International Atomic Time and the clock associated with the Global Positioning System.

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

Inheritance diagram for jeod::TimeConverter\_TAI\_GPS:



### Public Member Functions

- [TimeConverter\\_TAI\\_GPS](#) ()  
Construct a [TimeConverter\\_TAI\\_GPS](#).
- [~TimeConverter\\_TAI\\_GPS](#) ()  
Destroy a [TimeConverter\\_TAI\\_GPS](#).
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
Initialize the converter.
- void [convert\\_a\\_to\\_b](#) (void)  
Convert from [TimeTAI](#) to [TimeGPS](#).
- void [convert\\_b\\_to\\_a](#) (void)  
Convert from [TimeGPS](#) to [TimeTAI](#).

### Private Member Functions

- [TimeConverter\\_TAI\\_GPS](#) (const [TimeConverter\\_TAI\\_GPS](#) &)
- [TimeConverter\\_TAI\\_GPS](#) & operator= (const [TimeConverter\\_TAI\\_GPS](#) &)

### Private Attributes

- [TimeTAI](#) \* tai\_ptr  
Converter parent time, always a [TimeTAI](#) for this converter.
- [TimeGPS](#) \* gps\_ptr  
Converter parent time, always a [TimeGPS](#) for this converter.

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_GPS](#) ()

## Additional Inherited Members

### 8.7.1 Detailed Description

Define class [TimeConverter\\_TAI\\_GPS](#), which converts between International Atomic Time and the clock associated with the Global Positioning System.

Definition at line 88 of file `time_converter_tai_gps.hh`.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 TimeConverter\_TAI\_GPS() [1/2]

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

Construct a [TimeConverter\\_TAI\\_GPS](#).

Definition at line 59 of file `time_converter_tai_gps.cc`.

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

#### 8.7.2.2 ~TimeConverter\_TAI\_GPS()

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

Destroy a [TimeConverter\\_TAI\\_GPS](#).

Definition at line 137 of file `time_converter_tai_gps.cc`.

#### 8.7.2.3 TimeConverter\_TAI\_GPS() [2/2]

```
jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS (
    const TimeConverter\_TAI\_GPS & ) [private]
```

### 8.7.3 Member Function Documentation

### 8.7.3.1 convert\_a\_to\_b()

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

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

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

Definition at line 113 of file `time_converter_tai_gps.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gps\\_ptr](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeGPS::set\\_time\\_by\\_seconds\(\)](#), and [tai\\_ptr](#).

### 8.7.3.2 convert\_b\_to\_a()

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

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

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

Definition at line 125 of file `time_converter_tai_gps.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gps\\_ptr](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), and [tai\\_ptr](#).

### 8.7.3.3 initialize()

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

Initialize the converter.

#### Assumptions and Limitations

- None

#### Parameters

in	<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 80 of file `time_converter_tai_gps.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gps\\_ptr](#), [jeod::TimeConverter::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [tai\\_ptr](#), [jeod::TimeStandard::tjt\\_at\\_epoch](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).

#### 8.7.3.4 operator=()

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

### 8.7.4 Friends And Related Function Documentation

#### 8.7.4.1 init\_attrjeod\_\_TimeConverter\_TAI\_GPS

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

#### 8.7.4.2 InputProcessor

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

Definition at line 90 of file `time_converter_tai_gps.hh`.

### 8.7.5 Field Documentation

#### 8.7.5.1 gps\_ptr

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

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

`trick_units(-)`

Definition at line 102 of file `time_converter_tai_gps.hh`.

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

### 8.7.5.2 tai\_ptr

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

Converter parent time, always a `TimeTAI` for this converter.

`trick_units(-)`

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

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

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

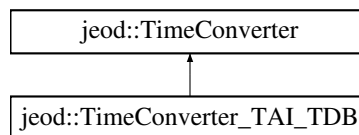
- [time\\_converter\\_tai\\_gps.hh](#)
- [time\\_converter\\_tai\\_gps.cc](#)

## 8.8 jeod::TimeConverter\_TAI\_TDB Class Reference

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

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

Inheritance diagram for `jeod::TimeConverter_TAI_TDB`:



### Public Member Functions

- `TimeConverter_TAI_TDB ()`
- `~TimeConverter_TAI_TDB ()`
- `void initialize (JeodBaseTime *parent, JeodBaseTime *child, const int direction)`  
*Initialize the converter.*
- `void set_a_to_b_offset (void)`
- `void convert_a_to_b (void)`  
*Default converter from time 'a' to time 'b'.*
- `void convert_b_to_a (void)`  
*Default converter from time 'b' to time 'a'.*

### Private Member Functions

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

## Private Attributes

- double [TAI\\_to\\_TT\\_offset](#)  
*The offset from TAI to TT.*
- double [a\\_to\\_b\\_offset\\_epoch](#)  
*The epoch value of a\_to\_b\_offset.*
- double [a\\_to\\_b\\_offset](#)  
*Calculated value of a\_to\_b\_offset.*
- double [prev\\_tai\\_seconds](#)  
*TAI seconds from previous loop iteration.*
- double [prev\\_tdb\\_seconds](#)  
*TDB seconds from previous loop iteration.*
- int [nSteps](#)  
*Counter for number of steps in iteration.*
- int [nlter](#)  
*Counter for number of iterations.*
- [TimeTAI](#) \* [tai\\_ptr](#)  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeTDB](#) \* [tdb\\_ptr](#)  
*Converter parent time, always a [TimeTDB](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_TDB](#) ()

## Additional Inherited Members

### 8.8.1 Detailed Description

Define class [TimeConverter\\_TAI\\_TDB](#), which converts from International Atomic Time to Barycentric Dynamic Time.

Definition at line 90 of file `time_converter_tai_tdb.hh`.

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 TimeConverter\_TAI\_TDB() [1/2]

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

Definition at line 68 of file `time_converter_tai_tdb.cc`.

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

### 8.8.2.2 ~TimeConverter\_TAI\_TDB()

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

Definition at line 194 of file `time_converter_tai_tdb.cc`.

### 8.8.2.3 TimeConverter\_TAI\_TDB() [2/2]

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

## 8.8.3 Member Function Documentation

### 8.8.3.1 convert\_a\_to\_b()

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

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

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

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

Definition at line 152 of file `time_converter_tai_tdb.cc`.

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

### 8.8.3.2 convert\_b\_to\_a()

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

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

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

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

Definition at line 168 of file `time_converter_tai_tdb.cc`.

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

### 8.8.3.3 initialize()

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

Initialize the converter.

## Parameters

in	<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 97 of file `time_converter_tai_tdb.cc`.

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

#### 8.8.3.4 operator=()

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

#### 8.8.3.5 set\_a\_to\_b\_offset()

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

Definition at line 133 of file `time_converter_tai_tdb.cc`.

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

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

### 8.8.4 Friends And Related Function Documentation

#### 8.8.4.1 init\_attrjeod\_\_TimeConverter\_TAI\_TDB

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

#### 8.8.4.2 InputProcessor

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

Definition at line 92 of file `time_converter_tai_tdb.hh`.

### 8.8.5 Field Documentation

#### 8.8.5.1 a\_to\_b\_offset

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

Calculated value of a\_to\_b\_offset.

trick\_units(-)

Definition at line 107 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), set\_a\_to\_b\_offset(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.2 a\_to\_b\_offset\_epoch

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

The epoch value of a\_to\_b\_offset.

trick\_units(s)

Definition at line 103 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.3 nIter

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

Counter for number of iterations.

trick\_units(-)

Definition at line 123 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_b\_to\_a(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.4 nSteps

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

Counter for number of steps in iteration.

trick\_units(-)

Definition at line 119 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_b\_to\_a(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.5 prev\_tai\_seconds

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

TAI seconds from previous loop iteration.

trick\_units(s)

Definition at line 111 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_b\_to\_a(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.6 prev\_tdb\_seconds

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

TDB seconds from previous loop iteration.

trick\_units(s)

Definition at line 115 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_b\_to\_a(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.7 tai\_ptr

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

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

trick\_units(-)

Definition at line 127 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), set\_a\_to\_b\_offset(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.8 TAI\_to\_TT\_offset

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

The offset from TAI to TT.

trick\_units(s)

Definition at line 99 of file time\_converter\_tai\_tdb.hh.

Referenced by initialize(), and TimeConverter\_TAI\_TDB().

#### 8.8.5.9 tdb\_ptr

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

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

trick\_units(-)

Definition at line 131 of file time\_converter\_tai\_tdb.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), and TimeConverter\_TAI\_TDB().

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

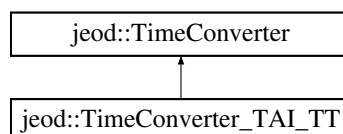
- [time\\_converter\\_tai\\_tdb.hh](#)
- [time\\_converter\\_tai\\_tdb.cc](#)

## 8.9 jeod::TimeConverter\_TAI\_TT Class Reference

Converts between International Atomic Time and Terrestrial Time.

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

Inheritance diagram for jeod::TimeConverter\_TAI\_TT:





## Public Member Functions

- [TimeConverter\\_TAI\\_TT](#) ()  
*Construct a [TimeConverter\\_TAI\\_TT](#).*
- [~TimeConverter\\_TAI\\_TT](#) ()  
*Destroy a [TimeConverter\\_TAI\\_TT](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeTAI](#) to [TimeTT](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeTT](#) to [TimeTAI](#).*

## Private Member Functions

- [TimeConverter\\_TAI\\_TT](#) (const [TimeConverter\\_TAI\\_TT](#) &)
- [TimeConverter\\_TAI\\_TT](#) & [operator=](#) (const [TimeConverter\\_TAI\\_TT](#) &)

## Private Attributes

- [TimeTAI](#) \* [tai\\_ptr](#)  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeTT](#) \* [tt\\_ptr](#)  
*Converter parent time, always a [TimeTT](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_TT](#) ()

## Additional Inherited Members

### 8.9.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

Definition at line 87 of file `time_converter_tai_tt.hh`.

### 8.9.2 Constructor & Destructor Documentation

**8.9.2.1 TimeConverter\_TAI\_TT()** [1/2]

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

Construct a [TimeConverter\\_TAI\\_TT](#).

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

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

**8.9.2.2 ~TimeConverter\_TAI\_TT()**

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

Destroy a [TimeConverter\\_TAI\\_TT](#).

Definition at line 137 of file `time_converter_tai_tt.cc`.

**8.9.2.3 TimeConverter\_TAI\_TT()** [2/2]

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

**8.9.3 Member Function Documentation****8.9.3.1 convert\_a\_to\_b()**

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

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

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

Definition at line 108 of file `time_converter_tai_tt.cc`.

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

## 8.9.3.2 convert\_b\_to\_a()

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

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

## Assumptions and Limitations

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

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

Definition at line 125 of file `time_converter_tai_tt.cc`.

References [jeod::JeodBaseTime::seconds](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), [tai\\_ptr](#), and [tt\\_ptr](#).

## 8.9.3.3 initialize()

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

Initialize the converter.

## Parameters

in	<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_tt.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::TimeConverter::initialized](#), [jeod::TimeMessages::invalid\\_](#)↔  
[setup\\_error](#), [tai\\_ptr](#), [tt\\_ptr](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).

## 8.9.3.4 operator=()

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

## 8.9.4 Friends And Related Function Documentation

#### 8.9.4.1 `init_attrjeod__TimeConverter_TAI_TT`

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

#### 8.9.4.2 `InputProcessor`

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

Definition at line 89 of file `time_converter_tai_tt.hh`.

### 8.9.5 Field Documentation

#### 8.9.5.1 `tai_ptr`

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

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

`trick_units(-)`

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

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

#### 8.9.5.2 `tt_ptr`

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

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

`trick_units(-)`

Definition at line 101 of file `time_converter_tai_tt.hh`.

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

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

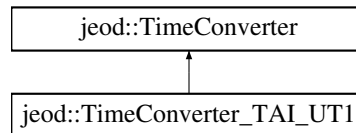
- [time\\_converter\\_tai\\_tt.hh](#)
- [time\\_converter\\_tai\\_tt.cc](#)

## 8.10 jeod::TimeConverter\_TAI\_UT1 Class Reference

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

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

Inheritance diagram for jeod::TimeConverter\_TAI\_UT1:



### Public Member Functions

- [TimeConverter\\_TAI\\_UT1](#) ()  
Construct a [TimeConverter\\_TAI\\_UT1](#).
- [~TimeConverter\\_TAI\\_UT1](#) ()  
Destroy a [TimeConverter\\_TAI\\_UT1](#).
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
Initialize the converter.
- void [convert\\_a\\_to\\_b](#) (void)  
Convert from [TimeTAI](#) to [TimeUT1](#).
- void [convert\\_b\\_to\\_a](#) (void)  
Convert from [TimeUT1](#) to [TimeTAI](#).

### Data Fields

- bool [override\\_data\\_table](#)  
"True" to enter user-specified tai-ut1 offset
- double [tai\\_to\\_ut1\\_override\\_val](#)  
User specified value (UT1 - TAI)
- int [last\\_index](#)  
Index of last datum in table.
- int [index](#)  
Current location in table.
- double \* [val\\_vec](#)  
Vector of values of difference between TAI-UT1.
- double \* [when\\_vec](#)  
Vector of corresponding times.

### Private Member Functions

- void [initialize\\_tai\\_to\\_ut1](#) (void)  
The conversion from Atomic Time (TAI) to Universal Time (UT1) involves the addition of value that is a continuous function of TAI.
- void [verify\\_table\\_lookup\\_ends](#) (void)  
Used when time reverses direction.
- [TimeConverter\\_TAI\\_UT1](#) (const [TimeConverter\\_TAI\\_UT1](#) &)
- [TimeConverter\\_TAI\\_UT1](#) & [operator=](#) (const [TimeConverter\\_TAI\\_UT1](#) &)

## Private Attributes

- [TimeTAI](#) \* [tai\\_ptr](#)  
Converter parent time, always a [TimeTAI](#) for this converter.
- [TimeUT1](#) \* [ut1\\_ptr](#)  
Converter parent time, always a [TimeUT1](#) for this converter.
- double [prev\\_when](#)  
Time of previous calibrated datum.
- double [prev\\_value](#)  
Offset value of previous datum.
- double [next\\_when](#)  
Time of next calibrated datum.
- double [next\\_value](#)  
Offset value of next datum.
- double [gradient](#)  
Rate at which "value" changes wrt "when".
- bool [off\\_table\\_end](#)  
Gone past the end of the table.

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_UT1](#) ()

## Additional Inherited Members

### 8.10.1 Detailed Description

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

Definition at line 89 of file `time_converter_tai_ut1.hh`.

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 [TimeConverter\\_TAI\\_UT1\(\)](#) [1/2]

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

Construct a [TimeConverter\\_TAI\\_UT1](#).

Definition at line 60 of file `time_converter_tai_ut1.cc`.

References [jeod::TimeConverter::a\\_name](#), [jeod::TimeConverter::ANY\\_DIRECTION](#), [jeod::TimeConverter::b\\_name](#), [gradient](#), [index](#), [last\\_index](#), [next\\_value](#), [next\\_when](#), [off\\_table\\_end](#), [override\\_data\\_table](#), [prev\\_value](#), [prev\\_when](#), [tai\\_ptr](#), [tai\\_to\\_ut1\\_override\\_val](#), [ut1\\_ptr](#), [val\\_vec](#), [jeod::TimeConverter::valid\\_directions](#), and [when\\_vec](#).

## 8.10.2.2 ~TimeConverter\_TAI\_UT1()

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

Destroy a [TimeConverter\\_TAI\\_UT1](#).

Definition at line 491 of file `time_converter_tai_ut1.cc`.

References `val_vec`, and `when_vec`.

## 8.10.2.3 TimeConverter\_TAI\_UT1() [2/2]

```
jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1 (
    const TimeConverter\_TAI\_UT1 & ) [private]
```

## 8.10.3 Member Function Documentation

## 8.10.3.1 convert\_a\_to\_b()

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

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

## Assumptions and Limitations

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

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

Definition at line 258 of file `time_converter_tai_ut1.cc`.

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

### 8.10.3.2 convert\_b\_to\_a()

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

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

#### Assumptions and Limitations

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

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

Definition at line 353 of file `time_converter_tai_ut1.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gradient](#), [index](#), [jeod::TimeMessages::invalid\\_data\\_error](#), [last\\_index](#), [next\\_value](#), [next\\_when](#), [off\\_table\\_end](#), [prev\\_value](#), [prev\\_when](#), [jeod::TimeStandard::set\\_time\\_by\\_trunc\\_julian\(\)](#), [tai\\_ptr](#), [jeod::TimeUT1::true\\_ut1](#), [jeod::TimeStandard::trunc\\_julian\\_time](#), [ut1\\_ptr](#), [val\\_vec](#), and [when\\_vec](#).

### 8.10.3.3 initialize()

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

Initialize the converter.

#### Assumptions and Limitations

- None

#### Parameters

in	<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 95 of file `time_converter_tai_ut1.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::TimeMessages::initialization\\_error](#), [initialize\\_tai\\_to\\_ut1\(\)](#), [jeod::TimeConverter::initialized](#), [next\\_when](#), [prev\\_when](#), [tai\\_ptr](#), [jeod::TimeStandard::trunc\\_julian\\_time](#), [ut1\\_ptr](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).



#### 8.10.3.4 initialize\_tai\_to\_ut1()

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

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

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

##### Assumptions and Limitations

- The table does not go into the future.

Definition at line 145 of file `time_converter_tai_ut1.cc`.

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

Referenced by `initialize()`.

#### 8.10.3.5 operator=()

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

#### 8.10.3.6 verify\_table\_lookup\_ends()

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

Used when time reverses direction.

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

##### Assumptions and Limitations

- None

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

Definition at line 449 of file `time_converter_tai_ut1.cc`.

References `jeod::TimeManager::dyn_time`, `index`, `last_index`, `next_when`, `off_table_end`, `prev_when`, `jeod::TimeManager::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 155 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), and TimeConverter\_TAI\_UT1().

### 8.10.5.2 index

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

Current location in table.

trick\_units(—)

Definition at line 125 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify↵\_table\_lookup\_ends().

### 8.10.5.3 last\_index

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

Index of last datum in table.

trick\_units(-)

Definition at line 121 of file time\_converter\_tai\_ut1.hh.

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

### 8.10.5.4 next\_value

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

Offset value of next datum.

trick\_units(s)

Definition at line 151 of file time\_converter\_tai\_ut1.hh.

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

### 8.10.5.5 next\_when

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

Time of next calibrated datum.

trick\_units(day)

Definition at line 147 of file time\_converter\_tai\_ut1.hh.

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

### 8.10.5.6 off\_table\_end

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

Gone past the end of the table.

trick\_units(-)

Definition at line 159 of file time\_converter\_tai\_ut1.hh.

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

#### 8.10.5.7 `override_data_table`

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

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

`trick_units(-)`

Definition at line 98 of file `time_converter_tai_ut1.hh`.

Referenced by `jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, and `jeod::TimeManagerInit::verify_converter_setup()`.

#### 8.10.5.8 `prev_value`

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

Offset value of previous datum.

`trick_units(s)`

Definition at line 143 of file `time_converter_tai_ut1.hh`.

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

#### 8.10.5.9 `prev_when`

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

Time of previous calibrated datum.

`trick_units(day)`

Definition at line 139 of file `time_converter_tai_ut1.hh`.

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

#### 8.10.5.10 `tai_ptr`

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

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

`trick_units(-)`

Definition at line 105 of file `time_converter_tai_ut1.hh`.

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

**8.10.5.11 tai\_to\_ut1\_override\_val**

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

User specified value (UT1 - TAI)

trick\_units(s)

Definition at line 116 of file time\_converter\_tai\_ut1.hh.

Referenced by initialize\_tai\_to\_ut1(), and TimeConverter\_TAI\_UT1().

**8.10.5.12 ut1\_ptr**

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

Converter parent time, always a [TimeUT1](#) for this converter.

trick\_units(-)

Definition at line 110 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify\_table\_lookup\_ends().

**8.10.5.13 val\_vec**

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

Vector of values of difference between TAI-UT1.

trick\_units(s)

Definition at line 129 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data↵::initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and ~TimeConverter\_TAI\_UT1().

**8.10.5.14 when\_vec**

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

Vector of corresponding times.

trick\_units(day)

Definition at line 133 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data↵::initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), verify\_table\_lookup\_ends(), and ~TimeConverter↵\_TAI\_UT1().

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

- [time\\_converter\\_tai\\_ut1.hh](#)
- [time\\_converter\\_tai\\_ut1.cc](#)

## 8.11 jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data Class Reference

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

### Public Member Functions

- void [initialize](#) ([TimeConverter\\_TAI\\_UT1](#) \*)

### 8.11.1 Detailed Description

Definition at line 54 of file [tai\\_to\\_ut1.hh](#).

### 8.11.2 Member Function Documentation

#### 8.11.2.1 initialize()

```
void jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize (
    TimeConverter\_TAI\_UT1 * TimeConverter_TAI_UT1_ptr )
```

Definition at line 41 of file [tai\\_to\\_ut1.cc](#).

References [jeod::TimeConverter\\_TAI\\_UT1::last\\_index](#), [jeod::TimeConverter\\_TAI\\_UT1::override\\_data\\_table](#), [jeod::TimeConverter\\_TAI\\_UT1::val\\_vec](#), and [jeod::TimeConverter\\_TAI\\_UT1::when\\_vec](#).

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

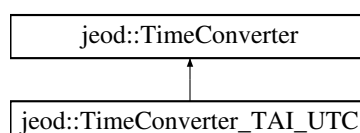
- [tai\\_to\\_ut1.hh](#)
- [tai\\_to\\_ut1.cc](#)

## 8.12 jeod::TimeConverter\_TAI\_UTC Class Reference

Converts between International Atomic Time and Coordinated Universal Time.

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

Inheritance diagram for [jeod::TimeConverter\\_TAI\\_UTC](#):



## Public Member Functions

- [TimeConverter\\_TAI\\_UTC](#) ()  
*Construct a [TimeConverter\\_TAI\\_UTC](#).*
- [~TimeConverter\\_TAI\\_UTC](#) ()  
*Destroy a [TimeConverter\\_TAI\\_UTC](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeTAI](#) to [TimeUTC](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeUTC](#) to [TimeTAI](#).*

## Data Fields

- bool [override\\_data\\_table](#)  
*"True" to enter user-specified tai-utc offset*
- double [leap\\_sec\\_override\\_val](#)  
*User specified value (TAI - UTC)*
- int [last\\_index](#)  
*Maximum index in the leap tables.*
- int [index](#)  
*Current index in the leap tables.*
- int \* [val\\_vec](#)  
*Tabulated values of leap\_value.*
- double \* [when\\_vec](#)  
*Tabulated values of Julian time corresponding to changes in leap\_value.*

## Private Member Functions

- void [initialize\\_leap\\_second](#) (void)  
*The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.*
- void [verify\\_table\\_lookup\\_ends](#) (void)  
*Used when time reverses direction.*
- [TimeConverter\\_TAI\\_UTC](#) (const [TimeConverter\\_TAI\\_UTC](#) &)
- [TimeConverter\\_TAI\\_UTC](#) & [operator=](#) (const [TimeConverter\\_TAI\\_UTC](#) &)

## Private Attributes

- [TimeTAI](#) \* [tai\\_ptr](#)  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeUTC](#) \* [utc\\_ptr](#)  
*Converter parent time, always a [TimeUTC](#) for this converter.*
- double [next\\_when](#)  
*The next (future) UTC time of a leap second instance.*
- double [prev\\_when](#)  
*The most recent (past) UTC time of a leap second instance.*
- bool [off\\_table\\_end](#)  
*Flag to indicate that the current time is not covered by the leap-second tables.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI.UTC](#) ()

## Additional Inherited Members

### 8.12.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

Definition at line 89 of file `time_converter_tai_utc.hh`.

### 8.12.2 Constructor & Destructor Documentation

#### 8.12.2.1 `TimeConverter_TAI.UTC()` [1/2]

```
jeod::TimeConverter_TAI.UTC::TimeConverter_TAI.UTC (
    void )
```

Construct a [TimeConverter\\_TAI.UTC](#).

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

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

#### 8.12.2.2 `~TimeConverter_TAI.UTC()`

```
jeod::TimeConverter_TAI.UTC::~~TimeConverter_TAI.UTC (
    void )
```

Destroy a [TimeConverter\\_TAI.UTC](#).

Definition at line 461 of file `time_converter_tai_utc.cc`.

References `val_vec`, and `when_vec`.

#### 8.12.2.3 `TimeConverter_TAI.UTC()` [2/2]

```
jeod::TimeConverter_TAI.UTC::TimeConverter_TAI.UTC (
    const TimeConverter\_TAI.UTC & ) [private]
```



### 8.12.3 Member Function Documentation

#### 8.12.3.1 convert\_a\_to\_b()

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

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

#### Assumptions and Limitations

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

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

Definition at line 266 of file `time_converter_tai_utc.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::TimeManager::dyn\\_time](#), [index](#), [jeod::TimeMessages::invalid\\_data\\_error](#), [last\\_index](#), [next\\_when](#), [off\\_table\\_end](#), [prev\\_when](#), [jeod::TimeDyn::scale\\_factor](#), [jeod::TimeStandard::set\\_time\\_by\\_trunc\\_julian\(\)](#), [tai\\_ptr](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::TimeUTC::true\\_utc](#), [jeod::TimeStandard::trunc\\_julian\\_time](#), [utc\\_ptr](#), [val\\_vec](#), and [when\\_vec](#).

#### 8.12.3.2 convert\_b\_to\_a()

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

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

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

Definition at line 352 of file `time_converter_tai_utc.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [jeod::TimeManager::dyn\\_time](#), [index](#), [jeod::TimeMessages::invalid\\_data\\_error](#), [last\\_index](#), [next\\_when](#), [off\\_table\\_end](#), [prev\\_when](#), [jeod::TimeDyn::scale\\_factor](#), [jeod::TimeStandard::set\\_time\\_by\\_trunc\\_julian\(\)](#), [tai\\_ptr](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::TimeUTC::true\\_utc](#), [jeod::TimeStandard::trunc\\_julian\\_time](#), [utc\\_ptr](#), [val\\_vec](#), and [when\\_vec](#).

#### 8.12.3.3 initialize()

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

Initialize the converter.

**Parameters**

in	<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 90 of file `time_converter_tai_utc.cc`.

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

**8.12.3.4 initialize\_leap\_second()**

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

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

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

**Assumptions and Limitations**

- The table does not go into the future.

Definition at line 150 of file `time_converter_tai_utc.cc`.

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

Referenced by `initialize()`.

**8.12.3.5 operator=()**

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

### 8.12.3.6 verify\_table\_lookup\_ends()

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

Used when time reverses direction.

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

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

Definition at line 420 of file time\_converter\_tai\_utc.cc.

References [jeod::TimeManager::dyn\\_time](#), [index](#), [last\\_index](#), [next\\_when](#), [off\\_table\\_end](#), [prev\\_when](#), [jeod::Time↔Dyn::scale\\_factor](#), [tai\\_ptr](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::TimeUTC::true\\_utc](#), [jeod::TimeStandard↔::trunc\\_julian\\_time](#), [utc\\_ptr](#), and [when\\_vec](#).

## 8.12.4 Friends And Related Function Documentation

### 8.12.4.1 init\_attrjeod\_\_TimeConverter\_TAI\_UTC

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

### 8.12.4.2 InputProcessor

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

Definition at line 91 of file time\_converter\_tai\_utc.hh.

## 8.12.5 Field Documentation

### 8.12.5.1 index

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

Current index in the leap tables.

trick\_units(-)

Definition at line 123 of file time\_converter\_tai\_utc.hh.

Referenced by [convert\\_a\\_to\\_b\(\)](#), [convert\\_b\\_to\\_a\(\)](#), [initialize\(\)](#), [initialize\\_leap\\_second\(\)](#), [TimeConverter\\_TAI\\_UT↔C\(\)](#), and [verify\\_table\\_lookup\\_ends\(\)](#).

#### 8.12.5.2 last\_index

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

Maximum index in the leap tables.

trick\_units(-)

Definition at line 119 of file time\_converter\_tai\_utc.hh.

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

#### 8.12.5.3 leap\_sec\_override\_val

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

User specified value (TAI - UTC)

trick\_units(s)

Definition at line 115 of file time\_converter\_tai\_utc.hh.

Referenced by `initialize_leap_second()`, and `TimeConverter_TAI_UTC()`.

#### 8.12.5.4 next\_when

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

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

trick\_units(-)

Definition at line 137 of file time\_converter\_tai\_utc.hh.

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

#### 8.12.5.5 off\_table\_end

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

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

trick\_units(-)

Definition at line 147 of file time\_converter\_tai\_utc.hh.

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

## 8.12.5.6 override\_data\_table

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

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

```
trick_units(-)
```

Definition at line 98 of file time\_converter\_tai\_utc.hh.

Referenced by jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data::initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and jeod::TimeManagerInit::verify\_converter\_setup().

## 8.12.5.7 prev\_when

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

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

```
trick_units(-)
```

Definition at line 142 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

## 8.12.5.8 tai\_ptr

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

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

```
trick_units(-)
```

Definition at line 104 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

## 8.12.5.9 utc\_ptr

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

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

```
trick_units(-)
```

Definition at line 109 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

#### 8.12.5.10 val\_vec

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

Tabulated values of leap\_value.

trick\_units(s)

Definition at line 127 of file time\_converter\_tai\_utc.hh.

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

#### 8.12.5.11 when\_vec

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

Tabulated values of Julian time corresponding to changes in leap\_value.

trick\_units(day)

Definition at line 132 of file time\_converter\_tai\_utc.hh.

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

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

- [time\\_converter\\_tai\\_utc.hh](#)
- [time\\_converter\\_tai\\_utc.cc](#)

## 8.13 jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data Class Reference

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

### Public Member Functions

- void [initialize](#) ([TimeConverter\\_TAI\\_UTC](#) \*)

#### 8.13.1 Detailed Description

Definition at line 54 of file tai\_to\_utc.hh.

### 8.13.2 Member Function Documentation

#### 8.13.2.1 initialize()

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

Definition at line 40 of file tai\_to\_utc.cc.

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

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

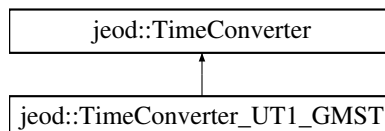
- [tai\\_to\\_utc.hh](#)
- [tai\\_to\\_utc.cc](#)

## 8.14 jeod::TimeConverter\_UT1\_GMST Class Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

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

Inheritance diagram for `jeod::TimeConverter_UT1_GMST`:



### Public Member Functions

- [TimeConverter\\_UT1\\_GMST](#) ()  
*Construct a [TimeConverter\\_UT1\\_GMST](#).*
- [~TimeConverter\\_UT1\\_GMST](#) ()  
*Destroy a [TimeConverter\\_UT1\\_GMST](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeUT1](#) to [TimeGMST](#).*

### Private Member Functions

- [TimeConverter\\_UT1\\_GMST](#) (const [TimeConverter\\_UT1\\_GMST](#) &)
- [TimeConverter\\_UT1\\_GMST](#) & [operator=](#) (const [TimeConverter\\_UT1\\_GMST](#) &)

## Private Attributes

- [TimeUT1](#) \* [ut1\\_ptr](#)  
Converter parent time, always a [TimeUT1](#) for this converter.
- [TimeGMST](#) \* [gmst\\_ptr](#)  
Converter parent time, always a [TimeGMST](#) for this converter.

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_UT1\\_GMST](#) ()

## Additional Inherited Members

### 8.14.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

Definition at line 87 of file `time_converter_ut1_gmst.hh`.

### 8.14.2 Constructor & Destructor Documentation

#### 8.14.2.1 [TimeConverter\\_UT1\\_GMST](#)() [1/2]

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

Construct a [TimeConverter\\_UT1\\_GMST](#).

Definition at line 58 of file `time_converter_ut1_gmst.cc`.

References [jeod::TimeConverter::a\\_name](#), [jeod::TimeConverter::A\\_TO\\_B](#), [jeod::TimeConverter::b\\_name](#), [gmst\\_ptr](#), [ut1\\_ptr](#), and [jeod::TimeConverter::valid\\_directions](#).

#### 8.14.2.2 [~TimeConverter\\_UT1\\_GMST](#)()

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

Destroy a [TimeConverter\\_UT1\\_GMST](#).

Definition at line 148 of file `time_converter_ut1_gmst.cc`.



## 8.14.2.3 TimeConverter\_UT1\_GMST() [2/2]

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

## 8.14.3 Member Function Documentation

## 8.14.3.1 convert\_a\_to\_b()

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

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

## Assumptions and Limitations

- None

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

Definition at line 111 of file `time_converter_ut1_gmst.cc`.

References [jeod::TimeUT1::get\\_days\(\)](#), [gmst\\_ptr](#), [jeod::TimeStandard::set\\_time\\_by\\_days\(\)](#), and [ut1\\_ptr](#).

## 8.14.3.2 initialize()

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

Initialize the converter.

## Assumptions and Limitations

- None

## Parameters

in	<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 79 of file `time_converter_ut1_gmst.cc`.

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

#### 8.14.3.3 `operator=()`

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

### 8.14.4 Friends And Related Function Documentation

#### 8.14.4.1 `init_attrjeod__TimeConverter_UT1_GMST`

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

#### 8.14.4.2 `InputProcessor`

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

Definition at line 89 of file `time_converter_ut1_gmst.hh`.

### 8.14.5 Field Documentation

#### 8.14.5.1 `gmst_ptr`

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

Converter parent time, always a `TimeGMST` for this converter.

`trick_units(-)`

Definition at line 101 of file `time_converter_ut1_gmst.hh`.

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

## 8.14.5.2 ut1\_ptr

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

Converter parent time, always a [TimeUT1](#) for this converter.

trick\_units(-)

Definition at line 96 of file `time_converter_ut1_gmst.hh`.

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

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

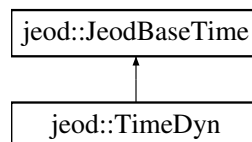
- [time\\_converter\\_ut1\\_gmst.hh](#)
- [time\\_converter\\_ut1\\_gmst.cc](#)

## 8.15 jeod::TimeDyn Class Reference

Represents the Dynamic Time in the simulation.

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

Inheritance diagram for `jeod::TimeDyn`:



## Public Member Functions

- [TimeDyn](#) ()  
*Construct a Time\_Dyn.*
- [~TimeDyn](#) ()  
*Destroy a Time\_Dyn.*
- `bool` [update\\_offset](#) (void)  
*Changeing time direction and/or scale factor.*

## Data Fields

- `double` [scale\\_factor](#)  
*Multiplicative difference between sim-time and dyn-time.*

## Private Member Functions

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

## Private Attributes

- double `ref_scale`  
*Private copy of scale\_factor.*
- double `offset`  
*Extrapolated difference between sim-time and dyn-time at the sim-start (0 if there are no changes to direction or scale)*

## Friends

- class `InputProcessor`
- void `init_attrjeod__TimeDyn` ()

## Additional Inherited Members

### 8.15.1 Detailed Description

Represents the Dynamic Time in the simulation.

Definition at line 86 of file `time_dyn.hh`.

### 8.15.2 Constructor & Destructor Documentation

#### 8.15.2.1 TimeDyn() [1/2]

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

Construct a `Time_Dyn`.

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

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

## 8.15.2.2 ~TimeDyn()

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

Destroy a Time\_Dyn.

Definition at line 149 of file time\_dyn.cc.

## 8.15.2.3 TimeDyn() [2/2]

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

## 8.15.3 Member Function Documentation

## 8.15.3.1 initialize\_initializer\_time()

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

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

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

## Assumptions and Limitations

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

## Parameters

in	<a href="#">time_manager_init</a>	TM initializer
----	-----------------------------------	----------------

Implements [jeod::JeodBaseTime](#).

Definition at line 82 of file time\_dyn.cc.

References [jeod::JeodBaseTime::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::seconds](#), [jeod::JeodBaseTime::time\\_manager](#), and [jeod::TimeManager::time\\_standards\\_exist\(\)](#).

### 8.15.3.2 operator=()

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

### 8.15.3.3 update()

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

[TimeDyn](#) updates directly from simtime, and everything else from [TimeDyn](#).

This function does that first update from simtime

#### Assumptions and Limitations

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

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

Definition at line 110 of file `time_dyn.cc`.

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

### 8.15.3.4 update\_offset()

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

Changing time direction and/or scale factor.

#### Returns

Void

Definition at line 124 of file `time_dyn.cc`.

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

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

## 8.15.4 Friends And Related Function Documentation

#### 8.15.4.1 init\_attrjeod\_\_TimeDyn

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

#### 8.15.4.2 InputProcessor

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

Definition at line 88 of file time\_dyn.hh.

### 8.15.5 Field Documentation

#### 8.15.5.1 offset

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

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

trick\_units(–)

Definition at line 109 of file time\_dyn.hh.

Referenced by TimeDyn(), update(), and update\_offset().

#### 8.15.5.2 ref\_scale

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

Private copy of scale\_factor.

This value should not be changed externally; it is used for comparison purposes to identify when "scale\_factor" has changed.trick\_units(–)

Definition at line 104 of file time\_dyn.hh.

Referenced by TimeDyn(), update(), and update\_offset().

### 8.15.5.3 scale\_factor

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

Multiplicative difference between sim-time and dyn-time.

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

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

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeManager::get_time_scale_factor()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `TimeDyn()`, `update_offset()`, `jeod::TimeConverter_TAI_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 78 of file `time_enum.hh`.

### 8.16.2 Member Enumeration Documentation

#### 8.16.2.1 TimeFormat

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

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



## Enumerator

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

Definition at line 86 of file `time_enum.hh`.

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

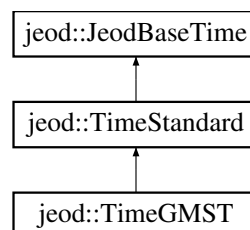
- [time\\_enum.hh](#)

## 8.17 jeod::TimeGMST Class Reference

To represent the clock known as Greenwich Mean Sidereal Time.

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

Inheritance diagram for `jeod::TimeGMST`:



### Public Member Functions

- [TimeGMST](#) ()  
*Construct a Time\_GMST.*
- [~TimeGMST](#) ()  
*Destroy a Time\_GMST.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double nonsense)  
*TJT does not function in GMST.*

### Private Member Functions

- void [calculate\\_calendar\\_values](#) (void)  
*Protection against inheriting nonsense function.*
- void [set\\_epoch](#) (void)  
*No action.*
- [TimeGMST](#) (const [TimeGMST](#) &)
- [TimeGMST](#) & [operator=](#) (const [TimeGMST](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeGMST](#) ()

## Additional Inherited Members

### 8.17.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

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

### 8.17.2 Constructor & Destructor Documentation

#### 8.17.2.1 TimeGMST() [1/2]

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

Construct a Time\_GMST.

Definition at line 55 of file `time_gmst.cc`.

References `jeod::JeodBaseTime::name`.

#### 8.17.2.2 ~TimeGMST()

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

Destroy a Time\_GMST.

Definition at line 102 of file `time_gmst.cc`.

#### 8.17.2.3 TimeGMST() [2/2]

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

### 8.17.3 Member Function Documentation

#### 8.17.3.1 calculate\_calendar\_values()

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

Protection against inheriting nonsense function.

#### Assumptions and Limitations

- GMST does not have a conventional calendar

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

Definition at line 68 of file `time_gmst.cc`.

References `jeod::TimeMessages::invalid_data_error`.

#### 8.17.3.2 operator=()

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

#### 8.17.3.3 set\_epoch()

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

No action.

Function is required to make this class instantiable.

Implements [jeod::TimeStandard](#).

Definition at line 104 of file `time_gmst.hh`.

#### 8.17.3.4 set\_time\_by\_trunc\_julian()

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

TJT does not function in GMST.

#### Assumptions and Limitations

- 86400 seconds = 1 day

**Parameters**

in	<i>nonsense</i>	Any old invalid value
----	-----------------	-----------------------

Definition at line 86 of file `time_gmst.cc`.

References `jeod::TimeMessages::invalid_data_error`.

## 8.17.4 Friends And Related Function Documentation

### 8.17.4.1 `init_attrjeod__TimeGMST`

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

### 8.17.4.2 `InputProcessor`

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

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

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

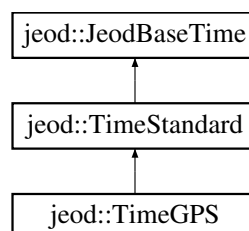
- [time\\_gmst.hh](#)
- [time\\_gmst.cc](#)

## 8.18 `jeod::TimeGPS` Class Reference

To represent the time associated with the Global Positioning System.

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

Inheritance diagram for `jeod::TimeGPS`:



## Public Member Functions

- [TimeGPS](#) ()  
*Construct a Time\_GPS.*
- [~TimeGPS](#) ()  
*Destroy a TimeGPS.*
- void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)  
*Given a value of seconds, propagate to other reps.*
- void [set\\_time\\_by\\_days](#) (const double new\_seconds)  
*Given a value of days, propagate to other values.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double new\_tjt)  
*TJT does not function in GPS.*

## Data Fields

- double [seconds\\_of\\_day](#)  
*Seconds elapsed in last (partial) day.*
- double [seconds\\_of\\_week](#)  
*Seconds elapsed in last (partial) week.*
- int [day\\_of\\_week](#)  
*Number of whole days this week.*
- int [rollover\\_count](#)  
*Number of rollovers (1024 week blocks) since epoch.*
- int [week](#)  
*Number of weeks in current 1024-week block.*
- int [rollover\\_count\\_13\\_bit](#)  
*Number of rollovers (8192 week blocks) since epoch.*
- int [week\\_13\\_bit](#)  
*Number of weeks in current 8192-week block.*

## Private Member Functions

- void [calculate\\_calendar\\_values](#) (void)  
*Protection against inheriting nonsense function.*
- void [convert\\_from\\_calendar](#) (void)  
*Protection against inheriting nonsense function.*
- void [set\\_epoch](#) (void)  
*Sets the epoch for GPS time.*
- [TimeGPS](#) (const [TimeGPS](#) &)
- [TimeGPS](#) & [operator=](#) (const [TimeGPS](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeGPS](#) ()

## Additional Inherited Members

### 8.18.1 Detailed Description

To represent the time associated with the Global Positioning System.

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

### 8.18.2 Constructor & Destructor Documentation

#### 8.18.2.1 TimeGPS() [1/2]

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

Construct a `Time_GPS`.

Definition at line 55 of file `time_gps.cc`.

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

#### 8.18.2.2 ~TimeGPS()

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

Destroy a [TimeGPS](#).

Definition at line 193 of file `time_gps.cc`.

#### 8.18.2.3 TimeGPS() [2/2]

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

### 8.18.3 Member Function Documentation

### 8.18.3.1 calculate\_calendar\_values()

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

Protection against inheriting nonsense function.

#### Assumptions and Limitations

- GPS does not have a conventional calendar

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

Definition at line 110 of file time\_gps.cc.

References [jeod::TimeMessages::invalid\\_data\\_error](#).

### 8.18.3.2 convert\_from\_calendar()

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

Protection against inheriting nonsense function.

#### Assumptions and Limitations

- GPS does not have a conventional calendar

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

Definition at line 92 of file time\_gps.cc.

References [jeod::TimeMessages::invalid\\_data\\_error](#).

### 8.18.3.3 operator=()

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

#### 8.18.3.4 `set_epoch()`

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

Sets the epoch for GPS time.

Implements [jeod::TimeStandard](#).

Definition at line 75 of file `time_gps.cc`.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by `TimeGPS()`.

#### 8.18.3.5 `set_time_by_days()`

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

Given a value of days, propagate to other values.

##### Assumptions and Limitations

- 86400 seconds = 1 day

##### Parameters

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

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

Definition at line 163 of file `time_gps.cc`.

References `set_time_by_seconds()`.

#### 8.18.3.6 `set_time_by_seconds()`

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

Given a value of seconds, propagate to other reps.

##### Assumptions and Limitations

- 86400 seconds = 1 day



## Parameters

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

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

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

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

Referenced by `jeod::TimeConverter_TAI_GPS::convert_a_to_b()`, `set_time_by_days()`, and `set_time_by_trunc_↵julian()`.

### 8.18.3.7 set\_time\_by\_trunc\_julian()

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

TJT does not function in GPS.

#### Assumptions and Limitations

- 86400 seconds = 1 day

## Parameters

in	<i>new_↵_tjt</i>	new value for Truncated Julian Time Units: day
----	------------------	---

Definition at line 179 of file `time_gps.cc`.

References `jeod::JeodBaseTime::seconds`, `set_time_by_seconds()`, and `jeod::TimeStandard::set_time_by_trunc_↵_julian()`.

## 8.18.4 Friends And Related Function Documentation

### 8.18.4.1 init\_attrjeod\_\_TimeGPS

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

#### 8.18.4.2 InputProcessor

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

Definition at line 84 of file time\_gps.hh.

### 8.18.5 Field Documentation

#### 8.18.5.1 day\_of\_week

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

Number of whole days this week.

trick\_units(day)

Definition at line 100 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.2 rollover\_count

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

Number of rollovers (1024 week blocks) since epoch.

trick\_units(-)

Definition at line 104 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.3 rollover\_count\_13\_bit

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

Number of rollovers (8192 week blocks) since epoch.

trick\_units(-)

Definition at line 112 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.4 seconds\_of\_day

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

Seconds elapsed in last (partial) day.

trick\_units(s)

Definition at line 92 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.5 seconds\_of\_week

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

Seconds elapsed in last (partial) week.

trick\_units(s)

Definition at line 96 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.6 week

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

Number of weeks in current 1024-week block.

trick\_units(-)

Definition at line 108 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

#### 8.18.5.7 week\_13\_bit

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

Number of weeks in current 8192-week block.

trick\_units(-)

Definition at line 116 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

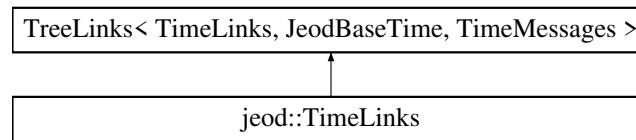
The documentation for this class was generated from the following files:

- [time\\_gps.hh](#)
- [time\\_gps.cc](#)

## 8.19 jeod::TimeLinks Class Reference

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

Inheritance diagram for jeod::TimeLinks:



### Public Member Functions

- [TimeLinks](#) ([JeodBaseTime](#) &time\_in)
  - [TimeLinks](#) ()=delete
  - [TimeLinks](#) (const [TimeLinks](#) &)=delete
  - void [operator=](#) (const [TimeLinks](#) &)=delete
  - virtual [~TimeLinks](#) ()=default
- Default destructor.*

### Static Private Attributes

- static const unsigned int [default\\_path\\_size](#) = 8
- Default allocated number of entries in linkage container.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeLinks](#) ()

### 8.19.1 Detailed Description

Definition at line 78 of file `time_links.hh`.

### 8.19.2 Constructor & Destructor Documentation

#### 8.19.2.1 TimeLinks() [1/3]

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

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

### 8.19.2.2 TimeLinks() [2/3]

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

### 8.19.2.3 TimeLinks() [3/3]

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

### 8.19.2.4 ~TimeLinks()

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

Default destructor.

## 8.19.3 Member Function Documentation

### 8.19.3.1 operator=()

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

## 8.19.4 Friends And Related Function Documentation

### 8.19.4.1 init\_attrjeod\_\_TimeLinks

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

### 8.19.4.2 InputProcessor

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

Definition at line 80 of file time\_links.hh.

### 8.19.5 Field Documentation

#### 8.19.5.1 default\_path\_size

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

Default allocated number of entries in linkage container.

trick\_units(—)

Definition at line 104 of file time\_links.hh.

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

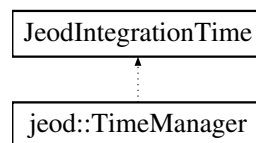
- [time\\_links.hh](#)

## 8.20 jeod::TimeManager Class Reference

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

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

Inheritance diagram for jeod::TimeManager:



### Public Member Functions

- [TimeManager](#) ()  
*Construct a [TimeManager](#).*
- [~TimeManager](#) ()  
*Destroy a [TimeManager](#).*
- void [initialize](#) ([TimeManagerInit](#) \*time\_manager\_init)  
*initializes the time manager*
- int [time\\_lookup](#) (const std::string &name) const  
*Uses a string comparison to find where in the [TimeManager](#) record a time type of a particular name is located.*
- bool [get\\_time\\_change\\_flag](#) () const  
*Returns the boolean value time\_change\_flag.*
- [JeodBaseTime](#) \* [get\\_time\\_ptr](#) (const std::string &name) const  
*Return a pointer to the Time object with the provided name, or NULL if no such Time object has been registered.*
- [JeodBaseTime](#) \* [get\\_time\\_ptr](#) (const int index) const  
*Return a pointer to the Time object with the provided index, or NULL if no such Time object has been registered.*
- [TimeConverter](#) \* [get\\_converter\\_ptr](#) (const int index) const

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

- bool [time\\_standards\\_exist](#) (void)  
Tests for the existence in the registry of time types that inherit from [TimeStandard](#).
- virtual void [update](#) (double time)  
This function manages the time update process.
- void [verify\\_table\\_lookup\\_ends](#) (void)  
This function is called when the simulation reverses direction (in time).
- void [register\\_time](#) ([JeodBaseTime](#) &time\_ref)  
Registers the time representation with the Time Manager.
- void [register\\_time\\_named](#) ([JeodBaseTime](#) &time\_ref, const std::string &name)  
Reassigns the name to the type; this is used when there are multiple instances of a time type such as a MET or UDE.
- void [register\\_converter](#) ([TimeConverter](#) &converter\_ref, std::string name\_a="", std::string name\_b="")  
Registers the time converters with the Time Manager.
- [JeodIntegrationTime](#) & [get\\_jeod\\_integration\\_time](#) ()  
Expose the private inheritance from [JeodIntegrationTime](#).
- virtual double [get\\_time\\_scale\\_factor](#) () const  
Returns the scale factor from sim time to dynamic time.
- virtual double [get\\_timestamp\\_time](#) () const  
Returns the time used to timestamp objects, currently dynamic time seconds.

## Data Fields

- double [simtime](#)  
Simulation time (sys.exec.out.time).
- [TimeDyn](#) dyn\_time  
The instance of [TimeDyn](#), the dynamic time that is used as the integration time.
- int [num\\_types](#)  
Size of time\_types\_ptrs vector.

## Private Member Functions

- virtual void [update\\_time](#) (double time)  
Update each of the representations of time, calling the update functions for each such representation in dependency order.
- [TimeManager](#) (const [TimeManager](#) &)
- [TimeManager](#) & operator= (const [TimeManager](#) &)

## Private Attributes

- bool [time\\_change\\_flag](#)  
Indicates that the dynamic scale factor changed.
- std::vector< [JeodBaseTime](#) \* > [time\\_vector](#)  
List of pointers to time-types.
- std::vector< [TimeConverter](#) \* > [converter\\_vector](#)  
List of pointers to time-converters.

## Friends

- class [InputProcessor](#)
- class [TimeManagerInit](#)
- void [init\\_attrjeod\\_\\_TimeManager](#) ()

### 8.20.1 Detailed Description

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

Definition at line 94 of file `time_manager.hh`.

### 8.20.2 Constructor & Destructor Documentation

#### 8.20.2.1 TimeManager() [1/2]

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

Construct a [TimeManager](#).

Definition at line 65 of file `time_manager.cc`.

#### 8.20.2.2 ~TimeManager()

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

Destroy a [TimeManager](#).

Definition at line 505 of file `time_manager.cc`.

References `converter_vector`, and `time_vector`.

#### 8.20.2.3 TimeManager() [2/2]

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

### 8.20.3 Member Function Documentation

#### 8.20.3.1 get\_converter\_ptr()

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

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

#### Returns

[TimeConverter](#) object corresponding to index in the vector of such types.



## Parameters

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

Definition at line 88 of file time\_manager.cc.

References converter\_vector.

Referenced by jeod::JeodBaseTime::add\_type\_update(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), and jeod::TimeUDE::initialize\_initializer\_time().

## 8.20.3.2 get\_jeod\_integration\_time()

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

Expose the private inheritance from JeodIntegrationTime.

Definition at line 108 of file time\_manager.cc.

## 8.20.3.3 get\_time\_change\_flag()

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

Returns the boolean value time\_change\_flag.

## Returns

time\_change\_flag

Definition at line 120 of file time\_manager.cc.

References time\_change\_flag.

## 8.20.3.4 get\_time\_ptr() [1/2]

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

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

## Returns

Time object corresponding to name

**Parameters**

in	<i>name</i>	Name of time object
----	-------------	---------------------

Definition at line 162 of file time\_manager.cc.

References time\_lookup().

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), jeod::JeodBase↵  
Time::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 177 of file time\_manager.cc.

References time\_vector.

**8.20.3.6 get\_time\_scale\_factor()**

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

Returns the scale factor from sim time to dynamic time.

**Returns**

dyn\_time.scale\_factor

Definition at line 133 of file time\_manager.cc.

References dyn\_time, and jeod::TimeDyn::scale\_factor.

### 8.20.3.7 get\_timestamp\_time()

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

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

#### Returns

dyn\_time.seconds

Definition at line 147 of file time\_manager.cc.

References dyn\_time, and jeod::JeodBaseTime::seconds.

### 8.20.3.8 initialize()

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

initializes the time manager

#### Parameters

in	<i>time_manager_init</i>	Initialization parameters
----	--------------------------	---------------------------

Definition at line 64 of file time\_manager\_\_initialize.cc.

References dyn\_time, jeod::TimeManagerInit::initialize\_manager(), num\_types, register\_time(), time\_vector, and update().

### 8.20.3.9 operator=()

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

### 8.20.3.10 register\_converter()

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

Registers the time converters with the Time Manager.

#### Assumptions and Limitations

- the input values name\_a and name\_b will only be used if the converter-type names have not already been set. So registering a Dyn\_UDE converter will ignore name\_a completely because it is already set.

**Parameters**

in, out	<i>conv_ref</i>	ref. to converter being registered
in	<i>name</i> ↔ <i>_a</i>	name of type-a in the converter
in	<i>name</i> ↔ <i>_b</i>	name of type-b in the converter

Definition at line 266 of file time\_manager.cc.

References jeod::TimeConverter::a\_name, jeod::TimeConverter::b\_name, converter\_vector, jeod::Time↔Messages::incomplete\_setup\_error, and jeod::TimeMessages::redundancy\_error.

**8.20.3.11 register\_time()**

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

Registers the time representation with the Time Manager.

Records the frequency at which the representation should be updated.

**Assumptions and Limitations**

- None

**Parameters**

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

Definition at line 200 of file time\_manager.cc.

References jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy\_error, jeod::JeodBaseTime::set↔index(), jeod::JeodBaseTime::time\_manager, and time\_vector.

Referenced by initialize(), and register\_time\_named().

**8.20.3.12 register\_time\_named()**

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

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

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

**Assumptions and Limitations**

- None

## Parameters

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

Definition at line 234 of file `time_manager.cc`.

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

8.20.3.13 `time_lookup()`

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

Uses a string comparison to find where in the [TimeManager](#) record a time type of a particular name is located.

Returns the integer corresponding to the time type's index in the [TimeManager](#).

## Assumptions and Limitations

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

## Returns

index value of time-type

## Parameters

in	<i>name</i>	name of time-type
----	-------------	-------------------

Definition at line 355 of file `time_manager.cc`.

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

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

8.20.3.14 `time_standards_exist()`

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

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

**Assumptions and Limitations**

- None

**Returns**

true/false

Definition at line 330 of file `time_manager.cc`.

References `time_vector`.

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

**8.20.3.15 update()**

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

This function manages the time update process.

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

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

**Assumptions and Limitations**

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

**Parameters**

in	<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 419 of file `time_manager.cc`.

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

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

## 8.20.3.16 update\_time()

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

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

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

## Assumptions and Limitations

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

## Parameters

in	<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 461 of file time\_manager.cc.

References [num\\_types](#), [simtime](#), and [time\\_vector](#).

## 8.20.3.17 verify\_table\_lookup\_ends()

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

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

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

## Assumptions and Limitations

- None

Definition at line 492 of file time\_manager.cc.

References [converter\\_vector](#).

Referenced by [jeod::TimeDyn::update\\_offset\(\)](#).

## 8.20.4 Friends And Related Function Documentation

#### 8.20.4.1 init\_attrjeod\_\_TimeManager

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

#### 8.20.4.2 InputProcessor

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

Definition at line 96 of file time\_manager.hh.

#### 8.20.4.3 TimeManagerInit

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

Definition at line 98 of file time\_manager.hh.

### 8.20.5 Field Documentation

#### 8.20.5.1 converter\_vector

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

List of pointers to time-converters.

Definition at line 136 of file time\_manager.hh.

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

#### 8.20.5.2 dyn\_time

```
TimeDyn jeod::TimeManager::dyn_time
```

The instance of `TimeDyn`, the dynamic time that is used as the integration time.

`trick_units(-)`

Definition at line 113 of file time\_manager.hh.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `get_time_scale_factor()`, `get_timestamp_time()`, `initialize()`, `jeod::TimeManagerInit::initialize()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `update()`, `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.



### 8.20.5.3 num\_types

```
int jeod::TimeManager::num_types
```

Size of time\_types\_ptrs vector.

trick\_units(-)

Definition at line 118 of file time\_manager.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), jeod::JeodBase↵  
Time::add\_type\_update(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeManagerInit::create\_init\_tree(),  
jeod::TimeManagerInit::create\_update\_tree(), initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::↵  
TimeUDE::initialize\_from\_parent(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeManagerInit::initialize↵  
\_time\_types(), jeod::TimeManagerInit::organize\_update\_list(), jeod::TimeManagerInit::populate\_converter\_↵  
registry(), update(), update\_time(), jeod::TimeManagerInit::verify\_converter\_setup(), and jeod::TimeManager↵  
Init::verify\_times\_setup().

### 8.20.5.4 simtime

```
double jeod::TimeManager::simtime
```

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

trick\_units(-)

Definition at line 107 of file time\_manager.hh.

Referenced by jeod::TimeStandard::calendar\_update(), jeod::TimeStandard::seconds\_of\_year(), jeod::TimeDyn↵  
::update(), update(), jeod::TimeDyn::update\_offset(), and update\_time().

### 8.20.5.5 time\_change\_flag

```
bool jeod::TimeManager::time_change_flag [private]
```

Indicates that the dynamic scale factor changed.

trick\_units(-)

Definition at line 126 of file time\_manager.hh.

Referenced by get\_time\_change\_flag(), and update().

### 8.20.5.6 time\_vector

```
std::vector<JeodBaseTime*> jeod::TimeManager::time_vector [private]
```

List of pointers to time-types.

Definition at line 131 of file time\_manager.hh.

Referenced by jeod::TimeManagerInit::create\_init\_tree(), jeod::TimeManagerInit::create\_update\_tree(), get\_time\_ptr(), initialize(), jeod::TimeManagerInit::initialize(), jeod::TimeManagerInit::initialize\_time\_types(), jeod::TimeManagerInit::organize\_update\_list(), jeod::TimeManagerInit::populate\_converter\_registry(), register\_time(), time\_lookup(), time\_standards\_exist(), update(), update\_time(), jeod::TimeManagerInit::verify\_times\_setup(), and ~TimeManager().

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

- [time\\_manager.hh](#)
- [time\\_manager.cc](#)
- [time\\_manager\\_\\_initialize.cc](#)

## 8.21 jeod::TimeManagerInit Class Reference

To initialize the Time Manager.

```
#include <time_manager_init.hh>
```

### Public Member Functions

- [TimeManagerInit \(\)](#)  
*Construct a [TimeManagerInit](#).*
- [~TimeManagerInit \(\)](#)  
*Destroy a [TimeManagerInit](#).*
- [int get\\_conv\\_ptr\\_index \(const int conv\\_index\)](#)  
*Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the time\_manager's vector of converters that corresponds to those two time-types.*
- [int get\\_conv\\_dir\\_init \(const int conv\\_index\)](#)  
*Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".*
- [int get\\_conv\\_dir\\_upd \(const int conv\\_index\)](#)  
*Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".*
- [int get\\_status \(const int index\)](#)  
*Returns the status of a time-type.*
- [void set\\_status \(const int index, const int status\\_value\)](#)  
*Receives an updated value for the status of a time-type.*
- [void increment\\_status \(const int slave\\_index, const int master\\_index\)](#)  
*Modifies the status of one time-type to be one higher than that of another type for initialization purposes.*
- [void initialize\\_manager \(TimeManager \\*time\\_mgr\)](#)  
*The master program behind the initialization of the time types and the time converters.*
- [void organize\\_update\\_list \(\)](#)  
*Reorganizes the update list according to initialization status.*

## Data Fields

- int [num\\_added\\_total](#)  
*Count of the total number of time-types placed in the update tree or in the initialization tree.*
- [TimeEnum::TimeFormat](#) [sim\\_start\\_format](#)  
*Calendar, truncated\_julian, etc.*
- [TimeManager](#) \* [time\\_manager](#)  
*Pointer to the Time Manager.*
- std::string [initializer](#)  
*Name of the time-type used for initialization.*

## Protected Attributes

- int [initializer\\_index](#)  
*Index-value of the initializer.*
- int [dyn\\_time\\_index](#)  
*Index-value of the type dyn-time.*
- int [num\\_added\\_pass](#)  
*Count of number of time-types placed in the update tree or in the initialization tree in any given pass.*
- int \* [converter\\_ptrs\\_index](#)  
*List of the indices (in the TimeManager->time\_converter\_ptrs vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)*
- int \* [init\\_converter\\_dir\\_table](#)  
*List of directions available for initialization for each of the converters listed in converter\_class\_ptrs.*
- int \* [update\\_converter\\_dir\\_table](#)  
*List of directions available for run-time updates for each of the converters listed in converter\_class\_ptrs.*
- int \* [status](#)  
*A running ledger of properly linked times during update tree and initialization tree construction.*

## Private Member Functions

- void [initialize](#) (void)  
*The TimeManagerInit determines initialization and update paths for conversions between time-types.*
- void [verify\\_times\\_setup](#) (void)  
*A number of checks that the setup is self-consistent.*
- void [populate\\_converter\\_registry](#) (void)  
*The converter registry accounts for all of the converter functions that provide conversions between time types.*
- void [verify\\_converter\\_setup](#) (void)  
*To verify that there are no incompatibilities between specific converters.*
- void [initialize\\_time\\_types](#) (void)  
*Initialize each time type so that it has a starting value corresponding to dynamic\_time = 0 and such that the starting values are consistent.*
- void [create\\_init\\_tree](#) (void)  
*Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.*
- void [create\\_update\\_tree](#) (void)  
*(To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - create\_update\_tree builds the tree, populated recursively by add\_type\_↔ update.*
- [TimeManagerInit](#) (const [TimeManagerInit](#) &)
- [TimeManagerInit](#) & [operator=](#) (const [TimeManagerInit](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeManagerInit](#) ()

### 8.21.1 Detailed Description

To initialize the Time Manager.

Definition at line 85 of file `time_manager_init.hh`.

### 8.21.2 Constructor & Destructor Documentation

#### 8.21.2.1 TimeManagerInit() [1/2]

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

Construct a [TimeManagerInit](#).

Definition at line 64 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `dyn_time_index`, `init_converter_dir_table`, `initializer`, `initializer_index`, `num_↵`  
`added_pass`, `num_added_total`, `sim_start_format`, `status`, `jeod::TimeEnum::undefined`, and `update_converter_↵`  
`dir_table`.

#### 8.21.2.2 ~TimeManagerInit()

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

Destroy a [TimeManagerInit](#).

Definition at line 811 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `init_converter_dir_table`, `status`, and `update_converter_dir_table`.

#### 8.21.2.3 TimeManagerInit() [2/2]

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

### 8.21.3 Member Function Documentation

#### 8.21.3.1 create\_init\_tree()

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

Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.

Create\_init\_tree builds the tree, using add\_type\_initialize to populate the tree recursively.

#### Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize\_from" for each time representation, except the top-level initializer type
- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

Definition at line 430 of file time\_manager\_init.cc.

References dyn\_time\_index, jeod::TimeMessages::initialization\_error, initializer\_index, jeod::TimeMessages::invalid\_setup\_error, num\_added\_pass, num\_added\_total, jeod::TimeManager::num\_types, status, jeod::TimeManager::time\_lookup(), time\_manager, jeod::TimeManager::time\_vector, and jeod::JeodBaseTime::update\_from\_name.

Referenced by initialize\_manager().

#### 8.21.3.2 create\_update\_tree()

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

(To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - create\_update\_tree builds the tree, populated recursively by add\_type\_update.

record\_update records the update paths to facilitate runtime updates)

#### Assumptions and Limitations

- None

Definition at line 573 of file time\_manager\_init.cc.

References dyn\_time\_index, jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeMessages::initialization\_error, num\_added\_pass, num\_added\_total, jeod::TimeManager::num\_types, organize\_update\_list(), status, time\_manager, and jeod::TimeManager::time\_vector.

Referenced by initialize\_manager().

### 8.21.3.3 `get_conv_dir_init()`

```
int jeod::TimeManagerInit::get_conv_dir_init (
    const int index )
```

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".

#### Assumptions and Limitations

- Returns 0 if no suitable converter available at initialization

#### Returns

Index corresponding to [TimeConverter](#)

#### Parameters

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

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

References `init_converter_dir_table`.

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

### 8.21.3.4 `get_conv_dir_upd()`

```
int jeod::TimeManagerInit::get_conv_dir_upd (
    const int index )
```

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".

#### Assumptions and Limitations

- Returns 0 if no suitable converter available at update

#### Returns

Index corresponding to [TimeConverter](#)

#### Parameters

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

Definition at line 738 of file `time_manager_init.cc`.

References `update_converter_dir_table`.

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

#### 8.21.3.5 `get_conv_ptr_index()`

```
int jeod::TimeManagerInit::get_conv_ptr_index (
    const int index_in )
```

Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the `time_manager`'s vector of converters that corresponds to those two time-types.

##### Returns

Index corresponding to [TimeConverter](#)

##### Parameters

in	<i>index</i> ↔ _in	Index of object
----	-----------------------	-----------------

Definition at line 686 of file `time_manager_init.cc`.

References `converter_ptrs_index`.

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

#### 8.21.3.6 `get_status()`

```
int jeod::TimeManagerInit::get_status (
    const int index )
```

Returns the status of a time-type.

##### Returns

Integer corresponding to Status

##### Parameters

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

Definition at line 760 of file `time_manager_init.cc`.

References status.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::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 797 of file `time_manager_init.cc`.

References `num_added_pass`, and `status`.

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

#### 8.21.3.8 initialize()

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

The [TimeManagerInit](#) determines initialization and update paths for conversions between time-types.

This function creates and initializes the data structures necessary for these determinations

##### Assumptions and Limitations

- None

Definition at line 139 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `jeod::TimeManager::dyn_time`, `dyn_time_index`, `jeod::JeodBaseTime::index`, `init_converter_dir_table`, `jeod::JeodBaseTime::initialized`, `initializer`, `initializer_index`, `jeod::JeodBaseTime::seconds`, `status`, `jeod::TimeManager::time_lookup()`, `time_manager`, `jeod::TimeManager::time_vector`, `update_converter↔dir_table`, and `verify_times_setup()`.

Referenced by `initialize_manager()`.



## 8.21.3.9 initialize\_manager()

```
void jeod::TimeManagerInit::initialize_manager (
    TimeManager * time_mgr )
```

The master program behind the initialization of the time types and the time converters.

## Assumptions and Limitations

- None

## Parameters

in, out	time_mgr	The time manager
---------	----------	------------------

Definition at line 94 of file time\_manager\_init.cc.

References `create_init_tree()`, `create_update_tree()`, `initialize()`, `initialize_time_types()`, `populate_converter_registry()`, `time_manager`, and `verify_converter_setup()`.

Referenced by `jeod::TimeManager::initialize()`.

## 8.21.3.10 initialize\_time\_types()

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

Initialize each time type so that it has a starting value corresponding to `dynamic_time = 0` and such that the starting values are consistent.

`Initialize_time_types` repetitively calls `initialize_from_parent` for each time type; `initialize_from_parent` recursively adds types moving up the tree as necessary

## Assumptions and Limitations

- An initializer time defined by the user

Definition at line 540 of file time\_manager\_init.cc.

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

**8.21.3.12 organize\_update\_list()**

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

Reorganizes the update list according to initialization status.

Definition at line 636 of file `time_manager_init.cc`.

References `jeod::TimeManager::num_types`, `status`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `create_update_tree()`.

**8.21.3.13 populate\_converter\_registry()**

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

The converter registry accounts for all of the converter functions that provide conversions between time types.

This function populates that registry so that the existence of functional converter functions can be tested efficiently.

**Assumptions and Limitations**

- None

Definition at line 275 of file `time_manager_init.cc`.

References `jeod::TimeConverter::A_TO_B_INIT`, `jeod::TimeConverter::A_TO_B_UPDATE`, `jeod::TimeConverter::B_TO_A_INIT`, `jeod::TimeConverter::B_TO_A_UPDATE`, `converter_ptrs_index`, `jeod::TimeManager::converter_vector`, `init_converter_dir_table`, `jeod::TimeManager::num_types`, `jeod::TimeMessages::redundancy_error`, `jeod::TimeManager::time_lookup()`, `time_manager`, `jeod::TimeManager::time_vector`, and `update_converter_dir_table`.

Referenced by `initialize_manager()`.

**8.21.3.14 set\_status()**

```
void jeod::TimeManagerInit::set_status (
    const int index,
    const int new_status )
```

Receives an updated value for the status of a time-type.

## Parameters

in	<i>index</i>	Index of object
in	<i>new_status</i>	New status value

Definition at line 781 of file `time_manager_init.cc`.

References `status`.

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

8.21.3.15 `verify_converter_setup()`

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

To verify that there are no incompatibilities between specific converters.

## Assumptions and Limitations

- The instance of `TimeTAI`, if it exists, has name "TAI"
- The instance of `TimeUTC`, if it exists, has name "UTC"
- The instance of `TimeUT1`, if it exists, has name "UT1"

Definition at line 367 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeManager::num_types`, `jeod::TimeConverter_TAI_UTC::override_data_table`, `jeod::TimeConverter_TAI_UT1::override_data_table`, `jeod::↵TimeManager::time_lookup()`, and `time_manager`.

Referenced by `initialize_manager()`.

8.21.3.16 `verify_times_setup()`

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

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

## Assumptions and Limitations

- None

Definition at line 205 of file `time_manager_init.cc`.

References `jeod::TimeMessages::incomplete_setup_error`, `initializer`, `initializer_index`, `jeod::TimeMessages↵::invalid_setup_error`, `jeod::TimeManager::num_types`, `jeod::TimeMessages::redundancy_error`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `initialize()`.

## 8.21.4 Friends And Related Function Documentation

### 8.21.4.1 init\_attrjeod\_\_TimeManagerInit

```
void init_attrjeod__TimeManagerInit ( ) [friend]
```

### 8.21.4.2 InputProcessor

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

Definition at line 87 of file time\_manager\_init.hh.

## 8.21.5 Field Documentation

### 8.21.5.1 converter\_ptrs\_index

```
int* jeod::TimeManagerInit::converter_ptrs_index [protected]
```

List of the indices (in the TimeManager->time\_converter\_ptrs vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)

trick\_units(-)

Definition at line 132 of file time\_manager\_init.hh.

Referenced by get\_conv\_ptr\_index(), initialize(), populate\_converter\_registry(), TimeManagerInit(), verify\_↔converter\_setup(), and ~TimeManagerInit().

### 8.21.5.2 dyn\_time\_index

```
int jeod::TimeManagerInit::dyn_time_index [protected]
```

Index-value of the type dyn-time.

trick\_units(-)

Definition at line 119 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), create\_update\_tree(), initialize(), and TimeManagerInit().

### 8.21.5.3 init\_converter\_dir\_table

```
int* jeod::TimeManagerInit::init_converter_dir_table [protected]
```

List of directions available for initialization for each of the converters listed in converter\_class\_ptrs.

trick\_units(-)

Definition at line 137 of file time\_manager\_init.hh.

Referenced by get\_conv\_dir\_init(), initialize(), populate\_converter\_registry(), TimeManagerInit(), and ~TimeManagerInit().

### 8.21.5.4 initializer

```
std::string jeod::TimeManagerInit::initializer
```

Name of the time-type used for initialization.

trick\_units(-)

Definition at line 109 of file time\_manager\_init.hh.

Referenced by initialize(), TimeManagerInit(), and verify\_times\_setup().

### 8.21.5.5 initializer\_index

```
int jeod::TimeManagerInit::initializer_index [protected]
```

Index-value of the initializer.

trick\_units(-)

Definition at line 114 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), initialize(), initialize\_time\_types(), TimeManagerInit(), and verify\_times\_setup().

### 8.21.5.6 num\_added\_pass

```
int jeod::TimeManagerInit::num_added_pass [protected]
```

Count of number of time-types placed in the update tree or in the initialization tree in any given pass.

trick\_units(-)

Definition at line 124 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), create\_update\_tree(), increment\_status(), and TimeManagerInit().

### 8.21.5.7 num\_added\_total

```
int jeod::TimeManagerInit::num_added_total
```

Count of the total number of time-types placed in the update tree or in the initialization tree.

trick\_units(-)

Definition at line 96 of file time\_manager\_init.hh.

Referenced by jeod::JeodBaseTime::add\_type\_update(), create\_init\_tree(), create\_update\_tree(), and TimeManagerInit().

### 8.21.5.8 sim\_start\_format

```
TimeEnum::TimeFormat jeod::TimeManagerInit::sim_start_format
```

Calendar, truncated\_julian, etc.

trick\_units(-)

Definition at line 100 of file time\_manager\_init.hh.

Referenced by jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), and TimeManagerInit().

### 8.21.5.9 status

```
int* jeod::TimeManagerInit::status [protected]
```

A running ledger of properly linked times during update tree and initialization tree construction.

Entries correspond to times of shared indexes in time\_vector ( e.g. status[2] : status of time\_manager->time\_vector[2] ) Update tree encoding: -2: undefined. Requires auto-assignment or causes error. -1: definitive error. Process will terminate. 0: uninitialized 1: THE 1st generation (root) time. dyn\_time for update tree. 2: a 2nd generation time, converted from root time. n: a nth gen time, converted from (n-1)th gen time.trick\_units(-)

Definition at line 156 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), create\_update\_tree(), get\_status(), increment\_status(), initialize(), organize\_update\_list(), set\_status(), TimeManagerInit(), and ~TimeManagerInit().

## 8.21.5.10 time\_manager

`TimeManager*` jeod::TimeManagerInit::time\_manager

Pointer to the Time Manager.

Automatically linked during init routines.trick\_units(-)

Definition at line 104 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), create\_update\_tree(), initialize(), initialize\_manager(), initialize\_time\_types(), organize\_update\_list(), populate\_converter\_registry(), verify\_converter\_setup(), and verify\_times\_setup().

## 8.21.5.11 update\_converter\_dir\_table

`int*` jeod::TimeManagerInit::update\_converter\_dir\_table [protected]

List of directions available for run-time updates for each of the converters listed in converter\_class\_ptrs.

trick\_units(-)

Definition at line 142 of file time\_manager\_init.hh.

Referenced by get\_conv\_dir\_upd(), initialize(), populate\_converter\_registry(), TimeManagerInit(), and ~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>
```

## Static Public Attributes

- static char const \* [initialization\\_error](#)  
*Error issued when initialization fails due to some non-obvious cause.*
- static char const \* [memory\\_error](#)  
*Error issued when system fails because something is not where it should be, or has a value other than its assumed value.*
- static char const \* [invalid\\_setup\\_error](#)  
*Error issued when user tries to use something that doesn't exist in the model.*
- static char const \* [invalid\\_data\\_error](#)  
*Error issued when a variable is found with an illegal value.*
- static char const \* [invalid\\_node](#)  
*Issued when a [TimeLinks](#) node is improperly linked.*
- static char const \* [incomplete\\_setup\\_error](#)  
*Error issued when user tries to use something that doesn't exist in the simulation. This is usually a user error, brought about by not having registered something that is later needed (e.g.*
- static char const \* [redundancy\\_error](#)  
*Error issued when some value is multiply defined, and the code cannot determine which value to use.*
- static char const \* [duplicate\\_methods](#)  
*Informational only.*
- static char const \* [extension\\_error](#)  
*Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.*

## Private Member Functions

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

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeMessages](#) ()

### 8.22.1 Detailed Description

Specify the message IDs used in the Time model.

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

### 8.22.2 Constructor & Destructor Documentation

#### 8.22.2.1 TimeMessages() [1/2]

```
jeod::TimeMessages::TimeMessages (
    void ) [private]
```

#### 8.22.2.2 TimeMessages() [2/2]

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

### 8.22.3 Member Function Documentation

#### 8.22.3.1 operator=()

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

### 8.22.4 Friends And Related Function Documentation



#### 8.22.4.1 init\_attrjeod\_\_TimeMessages

```
void init_attrjeod__TimeMessages ( ) [friend]
```

#### 8.22.4.2 InputProcessor

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

Definition at line 87 of file time\_messages.hh.

### 8.22.5 Field Documentation

#### 8.22.5.1 duplicate\_methods

```
char const * jeod::TimeMessages::duplicate_methods [static]
```

##### Initial value:

```
=  
    "environment/time/" "duplicate_methods"
```

Informational only.

Issued when there are multiple equivalent methods for doing something, and one method is chosen over another. Wherever the code fills in data to accommodate, it sends this informational broadcast.trick\_units(—)

Definition at line 146 of file time\_messages.hh.

Referenced by jeod::TimeUDE::initialize\_initializer\_time().

#### 8.22.5.2 extension\_error

```
char const * jeod::TimeMessages::extension_error [static]
```

##### Initial value:

```
=  
    "environment/time/" "extension_error"
```

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

trick\_units(—)

Definition at line 152 of file time\_messages.hh.

### 8.22.5.3 incomplete\_setup\_error

```
char const * jeod::TimeMessages::incomplete_setup_error [static]
```

#### Initial value:

```
=
    "environment/time/" "incomplete_setup_error"
```

Error issued when user tries to use something that doesn't exist in the simulation. This is usually a user error, brought about by not having registered something that is later needed (e.g.

not registering a TAI-UTC converter, but specifying that UTC updates from TAI). Note the distinction between invalid (typically, cannot exist) and incomplete (typically, did not define) `trick_units(-)`

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

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, `jeod::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::TimeUDE::initialize_from_parent()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeUDE::initialize_initializer_time()`, `jeod::TimeManager::register_converter()`, `jeod::TimeUDE::set_epoch_dyn()`, `jeod::TimeUDE::set_epoch_std()`, `jeod::TimeUDE::set_epoch_ude()`, `jeod::TimeUDE::set_initial_times()`, `jeod::TimeUDE::verify_epoch()`, `jeod::TimeManagerInit::verify_times_setup()`, and `jeod::TimeUDE::verify_update()`.

### 8.22.5.4 initialization\_error

```
char const * jeod::TimeMessages::initialization_error [static]
```

#### Initial value:

```
=
    "environment/time/" "initialization_error"
```

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

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

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

Referenced by `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeManagerInit::create_update_tree()`, `jeod::TimeConverter_Dyn_TAI::initialize()`, `jeod::TimeConverter_Dyn_TDB::initialize()`, `jeod::TimeConverter_TAI_UTC::initialize()`, `jeod::TimeConverter_TAI_UT1::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeConverter::verify_setup()`.

## 8.22.5.5 invalid\_data\_error

```
char const * jeod::TimeMessages::invalid_data_error [static]
```

**Initial value:**

```
=
    "environment/time/" "invalid_data_error"
```

Error issued when a variable is found with an illegal value.

This is usually a user error, having set some value externally to some unrecognizable value.`trick_units(-)`

Definition at line 117 of file `time_messages.hh`.

Referenced by `jeod::TimeGMST::calculate_calendar_values()`, `jeod::TimeGPS::calculate_calendar_values()`, `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `jeod::TimeGPS::convert_from_calendar()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, and `jeod::TimeGMST::set_time_by_trunc_julian()`.

## 8.22.5.6 invalid\_node

```
char const * jeod::TimeMessages::invalid_node [static]
```

**Initial value:**

```
=
    "environment/time/" "invalid_node"
```

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

`trick_units(-)`

Definition at line 122 of file `time_messages.hh`.

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

### 8.22.5.7 invalid\_setup\_error

```
char const * jeod::TimeMessages::invalid_setup_error [static]
```

#### Initial value:

```
=
    "environment/time/" "invalid_setup_error"
```

Error issued when user tries to use something that doesn't exist in the model.

This is usually a user error. Note the distinction between invalid and incompletetrack\_units(—)

Definition at line 110 of file time\_messages.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), jeod::JeodBaseTime::add\_type\_initialize(), jeod::JeodBaseTime::add\_type\_update(), jeod::TimeConverter::convert\_a\_to\_b(), jeod::TimeConverter::convert\_b\_to\_a(), jeod::TimeManagerInit::create\_init\_tree(), jeod::TimeConverter\_TAI::initialize(), jeod::TimeConverter\_UT1\_GMST::initialize(), jeod::TimeConverter\_TAI\_GPS::initialize(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::TimeConverter\_Dyn\_TDB::initialize(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeConverter\_TAI\_TDB::initialize(), jeod::JeodBaseTime::initialize\_from\_parent(), jeod::TimeDyn::initialize\_initializer\_time(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeConverter\_TAI\_UTC::initialize\_leap\_second(), jeod::TimeUDE::set\_epoch\_dyn(), jeod::TimeUDE::set\_epoch\_initializing\_value(), jeod::TimeUDE::set\_epoch\_std(), jeod::TimeUDE::set\_epoch\_times(), jeod::TimeUDE::set\_epoch\_ude(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeManager::time\_lookup(), jeod::TimeManagerInit::verify\_converter\_setup(), jeod::TimeUDE::verify\_epoch(), jeod::TimeConverter::verify\_setup(), jeod::TimeManagerInit::verify\_times\_setup(), and jeod::TimeUDE::verify\_update().

### 8.22.5.8 memory\_error

```
char const * jeod::TimeMessages::memory_error [static]
```

#### Initial value:

```
=
    "environment/time/" "memory_error"
```

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

track\_units(—)

Definition at line 103 of file time\_messages.hh.

Referenced by jeod::JeodBaseTime::add\_type\_update(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::set\_initial\_times(), and jeod::JeodBaseTime::update().

## 8.22.5.9 redundancy\_error

```
char const * jeod::TimeMessages::redundancy_error [static]
```

**Initial value:**

```
=  
    "environment/time/" "redundancy_error"
```

Error issued when some value is multiply defined, and the code cannot determine which value to use.

Usually a user-error, from attempting to use too many of the initialization options simultaneously.`trick_units(-)`

Definition at line 139 of file `time_messages.hh`.

Referenced by `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeManagerInit::populate_converter_registry()`, `jeod::TimeManager::register_converter()`, `jeod::TimeManager::register_time()`, `jeod::TimeUDE::set_epoch_dyn()`, `jeod::TimeUDE::set_initial_times()`, `jeod::TimeUDE::verify_epoch()`, `jeod::TimeUDE::verify_init()`, and `jeod::TimeManagerInit::verify_times_setup()`.

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

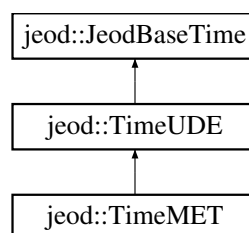
- [time\\_messages.hh](#)
- [time\\_messages.cc](#)

## 8.23 jeod::TimeMET Class Reference

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

```
#include <time_met.hh>
```

Inheritance diagram for `jeod::TimeMET`:

**Public Member Functions**

- [TimeMET](#) ()
- [~TimeMET](#) ()  
*Destroy a Time\_MET.*
- void [update](#) (void)  
*Updates to current time.*

## Data Fields

- bool `hold`

*Flags whether to hold time at current value.*

## Private Member Functions

- `TimeMET` (const `TimeMET` &)
- `TimeMET` & `operator=` (const `TimeMET` &)

## Private Attributes

- bool `previous_hold`

*Previously known value of hold, used for recalculating converters.*

## Friends

- class `InputProcessor`
- void `init_attrjeod__TimeMET` ()

## Additional Inherited Members

### 8.23.1 Detailed Description

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

Definition at line 83 of file `time_met.hh`.

### 8.23.2 Constructor & Destructor Documentation

#### 8.23.2.1 `TimeMET()` [1/2]

```
jeod::TimeMET::TimeMET (  
    void )
```

Definition at line 68 of file `time_met.cc`.

References `jeod::JeodBaseTime::name`.

#### 8.23.2.2 ~TimeMET()

```
jeod::TimeMET::~~TimeMET (
    void )
```

Destroy a Time\_MET.

Definition at line 110 of file time\_met.cc.

#### 8.23.2.3 TimeMET() [2/2]

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

### 8.23.3 Member Function Documentation

#### 8.23.3.1 operator=()

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

#### 8.23.3.2 update()

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

Updates to current time.

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

Definition at line 82 of file time\_met.cc.

References [hold](#), [previous\\_hold](#), [jeod::TimeConverter::reset\\_a\\_to\\_b\\_offset\(\)](#), [jeod::JeodBaseTime::update\(\)](#), and [jeod::JeodBaseTime::update\\_converter\\_ptr](#).

### 8.23.4 Friends And Related Function Documentation

#### 8.23.4.1 init\_attrjeod\_\_TimeMET

```
void init_attrjeod__TimeMET ( ) [friend]
```

#### 8.23.4.2 InputProcessor

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

Definition at line 85 of file time\_met.hh.

### 8.23.5 Field Documentation

#### 8.23.5.1 hold

```
bool jeod::TimeMET::hold
```

Flags whether to hold time at current value.

trick\_units(-)

Definition at line 92 of file time\_met.hh.

Referenced by update().

#### 8.23.5.2 previous\_hold

```
bool jeod::TimeMET::previous_hold [private]
```

Previously known value of hold, used for recalculating converters.

trick\_units(-)

Definition at line 98 of file time\_met.hh.

Referenced by update().

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

- [time\\_met.hh](#)
- [time\\_met.cc](#)

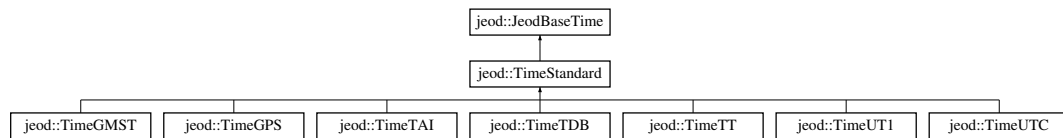


## 8.24 jeod::TimeStandard Class Reference

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

```
#include <time_standard.hh>
```

Inheritance diagram for jeod::TimeStandard:



### Public Member Functions

- [TimeStandard](#) ()  
*Construct a [TimeStandard](#).*
- virtual [~TimeStandard](#) ()  
*Destroy a [TimeStandard](#).*
- void [calendar\\_update](#) (double simtime)  
*Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation.*
- void [initialize\\_initializer\\_time](#) ([TimeManagerInit](#) \*tm\_init)  
*Each time type is initialized from its parent in the initialization tree, except one.*
- void [add\\_type\\_initialize](#) (const int seeking\_status, [TimeManagerInit](#) \*tm\_init)  
*Recursively adds elements to the initialization tree.*
- void [initialize\\_from\\_parent](#) ([TimeManagerInit](#) \*tm\_init)  
*Initialize a time type from its parent on the initialization tree.*
- virtual void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)  
*Given a value of seconds, propagate to days and trunc\_julian\_time.*
- virtual void [set\\_time\\_by\\_days](#) (const double new\_days)  
*Given a value of days, propagate to seconds and trunc\_julian\_time.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double new\_tjt)  
*Given a value of tjt, propagate to seconds and days.*
- double [julian\\_date\\_at\\_epoch](#) (void)  
*Returns the full Julian date at epoch, rather than the Truncated Julian Time.*
- double [seconds\\_of\\_year](#) (void)  
*Generate the number of seconds elapsed this year.*

### Data Fields

- double [last\\_calendar\\_update](#)  
*The simtime when the calendar update was last run.*
- int [prev\\_julian\\_day](#)  
*Used for determining whether to update the date in the calendar function.*
- double [seconds\\_at\\_year\\_start](#)  
*The value of "seconds" at the start of the year in which the last seconds\_of\_year calculation was made.*
- int [year\\_of\\_last\\_soy](#)  
*The year in which the last seconds\_of\_year calculation was made.*
- bool [send\\_warning\\_pre\\_1968](#)

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

- const double [tjt\\_mjt\\_offset](#)

*Difference between Truncated Julian and Modified Julian.*

- const double [tjt\\_jd\\_offset](#)

*Difference between Julian and Truncated Julian.*

- double [trunc\\_julian\\_time](#)

*Truncated Julian time for this time-type.*

- double [julian\\_date](#)

*Conventional Julian Date.*

- double [tjt\\_at\\_epoch](#)

*Truncated Julian Date at epoch.*

- int [calendar\\_day](#)

*Gregorian calendar date day number.*

- int [calendar\\_hour](#)

*24-hour clock hour number.*

- int [calendar\\_minute](#)

*Clock minute number.*

- double [calendar\\_second](#)

*Clock second number.*

- int [calendar\\_year](#)

*Gregorian calendar year.*

- int [calendar\\_month](#)

*Gregorian calendar month.*

## Protected Member Functions

- virtual void [convert\\_from\\_calendar](#) (void)

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

- virtual void [calculate\\_calendar\\_values](#) (void)

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

- virtual void [set\\_epoch](#) (void)=0

*Set the epoch time.*

## Private Member Functions

- [TimeStandard](#) (const [TimeStandard](#) &)
- [TimeStandard](#) & [operator=](#) (const [TimeStandard](#) &)

## Friends

- class [InputProcessor](#)
- class [TimeUDE](#)
- void [init\\_attrjeod\\_\\_TimeStandard](#) ()

## Additional Inherited Members

### 8.24.1 Detailed Description

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

Definition at line 89 of file `time_standard.hh`.

## 8.24.2 Constructor & Destructor Documentation

### 8.24.2.1 TimeStandard() [1/2]

```
jeod::TimeStandard::TimeStandard (  
    void )
```

Construct a [TimeStandard](#).

Definition at line 60 of file time\_standard.cc.

### 8.24.2.2 ~TimeStandard()

```
jeod::TimeStandard::~~TimeStandard (  
    void ) [virtual]
```

Destroy a [TimeStandard](#).

Definition at line 771 of file time\_standard.cc.

### 8.24.2.3 TimeStandard() [2/2]

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

## 8.24.3 Member Function Documentation

### 8.24.3.1 add\_type\_initialize()

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

Recursively adds elements to the initialization tree.

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

#### Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize\_from" for each time representation, except the top-level initializer type.
- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

**Parameters**

in	<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 176 of file `time_standard.cc`.

References `jeod::TimeManagerInit::get_conv_dir_init()`, `jeod::TimeManagerInit::get_status()`, `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeManagerInit::increment_status()`, `jeod::JeodBaseTime::index`, `jeod::JeodBaseTime::initialize_from_name`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::TimeManager::num_types`, `jeod::TimeManagerInit::set_status()`, `jeod::TimeManager::time_lookup()`, and `jeod::JeodBaseTime::time_manager`.

**8.24.3.2 calculate\_calendar\_values()**

```
void jeod::TimeStandard::calculate_calendar_values (
    void ) [protected], [virtual]
```

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

**Assumptions and Limitations**

- Coverage is from March 1, 1600 onward.
- Produces a time in 24-hour clock format.
- Assumes that the values year, month, day, hour, minute, second, and truncated\_julian\_time are all present in the same class.

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

Definition at line 295 of file `time_standard.cc`.

References `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `jeod::JeodBaseTime::clock_resolution`, `prev_julian_day`, and `trunc_julian_time`.

Referenced by `calendar_update()`, and `seconds_of_year()`.

**8.24.3.3 calendar\_update()**

```
void jeod::TimeStandard::calendar_update (
    double simtime )
```

Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation.

Makes sure that the time type on which it is called is up-to-date before doing so.

**Assumptions and Limitations**

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

## Parameters

in	<i>simtime</i>	Simulation elapsed time, on the simulation clock Units: s
----	----------------	--

Definition at line 395 of file `time_standard.cc`.

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

8.24.3.4 `convert_from_calendar()`

```
void jeod::TimeStandard::convert_from_calendar (
    void ) [protected], [virtual]
```

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

## Assumptions and Limitations

- Coverage s from March 1, 1600 onward.
- Assumes that time is in 24-hour clock format; 1:00:00 pm cannot be read correctly, but 13:00:00 can.
- Assumes that the values year, month, day, hour, minute, second, and truncated\_julian\_time are all present in the same class.

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

Definition at line 425 of file `time_standard.cc`.

References `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `jeod::JeodBaseTime::days`, `jeod::JeodBaseTime::seconds`, `tjt_at_epoch`, and `trunc_julian_time`.

Referenced by `initialize_initializer_time()`, `seconds_of_year()`, and `jeod::TimeUDE::set_epoch_std()`.

8.24.3.5 `initialize_from_parent()`

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

Initialize a time type from its parent on the initialization tree.

## Assumptions and Limitations

- More than 1 time-type defined, otherwise this is not called.

## Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

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

Definition at line 627 of file `time_standard.cc`.

References `jeod::TimeConverter::convert_a_to_b()`, `jeod::TimeConverter::convert_b_to_a()`, `jeod::TimeManagerInit::get_conv_dir_init()`, `jeod::TimeManagerInit::get_conv_ptr_index()`, `jeod::TimeManager::get_converter_ptr()`, `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::JeodBaseTime::index`, `jeod::JeodBaseTime::initial_value`, `jeod::TimeMessages::initialization_error`, `jeod::TimeConverter::initialize()`, `jeod::JeodBaseTime::initialize_from_name`, `jeod::JeodBaseTime::initialize_from_parent()`, `jeod::JeodBaseTime::initialized`, `jeod::TimeConverter::is_initialized()`, `jeod::JeodBaseTime::is_initialized()`, `jeod::TimeMessages::memory_error`, `jeod::JeodBaseTime::name`, `jeod::TimeManager::num_types`, `jeod::JeodBaseTime::seconds`, `jeod::TimeManager::time_lookup()`, and `jeod::JeodBaseTime::time_manager`.

### 8.24.3.6 initialize\_initializer\_time()

```
void jeod::TimeStandard::initialize_initializer_time (
    TimeManagerInit * time_manager_init ) [virtual]
```

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

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

#### Assumptions and Limitations

- [TimeDyn](#) cannot be used as the initializer time.
- Each time representation can have its own initializer function, or can inherit the one in [TimeStandard](#).

## Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 484 of file `time_standard.cc`.

References `jeod::TimeEnum::calendar`, `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `jeod::TimeEnum::clock`, `convert_from_calendar()`, `jeod::JeodBaseTime::days`, `jeod::TimeEnum::days_since_epoch`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::JeodBaseTime::initial_value`, `jeod::JeodBaseTime::initialize_from_name`, `jeod::JeodBaseTime::initialized`, `jeod::JeodBaseTime::initializing_value`, `jeod::TimeMessages::invalid_data_error`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeEnum::Julian`, `jeod::TimeEnum::julian`, `jeod::TimeEnum::modified_julian`, `jeod::JeodBaseTime::name`, `jeod::TimeMessages::redundancy_error`, `jeod::JeodBaseTime::seconds`, `jeod::TimeEnum::seconds_since_epoch`, `send_warning_pre_1968`, `jeod::TimeManagerInit::sim_start_format`, `tjt_at_epoch`, `trunc_julian_time`, `jeod::TimeEnum::truncated_julian`, and `jeod::TimeEnum::undefined`.

### 8.24.3.7 julian\_date\_at\_epoch()

```
double jeod::TimeStandard::julian_date_at_epoch (
    void )
```

Returns the full Julian date at epoch, rather than the Truncated Julian Time.

#### Returns

Truncated Julian Time at the epoch of the time-type.

Units: day

Definition at line 148 of file time\_standard.cc.

References `tjt_at_epoch`, and `tjt_jd_offset`.

### 8.24.3.8 operator=()

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

### 8.24.3.9 seconds\_of\_year()

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

Generate the number of seconds elapsed this year.

#### Assumptions and Limitations

- Relies on the accuracy of the JEOD2.0 calendar.

#### Returns

Current second of year.

Definition at line 709 of file time\_standard.cc.

References `calculate_calendar_values()`, `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `convert_from_calendar()`, `jeod::JeodBaseTime::days`, `last_calendar_update`, `jeod::JeodBaseTime::seconds`, `seconds_at_year_start`, `jeod::TimeManager::simtime`, `jeod::JeodBaseTime::time←_manager`, `trunc_julian_time`, and `year_of_last_soy`.

#### 8.24.3.10 `set_epoch()`

```
virtual void jeod::TimeStandard::set_epoch (
    void ) [protected], [pure virtual]
```

Set the epoch time.

Implemented in [jeod::TimeGPS](#), [jeod::TimeUT1](#), [jeod::TimeUTC](#), [jeod::TimeGMST](#), [jeod::TimeTDB](#), [jeod::TimeTT](#), and [jeod::TimeTAI](#).

#### 8.24.3.11 `set_time_by_days()`

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

Given a value of days, propagate to seconds and trunc\_julian\_time.

#### Assumptions and Limitations

- 86400 seconds = 1 day

#### Parameters

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

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

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

Definition at line 112 of file time\_standard.cc.

References [jeod::JeodBaseTime::days](#), [julian\\_date](#), [jeod::JeodBaseTime::set\\_time\\_by\\_days\(\)](#), [tjt\\_at\\_epoch](#), [tjt\\_↔jd\\_offset](#), and [trunc\\_julian\\_time](#).

Referenced by [jeod::TimeConverter\\_UT1\\_GMST::convert\\_a\\_to\\_b\(\)](#), and [jeod::TimeUDE::set\\_epoch\\_std\(\)](#).

#### 8.24.3.12 `set_time_by_seconds()`

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

Given a value of seconds, propagate to days and trunc\_julian\_time.

#### Assumptions and Limitations

- 86400 seconds = 1 day



## Parameters

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

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

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

Definition at line 93 of file `time_standard.cc`.

References [jeod::JeodBaseTime::days](#), [julian\\_date](#), [jeod::JeodBaseTime::set\\_time\\_by\\_seconds\(\)](#), [tjt\\_at\\_epoch](#), [tjt\\_jd\\_offset](#), and [trunc\\_julian\\_time](#).

Referenced by [jeod::TimeConverter\\_TAI\\_TT::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter\\_Dyn\\_TAI::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter\\_Dyn\\_TDB::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter\\_TAI\\_TDB::convert\\_a\\_to\\_b\(\)](#), [jeod::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::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_tjt</i>	new value for Truncated Julian Time Units: day
----	----------------	---

Definition at line 131 of file `time_standard.cc`.

References [jeod::JeodBaseTime::days](#), [julian\\_date](#), [jeod::JeodBaseTime::seconds](#), [tjt\\_at\\_epoch](#), [tjt\\_jd\\_offset](#), and [trunc\\_julian\\_time](#).

Referenced by [jeod::TimeConverter\\_TAI\\_UTC::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter\\_TAI\\_UT1::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter\\_TAI\\_UTC::convert\\_b\\_to\\_a\(\)](#), [jeod::TimeConverter\\_TAI\\_UT1::convert\\_b\\_to\\_a\(\)](#), [jeod::TimeUDE::set\\_epoch\\_std\(\)](#), and [jeod::TimeGPS::set\\_time\\_by\\_trunc\\_julian\(\)](#).

### 8.24.4 Friends And Related Function Documentation

#### 8.24.4.1 init\_attrjeod\_\_TimeStandard

```
void init_attrjeod__TimeStandard ( ) [friend]
```

#### 8.24.4.2 InputProcessor

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

Definition at line 91 of file time\_standard.hh.

#### 8.24.4.3 TimeUDE

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

Definition at line 93 of file time\_standard.hh.

### 8.24.5 Field Documentation

#### 8.24.5.1 calendar\_day

```
int jeod::TimeStandard::calendar_day
```

Gregorian calendar date day number.

trick\_units(day)

Definition at line 161 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_↔ year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.2 calendar\_hour

```
int jeod::TimeStandard::calendar_hour
```

24-hour clock hour number.

trick\_units(hr)

Definition at line 166 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_↔ year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.3 calendar\_minute

```
int jeod::TimeStandard::calendar_minute
```

Clock minute number.

trick\_units(min)

Definition at line 171 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.4 calendar\_month

```
int jeod::TimeStandard::calendar_month
```

Gregorian calendar month.

trick\_units(-)

Definition at line 186 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.5 calendar\_second

```
double jeod::TimeStandard::calendar_second
```

Clock second number.

trick\_units(s)

Definition at line 176 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.6 calendar\_year

```
int jeod::TimeStandard::calendar_year
```

Gregorian calendar year.

trick\_units(-)

Definition at line 181 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.7 julian\_date

```
double jeod::TimeStandard::julian_date
```

Conventional Julian Date.

NOTE - because this value is typically so large, it has very little room for fine-detail precision. It should only ever be used as an output for the likes of terminal displays and for input to legacy code. Never use for newly developed code.`trick_units(day)`

Definition at line 151 of file `time_standard.hh`.

Referenced by `set_time_by_days()`, `set_time_by_seconds()`, and `set_time_by_trunc_julian()`.

#### 8.24.5.8 last\_calendar\_update

```
double jeod::TimeStandard::last_calendar_update
```

The simtime when the calendar update was last run.

`trick_units(-)`

Definition at line 101 of file `time_standard.hh`.

Referenced by `calendar_update()`, and `seconds_of_year()`.

#### 8.24.5.9 prev\_julian\_day

```
int jeod::TimeStandard::prev_julian_day
```

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

`trick_units(day)`

Definition at line 107 of file `time_standard.hh`.

Referenced by `calculate_calendar_values()`.

#### 8.24.5.10 seconds\_at\_year\_start

```
double jeod::TimeStandard::seconds_at_year_start
```

The value of "seconds" at the start of the year in which the last `seconds_of_year` calculation was made.

Used for `seconds_of_year` calculations only.`trick_units(s)`

Definition at line 114 of file `time_standard.hh`.

Referenced by `seconds_of_year()`.

#### 8.24.5.11 send\_warning\_pre\_1968

```
bool jeod::TimeStandard::send_warning_pre_1968
```

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

The flag defaults to true - warning will be sent.`trick_units(-)`

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

Referenced by `initialize_initializer_time()`.

#### 8.24.5.12 tjt\_at\_epoch

```
double jeod::TimeStandard::tjt_at_epoch
```

Truncated Julian Date at epoch.

`trick_units(day)`

Definition at line 156 of file `time_standard.hh`.

Referenced by `convert_from_calendar()`, `jeod::TimeConverter_TAI_GPS::initialize()`, `jeod::TimeConverter_TAI_TDB::initialize()`, `initialize_initializer_time()`, `julian_date_at_epoch()`, `jeod::TimeConverter_TAI_TDB::set_a_to_b_offset()`, `jeod::TimeTAI::set_epoch()`, `jeod::TimeTT::set_epoch()`, `jeod::TimeTDB::set_epoch()`, `jeod::TimeUTC::set_epoch()`, `jeod::TimeUT1::set_epoch()`, `jeod::TimeGPS::set_epoch()`, `set_time_by_days()`, `set_time_by_seconds()`, and `set_time_by_trunc_julian()`.

#### 8.24.5.13 tjt\_jd\_offset

```
const double jeod::TimeStandard::tjt_jd_offset
```

Difference between Julian and Truncated Julian.

`trick_units(day)`

Definition at line 138 of file `time_standard.hh`.

Referenced by `julian_date_at_epoch()`, `set_time_by_days()`, `set_time_by_seconds()`, and `set_time_by_trunc_julian()`.

#### 8.24.5.14 tjt\_mjt\_offset

```
const double jeod::TimeStandard::tjt_mjt_offset
```

Difference between Truncated Julian and Modified Julian.

`trick_units(day)`

Definition at line 133 of file `time_standard.hh`.

#### 8.24.5.15 trunc\_julian\_time

```
double jeod::TimeStandard::trunc_julian_time
```

Truncated Julian time for this time-type.

```
trick_units(day)
```

Definition at line 143 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), jeod::TimeConverter\_TAI\_UTC::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UT1::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UTC::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1::convert\_b\_to\_a(), convert\_from\_calendar(), jeod::TimeConverter\_TAI\_UTC::initialize(), jeod::TimeConverter\_TAI\_UT1::initialize(), initialize\_initializer\_time(), jeod::TimeConverter\_TAI\_UTC::initialize\_leap\_second(), jeod::TimeConverter\_TAI\_UT1::initialize\_tai\_to\_ut1(), seconds\_of\_year(), jeod::TimeConverter\_TAI\_TDB::set\_a\_to\_b\_offset(), set\_time\_by\_days(), set\_time\_by\_seconds(), set\_time\_by\_trunc\_julian(), jeod::TimeConverter\_TAI\_UTC::verify\_table\_lookup\_ends(), and jeod::TimeConverter\_TAI\_UT1::verify\_table\_lookup\_ends().

#### 8.24.5.16 year\_of\_last\_soy

```
int jeod::TimeStandard::year_of_last_soy
```

The year in which the last seconds\_of\_year calculation was made.

At the start of this year, seconds had value seconds\_at\_year\_start. Used for seconds\_of\_year calculations only.↵  
trick\_units(-)

Definition at line 121 of file time\_standard.hh.

Referenced by seconds\_of\_year().

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

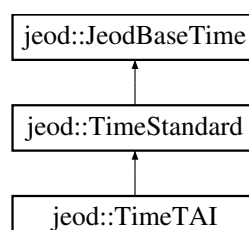
- [time\\_standard.hh](#)
- [time\\_standard.cc](#)

## 8.25 jeod::TimeTAI Class Reference

Represents International Atomic Time.

```
#include <time_tai.hh>
```

Inheritance diagram for jeod::TimeTAI:



## Public Member Functions

- [TimeTAI](#) ()  
*Construct a Time\_TAI.*
- [~TimeTAI](#) ()  
*Destroy a Time\_TAI.*

## Private Member Functions

- [TimeTAI](#) (const [TimeTAI](#) &)
- [TimeTAI](#) & [operator=](#) (const [TimeTAI](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for TAI time.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTAI](#) ()

## Additional Inherited Members

### 8.25.1 Detailed Description

Represents International Atomic Time.

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

### 8.25.2 Constructor & Destructor Documentation

#### 8.25.2.1 TimeTAI() [1/2]

```
jeod::TimeTAI::TimeTAI (  
    void )
```

Construct a Time\_TAI.

Definition at line 50 of file `time_tai.cc`.

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

#### 8.25.2.2 ~TimeTAI()

```
jeod::TimeTAI::~~TimeTAI (
    void )
```

Destroy a Time\_TAI.

Definition at line 74 of file time\_tai.cc.

#### 8.25.2.3 TimeTAI() [2/2]

```
jeod::TimeTAI::~TimeTAI (
    const TimeTAI & ) [private]
```

### 8.25.3 Member Function Documentation

#### 8.25.3.1 operator=()

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

#### 8.25.3.2 set\_epoch()

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

Sets the epoch for TAI time.

Implements [jeod::TimeStandard](#).

Definition at line 62 of file time\_tai.cc.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

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

### 8.25.4 Friends And Related Function Documentation



## 8.25.4.1 init\_attrjeod\_\_TimeTAI

```
void init_attrjeod__TimeTAI ( ) [friend]
```

## 8.25.4.2 InputProcessor

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

Definition at line 84 of file time\_tai.hh.

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

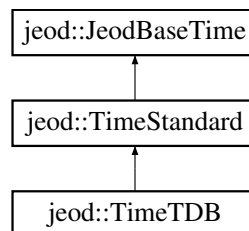
- [time\\_tai.hh](#)
- [time\\_tai.cc](#)

## 8.26 jeod::TimeTDB Class Reference

Represents Barycentric Dynamic Time.

```
#include <time_tdb.hh>
```

Inheritance diagram for jeod::TimeTDB:



## Public Member Functions

- [TimeTDB](#) ()  
*Construct a Time\_TDB.*
- [~TimeTDB](#) ()  
*Destroy a Time\_TDB.*

## Private Member Functions

- [TimeTDB](#) (const [TimeTDB](#) &)
- [TimeTDB](#) & [operator=](#) (const [TimeTDB](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for TDB time.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTDB](#) ()

## Additional Inherited Members

### 8.26.1 Detailed Description

Represents Barycentric Dynamic Time.

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

### 8.26.2 Constructor & Destructor Documentation

#### 8.26.2.1 TimeTDB() [1/2]

```
jeod::TimeTDB::TimeTDB (  
    void )
```

Construct a Time\_TDB.

Definition at line 51 of file `time_tdb.cc`.

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

#### 8.26.2.2 ~TimeTDB()

```
jeod::TimeTDB::~~TimeTDB (  
    void )
```

Destroy a Time\_TDB.

Definition at line 75 of file `time_tdb.cc`.

#### 8.26.2.3 TimeTDB() [2/2]

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

### 8.26.3 Member Function Documentation

#### 8.26.3.1 operator=()

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

#### 8.26.3.2 set\_epoch()

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

Sets the epoch for TDB time.

Implements [jeod::TimeStandard](#).

Definition at line 63 of file `time_tdb.cc`.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

Referenced by `TimeTDB()`.

### 8.26.4 Friends And Related Function Documentation

#### 8.26.4.1 init\_attrjeod\_\_TimeTDB

```
void init_attrjeod__TimeTDB ( ) [friend]
```

#### 8.26.4.2 InputProcessor

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

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

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

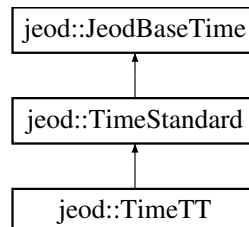
- [time\\_tdb.hh](#)
- [time\\_tdb.cc](#)

## 8.27 jeod::TimeTT Class Reference

Represents Terrestrial Time.

```
#include <time_tt.hh>
```

Inheritance diagram for jeod::TimeTT:



### Public Member Functions

- [TimeTT](#) ()  
*Construct a Time\_TT.*
- [~TimeTT](#) ()  
*Destroy a Time\_TT.*

### Private Member Functions

- [TimeTT](#) (const [TimeTT](#) &)
- [TimeTT](#) & [operator=](#) (const [TimeTT](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for TT time.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTT](#) ()

### Additional Inherited Members

#### 8.27.1 Detailed Description

Represents Terrestrial Time.

Definition at line 82 of file time\_tt.hh.

#### 8.27.2 Constructor & Destructor Documentation

### 8.27.2.1 TimeTT() [1/2]

```
jeod::TimeTT::TimeTT (  
    void )
```

Construct a Time\_TT.

Definition at line 51 of file time\_tt.cc.

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

### 8.27.2.2 ~TimeTT()

```
jeod::TimeTT::~~TimeTT (  
    void )
```

Destroy a Time\_TT.

Definition at line 75 of file time\_tt.cc.

### 8.27.2.3 TimeTT() [2/2]

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

## 8.27.3 Member Function Documentation

### 8.27.3.1 operator=()

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

### 8.27.3.2 set\_epoch()

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

Sets the epoch for TT time.

Implements [jeod::TimeStandard](#).

Definition at line 63 of file time\_tt.cc.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

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

## 8.27.4 Friends And Related Function Documentation

### 8.27.4.1 init\_attrjeod\_\_TimeTT

```
void init_attrjeod__TimeTT ( ) [friend]
```

### 8.27.4.2 InputProcessor

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

Definition at line 84 of file time\_tt.hh.

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

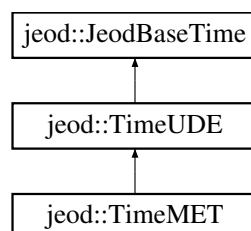
- [time\\_tt.hh](#)
- [time\\_tt.cc](#)

## 8.28 jeod::TimeUDE Class Reference

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

```
#include <time_ude.hh>
```

Inheritance diagram for jeod::TimeUDE:



## Public Member Functions

- [TimeUDE](#) ()  
*Constructor for class [TimeUDE](#).*
- virtual [~TimeUDE](#) ()  
*Destructor for [TimeUDE](#).*
- void [initialize\\_initializer\\_time](#) ([TimeManagerInit](#) \*tm\_init)  
*Each time type is initialized from its parent in the initialization tree, except one.*
- void [add\\_type\\_initialize](#) (const int seeking\_status, [TimeManagerInit](#) \*tm\_init)  
*Adds a UDE type to the initialization tree when it is appropriate to do so.*
- void [initialize\\_from\\_parent](#) ([TimeManagerInit](#) \*tm\_init)  
*Initializes this time-type.*
- void [set\\_time\\_by\\_clock](#) (void)  
*sets the decimal representation of time by the clock*
- void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)  
*Given a seconds value, sets days and clock values.*
- void [set\\_time\\_by\\_days](#) (const double new\_days)  
*Given a seconds value, sets days and clock values.*
- void [set\\_epoch\\_initializing\\_value](#) (const double simtime, const double [epoch\\_initializing\\_value](#))  
*sets the initial epoch value*

## Data Fields

- int [epoch\\_year](#)  
*Gregorian calendar year number at epoch.*
- int [epoch\\_month](#)  
*Gregorian calendar month number at epoch.*
- int [epoch\\_day](#)  
*Gregorian calendar day number at epoch.*
- int [epoch\\_hour](#)  
*24-hour clock hour number at epoch.*
- int [epoch\\_minute](#)  
*Clock minute number at epoch.*
- double [epoch\\_second](#)  
*Clock seconds value at epoch.*
- int [clock\\_day](#)  
*Whole number of days since epoch, in clock format.*
- int [clock\\_hour](#)  
*Whole number of hours since epoch, in clock format.*
- int [clock\\_minute](#)  
*Whole number of minutes since epoch, in clock format.*
- double [clock\\_second](#)  
*Number of seconds since epoch, in clock format.*
- double [last\\_clock\\_update](#)  
*Simtime at the last time the clock was updated.*
- [TimeEnum::TimeFormat](#) [epoch\\_format](#)  
*Format for expressing the epoch of this type (calendar, julian, etc)*
- [TimeEnum::TimeFormat](#) [initial\\_value\\_format](#)  
*Format for expressing the initial value of this type (calendar, julian, etc )*
- std::string [epoch\\_defined\\_in\\_name](#)  
*Name of time type in which epoch defined.*

## Protected Member Functions

- bool [must\\_be\\_singleton](#) ()  
*Returns false in response to the question "does this time class have to be a singleton".*
- void [convert\\_epoch\\_to\\_update](#) ([JeodBaseTime](#) \*epoch\_ptr, [JeodBaseTime](#) \*update\_ptr, [TimeManagerInit](#) \*tm\_init)  
*Converts the time, as specified in the epoch time-type to the update\_from time-type.*
- void [set\\_epoch\\_dyn](#) ([TimeDyn](#) \*epoch\_ptr)  
*Temporarily overwrites the simulation data in time type "epoch" with the epoch value.*
- void [set\\_epoch\\_times](#) ([JeodBaseTime](#) \*epoch\_ptr)  
*To set the times in the epoch time type coincident with the zero-point of this time-type.*
- void [set\\_epoch\\_ude](#) ([TimeUDE](#) \*epoch\_ptr)  
*Overwrites the data in time type "epoch" with that in this class that specifies the epoch.*
- void [set\\_epoch\\_std](#) ([TimeStandard](#) \*epoch\_ptr)  
*Overwrites the data in time type "epoch" with that in this class that specifies the epoch.*
- void [set\\_initial\\_times](#) (void)  
*Sets the initial value of this type from the myriad of initialization options.*
- void [clock\\_update](#) ()  
*converts the decimal seconds value to a clock interface*
- void [verify\\_epoch](#) (void)  
*Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.*
- void [verify\\_init](#) (void)  
*Verifies that any assignment to initialize\_from is flagged as inappropriate, and tests for the presence of initializing data.*
- void [verify\\_update](#) (void)  
*Ensures that the time-type identified as "update\_from" is legitimate.*

## Protected Attributes

- double [epoch\\_initializing\\_value](#)  
*Value of epoch in appropriate format.*
- bool [initializing\\_data\\_present](#)  
*Whether initializing data is present.*
- bool [epoch\\_data\\_present](#)  
*Whether epoch data is present.*
- bool [epoch\\_value\\_is\\_set\\_number](#)  
*Whether there is some numerical input that could set epoch.*
- bool [epoch\\_value\\_is\\_set\\_calendar](#)  
*Whether there is some calendar input that could set epoch.*
- bool [epoch\\_value\\_is\\_set\\_clock](#)  
*Whether there is some clock input that could set epoch.*
- int [update\\_index](#)  
*The index of the time-type from which this one is updated.*
- int [epoch\\_index](#)  
*The index of the time-type in which this one's epoch is defined.*

## Private Member Functions

- [TimeUDE](#) (const [TimeUDE](#) &)
- [TimeUDE](#) & [operator=](#) (const [TimeUDE](#) &)



## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUDE](#) ()

### 8.28.1 Detailed Description

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

Definition at line 89 of file `time_ude.hh`.

### 8.28.2 Constructor & Destructor Documentation

#### 8.28.2.1 TimeUDE() [1/2]

```
jeod::TimeUDE::TimeUDE (  
    void )
```

Constructor for class [TimeUDE](#).

#### Assumptions and Limitations

- None

Definition at line 64 of file `time_ude.cc`.

#### 8.28.2.2 ~TimeUDE()

```
jeod::TimeUDE::~~TimeUDE (  
    void ) [virtual]
```

Destructor for [TimeUDE](#).

Definition at line 1472 of file `time_ude.cc`.

#### 8.28.2.3 TimeUDE() [2/2]

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

### 8.28.3 Member Function Documentation

#### 8.28.3.1 add\_type\_initialize()

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

Adds a UDE type to the initialization tree when it is appropriate to do so.

#### Assumptions and Limitations

- The time type from which the UDE updates must be in the tree above the UDE.
- If the time type in which the epoch is defined is another UDE, it also must be in the tree above this UDE
- This function is only called when the UDE is NOT being used to initialize the simulation.

#### Parameters

in	<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 125 of file time\_ude.cc.

References epoch\_defined\_in\_name, epoch\_index, jeod::TimeManagerInit::get\_conv\_dir\_init(), jeod::TimeManagerInit::get\_status(), jeod::TimeManager::get\_time\_ptr(), jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeManagerInit::increment\_status(), jeod::JeodBaseTime::index, jeod::TimeMessages::invalid\_setup\_error, jeod::JeodBaseTime::name, jeod::TimeManager::num\_types, jeod::TimeManagerInit::set\_status(), jeod::JeodBaseTime::time\_manager, jeod::JeodBaseTime::update\_from\_name, update\_index, verify\_epoch(), and verify\_update().

#### 8.28.3.2 clock\_update()

```
void jeod::TimeUDE::clock_update (
    void ) [protected]
```

converts the decimal seconds value to a clock interface

#### Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1280 of file time\_ude.cc.

References clock\_day, clock\_hour, clock\_minute, jeod::JeodBaseTime::clock\_resolution, clock\_second, and jeod::JeodBaseTime::seconds.

Referenced by set\_time\_by\_days(), and set\_time\_by\_seconds().

## 8.28.3.3 convert\_epoch\_to\_update()

```
void jeod::TimeUDE::convert_epoch_to_update (
    JeodBaseTime * epoch_ptr,
    JeodBaseTime * update_from_ptr,
    TimeManagerInit * time_manager_init ) [protected]
```

Converts the time, as specified in the epoch time-type to the update\_from time-type.

This sets the update\_from time at the epoch of "this", and allows for the initialization of the converter.

## Assumptions and Limitations

- That there is a converter available to do this in one step
- Future work may include an extension to this routine to cover other cases.

## Parameters

in	<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 252 of file time\_ude.cc.

References `jeod::TimeConverter::convert_a_to_b()`, `jeod::TimeConverter::convert_b_to_a()`, `epoch_defined_in_name`, `epoch_index`, `jeod::TimeManagerInit::get_conv_dir_init()`, `jeod::TimeManagerInit::get_conv_ptr_index()`, `jeod::TimeManager::get_converter_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeConverter::initialize()`, `jeod::JeodBaseTime::name`, `jeod::TimeManager::num_types`, `jeod::TimeConverter::override_initialized()`, `jeod::JeodBaseTime::override_initialized()`, `jeod::JeodBaseTime::time_manager`, `jeod::JeodBaseTime::update_from_name`, and `update_index`.

Referenced by `initialize_from_parent()`, and `initialize_initializer_time()`.

## 8.28.3.4 initialize\_from\_parent()

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

Initializes this time-type.

## Assumptions and Limitations

- The subject object has a parent, a time-type with which it ticks. This has already been tested for.

## Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

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

Definition at line 316 of file `time_ude.cc`.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [convert\\_epoch\\_to\\_↵update\(\)](#), [jeod::JeodBaseTime::days](#), [epoch\\_data\\_present](#), [epoch\\_index](#), [jeod::TimeManagerInit::get\\_conv\\_dir\\_↵init\(\)](#), [jeod::TimeManagerInit::get\\_conv\\_ptr\\_index\(\)](#), [jeod::TimeManager::get\\_converter\\_ptr\(\)](#), [jeod::TimeManager↵::get\\_time\\_ptr\(\)](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [jeod::JeodBaseTime::index](#), [jeod::JeodBase↵Time::initial\\_value](#), [jeod::TimeMessages::initialization\\_error](#), [jeod::TimeConverter::initialize\(\)](#), [jeod::JeodBase↵Time::initialize\\_from\\_parent\(\)](#), [jeod::JeodBaseTime::initialized](#), [initializing\\_data\\_present](#), [jeod::TimeConverter↵::is\\_initialized\(\)](#), [jeod::JeodBaseTime::is\\_initialized\(\)](#), [jeod::JeodBaseTime::name](#), [jeod::TimeManager::num\\_↵types](#), [jeod::JeodBaseTime::override\\_initialized\(\)](#), [jeod::TimeMessages::redundancy\\_error](#), [jeod::JeodBaseTime↵::seconds](#), [set\\_epoch\\_times\(\)](#), [jeod::JeodBaseTime::set\\_time\\_by\\_seconds\(\)](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::JeodBaseTime::update\\_from\\_name](#), [update\\_index](#), and [verify\\_init\(\)](#).

### 8.28.3.5 initialize\_initializer\_time()

```
void jeod::TimeUDE::initialize_initializer_time (
    TimeManagerInit * time_manager_init ) [virtual]
```

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

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

#### Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 511 of file `time_ude.cc`.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [convert\\_epoch\\_to\\_↵\\_update\(\)](#), [jeod::JeodBaseTime::days](#), [jeod::TimeMessages::duplicate\\_methods](#), [epoch\\_data\\_present](#), [epoch↵\\_index](#), [jeod::TimeManagerInit::get\\_conv\\_dir\\_init\(\)](#), [jeod::TimeManagerInit::get\\_conv\\_ptr\\_index\(\)](#), [jeod::Time↵Manager::get\\_converter\\_ptr\(\)](#), [jeod::TimeManager::get\\_time\\_ptr\(\)](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [jeod::JeodBaseTime::index](#), [initial\\_value\\_format](#), [jeod::TimeConverter::initialize\(\)](#), [jeod::JeodBaseTime::initialized](#), [initializing\\_data\\_present](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::name](#), [jeod::Time↵Manager::num\\_types](#), [jeod::JeodBaseTime::override\\_initialized\(\)](#), [jeod::JeodBaseTime::seconds](#), [set\\_epoch↵\\_times\(\)](#), [jeod::TimeManagerInit::sim\\_start\\_format](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::TimeManager↵::time\\_standards\\_exist\(\)](#), [jeod::TimeEnum::undefined](#), [jeod::JeodBaseTime::update\\_from\\_name](#), [update\\_index](#), [verify\\_epoch\(\)](#), [verify\\_init\(\)](#), and [verify\\_update\(\)](#).

### 8.28.3.6 must\_be\_singleton()

```
bool jeod::TimeUDE::must_be_singleton (
    void ) [protected], [virtual]
```

Returns false in response to the question "does this time class have to be a singleton".

**Assumptions and Limitations**

- There can be more than one UDE

**Returns**

false

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

Definition at line 104 of file time\_ude.cc.

**8.28.3.7 operator=()**

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

**8.28.3.8 set\_epoch\_dyn()**

```
void jeod::TimeUDE::set_epoch_dyn (
    TimeDyn * epoch_ptr ) [protected]
```

Temporarily overwrites the simulation data in time type "epoch" with the epoch value.

**Assumptions and Limitations**

- "Epoch" is DynTime

**Parameters**

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

Definition at line 754 of file time\_ude.cc.

References [jeod::TimeEnum::calendar](#), [jeod::TimeEnum::clock](#), [jeod::TimeEnum::days\\_since\\_epoch](#), [epoch\\_data\\_present](#), [epoch\\_day](#), [epoch\\_defined\\_in\\_name](#), [epoch\\_format](#), [epoch\\_hour](#), [epoch\\_initializing\\_value](#), [epoch\\_minute](#), [epoch\\_month](#), [epoch\\_value\\_is\\_set\\_number](#), [epoch\\_year](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [initializing\\_data\\_present](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::TimeEnum::Julian](#), [jeod::TimeEnum::julian](#), [jeod::TimeEnum::modified\\_julian](#), [jeod::JeodBaseTime::name](#), [jeod::TimeMessages::redundancy\\_error](#), [jeod::TimeEnum::seconds\\_since\\_epoch](#), [jeod::JeodBaseTime::set\\_time\\_by\\_days\(\)](#), [jeod::JeodBaseTime::set\\_time\\_by\\_seconds\(\)](#), [jeod::TimeEnum::truncated\\_julian](#), and [jeod::TimeEnum::undefined](#).

Referenced by [set\\_epoch\\_times\(\)](#).

### 8.28.3.9 set\_epoch\_initializing\_value()

```
void jeod::TimeUDE::set_epoch_initializing_value (
    const double simtime,
    const double epoch )
```

sets the initial epoch value

#### Assumptions and Limitations

- Assumes that the number that is passed in is correctly entered with the correct units interpretation.

#### Parameters

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

Definition at line 1256 of file time\_ude.cc.

References epoch\_initializing\_value, jeod::TimeMessages::invalid\_setup\_error, and jeod::JeodBaseTime::name.

### 8.28.3.10 set\_epoch\_std()

```
void jeod::TimeUDE::set_epoch_std (
    TimeStandard * epoch_ptr ) [protected]
```

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

#### Assumptions and Limitations

- "Epoch" is Absolute Derived Time

#### Parameters

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

Definition at line 859 of file time\_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeStandard::calendar\_day, jeod::TimeStandard::calendar\_hour, jeod::TimeStandard::calendar\_minute, jeod::TimeStandard::calendar\_month, jeod::TimeStandard::calendar\_↵\_second, jeod::TimeStandard::calendar\_year, jeod::TimeEnum::clock, jeod::TimeStandard::convert\_from\_↵\_calendar(), jeod::TimeEnum::days\_since\_epoch, epoch\_day, epoch\_defined\_in\_name, epoch\_format, epoch\_hour, epoch\_initializing\_value, epoch\_minute, epoch\_month, epoch\_second, epoch\_value\_is\_set\_calendar, epoch\_↵\_value\_is\_set\_number, epoch\_year, jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeMessages::invalid\_↵\_setup\_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified\_julian, jeod::Jeod\_↵\_BaseTime::name, jeod::TimeEnum::seconds\_since\_epoch, jeod::TimeStandard::set\_time\_by\_days(), jeod::Time\_↵\_Standard::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 717 of file `time_ude.cc`.

References `jeod::TimeMessages::invalid_setup_error`, `set_epoch_dyn()`, `set_epoch_std()`, and `set_epoch_ude()`.

Referenced by `initialize_from_parent()`, and `initialize_initializer_time()`.

### 8.28.3.12 `set_epoch_ude()`

```
void jeod::TimeUDE::set_epoch_ude (
    TimeUDE * epoch_ptr ) [protected]
```

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

#### Assumptions and Limitations

- "Epoch" is a User-Defined-Epoch Time.

#### Parameters

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

Definition at line 993 of file `time_ude.cc`.

References `jeod::TimeEnum::calendar`, `jeod::TimeEnum::clock`, `clock_day`, `clock_hour`, `clock_minute`, `clock_↵`  
`second`, `jeod::TimeEnum::days_since_epoch`, `epoch_day`, `epoch_defined_in_name`, `epoch_format`, `epoch_↵`  
`hour`, `epoch_initializing_value`, `epoch_minute`, `epoch_second`, `epoch_value_is_set_clock`, `epoch_value_is_set_↵`  
`_number`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeMessages::invalid_setup_error`, `jeod::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 (
    void ) [protected]
```

Sets the initial value of this type from the myriad of initialization options.

##### Assumptions and Limitations

- At least one of the following is non-zero: initializing value, clock\_day, clock\_hour, clock\_minute, clock\_↵second, seconds, days

Definition at line 1095 of file time\_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, clock\_day, clock\_hour, clock\_minute, clock\_↵second, jeod::JeodBaseTime::days, jeod::TimeEnum::days\_since\_epoch, jeod::TimeMessages::incomplete\_↵setup\_error, initial\_value\_format, initializing\_data\_present, jeod::JeodBaseTime::initializing\_value, jeod::Time\_↵Messages::invalid\_setup\_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeMessages::memory\_↵\_error, jeod::TimeEnum::modified\_julian, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy\_error, jeod::JeodBaseTime::seconds, jeod::TimeEnum::seconds\_since\_epoch, jeod::TimeEnum::truncated\_julian, and jeod::TimeEnum::undefined.

Referenced by verify\_init().

#### 8.28.3.14 set\_time\_by\_clock()

```
void jeod::TimeUDE::set_time_by_clock (
    void )
```

sets the decimal representation of time by the clock

##### Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1236 of file time\_ude.cc.

References clock\_day, clock\_hour, clock\_minute, clock\_second, jeod::JeodBaseTime::days, and jeod::JeodBase\_↵Time::seconds.

Referenced by set\_epoch\_ude().

#### 8.28.3.15 set\_time\_by\_days()

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

Given a seconds value, sets days and clock values.



## Parameters

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

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

Definition at line 1203 of file `time_ude.cc`.

References `clock_update()`, and `jeod::JeodBaseTime::set_time_by_days()`.

8.28.3.16 `set_time_by_seconds()`

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

Given a seconds value, sets days and clock values.

## Parameters

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

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

Definition at line 1217 of file `time_ude.cc`.

References `clock_update()`, and `jeod::JeodBaseTime::set_time_by_seconds()`.

Referenced by `jeod::TimeConverter_Dyn_UDE::convert_a_to_b()`, `jeod::TimeConverter_STD_UDE::convert_a_to_b()`, and `set_epoch_ude()`.

8.28.3.17 `verify_epoch()`

```
void jeod::TimeUDE::verify_epoch (
    void ) [protected]
```

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

Definition at line 1312 of file `time_ude.cc`.

References `epoch_data_present`, `epoch_day`, `epoch_defined_in_name`, `epoch_format`, `epoch_hour`, `epoch_index`, `epoch_initializing_value`, `epoch_minute`, `epoch_month`, `epoch_second`, `epoch_value_is_set_calendar`, `epoch_value_is_set_clock`, `epoch_value_is_set_number`, `epoch_year`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::JeodBaseTime::index`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::TimeMessages::redundancy_error`, `jeod::TimeManager::time_lookup()`, `jeod::JeodBaseTime::time_manager`, and `jeod::TimeEnum::undefined`.

Referenced by `add_type_initialize()`, and `initialize_initializer_time()`.

#### 8.28.3.18 verify\_init()

```
void jeod::TimeUDE::verify_init (
    void ) [protected]
```

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

Definition at line 1406 of file `time_ude.cc`.

References `jeod::JeodBaseTime::initialize_from_name`, `jeod::JeodBaseTime::name`, `jeod::TimeMessages::redundancy_error`, `set_initial_times()`, `jeod::TimeManager::time_lookup()`, and `jeod::JeodBaseTime::time_manager`.

Referenced by `initialize_from_parent()`, and `initialize_initializer_time()`.

#### 8.28.3.19 verify\_update()

```
void jeod::TimeUDE::verify_update (
    void ) [protected]
```

Ensures that the time-type identified as "update\_from" is legitimate.

Definition at line 1440 of file `time_ude.cc`.

References `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::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 123 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.2 clock\_hour

```
int jeod::TimeUDE::clock_hour
```

Whole number of hours since epoch, in clock format.

trick\_units(-)

Definition at line 128 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.3 clock\_minute

```
int jeod::TimeUDE::clock_minute
```

Whole number of minutes since epoch, in clock format.

trick\_units(-)

Definition at line 133 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.4 clock\_second

```
double jeod::TimeUDE::clock_second
```

Number of seconds since epoch, in clock format.

trick\_units(s)

Definition at line 138 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.5 epoch\_data\_present

```
bool jeod::TimeUDE::epoch_data_present [protected]
```

Whether epoch data is present.

trick\_units(-)

Definition at line 174 of file time\_ude.hh.

Referenced by initialize\_from\_parent(), initialize\_initializer\_time(), set\_epoch\_dyn(), and verify\_epoch().

#### 8.28.5.6 epoch\_day

```
int jeod::TimeUDE::epoch_day
```

Gregorian calendar day number at epoch.

trick\_units(day)

Definition at line 106 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.7 epoch\_defined\_in\_name

```
std::string jeod::TimeUDE::epoch_defined_in_name
```

Name of time type in which epoch defined.

trick\_units(-)

Definition at line 158 of file time\_ude.hh.

Referenced by add\_type\_initialize(), convert\_epoch\_to\_update(), set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.8 epoch\_format

`TimeEnum::TimeFormat jeod::TimeUDE::epoch_format`

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

`trick_units(-)`

Definition at line 148 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.9 epoch\_hour

`int jeod::TimeUDE::epoch_hour`

24-hour clock hour number at epoch.

`trick_units(hr)`

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

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.10 epoch\_index

`int jeod::TimeUDE::epoch_index [protected]`

The index of the time-type in which this one's epoch is defined.

`trick_units(-)`

Definition at line 199 of file `time_ude.hh`.

Referenced by `add_type_initialize()`, `convert_epoch_to_update()`, `initialize_from_parent()`, `initialize_initializer_↔time()`, and `verify_epoch()`.

#### 8.28.5.11 epoch\_initializing\_value

`double jeod::TimeUDE::epoch_initializing_value [protected]`

Value of epoch in appropriate format.

`trick_units(-)`

Definition at line 164 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_initializing_value()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_↔epoch()`.

#### 8.28.5.12 epoch\_minute

```
int jeod::TimeUDE::epoch_minute
```

Clock minute number at epoch.

trick\_units(min)

Definition at line 114 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.13 epoch\_month

```
int jeod::TimeUDE::epoch_month
```

Gregorian calendar month number at epoch.

trick\_units(-)

Definition at line 102 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), and verify\_epoch().

#### 8.28.5.14 epoch\_second

```
double jeod::TimeUDE::epoch_second
```

Clock seconds value at epoch.

trick\_units(s)

Definition at line 118 of file time\_ude.hh.

Referenced by set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.15 epoch\_value\_is\_set\_calendar

```
bool jeod::TimeUDE::epoch_value_is_set_calendar [protected]
```

Whether there is some calendar input that could set epoch.

trick\_units(-)

Definition at line 184 of file time\_ude.hh.

Referenced by set\_epoch\_std(), and verify\_epoch().

#### 8.28.5.16 epoch\_value\_is\_set\_clock

```
bool jeod::TimeUDE::epoch_value_is_set_clock [protected]
```

Whether there is some clock input that could set epoch.

trick\_units(—)

Definition at line 189 of file time\_ude.hh.

Referenced by set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.17 epoch\_value\_is\_set\_number

```
bool jeod::TimeUDE::epoch_value_is_set_number [protected]
```

Whether there is some numerical input that could set epoch.

trick\_units(—)

Definition at line 179 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.18 epoch\_year

```
int jeod::TimeUDE::epoch_year
```

Gregorian calendar year number at epoch.

trick\_units(—)

Definition at line 98 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), and verify\_epoch().

#### 8.28.5.19 initial\_value\_format

```
TimeEnum::TimeFormat jeod::TimeUDE::initial_value_format
```

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

trick\_units(—)

Definition at line 153 of file time\_ude.hh.

Referenced by initialize\_initializer\_time(), and set\_initial\_times().

#### 8.28.5.20 initializing\_data\_present

```
bool jeod::TimeUDE::initializing_data_present [protected]
```

Whether initializing data is present.

trick\_units(-)

Definition at line 169 of file time\_ude.hh.

Referenced by initialize\_from\_parent(), initialize\_initializer\_time(), set\_epoch\_dyn(), and set\_initial\_times().

#### 8.28.5.21 last\_clock\_update

```
double jeod::TimeUDE::last_clock_update
```

Simtime at the last time the clock was updated.

trick\_units(s)

Definition at line 143 of file time\_ude.hh.

#### 8.28.5.22 update\_index

```
int jeod::TimeUDE::update_index [protected]
```

The index of the time-type from which this one is updated.

trick\_units(-)

Definition at line 194 of file time\_ude.hh.

Referenced by add\_type\_initialize(), convert\_epoch\_to\_update(), initialize\_from\_parent(), initialize\_initializer\_time(), and verify\_update().

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

- [time\\_ude.hh](#)
- [time\\_ude.cc](#)

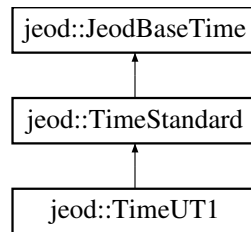


## 8.29 jeod::TimeUT1 Class Reference

Represents Universal Time.

```
#include <time_ut1.hh>
```

Inheritance diagram for jeod::TimeUT1:



### Public Member Functions

- [TimeUT1](#) ()  
*Construct a Time\_UT1.*
- [~TimeUT1](#) ()  
*Destroy a Time\_UT1.*
- double [get\\_days](#) ()  
*Accesses days.*

### Data Fields

- bool [true\\_ut1](#)  
*"False" for comparison with older versions of JEOD*

### Private Member Functions

- [TimeUT1](#) (const [TimeUT1](#) &)
- [TimeUT1](#) & [operator=](#) (const [TimeUT1](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for UT1 time.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUT1](#) ()

### Additional Inherited Members

#### 8.29.1 Detailed Description

Represents Universal Time.

Definition at line 82 of file time\_ut1.hh.

## 8.29.2 Constructor & Destructor Documentation

### 8.29.2.1 TimeUT1() [1/2]

```
jeod::TimeUT1::TimeUT1 (  
    void )
```

Construct a Time\_UT1.

Definition at line 51 of file time\_ut1.cc.

References jeod::JeodBaseTime::name, set\_epoch(), and true\_ut1.

### 8.29.2.2 ~TimeUT1()

```
jeod::TimeUT1::~~TimeUT1 (  
    void )
```

Destroy a Time\_UT1.

Definition at line 85 of file time\_ut1.cc.

### 8.29.2.3 TimeUT1() [2/2]

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

## 8.29.3 Member Function Documentation

### 8.29.3.1 get\_days()

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

Accesses days.

#### Returns

days value  
Units: d

Definition at line 75 of file time\_ut1.cc.

References jeod::JeodBaseTime::days.

Referenced by jeod::TimeConverter\_UT1\_GMST::convert\_a\_to\_b().

### 8.29.3.2 operator=()

```
TimeUT1& jeod::TimeUT1::operator= (
    const TimeUT1 & ) [private]
```

### 8.29.3.3 set\_epoch()

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

Sets the epoch for UT1 time.

Implements [jeod::TimeStandard](#).

Definition at line 63 of file time\_ut1.cc.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

Referenced by [TimeUT1\(\)](#).

## 8.29.4 Friends And Related Function Documentation

### 8.29.4.1 init\_attrjeod\_\_TimeUT1

```
void init_attrjeod__TimeUT1 ( ) [friend]
```

### 8.29.4.2 InputProcessor

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

Definition at line 84 of file time\_ut1.hh.

## 8.29.5 Field Documentation

### 8.29.5.1 true\_ut1

```
bool jeod::TimeUT1::true_ut1
```

"False" for comparison with older versions of JEOD

trick\_units(-)

Definition at line 91 of file time\_ut1.hh.

Referenced by `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `TimeUT1()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

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

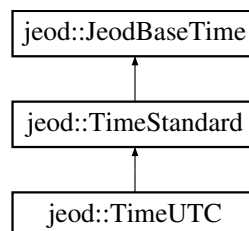
- [time\\_ut1.hh](#)
- [time\\_ut1.cc](#)

## 8.30 jeod::TimeUTC Class Reference

Represents Coordinated Universal Time.

```
#include <time_utc.hh>
```

Inheritance diagram for `jeod::TimeUTC`:



### Public Member Functions

- [TimeUTC\(\)](#)  
*Construct a Time\_UTC.*
- [~TimeUTC\(\)](#)  
*Destroy a Time\_UTC.*

### Data Fields

- bool [true\\_utc](#)  
*"False" for comparison with older versions of JEOD*

## Private Member Functions

- [TimeUTC](#) (const [TimeUTC](#) &)
- [TimeUTC](#) & [operator=](#) (const [TimeUTC](#) &)
- void [set\\_epoch](#) (void)

*Sets the epoch for UTC time.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUTC](#) ()

## Additional Inherited Members

### 8.30.1 Detailed Description

Represents Coordinated Universal Time.

Definition at line 83 of file `time_utc.hh`.

### 8.30.2 Constructor & Destructor Documentation

#### 8.30.2.1 [TimeUTC\(\)](#) [1/2]

```
jeod::TimeUTC::TimeUTC (  
    void )
```

Construct a `Time_UTC`.

Definition at line 51 of file `time_utc.cc`.

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

#### 8.30.2.2 [~TimeUTC\(\)](#)

```
jeod::TimeUTC::~~TimeUTC (  
    void )
```

Destroy a `Time_UTC`.

Definition at line 75 of file `time_utc.cc`.

### 8.30.2.3 TimeUTC() [2/2]

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

## 8.30.3 Member Function Documentation

### 8.30.3.1 operator=()

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

### 8.30.3.2 set\_epoch()

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

Sets the epoch for UTC time.

Implements [jeod::TimeStandard](#).

Definition at line 64 of file time\_utc.cc.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

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

## 8.30.4 Friends And Related Function Documentation

### 8.30.4.1 init\_attrjeod\_\_TimeUTC

```
void init_attrjeod__TimeUTC ( ) [friend]
```

### 8.30.4.2 InputProcessor

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

Definition at line 85 of file time\_utc.hh.

### 8.30.5 Field Documentation

#### 8.30.5.1 true\_utc

```
bool jeod::TimeUTC::true_utc
```

"False" for comparison with older versions of JEOD

trick\_units(−)

Definition at line 92 of file time\_utc.hh.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `TimeUTC()`, and `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`.

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

- [time\\_utc.hh](#)
- [time\\_utc.cc](#)





## Chapter 9

# File Documentation

### 9.1 class\_declarations.hh File Reference

Forward declaration of classes defined in [time.hh](#).

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.1.1 Detailed Description

Forward declaration of classes defined in [time.hh](#).

### 9.2 tai\_to\_ut1.cc File Reference

```
#include "environment/time/include/time_converter_tai_ut1.hh"  
#include "utils/memory/include/jeod_alloc.hh"  
#include "../include/tai_to_ut1.hh"
```

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### Macros

- `#define JEOD\_FRIEND\_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data`

## 9.2.1 Macro Definition Documentation

### 9.2.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data
```

Definition at line 25 of file tai\_to\_ut1.cc.

## 9.3 tai\_to\_ut1.hh File Reference

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UT1\\_tai\\_to\\_ut1\\_default\\_data](#)

### Namespaces

- [jeod](#)  
*Namespace jeod.*

## 9.4 tai\_to\_utc.cc File Reference

```
#include "environment/time/include/time_converter_tai_utc.hh"  
#include "utils/memory/include/jeod_alloc.hh"  
#include "../include/tai_to_utc.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data

### 9.4.1 Macro Definition Documentation

#### 9.4.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS TimeConverter_TAI_UTC_tai_to_utc_default_data
```

Definition at line 23 of file tai\_to\_utc.cc.

## 9.5 tai\_to\_utc.hh File Reference

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UTC\\_tai\\_to\\_utc\\_default\\_data](#)

### Namespaces

- [jeod](#)  
*Namespace jeod.*

## 9.6 time.cc File Reference

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

```
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.6.1 Detailed Description

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

## 9.7 time.hh File Reference

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

```
#include <string>
#include <utility>
#include "utils/named_item/include/named_item.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
#include "time_links.hh"
```

## Data Structures

- class [jeod::JeodBaseTime](#)

*[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.*

## Namespaces

- [jeod](#)

*Namespace [jeod](#).*

### 9.7.1 Detailed Description

[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.

## 9.8 [time\\_\\_add\\_type\\_update.cc](#) File Reference

Define [JeodBaseTime::add\\_type\\_update](#).

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace [jeod](#).*

### 9.8.1 Detailed Description

Define [JeodBaseTime::add\\_type\\_update](#).

This is a final method that draws in a lot of the time model functionality. Making this method a separate compilation unit enables models that only need the vtable for class [Time](#) can get that from [time.o](#) without pulling in the whole of the time model.

## 9.9 time\_converter.cc File Reference

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

```
#include <stddef>
#include <stdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter.hh"
#include "../include/time.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.9.1 Detailed Description

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

## 9.10 time\_converter.hh File Reference

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

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::TimeConverter](#)  
*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Functions

- TimeConverter::Direction [jeod::operator|](#) (TimeConverter::Direction a, TimeConverter::Direction b)  
*Bitwise or operator for combining multiple converter direction flags.*

### 9.10.1 Detailed Description

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

## 9.11 `time_converter_dyn_tai.cc` File Reference

Converts between International Atomic Time and Dynamic Time.

```
#include <cstdint>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tai.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tai.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.11.1 Detailed Description

Converts between International Atomic Time and Dynamic Time.

## 9.12 `time_converter_dyn_tai.hh` File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_TAI](#)  
*Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.12.1 Detailed Description

Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.

## 9.13 time\_converter\_dyn\_tdb.cc File Reference

Converts between Dynamic Time and Barycentric Dynamic Time.

```
#include <cstdint>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tdb.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.13.1 Detailed Description

Converts between Dynamic Time and Barycentric Dynamic Time.

## 9.14 time\_converter\_dyn\_tdb.hh File Reference

Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_TDB](#)  
*Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.14.1 Detailed Description

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

## 9.15 `time_converter_dyn_ude.cc` File Reference

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

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.15.1 Detailed Description

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

## 9.16 `time_converter_dyn_ude.hh` File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_UDE](#)  
*Define class [TimeConverter\\_Dyn\\_UDE](#), which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*



### 9.16.1 Detailed Description

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

There can be multiple instances of this class.

## 9.17 time\_converter\_std\_ude.cc File Reference

Define member functions for class TimeConverter\_STD\_UDE.

```
#include <cmath>
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_std_ude.hh"
#include "../include/time_standard.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.17.1 Detailed Description

Define member functions for class TimeConverter\_STD\_UDE.

## 9.18 time\_converter\_std\_ude.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_STD\\_UDE](#)

*Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.18.1 Detailed Description

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

There can be multiple such instances of this class.

## 9.19 time\_converter\_tai\_gps.cc File Reference

Converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_gps.hh"
#include "../include/time_tai.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.19.1 Detailed Description

Converts between International Atomic Time and the clock associated with the Global Positioning System.

## 9.20 time\_converter\_tai\_gps.hh File Reference

Define class TimeConverter\_TAI\_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_TAI\\_GPS](#)

Define class [TimeConverter\\_TAI\\_GPS](#), which converts between International Atomic Time and the clock associated with the Global Positioning System.

## Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.20.1 Detailed Description

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

## 9.21 time\_converter\_tai\_tdb.cc File Reference

Converts from International Atomic Time to Barycentric Dynamic Time.

```
#include <cmath>
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tdb.hh"
#include "../include/time_tai.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.21.1 Detailed Description

Converts from International Atomic Time to Barycentric Dynamic Time.

## 9.22 time\_converter\_tai\_tdb.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_TAI\\_TDB](#)

Define class [TimeConverter\\_TAI\\_TDB](#), which converts from International Atomic Time to Barycentric Dynamic Time.

## Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.22.1 Detailed Description

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

## 9.23 `time_converter_tai_tt.cc` File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tt.hh"
#include "../include/time_tai.hh"
#include "../include/time_tt.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.23.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

## 9.24 `time_converter_tai_tt.hh` File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_TAI\\_TT](#)  
*Converts between International Atomic Time and Terrestrial Time.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.24.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

## 9.25 time\_converter\_tai\_ut1.cc File Reference

Converts between International Atomic Time and Universal Time.

```
#include <cmath>
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_ut1.hh"
#include "../include/time_tai.hh"
#include "../include/time_ut1.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.25.1 Detailed Description

Converts between International Atomic Time and Universal Time.

## 9.26 time\_converter\_tai\_ut1.hh File Reference

Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_TAI\\_UT1](#)

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

## Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.26.1 Detailed Description

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

## 9.27 [time\\_converter\\_tai\\_utc.cc](#) File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_tai.hh"
#include "../include/time_utc.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.27.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

## 9.28 [time\\_converter\\_tai\\_utc.hh](#) File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_TAI\\_UTC](#)

*Converts between International Atomic Time and Coordinated Universal Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.28.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

## 9.29 time\_converter\_ut1\_gmst.cc File Reference

Define member functions for class TimeConverter\_UT1\_GMST.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_ut1_gmst.hh"
#include "../include/time_ut1.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.29.1 Detailed Description

Define member functions for class TimeConverter\_UT1\_GMST.

## 9.30 time\_converter\_ut1\_gmst.hh File Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

## Data Structures

- class [jeod::TimeConverter\\_UT1\\_GMST](#)

*Converts between Universal Time and Greenwich Mean Sidereal Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.30.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

## 9.31 time\_dyn.cc File Reference

Define member functions for Dynamic Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time_dyn.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.31.1 Detailed Description

Define member functions for Dynamic Time.

## 9.32 time\_dyn.hh File Reference

Represents the Dynamic Time in the simulation.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```



## Data Structures

- class [jeod::TimeDyn](#)  
*Represents the Dynamic Time in the simulation.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.32.1 Detailed Description

Represents the Dynamic Time in the simulation.

## 9.33 time\_enum.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::TimeEnum](#)  
*Contains an enumeration of the formats in which time can be represented.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.33.1 Detailed Description

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

## 9.34 time\_gmst.cc File Reference

Define member functions for Greenwich Mean Sidereal Time.

```
#include <cstdlib>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.34.1 Detailed Description

Define member functions for Greenwich Mean Sidereal Time.

## 9.35 time\_gmst.hh File Reference

To represent the clock known as Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

## Data Structures

- class [jeod::TimeGMST](#)

*To represent the clock known as Greenwich Mean Sidereal Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.35.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

## 9.36 time\_gps.cc File Reference

Define member functions for the clock associated with the Global Positioning System.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.36.1 Detailed Description

Define member functions for the clock associated with the Global Positioning System.

## 9.37 time\_gps.hh File Reference

To represent the time associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

## Data Structures

- class [jeod::TimeGPS](#)

*To represent the time associated with the Global Positioning System.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.37.1 Detailed Description

To represent the time associated with the Global Positioning System.

## 9.38 time\_links.hh File Reference

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

```
#include "class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/tree_links.hh"
```

## Data Structures

- class [jeod::TimeLinks](#)

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.38.1 Detailed Description

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

## 9.39 time\_manager.cc File Reference

Define member functions for class TimeManager.

```
#include <algorithm>
#include <cstdlib>
#include <cstring>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.39.1 Detailed Description

Define member functions for class TimeManager.

## 9.40 time\_manager.hh File Reference

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

```
#include <string>
#include <vector>
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/integration/include/jeod_integration_time.hh"
#include "time_dyn.hh"
```

## Data Structures

- class [jeod::TimeManager](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.40.1 Detailed Description

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

## 9.41 time\_manager\_\_initialize.cc File Reference

Define TimeManager::initialize.

```
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.41.1 Detailed Description

Define TimeManager::initialize.

This method allocates resources and invokes TimeManagerInit functionality. This method needs to be defined as a separate compilation unit.

## 9.42 time\_manager\_init.cc File Reference

Define member functions for the Time Manager Initialization.

```
#include <cstdlib>
#include <typeinfo>
#include <algorithm>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_ude.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_converter_tai_utl.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.42.1 Detailed Description

Define member functions for the Time Manager Initialization.

## 9.43 time\_manager\_init.hh File Reference

To initialize the Time Manager.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_enum.hh"
```

## Data Structures

- class [jeod::TimeManagerInit](#)

*To initialize the Time Manager.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.43.1 Detailed Description

To initialize the Time Manager.

## 9.44 time\_messages.cc File Reference

Implement the class TimeMessages.

```
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

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

### 9.44.1 Detailed Description

Implement the class TimeMessages.

## 9.45 time\_messages.hh File Reference

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class `jeod::TimeMessages`  
*Specify the message IDs used in the Time model.*

## Namespaces

- `jeod`  
*Namespace jeod.*

### 9.45.1 Detailed Description

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

## 9.46 time\_met.cc File Reference

Define member functions for Mission Elapsed Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_met.hh"
```

## Namespaces

- `jeod`  
*Namespace jeod.*

### 9.46.1 Detailed Description

Define member functions for Mission Elapsed Time.

## 9.47 time\_met.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_ude.hh"
```

### Data Structures

- class [jeod::TimeMET](#)  
*A type of UDE time that allows for deliberate holds, or pauses.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.47.1 Detailed Description

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

## 9.48 time\_standard.cc File Reference

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

```
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/math/include/numerical.hh"
#include "../include/time_standard.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*



### 9.48.1 Detailed Description

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

## 9.49 time\_standard.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```

### Data Structures

- class [jeod::TimeStandard](#)

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

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.49.1 Detailed Description

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

## 9.50 time\_tai.cc File Reference

Define member functions for International Atomic Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tai.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.50.1 Detailed Description

Define member functions for International Atomic Time.

## 9.51 time\_tai.hh File Reference

Represents International Atomic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeTAI](#)  
*Represents International Atomic Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.51.1 Detailed Description

Represents International Atomic Time.

## 9.52 time\_tdb.cc File Reference

Define member functions Barycentric Dynamic Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tdb.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.52.1 Detailed Description

Define member functions Barycentric Dynamic Time.

## 9.53 time\_tdb.hh File Reference

Represents Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeTDB](#)  
*Represents Barycentric Dynamic Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.53.1 Detailed Description

Represents Barycentric Dynamic Time.

## 9.54 time\_tt.cc File Reference

Define member functions for Terrestrial Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tt.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.54.1 Detailed Description

Define member functions for Terrestrial Time.

## 9.55 time\_tt.hh File Reference

Represents Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeTT](#)  
*Represents Terrestrial Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.55.1 Detailed Description

Represents Terrestrial Time.

## 9.56 time\_ude.cc File Reference

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

```
#include <cmath>
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.56.1 Detailed Description

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

## 9.57 time\_ude.hh File Reference

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

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
#include "time_enum.hh"
```

### Data Structures

- class [jeod::TimeUDE](#)

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

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.57.1 Detailed Description

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

## 9.58 time\_ut1.cc File Reference

Define member functions for Universal Time.

```
#include <cstdlib>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ut1.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.58.1 Detailed Description

Define member functions for Universal Time.

## 9.59 time\_ut1.hh File Reference

Represents Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeUT1](#)  
*Represents Universal Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.59.1 Detailed Description

Represents Universal Time.

## 9.60 time\_utc.cc File Reference

Define member functions for Coordinated Universal Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_utc.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.60.1 Detailed Description

Define member functions for Coordinated Universal Time.

## 9.61 time\_utc.hh File Reference

Represents Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeUTC](#)  
*Represents Coordinated Universal Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.61.1 Detailed Description

Represents Coordinated Universal Time.





# Index

- ~JeodBaseTime
  - jeod::JeodBaseTime, [25](#)
- ~TimeConverter
  - jeod::TimeConverter, [41](#)
- ~TimeConverter\_Dyn\_TAI
  - jeod::TimeConverter\_Dyn\_TAI, [50](#)
- ~TimeConverter\_Dyn\_TDB
  - jeod::TimeConverter\_Dyn\_TDB, [54](#)
- ~TimeConverter\_Dyn\_UDE
  - jeod::TimeConverter\_Dyn\_UDE, [58](#)
- ~TimeConverter\_STD\_UDE
  - jeod::TimeConverter\_STD\_UDE, [62](#)
- ~TimeConverter\_TAI\_GPS
  - jeod::TimeConverter\_TAI\_GPS, [67](#)
- ~TimeConverter\_TAI\_TT
  - jeod::TimeConverter\_TAI\_TT, [78](#)
- ~TimeConverter\_TAI\_TDB
  - jeod::TimeConverter\_TAI\_TDB, [71](#)
- ~TimeConverter\_TAI\_UT1
  - jeod::TimeConverter\_TAI\_UT1, [82](#)
- ~TimeConverter\_TAI\_UTC
  - jeod::TimeConverter\_TAI\_UTC, [92](#)
- ~TimeConverter\_UT1\_GMST
  - jeod::TimeConverter\_UT1\_GMST, [100](#)
- ~TimeDyn
  - jeod::TimeDyn, [104](#)
- ~TimeGMST
  - jeod::TimeGMST, [110](#)
- ~TimeGPS
  - jeod::TimeGPS, [114](#)
- ~TimeLinks
  - jeod::TimeLinks, [121](#)
- ~TimeMET
  - jeod::TimeMET, [154](#)
- ~TimeManager
  - jeod::TimeManager, [124](#)
- ~TimeManagerInit
  - jeod::TimeManagerInit, [136](#)
- ~TimeStandard
  - jeod::TimeStandard, [159](#)
- ~TimeTT
  - jeod::TimeTT, [177](#)
- ~TimeTAI
  - jeod::TimeTAI, [171](#)
- ~TimeTDB
  - jeod::TimeTDB, [174](#)
- ~TimeUDE
  - jeod::TimeUDE, [181](#)
- ~TimeUT1
  - jeod::TimeUT1, [198](#)
- ~TimeUTC
  - jeod::TimeUTC, [201](#)
- a\_name
  - jeod::TimeConverter, [46](#)
- a\_to\_b\_offset
  - jeod::TimeConverter, [47](#)
  - jeod::TimeConverter\_TAI\_TDB, [74](#)
- a\_to\_b\_offset\_epoch
  - jeod::TimeConverter\_TAI\_TDB, [74](#)
- add\_parent
  - jeod::JeodBaseTime, [26](#)
- add\_type\_initialize
  - jeod::JeodBaseTime, [26](#)
  - jeod::TimeStandard, [159](#)
  - jeod::TimeUDE, [182](#)
- add\_type\_update
  - jeod::JeodBaseTime, [27](#)
- b\_name
  - jeod::TimeConverter, [47](#)
- calculate\_calendar\_values
  - jeod::TimeGMST, [111](#)
  - jeod::TimeGPS, [114](#)
  - jeod::TimeStandard, [160](#)
- calendar\_day
  - jeod::TimeStandard, [166](#)
- calendar\_hour
  - jeod::TimeStandard, [166](#)
- calendar\_minute
  - jeod::TimeStandard, [166](#)
- calendar\_month
  - jeod::TimeStandard, [167](#)
- calendar\_second
  - jeod::TimeStandard, [167](#)
- calendar\_update
  - jeod::TimeStandard, [160](#)
- calendar\_year
  - jeod::TimeStandard, [167](#)
- can\_convert
  - jeod::TimeConverter, [42](#)
- class\_declarations.hh, [205](#)
- clock\_day
  - jeod::TimeUDE, [191](#)
- clock\_hour
  - jeod::TimeUDE, [191](#)
- clock\_minute
  - jeod::TimeUDE, [191](#)

- clock\_resolution
  - jeod::JeodBaseTime, [34](#)
- clock\_second
  - jeod::TimeUDE, [191](#)
- clock\_update
  - jeod::TimeUDE, [182](#)
- convert\_a\_to\_b
  - jeod::TimeConverter, [42](#)
  - 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, [72](#)
  - jeod::TimeConverter\_TAI\_TT, [78](#)
  - jeod::TimeConverter\_TAI\_UT1, [83](#)
  - jeod::TimeConverter\_TAI\_UTC, [93](#)
  - jeod::TimeConverter\_UT1\_GMST, [101](#)
- convert\_b\_to\_a
  - jeod::TimeConverter, [43](#)
  - jeod::TimeConverter\_STD\_UDE, [63](#)
  - jeod::TimeConverter\_TAI\_GPS, [68](#)
  - jeod::TimeConverter\_TAI\_TDB, [72](#)
  - jeod::TimeConverter\_TAI\_TT, [78](#)
  - jeod::TimeConverter\_TAI\_UT1, [83](#)
  - jeod::TimeConverter\_TAI\_UTC, [93](#)
- convert\_epoch\_to\_update
  - jeod::TimeUDE, [182](#)
- convert\_from\_calendar
  - jeod::TimeGPS, [115](#)
  - jeod::TimeStandard, [161](#)
- converter\_ptrs\_index
  - jeod::TimeManagerInit, [144](#)
- converter\_vector
  - jeod::TimeManager, [132](#)
- create\_init\_tree
  - jeod::TimeManagerInit, [137](#)
- create\_update\_tree
  - jeod::TimeManagerInit, [137](#)
- day\_of\_week
  - jeod::TimeGPS, [118](#)
- days
  - jeod::JeodBaseTime, [34](#)
- default\_path\_size
  - jeod::TimeLinks, [122](#)
- Direction
  - jeod::TimeConverter, [41](#)
- duplicate\_methods
  - jeod::TimeMessages, [149](#)
- dyn\_ptr
  - jeod::TimeConverter\_Dyn\_TAI, [51](#)
  - jeod::TimeConverter\_Dyn\_TDB, [55](#)
  - jeod::TimeConverter\_Dyn\_UDE, [60](#)
- dyn\_time
  - jeod::TimeManager, [132](#)
- dyn\_time\_index
  - jeod::TimeManagerInit, [144](#)
- Environment, [14](#)
- epoch\_data\_present
  - jeod::TimeUDE, [192](#)
- epoch\_day
  - jeod::TimeUDE, [192](#)
- epoch\_defined\_in\_name
  - jeod::TimeUDE, [192](#)
- epoch\_format
  - jeod::TimeUDE, [192](#)
- epoch\_hour
  - jeod::TimeUDE, [193](#)
- epoch\_index
  - jeod::TimeUDE, [193](#)
- epoch\_initializing\_value
  - jeod::TimeUDE, [193](#)
- epoch\_minute
  - jeod::TimeUDE, [193](#)
- epoch\_month
  - jeod::TimeUDE, [194](#)
- epoch\_second
  - jeod::TimeUDE, [194](#)
- epoch\_value\_is\_set\_calendar
  - jeod::TimeUDE, [194](#)
- epoch\_value\_is\_set\_clock
  - jeod::TimeUDE, [194](#)
- epoch\_value\_is\_set\_number
  - jeod::TimeUDE, [195](#)
- epoch\_year
  - jeod::TimeUDE, [195](#)
- extension\_error
  - jeod::TimeMessages, [149](#)
- failed\_null\_test
  - jeod::TimeConverter\_STD\_UDE, [65](#)
- get\_a\_to\_b\_offset
  - jeod::TimeConverter, [43](#)
- get\_conv\_dir\_init
  - jeod::TimeManagerInit, [137](#)
- get\_conv\_dir\_upd
  - jeod::TimeManagerInit, [138](#)
- get\_conv\_ptr\_index
  - jeod::TimeManagerInit, [139](#)
- get\_converter\_ptr
  - jeod::TimeManager, [124](#)
- get\_days
  - jeod::TimeUT1, [198](#)
- get\_index
  - jeod::JeodBaseTime, [28](#)
- get\_jeod\_integration\_time
  - jeod::TimeManager, [125](#)
- get\_status
  - jeod::TimeManagerInit, [139](#)
- get\_time\_change\_flag
  - jeod::TimeManager, [125](#)
- get\_time\_ptr
  - jeod::TimeManager, [125](#), [126](#)
- get\_time\_scale\_factor
  - jeod::TimeManager, [126](#)

- get\_timestamp\_time
  - jeod::TimeManager, 126
- gmst\_ptr
  - jeod::TimeConverter\_UT1\_GMST, 102
- gps\_ptr
  - jeod::TimeConverter\_TAI\_GPS, 69
- gradient
  - jeod::TimeConverter\_TAI\_UT1, 86
- hold
  - jeod::TimeMET, 156
- incomplete\_setup\_error
  - jeod::TimeMessages, 149
- increment\_status
  - jeod::TimeManagerInit, 140
- index
  - jeod::JeodBaseTime, 34
  - jeod::TimeConverter\_TAI\_UT1, 86
  - jeod::TimeConverter\_TAI\_UTC, 95
- init\_attrjeod\_\_JeodBaseTime
  - jeod::JeodBaseTime, 33
- init\_attrjeod\_\_TimeConverter
  - jeod::TimeConverter, 46
- init\_attrjeod\_\_TimeConverter\_Dyn\_TAI
  - jeod::TimeConverter\_Dyn\_TAI, 51
- init\_attrjeod\_\_TimeConverter\_Dyn\_TDB
  - jeod::TimeConverter\_Dyn\_TDB, 55
- init\_attrjeod\_\_TimeConverter\_Dyn\_UDE
  - jeod::TimeConverter\_Dyn\_UDE, 60
- init\_attrjeod\_\_TimeConverter\_STD\_UDE
  - jeod::TimeConverter\_STD\_UDE, 64
- init\_attrjeod\_\_TimeConverter\_TAI\_GPS
  - jeod::TimeConverter\_TAI\_GPS, 69
- init\_attrjeod\_\_TimeConverter\_TAI\_TDB
  - jeod::TimeConverter\_TAI\_TDB, 73
- init\_attrjeod\_\_TimeConverter\_TAI\_TT
  - jeod::TimeConverter\_TAI\_TT, 79
- init\_attrjeod\_\_TimeConverter\_TAI\_UT1
  - jeod::TimeConverter\_TAI\_UT1, 86
- init\_attrjeod\_\_TimeConverter\_TAI\_UTC
  - jeod::TimeConverter\_TAI\_UTC, 95
- init\_attrjeod\_\_TimeConverter\_UT1\_GMST
  - jeod::TimeConverter\_UT1\_GMST, 102
- init\_attrjeod\_\_TimeDyn
  - jeod::TimeDyn, 106
- init\_attrjeod\_\_TimeGMST
  - jeod::TimeGMST, 112
- init\_attrjeod\_\_TimeGPS
  - jeod::TimeGPS, 117
- init\_attrjeod\_\_TimeLinks
  - jeod::TimeLinks, 121
- init\_attrjeod\_\_TimeMET
  - jeod::TimeMET, 155
- init\_attrjeod\_\_TimeManager
  - jeod::TimeManager, 131
- init\_attrjeod\_\_TimeManagerInit
  - jeod::TimeManagerInit, 144
- init\_attrjeod\_\_TimeMessages
  - jeod::TimeMessages, 148
- init\_attrjeod\_\_TimeStandard
  - jeod::TimeStandard, 165
- init\_attrjeod\_\_TimeTAI
  - jeod::TimeTAI, 172
- init\_attrjeod\_\_TimeTDB
  - jeod::TimeTDB, 175
- init\_attrjeod\_\_TimeTT
  - jeod::TimeTT, 178
- init\_attrjeod\_\_TimeUDE
  - jeod::TimeUDE, 190
- init\_attrjeod\_\_TimeUT1
  - jeod::TimeUT1, 199
- init\_attrjeod\_\_TimeUTC
  - jeod::TimeUTC, 202
- init\_converter\_dir\_table
  - jeod::TimeManagerInit, 144
- initial\_value
  - jeod::JeodBaseTime, 34
- initial\_value\_format
  - jeod::TimeUDE, 195
- initialization\_error
  - jeod::TimeMessages, 150
- initialize
  - jeod::TimeConverter, 43
  - jeod::TimeConverter\_Dyn\_TAI, 50
  - jeod::TimeConverter\_Dyn\_TDB, 54
  - jeod::TimeConverter\_Dyn\_UDE, 58
  - jeod::TimeConverter\_STD\_UDE, 63
  - jeod::TimeConverter\_TAI\_GPS, 68
  - jeod::TimeConverter\_TAI\_TDB, 72
  - jeod::TimeConverter\_TAI\_TT, 79
  - jeod::TimeConverter\_TAI\_UT1, 84
  - jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1 ↔
    - default\_data, 90
  - jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc ↔
    - default\_data, 99
  - jeod::TimeConverter\_TAI\_UTC, 93
  - jeod::TimeConverter\_UT1\_GMST, 101
  - jeod::TimeManager, 127
  - jeod::TimeManagerInit, 140
- initialize\_from\_name
  - jeod::JeodBaseTime, 35
- initialize\_from\_parent
  - jeod::JeodBaseTime, 28
  - jeod::TimeStandard, 161
  - jeod::TimeUDE, 183
- initialize\_initializer\_time
  - jeod::JeodBaseTime, 28
  - jeod::TimeDyn, 105
  - jeod::TimeStandard, 162
  - jeod::TimeUDE, 184
- initialize\_leap\_second
  - jeod::TimeConverter\_TAI\_UTC, 94
- initialize\_manager
  - jeod::TimeManagerInit, 140
- initialize\_tai\_to\_ut1
  - jeod::TimeConverter\_TAI\_UT1, 84

- initialize\_time\_types
  - jeod::TimeManagerInit, 141
- initialized
  - jeod::JeodBaseTime, 35
  - jeod::TimeConverter, 47
- initializer
  - jeod::TimeManagerInit, 145
- initializer\_index
  - jeod::TimeManagerInit, 145
- initializing\_data\_present
  - jeod::TimeUDE, 195
- initializing\_value
  - jeod::JeodBaseTime, 35
- InputProcessor
  - jeod::JeodBaseTime, 33
  - jeod::TimeConverter, 46
  - jeod::TimeConverter\_Dyn\_TAI, 51
  - jeod::TimeConverter\_Dyn\_TDB, 55
  - jeod::TimeConverter\_Dyn\_UDE, 60
  - jeod::TimeConverter\_STD\_UDE, 64
  - jeod::TimeConverter\_TAI\_GPS, 69
  - jeod::TimeConverter\_TAI\_TDB, 73
  - jeod::TimeConverter\_TAI\_TT, 80
  - jeod::TimeConverter\_TAI\_UT1, 86
  - jeod::TimeConverter\_TAI\_UTC, 95
  - jeod::TimeConverter\_UT1\_GMST, 102
  - jeod::TimeDyn, 107
  - jeod::TimeGMST, 112
  - jeod::TimeGPS, 117
  - jeod::TimeLinks, 121
  - jeod::TimeMET, 156
  - jeod::TimeManager, 132
  - jeod::TimeManagerInit, 144
  - jeod::TimeMessages, 149
  - jeod::TimeStandard, 166
  - jeod::TimeTAI, 173
  - jeod::TimeTDB, 175
  - jeod::TimeTT, 178
  - jeod::TimeUDE, 190
  - jeod::TimeUT1, 199
  - jeod::TimeUTC, 202
- invalid\_data\_error
  - jeod::TimeMessages, 150
- invalid\_node
  - jeod::TimeMessages, 151
- invalid\_setup\_error
  - jeod::TimeMessages, 151
- is\_initialized
  - jeod::JeodBaseTime, 30
  - jeod::TimeConverter, 44
- JEOD\_FRIEND\_CLASS
  - tai\_to\_ut1.cc, 206
  - tai\_to\_utc.cc, 206
- 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, 47
  - b\_name, 47
  - can\_convert, 42
  - convert\_a\_to\_b, 42
  - convert\_b\_to\_a, 43
  - 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, 42
  - valid\_directions, 48
  - verify\_setup, 45
  - verify\_table\_lookup\_ends, 45
- jeod::TimeConverter\_Dyn\_TAI, 48
  - ~TimeConverter\_Dyn\_TAI, 50

- convert\_a\_to\_b, 50
- dyn\_ptr, 51
- init\_attrjeod\_\_TimeConverter\_Dyn\_TAI, 51
- initialize, 50
- InputProcessor, 51
- operator=, 51
- tai\_ptr, 52
- TimeConverter\_Dyn\_TAI, 49, 50
- jeod::TimeConverter\_Dyn\_TDB, 52
  - ~TimeConverter\_Dyn\_TDB, 54
  - convert\_a\_to\_b, 54
  - dyn\_ptr, 55
  - init\_attrjeod\_\_TimeConverter\_Dyn\_TDB, 55
  - initialize, 54
  - InputProcessor, 55
  - operator=, 55
  - tdb\_ptr, 56
  - TimeConverter\_Dyn\_TDB, 53, 54
- jeod::TimeConverter\_Dyn\_UDE, 56
  - ~TimeConverter\_Dyn\_UDE, 58
  - convert\_a\_to\_b, 58
  - dyn\_ptr, 60
  - init\_attrjeod\_\_TimeConverter\_Dyn\_UDE, 60
  - initialize, 58
  - InputProcessor, 60
  - operator=, 59
  - reset\_a\_to\_b\_offset, 59
  - TimeConverter\_Dyn\_UDE, 57, 58
  - ude\_ptr, 60
- jeod::TimeConverter\_STD\_UDE, 61
  - ~TimeConverter\_STD\_UDE, 62
  - convert\_a\_to\_b, 62
  - convert\_b\_to\_a, 63
  - failed\_null\_test, 65
  - init\_attrjeod\_\_TimeConverter\_STD\_UDE, 64
  - initialize, 63
  - InputProcessor, 64
  - operator=, 64
  - reset\_a\_to\_b\_offset, 64
  - std\_ptr, 65
  - TimeConverter\_STD\_UDE, 62
  - ude\_ptr, 65
- jeod::TimeConverter\_TAI\_GPS, 66
  - ~TimeConverter\_TAI\_GPS, 67
  - convert\_a\_to\_b, 67
  - convert\_b\_to\_a, 68
  - gps\_ptr, 69
  - init\_attrjeod\_\_TimeConverter\_TAI\_GPS, 69
  - initialize, 68
  - InputProcessor, 69
  - operator=, 69
  - tai\_ptr, 69
  - TimeConverter\_TAI\_GPS, 67
- jeod::TimeConverter\_TAI\_TDB, 70
  - ~TimeConverter\_TAI\_TDB, 71
  - a\_to\_b\_offset, 74
  - a\_to\_b\_offset\_epoch, 74
  - convert\_a\_to\_b, 72
  - convert\_b\_to\_a, 72
  - init\_attrjeod\_\_TimeConverter\_TAI\_TDB, 73
  - initialize, 72
  - InputProcessor, 73
  - nlter, 74
  - nSteps, 74
  - operator=, 73
  - prev\_tai\_seconds, 75
  - prev\_tdb\_seconds, 75
  - set\_a\_to\_b\_offset, 73
  - TAI\_to\_TT\_offset, 75
  - tai\_ptr, 75
  - tdb\_ptr, 76
  - TimeConverter\_TAI\_TDB, 71, 72
- jeod::TimeConverter\_TAI\_TT, 76
  - ~TimeConverter\_TAI\_TT, 78
  - convert\_a\_to\_b, 78
  - convert\_b\_to\_a, 78
  - init\_attrjeod\_\_TimeConverter\_TAI\_TT, 79
  - initialize, 79
  - InputProcessor, 80
  - operator=, 79
  - tai\_ptr, 80
  - TimeConverter\_TAI\_TT, 77, 78
  - tt\_ptr, 80
- jeod::TimeConverter\_TAI\_UT1, 81
  - ~TimeConverter\_TAI\_UT1, 82
  - convert\_a\_to\_b, 83
  - convert\_b\_to\_a, 83
  - gradient, 86
  - index, 86
  - init\_attrjeod\_\_TimeConverter\_TAI\_UT1, 86
  - initialize, 84
  - initialize\_tai\_to\_ut1, 84
  - InputProcessor, 86
  - last\_index, 86
  - next\_value, 87
  - next\_when, 87
  - off\_table\_end, 87
  - operator=, 85
  - override\_data\_table, 87
  - prev\_value, 88
  - prev\_when, 88
  - tai\_ptr, 88
  - tai\_to\_ut1\_override\_val, 88
  - TimeConverter\_TAI\_UT1, 82, 83
  - ut1\_ptr, 89
  - val\_vec, 89
  - verify\_table\_lookup\_ends, 85
  - when\_vec, 89
- jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_↔
  - data, 90
  - initialize, 90
- jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_↔
  - data, 98
  - initialize, 99
- jeod::TimeConverter\_TAI\_UTC, 90
  - ~TimeConverter\_TAI\_UTC, 92

- convert\_a\_to\_b, 93
- convert\_b\_to\_a, 93
- index, 95
- init\_attrjeod\_\_TimeConverter\_TAI.UTC, 95
- initialize, 93
- initialize\_leap\_second, 94
- InputProcessor, 95
- last\_index, 95
- leap\_sec\_override\_val, 96
- next\_when, 96
- off\_table\_end, 96
- operator=, 94
- override\_data\_table, 96
- prev\_when, 97
- tai\_ptr, 97
- TimeConverter\_TAI.UTC, 92
- utc\_ptr, 97
- val\_vec, 97
- verify\_table\_lookup\_ends, 94
- when\_vec, 98
- jeod::TimeConverter\_UT1\_GMST, 99
  - ~TimeConverter\_UT1\_GMST, 100
  - convert\_a\_to\_b, 101
  - gmst\_ptr, 102
  - init\_attrjeod\_\_TimeConverter\_UT1\_GMST, 102
  - initialize, 101
  - InputProcessor, 102
  - operator=, 102
  - TimeConverter\_UT1\_GMST, 100
  - ut1\_ptr, 102
- jeod::TimeDyn, 103
  - ~TimeDyn, 104
  - init\_attrjeod\_\_TimeDyn, 106
  - initialize\_initializer\_time, 105
  - InputProcessor, 107
  - offset, 107
  - operator=, 105
  - ref\_scale, 107
  - scale\_factor, 107
  - TimeDyn, 104, 105
  - update, 106
  - update\_offset, 106
- jeod::TimeEnum, 108
  - TimeFormat, 108
- jeod::TimeGMST, 109
  - ~TimeGMST, 110
  - calculate\_calendar\_values, 111
  - init\_attrjeod\_\_TimeGMST, 112
  - InputProcessor, 112
  - operator=, 111
  - set\_epoch, 111
  - set\_time\_by\_trunc\_julian, 111
  - TimeGMST, 110
- jeod::TimeGPS, 112
  - ~TimeGPS, 114
  - calculate\_calendar\_values, 114
  - convert\_from\_calendar, 115
  - day\_of\_week, 118
  - init\_attrjeod\_\_TimeGPS, 117
  - InputProcessor, 117
  - operator=, 115
  - rollover\_count, 118
  - rollover\_count\_13\_bit, 118
  - seconds\_of\_day, 118
  - seconds\_of\_week, 119
  - set\_epoch, 115
  - set\_time\_by\_days, 116
  - set\_time\_by\_seconds, 116
  - set\_time\_by\_trunc\_julian, 117
  - TimeGPS, 114
  - week, 119
  - week\_13\_bit, 119
- jeod::TimeLinks, 120
  - ~TimeLinks, 121
  - default\_path\_size, 122
  - init\_attrjeod\_\_TimeLinks, 121
  - InputProcessor, 121
  - operator=, 121
  - TimeLinks, 120, 121
- jeod::TimeMET, 153
  - ~TimeMET, 154
  - hold, 156
  - init\_attrjeod\_\_TimeMET, 155
  - InputProcessor, 156
  - operator=, 155
  - previous\_hold, 156
  - TimeMET, 154, 155
  - update, 155
- jeod::TimeManager, 122
  - ~TimeManager, 124
  - converter\_vector, 132
  - dyn\_time, 132
  - get\_converter\_ptr, 124
  - get\_jeod\_integration\_time, 125
  - get\_time\_change\_flag, 125
  - get\_time\_ptr, 125, 126
  - get\_time\_scale\_factor, 126
  - get\_timestamp\_time, 126
  - init\_attrjeod\_\_TimeManager, 131
  - initialize, 127
  - InputProcessor, 132
  - num\_types, 132
  - operator=, 127
  - register\_converter, 127
  - register\_time, 128
  - register\_time\_named, 128
  - simtime, 133
  - time\_change\_flag, 133
  - time\_lookup, 129
  - time\_standards\_exist, 129
  - time\_vector, 133
  - TimeManager, 124
  - TimeManagerInit, 132
  - update, 130
  - update\_time, 130
  - verify\_table\_lookup\_ends, 131

- jeod::TimeManagerInit, 134
  - ~TimeManagerInit, 136
  - converter\_ptrs\_index, 144
  - create\_init\_tree, 137
  - create\_update\_tree, 137
  - dyn\_time\_index, 144
  - get\_conv\_dir\_init, 137
  - get\_conv\_dir\_upd, 138
  - get\_conv\_ptr\_index, 139
  - get\_status, 139
  - increment\_status, 140
  - init\_attrjeod\_\_TimeManagerInit, 144
  - init\_converter\_dir\_table, 144
  - initialize, 140
  - initialize\_manager, 140
  - initialize\_time\_types, 141
  - initializer, 145
  - initializer\_index, 145
  - InputProcessor, 144
  - num\_added\_pass, 145
  - num\_added\_total, 145
  - operator=, 141
  - organize\_update\_list, 142
  - populate\_converter\_registry, 142
  - set\_status, 142
  - sim\_start\_format, 146
  - status, 146
  - time\_manager, 146
  - TimeManagerInit, 136
  - update\_converter\_dir\_table, 147
  - verify\_converter\_setup, 143
  - verify\_times\_setup, 143
- jeod::TimeMessages, 147
  - duplicate\_methods, 149
  - extension\_error, 149
  - incomplete\_setup\_error, 149
  - init\_attrjeod\_\_TimeMessages, 148
  - initialization\_error, 150
  - InputProcessor, 149
  - invalid\_data\_error, 150
  - invalid\_node, 151
  - invalid\_setup\_error, 151
  - memory\_error, 152
  - operator=, 148
  - redundancy\_error, 152
  - TimeMessages, 148
- jeod::TimeStandard, 157
  - ~TimeStandard, 159
  - add\_type\_initialize, 159
  - calculate\_calendar\_values, 160
  - calendar\_day, 166
  - calendar\_hour, 166
  - calendar\_minute, 166
  - calendar\_month, 167
  - calendar\_second, 167
  - calendar\_update, 160
  - calendar\_year, 167
  - convert\_from\_calendar, 161
  - init\_attrjeod\_\_TimeStandard, 165
  - initialize\_from\_parent, 161
  - initialize\_initializer\_time, 162
  - InputProcessor, 166
  - julian\_date, 167
  - julian\_date\_at\_epoch, 162
  - last\_calendar\_update, 168
  - operator=, 163
  - prev\_julian\_day, 168
  - seconds\_at\_year\_start, 168
  - seconds\_of\_year, 163
  - send\_warning\_pre\_1968, 168
  - set\_epoch, 163
  - set\_time\_by\_days, 164
  - set\_time\_by\_seconds, 164
  - set\_time\_by\_trunc\_julian, 165
  - TimeStandard, 159
  - TimeUDE, 166
  - tjt\_at\_epoch, 169
  - tjt\_jd\_offset, 169
  - tjt\_mjt\_offset, 169
  - trunc\_julian\_time, 169
  - year\_of\_last\_soy, 170
- jeod::TimeTAI, 170
  - ~TimeTAI, 171
  - init\_attrjeod\_\_TimeTAI, 172
  - InputProcessor, 173
  - operator=, 172
  - set\_epoch, 172
  - TimeTAI, 171, 172
- jeod::TimeTDB, 173
  - ~TimeTDB, 174
  - init\_attrjeod\_\_TimeTDB, 175
  - InputProcessor, 175
  - operator=, 175
  - set\_epoch, 175
  - TimeTDB, 174
- jeod::TimeTT, 176
  - ~TimeTT, 177
  - init\_attrjeod\_\_TimeTT, 178
  - InputProcessor, 178
  - operator=, 177
  - set\_epoch, 177
  - TimeTT, 176, 177
- jeod::TimeUDE, 178
  - ~TimeUDE, 181
  - add\_type\_initialize, 182
  - clock\_day, 191
  - clock\_hour, 191
  - clock\_minute, 191
  - clock\_second, 191
  - clock\_update, 182
  - convert\_epoch\_to\_update, 182
  - epoch\_data\_present, 192
  - epoch\_day, 192
  - epoch\_defined\_in\_name, 192
  - epoch\_format, 192
  - epoch\_hour, 193



- epoch\_index, [193](#)
- epoch\_initializing\_value, [193](#)
- epoch\_minute, [193](#)
- epoch\_month, [194](#)
- epoch\_second, [194](#)
- epoch\_value\_is\_set\_calendar, [194](#)
- epoch\_value\_is\_set\_clock, [194](#)
- epoch\_value\_is\_set\_number, [195](#)
- epoch\_year, [195](#)
- init\_attrjeod\_\_TimeUDE, [190](#)
- initial\_value\_format, [195](#)
- initialize\_from\_parent, [183](#)
- initialize\_initializer\_time, [184](#)
- initializing\_data\_present, [195](#)
- InputProcessor, [190](#)
- last\_clock\_update, [196](#)
- must\_be\_singleton, [184](#)
- operator=, [185](#)
- set\_epoch\_dyn, [185](#)
- set\_epoch\_initializing\_value, [185](#)
- set\_epoch\_std, [186](#)
- set\_epoch\_times, [187](#)
- set\_epoch\_ude, [187](#)
- set\_initial\_times, [188](#)
- set\_time\_by\_clock, [188](#)
- set\_time\_by\_days, [188](#)
- set\_time\_by\_seconds, [189](#)
- TimeUDE, [181](#)
- update\_index, [196](#)
- verify\_epoch, [189](#)
- verify\_init, [189](#)
- verify\_update, [190](#)
- jeod::TimeUT1, [197](#)
  - ~TimeUT1, [198](#)
  - get\_days, [198](#)
  - init\_attrjeod\_\_TimeUT1, [199](#)
  - InputProcessor, [199](#)
  - operator=, [198](#)
  - set\_epoch, [199](#)
  - TimeUT1, [198](#)
  - true\_ut1, [199](#)
- jeod::TimeUTC, [200](#)
  - ~TimeUTC, [201](#)
  - init\_attrjeod\_\_TimeUTC, [202](#)
  - InputProcessor, [202](#)
  - operator=, [202](#)
  - set\_epoch, [202](#)
  - TimeUTC, [201](#)
  - true\_utc, [203](#)
- JeodBaseTime
  - jeod::JeodBaseTime, [25](#), [26](#)
  - jeod::TimeConverter, [46](#)
- julian\_date
  - jeod::TimeStandard, [167](#)
- julian\_date\_at\_epoch
  - jeod::TimeStandard, [162](#)
- last\_calendar\_update
  - jeod::TimeStandard, [168](#)
- last\_clock\_update
  - jeod::TimeUDE, [196](#)
- last\_index
  - jeod::TimeConverter\_TAI\_UT1, [86](#)
  - jeod::TimeConverter\_TAI\_UTC, [95](#)
- leap\_sec\_override\_val
  - jeod::TimeConverter\_TAI\_UTC, [96](#)
- links
  - jeod::JeodBaseTime, [36](#)
- memory\_error
  - jeod::TimeMessages, [152](#)
- Models, [13](#)
- must\_be\_singleton
  - jeod::JeodBaseTime, [30](#)
  - jeod::TimeUDE, [184](#)
- nIter
  - jeod::TimeConverter\_TAI\_TDB, [74](#)
- nSteps
  - jeod::TimeConverter\_TAI\_TDB, [74](#)
- name
  - jeod::JeodBaseTime, [36](#)
- next\_value
  - jeod::TimeConverter\_TAI\_UT1, [87](#)
- next\_when
  - jeod::TimeConverter\_TAI\_UT1, [87](#)
  - jeod::TimeConverter\_TAI\_UTC, [96](#)
- num\_added\_pass
  - jeod::TimeManagerInit, [145](#)
- num\_added\_total
  - jeod::TimeManagerInit, [145](#)
- num\_types
  - jeod::TimeManager, [132](#)
- off\_table\_end
  - jeod::TimeConverter\_TAI\_UT1, [87](#)
  - jeod::TimeConverter\_TAI\_UTC, [96](#)
- offset
  - jeod::TimeDyn, [107](#)
- operator=
  - jeod::JeodBaseTime, [30](#)
  - jeod::TimeConverter, [44](#)
  - jeod::TimeConverter\_Dyn\_TAI, [51](#)
  - jeod::TimeConverter\_Dyn\_TDB, [55](#)
  - jeod::TimeConverter\_Dyn\_UDE, [59](#)
  - jeod::TimeConverter\_STD\_UDE, [64](#)
  - jeod::TimeConverter\_TAI\_GPS, [69](#)
  - jeod::TimeConverter\_TAI\_TDB, [73](#)
  - jeod::TimeConverter\_TAI\_TT, [79](#)
  - jeod::TimeConverter\_TAI\_UT1, [85](#)
  - jeod::TimeConverter\_TAI\_UTC, [94](#)
  - jeod::TimeConverter\_UT1\_GMST, [102](#)
  - jeod::TimeDyn, [105](#)
  - jeod::TimeGMST, [111](#)
  - jeod::TimeGPS, [115](#)
  - jeod::TimeLinks, [121](#)
  - jeod::TimeMET, [155](#)
  - jeod::TimeManager, [127](#)



- jeod::TimeManagerInit, 141
- jeod::TimeMessages, 148
- jeod::TimeStandard, 163
- jeod::TimeTAI, 172
- jeod::TimeTDB, 175
- jeod::TimeTT, 177
- jeod::TimeUDE, 185
- jeod::TimeUT1, 198
- jeod::TimeUTC, 202
- operator |
  - jeod, 20
- organize\_update\_list
  - jeod::TimeManagerInit, 142
- override\_data\_table
  - jeod::TimeConverter\_TAI\_UT1, 87
  - jeod::TimeConverter\_TAI\_UTC, 96
- override\_initialized
  - jeod::JeodBaseTime, 30
  - jeod::TimeConverter, 44
- PATH
  - Time, 17
- populate\_converter\_registry
  - jeod::TimeManagerInit, 142
- prev\_julian\_day
  - jeod::TimeStandard, 168
- prev\_tai\_seconds
  - jeod::TimeConverter\_TAI\_TDB, 75
- prev\_tdb\_seconds
  - jeod::TimeConverter\_TAI\_TDB, 75
- prev\_value
  - jeod::TimeConverter\_TAI\_UT1, 88
- prev\_when
  - jeod::TimeConverter\_TAI\_UT1, 88
  - jeod::TimeConverter\_TAI\_UTC, 97
- previous\_hold
  - jeod::TimeMET, 156
- redundancy\_error
  - jeod::TimeMessages, 152
- ref\_scale
  - jeod::TimeDyn, 107
- register\_converter
  - jeod::TimeManager, 127
- register\_time
  - jeod::TimeManager, 128
- register\_time\_named
  - jeod::TimeManager, 128
- reset\_a\_to\_b\_offset
  - jeod::TimeConverter, 44
  - jeod::TimeConverter\_Dyn\_UDE, 59
  - jeod::TimeConverter\_STD\_UDE, 64
- rollover\_count
  - jeod::TimeGPS, 118
- rollover\_count\_13\_bit
  - jeod::TimeGPS, 118
- scale\_factor
  - jeod::TimeDyn, 107
- seconds
  - jeod::JeodBaseTime, 36
- seconds\_at\_year\_start
  - jeod::TimeStandard, 168
- seconds\_of\_day
  - jeod::TimeGPS, 118
- seconds\_of\_week
  - jeod::TimeGPS, 119
- seconds\_of\_year
  - jeod::TimeStandard, 163
- send\_warning\_pre\_1968
  - jeod::TimeStandard, 168
- set\_a\_to\_b\_offset
  - jeod::TimeConverter\_TAI\_TDB, 73
- set\_epoch
  - jeod::TimeGMST, 111
  - jeod::TimeGPS, 115
  - jeod::TimeStandard, 163
  - jeod::TimeTAI, 172
  - jeod::TimeTDB, 175
  - jeod::TimeTT, 177
  - jeod::TimeUT1, 199
  - jeod::TimeUTC, 202
- set\_epoch\_dyn
  - jeod::TimeUDE, 185
- set\_epoch\_initializing\_value
  - jeod::TimeUDE, 185
- set\_epoch\_std
  - jeod::TimeUDE, 186
- set\_epoch\_times
  - jeod::TimeUDE, 187
- set\_epoch\_ude
  - jeod::TimeUDE, 187
- set\_index
  - jeod::JeodBaseTime, 31
- set\_initial\_times
  - jeod::TimeUDE, 188
- set\_name
  - jeod::JeodBaseTime, 31
- set\_status
  - jeod::TimeManagerInit, 142
- set\_time\_by\_clock
  - jeod::TimeUDE, 188
- set\_time\_by\_days
  - jeod::JeodBaseTime, 31
  - jeod::TimeGPS, 116
  - jeod::TimeStandard, 164
  - jeod::TimeUDE, 188
- set\_time\_by\_seconds
  - jeod::JeodBaseTime, 32
  - jeod::TimeGPS, 116
  - jeod::TimeStandard, 164
  - jeod::TimeUDE, 189
- set\_time\_by\_trunc\_julian
  - jeod::TimeGMST, 111
  - jeod::TimeGPS, 117
  - jeod::TimeStandard, 165
- sim\_start\_format

- jeod::TimeManagerInit, 146
- simtime
  - jeod::TimeManager, 133
- status
  - jeod::TimeManagerInit, 146
- std\_ptr
  - jeod::TimeConverter\_STD\_UDE, 65
- TAI\_to\_TT\_offset
  - jeod::TimeConverter\_TAI\_TDB, 75
- tai\_ptr
  - jeod::TimeConverter\_Dyn\_TAI, 52
  - jeod::TimeConverter\_TAI\_GPS, 69
  - jeod::TimeConverter\_TAI\_TDB, 75
  - jeod::TimeConverter\_TAI\_TT, 80
  - jeod::TimeConverter\_TAI\_UT1, 88
  - jeod::TimeConverter\_TAI\_UTC, 97
- tai\_to\_ut1.cc, 205
  - JEOD\_FRIEND\_CLASS, 206
- tai\_to\_ut1.hh, 206
- tai\_to\_ut1\_override\_val
  - jeod::TimeConverter\_TAI\_UT1, 88
- tai\_to\_utc.cc, 206
  - JEOD\_FRIEND\_CLASS, 206
- tai\_to\_utc.hh, 207
- tdb\_ptr
  - jeod::TimeConverter\_Dyn\_TDB, 56
  - jeod::TimeConverter\_TAI\_TDB, 76
- Time, 15
  - PATH, 17
- time.cc, 207
- time.hh, 207
- time\_\_add\_type\_update.cc, 208
- time\_change\_flag
  - jeod::TimeManager, 133
- time\_converter.cc, 209
- time\_converter.hh, 209
- time\_converter\_dyn\_tai.cc, 210
- time\_converter\_dyn\_tai.hh, 210
- time\_converter\_dyn\_tdb.cc, 211
- time\_converter\_dyn\_tdb.hh, 211
- time\_converter\_dyn\_ude.cc, 212
- time\_converter\_dyn\_ude.hh, 212
- time\_converter\_std\_ude.cc, 213
- time\_converter\_std\_ude.hh, 213
- time\_converter\_tai\_gps.cc, 214
- time\_converter\_tai\_gps.hh, 214
- time\_converter\_tai\_tdb.cc, 215
- time\_converter\_tai\_tdb.hh, 215
- time\_converter\_tai\_tt.cc, 216
- time\_converter\_tai\_tt.hh, 216
- time\_converter\_tai\_ut1.cc, 217
- time\_converter\_tai\_ut1.hh, 217
- time\_converter\_tai\_utc.cc, 218
- time\_converter\_tai\_utc.hh, 218
- time\_converter\_ut1\_gmst.cc, 219
- time\_converter\_ut1\_gmst.hh, 219
- time\_dyn.cc, 220
- time\_dyn.hh, 220
- time\_enum.hh, 221
- time\_gmst.cc, 221
- time\_gmst.hh, 222
- time\_gps.cc, 222
- time\_gps.hh, 223
- time\_links.hh, 223
- time\_lookup
  - jeod::TimeManager, 129
- time\_manager
  - jeod::JeodBaseTime, 37
  - jeod::TimeManagerInit, 146
- time\_manager.cc, 224
- time\_manager.hh, 224
- time\_manager\_\_initialize.cc, 225
- time\_manager\_init.cc, 225
- time\_manager\_init.hh, 226
- time\_messages.cc, 226
- time\_messages.hh, 227
- time\_met.cc, 227
- time\_met.hh, 228
- time\_standard.cc, 228
- time\_standard.hh, 229
- time\_standards\_exist
  - jeod::TimeManager, 129
- time\_tai.cc, 229
- time\_tai.hh, 230
- time\_tdb.cc, 230
- time\_tdb.hh, 231
- time\_tt.cc, 231
- time\_tt.hh, 232
- time\_ude.cc, 232
- time\_ude.hh, 233
- time\_ut1.cc, 233
- time\_ut1.hh, 234
- time\_utc.cc, 234
- time\_utc.hh, 235
- time\_vector
  - jeod::TimeManager, 133
- TimeConverter
  - jeod::JeodBaseTime, 33
  - jeod::TimeConverter, 41, 42
- TimeConverter\_Dyn\_TAI
  - jeod::TimeConverter\_Dyn\_TAI, 49, 50
- TimeConverter\_Dyn\_TDB
  - jeod::TimeConverter\_Dyn\_TDB, 53, 54
- TimeConverter\_Dyn\_UDE
  - jeod::TimeConverter\_Dyn\_UDE, 57, 58
- TimeConverter\_STD\_UDE
  - jeod::TimeConverter\_STD\_UDE, 62
- TimeConverter\_TAI\_GPS
  - jeod::TimeConverter\_TAI\_GPS, 67
- TimeConverter\_TAI\_TDB
  - jeod::TimeConverter\_TAI\_TDB, 71, 72
- TimeConverter\_TAI\_TT
  - jeod::TimeConverter\_TAI\_TT, 77, 78
- TimeConverter\_TAI\_UT1
  - jeod::TimeConverter\_TAI\_UT1, 82, 83
- TimeConverter\_TAI\_UTC

- jeod::TimeConverter\_TAI\_UTC, 92
- TimeConverter\_UT1\_GMST
  - jeod::TimeConverter\_UT1\_GMST, 100
- TimeDyn
  - jeod::TimeDyn, 104, 105
- TimeFormat
  - jeod::TimeEnum, 108
- TimeGMST
  - jeod::TimeGMST, 110
- TimeGPS
  - jeod::TimeGPS, 114
- TimeLinks
  - jeod::TimeLinks, 120, 121
- TimeMET
  - jeod::TimeMET, 154, 155
- TimeManager
  - jeod::TimeManager, 124
- TimeManagerInit
  - jeod::JeodBaseTime, 33
  - jeod::TimeManager, 132
  - jeod::TimeManagerInit, 136
- TimeMessages
  - jeod::TimeMessages, 148
- TimeStandard
  - jeod::TimeStandard, 159
- TimeTAI
  - jeod::TimeTAI, 171, 172
- TimeTDB
  - jeod::TimeTDB, 174
- TimeTT
  - jeod::TimeTT, 176, 177
- TimeUDE
  - jeod::TimeStandard, 166
  - jeod::TimeUDE, 181
- TimeUT1
  - jeod::TimeUT1, 198
- TimeUTC
  - jeod::TimeUTC, 201
- tjt\_at\_epoch
  - jeod::TimeStandard, 169
- tjt\_jd\_offset
  - jeod::TimeStandard, 169
- tjt\_mjt\_offset
  - jeod::TimeStandard, 169
- true\_ut1
  - jeod::TimeUT1, 199
- true\_utc
  - jeod::TimeUTC, 203
- trunc\_julian\_time
  - jeod::TimeStandard, 169
- tt\_ptr
  - jeod::TimeConverter\_TAI\_TT, 80
- ude\_ptr
  - jeod::TimeConverter\_Dyn\_UDE, 60
  - jeod::TimeConverter\_STD\_UDE, 65
- update
  - jeod::JeodBaseTime, 32
  - jeod::TimeDyn, 106
  - jeod::TimeMET, 155
  - jeod::TimeManager, 130
- update\_converter\_dir\_table
  - jeod::TimeManagerInit, 147
- update\_converter\_direction
  - jeod::JeodBaseTime, 37
- update\_converter\_ptr
  - jeod::JeodBaseTime, 38
- update\_from\_name
  - jeod::JeodBaseTime, 38
- update\_index
  - jeod::TimeUDE, 196
- update\_offset
  - jeod::TimeDyn, 106
- update\_time
  - jeod::TimeManager, 130
- ut1\_ptr
  - jeod::TimeConverter\_TAI\_UT1, 89
  - jeod::TimeConverter\_UT1\_GMST, 102
- utc\_ptr
  - jeod::TimeConverter\_TAI\_UTC, 97
- val\_vec
  - jeod::TimeConverter\_TAI\_UT1, 89
  - jeod::TimeConverter\_TAI\_UTC, 97
- valid\_directions
  - jeod::TimeConverter, 48
- verify\_converter\_setup
  - jeod::TimeManagerInit, 143
- verify\_epoch
  - jeod::TimeUDE, 189
- verify\_init
  - jeod::TimeUDE, 189
- verify\_setup
  - jeod::TimeConverter, 45
- verify\_table\_lookup\_ends
  - jeod::TimeConverter, 45
  - jeod::TimeConverter\_TAI\_UT1, 85
  - jeod::TimeConverter\_TAI\_UTC, 94
  - jeod::TimeManager, 131
- verify\_times\_setup
  - jeod::TimeManagerInit, 143
- verify\_update
  - jeod::TimeUDE, 190
- week
  - jeod::TimeGPS, 119
- week\_13\_bit
  - jeod::TimeGPS, 119
- when\_vec
  - jeod::TimeConverter\_TAI\_UT1, 89
  - jeod::TimeConverter\_TAI\_UTC, 98
- year\_of\_last\_soy
  - jeod::TimeStandard, 170