

Rotation,Nutation,andPrecessionModel

5.0

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	11
6.1	Models	11
6.1.1	Detailed Description	11
6.2	Environment	12
6.2.1	Detailed Description	12
6.3	RNP	13
6.3.1	Detailed Description	13
6.4	GenericRNP	14
6.4.1	Detailed Description	14
6.4.2	Macro Definition Documentation	14
6.4.2.1	PATH	14
6.5	RNPJ2000	15
6.5.1	Detailed Description	15
6.6	RNPMars	16
6.6.1	Detailed Description	16

7 Namespace Documentation	17
7.1 jeod Namespace Reference	17
7.1.1 Detailed Description	18
8 Data Structure Documentation	19
8.1 jeod::NutationJ2000 Class Reference	19
8.1.1 Detailed Description	20
8.1.2 Constructor & Destructor Documentation	21
8.1.2.1 NutationJ2000() [1/2]	21
8.1.2.2 ~NutationJ2000()	21
8.1.2.3 NutationJ2000() [2/2]	21
8.1.3 Member Function Documentation	21
8.1.3.1 initialize()	21
8.1.3.2 operator=()	22
8.1.3.3 update_rotation()	22
8.1.4 Friends And Related Function Documentation	22
8.1.4.1 init_attrjeod__NutationJ2000	22
8.1.4.2 InputProcessor	23
8.1.5 Field Documentation	23
8.1.5.1 D	23
8.1.5.2 D_coeffs	23
8.1.5.3 epsilon_bar	23
8.1.5.4 equa_of_equi	24
8.1.5.5 F	24
8.1.5.6 F_coeffs	24
8.1.5.7 L	24
8.1.5.8 L_coeffs	25
8.1.5.9 long_coeffs	25
8.1.5.10 long_t_coeffs	25
8.1.5.11 M	25
8.1.5.12 M_coeffs	26

8.1.5.13	num_coeffs	26
8.1.5.14	nutaton_in_longitude	26
8.1.5.15	nutaton_in_obliquity	26
8.1.5.16	obliq_coeffs	27
8.1.5.17	obliq_t_coeffs	27
8.1.5.18	omega	27
8.1.5.19	omega_coeffs	27
8.2	jeod::NutationJ2000Init Class Reference	28
8.2.1	Detailed Description	29
8.2.2	Constructor & Destructor Documentation	29
8.2.2.1	NutationJ2000Init() [1/2]	29
8.2.2.2	~NutationJ2000Init()	29
8.2.2.3	NutationJ2000Init() [2/2]	29
8.2.3	Member Function Documentation	30
8.2.3.1	operator=()	30
8.2.4	Friends And Related Function Documentation	30
8.2.4.1	init_attrjeod__NutationJ2000Init	30
8.2.4.2	InputProcessor	30
8.2.5	Field Documentation	30
8.2.5.1	D_coeffs	30
8.2.5.2	F_coeffs	31
8.2.5.3	L_coeffs	31
8.2.5.4	long_coeffs	31
8.2.5.5	long_t_coeffs	31
8.2.5.6	M_coeffs	32
8.2.5.7	num_coeffs	32
8.2.5.8	obliq_coeffs	32
8.2.5.9	obliq_t_coeffs	32
8.2.5.10	omega_coeffs	33
8.3	jeod::NutationJ2000Init_nutation_j2000_default_data Class Reference	33

8.3.1	Detailed Description	33
8.3.2	Member Function Documentation	33
8.3.2.1	initialize()	33
8.4	jeod::NutationMars Class Reference	34
8.4.1	Detailed Description	35
8.4.2	Constructor & Destructor Documentation	35
8.4.2.1	NutationMars() [1/2]	35
8.4.2.2	~NutationMars()	35
8.4.2.3	NutationMars() [2/2]	36
8.4.3	Member Function Documentation	36
8.4.3.1	operator=()	36
8.4.3.2	update_rotation()	36
8.4.4	Friends And Related Function Documentation	36
8.4.4.1	init_attrjeod__NutationMars	36
8.4.4.2	InputProcessor	36
8.4.5	Field Documentation	37
8.4.5.1	I_at_j2000	37
8.4.5.2	I_dot	37
8.4.5.3	I_m_orig	37
8.4.5.4	int_to_double	38
8.4.5.5	mean_anomaly_j2000	38
8.4.5.6	mean_motion	38
8.4.5.7	nutaton_in_longitude	38
8.4.5.8	nutaton_in_obliquity	39
8.4.5.9	obliquity_angle	39
8.4.5.10	psi_m_orig	39
8.4.5.11	q_angle_j2000	39
8.5	jeod::PlanetOrientation Class Reference	40
8.5.1	Detailed Description	41
8.5.2	Constructor & Destructor Documentation	41

8.5.2.1	PlanetOrientation() [1/2]	41
8.5.2.2	~PlanetOrientation()	42
8.5.2.3	PlanetOrientation() [2/2]	42
8.5.3	Member Function Documentation	42
8.5.3.1	activate()	42
8.5.3.2	deactivate()	42
8.5.3.3	ephem_activate()	42
8.5.3.4	ephem_build_tree()	43
8.5.3.5	ephem_initialize()	43
8.5.3.6	get_name()	43
8.5.3.7	initialize()	44
8.5.3.8	operator=()	44
8.5.3.9	set_name()	44
8.5.4	Friends And Related Function Documentation	45
8.5.4.1	init_attrjeod__PlanetOrientation	45
8.5.4.2	InputProcessor	45
8.5.5	Field Documentation	45
8.5.5.1	active	45
8.5.5.2	name	45
8.5.5.3	orient_interface	46
8.5.5.4	planet	46
8.5.5.5	planet_omega	46
8.5.5.6	planet_rot_state	46
8.6	jeod::PlanetRNP Class Reference	47
8.6.1	Detailed Description	48
8.6.2	Member Enumeration Documentation	48
8.6.2.1	RNPFidelity	48
8.6.3	Constructor & Destructor Documentation	49
8.6.3.1	PlanetRNP() [1/2]	49
8.6.3.2	~PlanetRNP()	49

8.6.3.3	PlanetRNP() [2/2]	49
8.6.4	Member Function Documentation	49
8.6.4.1	get_name()	50
8.6.4.2	operator=()	50
8.6.4.3	propagate_rnp()	50
8.6.4.4	update_axial_rotation()	50
8.6.4.5	update_rnp()	51
8.6.5	Friends And Related Function Documentation	51
8.6.5.1	init_attrjeod__PlanetRNP	51
8.6.5.2	InputProcessor	51
8.6.6	Field Documentation	51
8.6.6.1	enable_polar	51
8.6.6.2	NP_matrix	52
8.6.6.3	nutaton	52
8.6.6.4	polar_motion	52
8.6.6.5	precession	52
8.6.6.6	rnp_type	53
8.6.6.7	rotation	53
8.6.6.8	scratch_matrix	53
8.7	jeod::PlanetRotation Class Reference	54
8.7.1	Detailed Description	55
8.7.2	Constructor & Destructor Documentation	55
8.7.2.1	PlanetRotation() [1/2]	55
8.7.2.2	~PlanetRotation()	55
8.7.2.3	PlanetRotation() [2/2]	55
8.7.3	Member Function Documentation	55
8.7.3.1	get_rotation()	55
8.7.3.2	get_rotation_transpose()	56
8.7.3.3	initialize()	56
8.7.3.4	operator=()	56

8.7.3.5	update_rotation()	57
8.7.3.6	update_time()	57
8.7.4	Friends And Related Function Documentation	57
8.7.4.1	init_attrjeod__PlanetRotation	57
8.7.4.2	InputProcessor	57
8.7.5	Field Documentation	58
8.7.5.1	current_time	58
8.7.5.2	DAYTOJULIANCENT	58
8.7.5.3	DEGTORAD	58
8.7.5.4	DEGTOSEC	59
8.7.5.5	JULIANCENTTODAY	59
8.7.5.6	JULIANDAYTOSEC	59
8.7.5.7	RADTODEG	59
8.7.5.8	rotation	60
8.7.5.9	SECTODEG	60
8.7.5.10	SECTOJULIANDAY	60
8.8	jeod::PlanetRotationInit Class Reference	61
8.8.1	Detailed Description	61
8.8.2	Constructor & Destructor Documentation	61
8.8.2.1	PlanetRotationInit() [1/2]	61
8.8.2.2	~PlanetRotationInit()	62
8.8.2.3	PlanetRotationInit() [2/2]	62
8.8.3	Member Function Documentation	62
8.8.3.1	operator=()	62
8.8.4	Friends And Related Function Documentation	62
8.8.4.1	init_attrjeod__PlanetRotationInit	62
8.8.4.2	InputProcessor	62
8.9	jeod::PolarMotionJ2000 Class Reference	63
8.9.1	Detailed Description	64
8.9.2	Constructor & Destructor Documentation	64

8.9.2.1	PolarMotionJ2000() [1/2]	64
8.9.2.2	~PolarMotionJ2000()	64
8.9.2.3	PolarMotionJ2000() [2/2]	64
8.9.3	Member Function Documentation	64
8.9.3.1	initialize()	64
8.9.3.2	operator=()	65
8.9.3.3	update_rotation()	65
8.9.4	Friends And Related Function Documentation	65
8.9.4.1	init_attrjeod__PolarMotionJ2000	65
8.9.4.2	InputProcessor	66
8.9.5	Field Documentation	66
8.9.5.1	last_table_index	66
8.9.5.2	override_table	66
8.9.5.3	polar_mjd	66
8.9.5.4	warn_table	67
8.9.5.5	xp	67
8.9.5.6	xp_tbl	67
8.9.5.7	yp	67
8.9.5.8	yp_tbl	68
8.10	jeod::PolarMotionJ2000Init Class Reference	68
8.10.1	Detailed Description	69
8.10.2	Constructor & Destructor Documentation	69
8.10.2.1	PolarMotionJ2000Init() [1/2]	69
8.10.2.2	~PolarMotionJ2000Init()	69
8.10.2.3	PolarMotionJ2000Init() [2/2]	70
8.10.3	Member Function Documentation	70
8.10.3.1	operator=()	70
8.10.4	Friends And Related Function Documentation	70
8.10.4.1	init_attrjeod__PolarMotionJ2000Init	70
8.10.4.2	InputProcessor	70

8.10.5	Field Documentation	70
8.10.5.1	last_table_index	70
8.10.5.2	override_table	71
8.10.5.3	polar_mjd	71
8.10.5.4	xp	71
8.10.5.5	xp_tbl	71
8.10.5.6	yp	72
8.10.5.7	yp_tbl	72
8.11	jeod::PolarMotionJ2000Init_xpyp_daily_default_data Class Reference	72
8.11.1	Detailed Description	72
8.11.2	Member Function Documentation	72
8.11.2.1	initialize()	73
8.12	jeod::PolarMotionJ2000Init_xpyp_monthly_default_data Class Reference	73
8.12.1	Detailed Description	73
8.12.2	Member Function Documentation	73
8.12.2.1	initialize()	73
8.13	jeod::PrecessionJ2000 Class Reference	74
8.13.1	Detailed Description	74
8.13.2	Constructor & Destructor Documentation	74
8.13.2.1	PrecessionJ2000() [1/2]	75
8.13.2.2	~PrecessionJ2000()	75
8.13.2.3	PrecessionJ2000() [2/2]	75
8.13.3	Member Function Documentation	75
8.13.3.1	operator=()	75
8.13.3.2	update_rotation()	75
8.13.4	Friends And Related Function Documentation	76
8.13.4.1	init_attrjeod__PrecessionJ2000	76
8.13.4.2	InputProcessor	76
8.14	jeod::PrecessionMars Class Reference	76
8.14.1	Detailed Description	77

8.14.2	Constructor & Destructor Documentation	77
8.14.2.1	PrecessionMars() [1/2]	78
8.14.2.2	~PrecessionMars()	78
8.14.2.3	PrecessionMars() [2/2]	78
8.14.3	Member Function Documentation	78
8.14.3.1	compute_fixed_matrices()	78
8.14.3.2	operator=()	79
8.14.3.3	update_rotation()	79
8.14.4	Friends And Related Function Documentation	79
8.14.4.1	init_attrjeod__PrecessionMars	79
8.14.4.2	InputProcessor	79
8.14.5	Field Documentation	79
8.14.5.1	J	80
8.14.5.2	N	80
8.14.5.3	NJ_matrix	80
8.14.5.4	nutaton	80
8.14.5.5	psi_at_j2000	81
8.14.5.6	psi_dot	81
8.14.5.7	psi_precess	81
8.15	jeod::RNPJ2000 Class Reference	82
8.15.1	Detailed Description	83
8.15.2	Constructor & Destructor Documentation	83
8.15.2.1	RNPJ2000() [1/2]	83
8.15.2.2	~RNPJ2000()	84
8.15.2.3	RNPJ2000() [2/2]	84
8.15.3	Member Function Documentation	84
8.15.3.1	ephem_update()	84
8.15.3.2	get_dyn_time_ptr()	84
8.15.3.3	get_name()	85
8.15.3.4	initialize()	85

8.15.3.5	operator=()	85
8.15.3.6	timestamp()	86
8.15.3.7	update_axial_rotation()	86
8.15.3.8	update_rnp()	86
8.15.4	Friends And Related Function Documentation	87
8.15.4.1	init_attrjeod__RNPJ2000	87
8.15.4.2	InputProcessor	87
8.15.5	Field Documentation	87
8.15.5.1	gmst_ptr	87
8.15.5.2	internal_name	87
8.15.5.3	last_updated_time_full	88
8.15.5.4	last_updated_time_rotational	88
8.15.5.5	never_updated_full	88
8.15.5.6	never_updated_rotational	88
8.15.5.7	NJ2000	89
8.15.5.8	PJ2000	89
8.15.5.9	PMJ2000	89
8.15.5.10	RJ2000	89
8.15.5.11	time_dyn_ptr	90
8.16	jeod::RNPJ2000_rnp_j2000_default_data Class Reference	90
8.16.1	Detailed Description	90
8.16.2	Member Function Documentation	90
8.16.2.1	initialize()	90
8.17	jeod::RNPMars Class Reference	91
8.17.1	Detailed Description	92
8.17.2	Constructor & Destructor Documentation	92
8.17.2.1	RNPMars() [1/2]	92
8.17.2.2	~RNPMars()	93
8.17.2.3	RNPMars() [2/2]	93
8.17.3	Member Function Documentation	93

8.17.3.1	ephem_update()	93
8.17.3.2	get_dyn_time_ptr()	93
8.17.3.3	get_name()	94
8.17.3.4	initialize()	94
8.17.3.5	operator=()	94
8.17.3.6	timestamp()	95
8.17.3.7	update_axial_rotation()	95
8.17.3.8	update_rnp()	95
8.17.4	Friends And Related Function Documentation	96
8.17.4.1	init_attrjeod__RNPMars	96
8.17.4.2	InputProcessor	96
8.17.5	Field Documentation	96
8.17.5.1	internal_name	96
8.17.5.2	last_updated_time_full	96
8.17.5.3	last_updated_time_rotational	97
8.17.5.4	never_updated_full	97
8.17.5.5	never_updated_rotational	97
8.17.5.6	NMars	97
8.17.5.7	PMars	98
8.17.5.8	RMars	98
8.17.5.9	time_dyn_ptr	98
8.17.5.10	tt_ptr	98
8.18	jeod::RNPMars_rnp_mars_default_data Class Reference	99
8.18.1	Detailed Description	99
8.18.2	Member Function Documentation	99
8.18.2.1	initialize()	99
8.19	jeod::RNPMessages Class Reference	99
8.19.1	Detailed Description	100
8.19.2	Constructor & Destructor Documentation	100
8.19.2.1	RNPMessages() [1/2]	100

8.19.2.2	RNPMessages() [2/2]	100
8.19.3	Member Function Documentation	101
8.19.3.1	operator=()	101
8.19.4	Friends And Related Function Documentation	101
8.19.4.1	init_attrjeod__RNPMessages	101
8.19.4.2	InputProcessor	101
8.19.5	Field Documentation	101
8.19.5.1	fidelity_error	101
8.19.5.2	initialization_error	102
8.19.5.3	polar_motion_table_warning	102
8.19.5.4	setup_error	102
8.20	jeod::RotationJ2000 Class Reference	103
8.20.1	Detailed Description	104
8.20.2	Constructor & Destructor Documentation	104
8.20.2.1	RotationJ2000() [1/2]	104
8.20.2.2	~RotationJ2000()	104
8.20.2.3	RotationJ2000() [2/2]	104
8.20.3	Member Function Documentation	104
8.20.3.1	operator=()	105
8.20.3.2	update_rotation()	105
8.20.4	Friends And Related Function Documentation	105
8.20.4.1	init_attrjeod__RotationJ2000	105
8.20.4.2	InputProcessor	105
8.20.5	Field Documentation	105
8.20.5.1	GMST	106
8.20.5.2	nutation	106
8.20.5.3	planet_rotational_velocity	106
8.20.5.4	theta_gast	106
8.20.5.5	use_full_rnp	107
8.21	jeod::RotationMars Class Reference	107

8.21.1	Detailed Description	108
8.21.2	Constructor & Destructor Documentation	108
8.21.2.1	RotationMars() [1/2]	108
8.21.2.2	~RotationMars()	108
8.21.2.3	RotationMars() [2/2]	108
8.21.3	Member Function Documentation	109
8.21.3.1	operator=()	109
8.21.3.2	update_rotation()	109
8.21.4	Friends And Related Function Documentation	109
8.21.4.1	init_attrjeod__RotationMars	109
8.21.4.2	InputProcessor	109
8.21.5	Field Documentation	109
8.21.5.1	nutaton	110
8.21.5.2	phi_at_j2000	110
8.21.5.3	phi_spin	110
8.21.5.4	planet_rotational_velocity	110
8.21.5.5	use_full_rnp	111
9	File Documentation	113
9.1	class_declarations.hh File Reference	113
9.1.1	Detailed Description	113
9.2	class_declarations.hh File Reference	113
9.2.1	Detailed Description	113
9.3	data_nutation_j2000.cc File Reference	114
9.3.1	Macro Definition Documentation	114
9.3.1.1	JEOD_FRIEND_CLASS	114
9.4	data_rnp_j2000.cc File Reference	114
9.4.1	Macro Definition Documentation	115
9.4.1.1	JEOD_FRIEND_CLASS	115
9.5	data_rnp_mars.cc File Reference	115
9.5.1	Macro Definition Documentation	115

9.5.1.1	JEOD_FRIEND_CLASS	115
9.6	nutaton_j2000.cc File Reference	116
9.6.1	Detailed Description	116
9.7	nutaton_j2000.hh File Reference	116
9.8	nutaton_j2000.hh File Reference	116
9.8.1	Detailed Description	117
9.9	nutaton_j2000_init.cc File Reference	117
9.9.1	Detailed Description	117
9.10	nutaton_j2000_init.hh File Reference	117
9.10.1	Detailed Description	118
9.11	nutaton_mars.cc File Reference	118
9.11.1	Detailed Description	118
9.12	nutaton_mars.hh File Reference	118
9.12.1	Detailed Description	119
9.13	planet_orientation.cc File Reference	119
9.13.1	Detailed Description	119
9.14	planet_orientation.hh File Reference	119
9.14.1	Detailed Description	120
9.15	planet_rnp.cc File Reference	120
9.15.1	Detailed Description	120
9.16	planet_rnp.hh File Reference	120
9.16.1	Detailed Description	121
9.17	planet_rotation.cc File Reference	121
9.17.1	Detailed Description	121
9.18	planet_rotation.hh File Reference	121
9.18.1	Detailed Description	122
9.19	planet_rotation_init.cc File Reference	122
9.19.1	Detailed Description	122
9.20	planet_rotation_init.hh File Reference	122
9.20.1	Detailed Description	123

9.21	polar_motion_j2000.cc File Reference	123
9.21.1	Detailed Description	123
9.22	polar_motion_j2000.hh File Reference	123
9.22.1	Detailed Description	124
9.23	polar_motion_j2000_init.cc File Reference	124
9.23.1	Detailed Description	124
9.24	polar_motion_j2000_init.hh File Reference	124
9.24.1	Detailed Description	124
9.25	precession_j2000.cc File Reference	125
9.25.1	Detailed Description	125
9.26	precession_j2000.hh File Reference	125
9.26.1	Detailed Description	125
9.27	precession_mars.cc File Reference	126
9.27.1	Detailed Description	126
9.28	precession_mars.hh File Reference	126
9.28.1	Detailed Description	126
9.29	rnp_j2000.cc File Reference	127
9.29.1	Detailed Description	127
9.30	rnp_j2000.hh File Reference	127
9.31	rnp_j2000.hh File Reference	127
9.31.1	Detailed Description	128
9.32	rnp_mars.cc File Reference	128
9.32.1	Detailed Description	128
9.33	rnp_mars.hh File Reference	128
9.34	rnp_mars.hh File Reference	129
9.34.1	Detailed Description	129
9.35	RNP_messages.cc File Reference	129
9.35.1	Detailed Description	130
9.36	RNP_messages.hh File Reference	130
9.36.1	Detailed Description	130

9.37	rotation_j2000.cc File Reference	130
9.37.1	Detailed Description	131
9.38	rotation_j2000.hh File Reference	131
9.38.1	Detailed Description	131
9.39	rotation_mars.cc File Reference	131
9.39.1	Detailed Description	132
9.40	rotation_mars.hh File Reference	132
9.40.1	Detailed Description	132
9.41	xyp_daily.cc File Reference	132
9.41.1	Macro Definition Documentation	133
9.41.1.1	JEOD_FRIEND_CLASS	133
9.42	xyp_daily.hh File Reference	133
9.43	xyp_monthly.cc File Reference	133
9.43.1	Macro Definition Documentation	133
9.43.1.1	JEOD_FRIEND_CLASS	134
9.44	xyp_monthly.hh File Reference	134
Index		135

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	11
Environment	12
RNP	13
GenericRNP	14
RNPJ2000	15
RNPMars	16

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	17
----------------------	--------------------------	--------------------

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

EphemerisInterface	
jeod::PlanetOrientation	40
jeod::PlanetRNP	47
jeod::RNPJ2000	82
jeod::RNPMars	91
jeod::NutationJ2000Init_nutation_j2000_default_data	33
jeod::PlanetRotation	54
jeod::NutationJ2000	19
jeod::NutationMars	34
jeod::PolarMotionJ2000	63
jeod::PrecessionJ2000	74
jeod::PrecessionMars	76
jeod::RotationJ2000	103
jeod::RotationMars	107
jeod::PlanetRotationInit	61
jeod::NutationJ2000Init	28
jeod::PolarMotionJ2000Init	68
jeod::PolarMotionJ2000Init_xpyp_daily_default_data	72
jeod::PolarMotionJ2000Init_xpyp_monthly_default_data	73
RefFrameOwner	
jeod::PlanetOrientation	40
jeod::RNPJ2000_rnp_j2000_default_data	90
jeod::RNPMars_rnp_mars_default_data	99
jeod::RNPMessages	99

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::NutationJ2000	Implements the nutation portion of the J2000 RNP	19
jeod::NutationJ2000Init	The NutationJ2000Init contains coefficients and other data used to initialize a NutationJ2000 object	28
jeod::NutationJ2000Init_nutation_j2000_default_data		33
jeod::NutationMars	Implements the nutation portion of the "Pathfinder" Mars RNP model	34
jeod::PlanetOrientation	The generic framework for orientation models to interact with a DynManager object	40
jeod::PlanetRNP	The generic framework for orientation models based on the RNP paradigm	47
jeod::PlanetRotation	The generic base class for all planet transformations	54
jeod::PlanetRotationInit	The generic base class for all initializing classes for PlanetRotation derived classes	61
jeod::PolarMotionJ2000	Implements the polar motion portion of the J2000 RNP	63
jeod::PolarMotionJ2000Init	Initializes the PolarMotionJ2000 object	68
jeod::PolarMotionJ2000Init_xpyp_daily_default_data		72
jeod::PolarMotionJ2000Init_xpyp_monthly_default_data		73
jeod::PrecessionJ2000	Implements the precession portion of the J2000 RNP	74
jeod::PrecessionMars	Implements the axial rotation portion of the "Pathfinder" Mars RNP model	76
jeod::RNPJ2000	Implements the J2000 RNP model using the generic RNP framework	82
jeod::RNPJ2000_rnp_j2000_default_data		90
jeod::RNPMars	Implements the "Pathfinder" Mars RNP model using the generic RNP framework	91
jeod::RNPMars_rnp_mars_default_data		99
jeod::RNPMessages	Describes messages used in the RNP model	99
jeod::RotationJ2000	Implements the axial rotation portion of J2000 RNP	103
jeod::RotationMars	Implements the axial rotation portion of the "Pathfinder" Mars RNP model	107

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

GenericRNP/include/class_declarations.hh	
Forward declarations of classes defined for JEOD 2.0 Generic RNP	113
RNPJ2000/include/class_declarations.hh	
Forward declarations of classes defined for JEOD 2.0 J2000 RNP	113
data_nutation_j2000.cc	114
data_rnp_j2000.cc	114
data_rnp_mars.cc	115
nutation_j2000.cc	
Implementation for the NutationJ2000 class	116
data/include/nutation_j2000.hh	116
include/nutation_j2000.hh	
Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52	116
nutation_j2000_init.cc	
Implementation of NutationJ2000Init	117
nutation_j2000_init.hh	
Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize	117
nutation_mars.cc	
Implementation for the NutationMars class	118
nutation_mars.hh	
Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model	118
planet_orientation.cc	
Implement PlanetOrientation	119
planet_orientation.hh	
Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame	119
planet_rnp.cc	
Implement PlanetRNP	120
planet_rnp.hh	
Establish a framework for rotation-nutation-precession style planet attitude models	120
planet_rotation.cc	
Implemenation for PlanetRotation	121
planet_rotation.hh	
Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation	121

planet_rotation_init.cc	
Implementation for PlanetRotationInit	122
planet_rotation_init.hh	
Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation	122
polar_motion_j2000.cc	
Implementation of PolarMotionJ2000	123
polar_motion_j2000.hh	
Model the polar motion portion of the RNP routine for the Standard Epoch J2000	123
polar_motion_j2000_init.cc	
Implementation for PolarMotionJ2000Init	124
polar_motion_j2000_init.hh	
Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize	124
precession_j2000.cc	
Implementation for PrecessionJ2000	125
precession_j2000.hh	
Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52	125
precession_mars.cc	
Implementation of PrecessionMars	126
precession_mars.hh	
Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model	126
rnp_j2000.cc	
Implementation for RNPJ2000	127
data/include/rnp_j2000.hh	127
include/rnp_j2000.hh	
A specific implementation of PlanetRNP, for Earth	127
rnp_mars.cc	
Implementation of RNPMars	128
data/include/rnp_mars.hh	128
include/rnp_mars.hh	
A specific implementation of PlanetRNP, for Mars	129
RNP_messages.cc	
Implement RNP_messages	129
RNP_messages.hh	
Implement RNP_messages	130
rotation_j2000.cc	
Implementation of RotationJ2000	130
rotation_j2000.hh	
Model the axial rotation portion of the RNP routine for the Standard Epoch J2000	131
rotation_mars.cc	
Implementation of RotationMars	131
rotation_mars.hh	
Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model	132
xpyp_daily.cc	132
xpyp_daily.hh	133
xpyp_monthly.cc	133
xpyp_monthly.hh	134

Chapter 6

Module Documentation

6.1 Models

Modules

- [Environment](#)

6.1.1 Detailed Description

6.2 Environment

Modules

- [RNP](#)

6.2.1 Detailed Description

6.3 RNP

Modules

- [GenericRNP](#)
- [RNPJ2000](#)
- [RNPMars](#)

6.3.1 Detailed Description

6.4 GenericRNP

Files

- file [GenericRNP/include/class_declarations.hh](#)
Forward declarations of classes defined for JEOD 2.0 Generic RNP.
- file [planet_orientation.hh](#)
Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.
- file [planet_rnp.hh](#)
Establish a framework for rotation-nutation-precession style planet attitude models.
- file [planet_rotation.hh](#)
Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation.
- file [planet_rotation_init.hh](#)
Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.
- file [RNP_messages.hh](#)
Implement RNP_messages.
- file [planet_orientation.cc](#)
Implement PlanetOrientation.
- file [planet_rnp.cc](#)
Implement PlanetRNP.
- file [planet_rotation.cc](#)
Implementation for PlanetRotation.
- file [planet_rotation_init.cc](#)
Implementation for PlanetRotationInit.
- file [RNP_messages.cc](#)
Implement RNP_messages.

Namespaces

- [jeod](#)
Namespace jeod.

Macros

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

6.4.1 Detailed Description

6.4.2 Macro Definition Documentation

6.4.2.1 PATH

```
#define PATH "environment/RNP/"
```

Definition at line 38 of file RNP_messages.cc.

6.5 RNPJ2000

Files

- file [RNPJ2000/include/class_declarations.hh](#)
Forward declarations of classes defined for JEOD 2.0 J2000 RNP.
- file [include/nutation_j2000.hh](#)
Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.
- file [nutation_j2000_init.hh](#)
Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.
- file [polar_motion_j2000.hh](#)
Model the polar motion portion of the RNP routine for the Standard Epoch J2000.
- file [polar_motion_j2000_init.hh](#)
Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.
- file [precession_j2000.hh](#)
Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.
- file [include/rnp_j2000.hh](#)
A specific implementation of PlanetRNP, for Earth.
- file [rotation_j2000.hh](#)
Model the axial rotation portion of the RNP routine for the Standard Epoch J2000.
- file [nutation_j2000.cc](#)
Implementation for the NutationJ2000 class.
- file [nutation_j2000_init.cc](#)
Implementation of NutationJ2000Init.
- file [polar_motion_j2000.cc](#)
Implementation of PolarMotionJ2000.
- file [polar_motion_j2000_init.cc](#)
Implementation for PolarMotionJ2000Init.
- file [precession_j2000.cc](#)
Implementation for PrecessionJ2000.
- file [rnp_j2000.cc](#)
Implementation for RNPJ2000.
- file [rotation_j2000.cc](#)
Implementation of RotationJ2000.

Namespaces

- [jeod](#)
Namespace jeod.

6.5.1 Detailed Description

6.6 RNPMars

Files

- file [nutation_mars.hh](#)
Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.
- file [precession_mars.hh](#)
Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.
- file [include/rnp_mars.hh](#)
A specific implementation of PlanetRNP, for Mars.
- file [rotation_mars.hh](#)
Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.
- file [nutation_mars.cc](#)
Implementation for the NutationMars class.
- file [precession_mars.cc](#)
Implementation of PrecessionMars.
- file [rnp_mars.cc](#)
Implementation of RNPMars.
- file [rotation_mars.cc](#)
Implementation of RotationMars.

Namespaces

- [jeod](#)
Namespace jeod.

6.6.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class [NutationJ2000](#)
Implements the nutation portion of the J2000 RNP.
- class [NutationJ2000Init](#)
The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.
- class [NutationJ2000Init_nutation_j2000_default_data](#)
- class [NutationMars](#)
Implements the nutation portion of the "Pathfinder" Mars RNP model.
- class [PlanetOrientation](#)
The generic framework for orientation models to interact with a [DynManager](#) object.
- class [PlanetRNP](#)
The generic framework for orientation models based on the RNP paradigm.
- class [PlanetRotation](#)
The generic base class for all planet transformations.
- class [PlanetRotationInit](#)
The generic base class for all initializing classes for [PlanetRotation](#) derived classes.
- class [PolarMotionJ2000](#)
Implements the polar motion portion of the J2000 RNP.
- class [PolarMotionJ2000Init](#)
Initializes the [PolarMotionJ2000](#) object.
- class [PolarMotionJ2000Init_xpyp_daily_default_data](#)
- class [PolarMotionJ2000Init_xpyp_monthly_default_data](#)
- class [PrecessionJ2000](#)
Implements the precession portion of the J2000 RNP.
- class [PrecessionMars](#)
Implements the axial rotation portion of the "Pathfinder" Mars RNP model.
- class [RNPJ2000](#)
Implements the J2000 RNP model using the generic RNP framework.
- class [RNPJ2000_rnp_j2000_default_data](#)

- class [RNPMars](#)
Implements the "Pathfinder" Mars RNP model using the generic RNP framework.
- class [RNPMars_rnp_mars_default_data](#)
- class [RNPMessages](#)
Describes messages used in the RNP model.
- class [RotationJ2000](#)
Implements the axial rotation portion of J2000 RNP.
- class [RotationMars](#)
Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

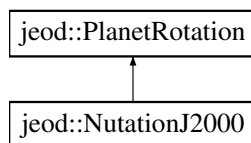
Data Structure Documentation

8.1 jeod::NutationJ2000 Class Reference

Implements the nutation portion of the J2000 RNP.

```
#include <nutation_j2000.hh>
```

Inheritance diagram for jeod::NutationJ2000:



Public Member Functions

- [NutationJ2000](#) ()
Constructor.
- virtual [~NutationJ2000](#) ()
destructor
- virtual void [update_rotation](#) ()
Specific implementation of update_rotation, from the polymorphic pure virtual base class [PlanetRotation](#).
- virtual void [initialize](#) ([PlanetRotationInit](#) *init)
Initialize the various coefficients needed for the calculation of nutation.

Data Fields

- unsigned int [num_coefs](#)
The number of coefficients in the 9 arrays directly following this declaration.
- double * [L_coefs](#)
The coefficients to calculate mean anomaly of the moon.
- double * [M_coefs](#)
The coefficients to calculate mean anomaly of the sun.
- double * [F_coefs](#)

- The coefficients to calculate mean argument of latitude of the moon.*

 - double * [D_coeffs](#)
- The coefficients to calculate mean elongation from the sun.*

 - double * [omega_coeffs](#)
- The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.*

 - double * [long_coeffs](#)
- Zero order (in time) coefficients for calculating the nutation in longitude.*

 - double * [long_t_coeffs](#)
- The first order (in time) coefficients for calculating the nutation in longitude.*

 - double * [obliq_coeffs](#)
- Zero order coefficients for calculating the nutation in obliquity.*

 - double * [obliq_t_coeffs](#)
- First order coefficients for calculating the nutation in obliquity.*

 - double [nutation_in_longitude](#)
- The nutation in longitude from the last call of update.*

 - double [nutation_in_obliquity](#)
- The nutation in obliquity from the last call of update.*

 - double [L](#)
- Last calculated mean anomaly of the moon.*

 - double [M](#)
- Last calculated mean anomaly of the sun.*

 - double [F](#)
- Last calculated mean argument of latitude of the moon.*

 - double [D](#)
- Last calculated mean elongation from the sun.*

 - double [omega](#)
- Last calculated ascension of the ascending node of the mean lunar orbit.*

 - double [epsilon_bar](#)
- Last calculated mean obliquity of the ecliptic.*

 - double [equa_of_equi](#)
- Last calculated equations of the equinox.*

Private Member Functions

- [NutationJ2000](#) & [operator=](#) (const [NutationJ2000](#) &rhs)
- [NutationJ2000](#) (const [NutationJ2000](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__NutationJ2000](#) ()

Additional Inherited Members

8.1.1 Detailed Description

Implements the nutation portion of the J2000 RNP.

Definition at line 92 of file include/nutation_j2000.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 NutationJ2000() [1/2]

```
jeod::NutationJ2000::NutationJ2000 (
    void )
```

Constructor.

Initialize all in class data

Definition at line 66 of file `nutations_j2000.cc`.

8.1.2.2 ~NutationJ2000()

```
jeod::NutationJ2000::~~NutationJ2000 (
    void ) [virtual]
```

destructor

Definition at line 96 of file `nutations_j2000.cc`.

References `D_coeffs`, `F_coeffs`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M_coeffs`, `obliq_coeffs`, `obliq_t_coeffs`, and `omega_coeffs`.

8.1.2.3 NutationJ2000() [2/2]

```
jeod::NutationJ2000::NutationJ2000 (
    const NutationJ2000 & rhs ) [private]
```

8.1.3 Member Function Documentation

8.1.3.1 initialize()

```
void jeod::NutationJ2000::initialize (
    PlanetRotationInit * init ) [virtual]
```

Initialize the various coefficients needed for the calculation of nutation.

init must be of type [NutationJ2000Init](#) or a fail message will occur

Parameters

<code>in</code>	<code>init</code>	NutationJ2000Init object with needed coefficients
-----------------	-------------------	---

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

Definition at line 269 of file `nutatation_j2000.cc`.

References `jeod::NutationJ2000Init::D_coeffs`, `D_coeffs`, `jeod::NutationJ2000Init::F_coeffs`, `F_coeffs`, `jeod::RN←PMessages::initialization_error`, `jeod::NutationJ2000Init::L_coeffs`, `L_coeffs`, `jeod::NutationJ2000Init::long_coeffs`, `long_coeffs`, `jeod::NutationJ2000Init::long_t_coeffs`, `long_t_coeffs`, `jeod::NutationJ2000Init::M_coeffs`, `M_coeffs`, `jeod::NutationJ2000Init::num_coeffs`, `num_coeffs`, `jeod::NutationJ2000Init::obliq_coeffs`, `obliq_coeffs`, `jeod::←NutationJ2000Init::obliq_t_coeffs`, `obliq_t_coeffs`, `jeod::NutationJ2000Init::omega_coeffs`, and `omega_coeffs`.

8.1.3.2 operator=()

```
NutationJ2000& jeod::NutationJ2000::operator= (
    const NutationJ2000 & rhs ) [private]
```

8.1.3.3 update_rotation()

```
void jeod::NutationJ2000::update_rotation (
    void ) [virtual]
```

Specific implementation of `update_rotation`, from the polymorphic pure virtual base class [PlanetRotation](#).

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

Definition at line 144 of file `nutatation_j2000.cc`.

References `jeod::PlanetRotation::current_time`, `D`, `D_coeffs`, `jeod::PlanetRotation::DEGTORAD`, `epsilon_bar`, `equa_of_equi`, `F`, `F_coeffs`, `L`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M`, `M_coeffs`, `num_coeffs`, `nutatation_in_longitude`, `nutatation_in_obliquity`, `obliq_coeffs`, `obliq_t_coeffs`, `omega`, `omega_coeffs`, and `jeod::PlanetRotation::rotation`.

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__NutationJ2000

```
void init_attrjeod__NutationJ2000 ( ) [friend]
```

8.1.4.2 InputProcessor

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

Definition at line 94 of file include/nutation_j2000.hh.

8.1.5 Field Documentation

8.1.5.1 D

```
double jeod::NutationJ2000::D
```

Last calculated mean elongation from the sun.

trick_units(-)

Definition at line 175 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.2 D_coeffs

```
double* jeod::NutationJ2000::D_coeffs
```

The coefficients to calculate mean elongation from the sun.

trick_units(-)

Definition at line 120 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.3 epsilon_bar

```
double jeod::NutationJ2000::epsilon_bar
```

Last calculated mean obliquity of the ecliptic.

trick_units(-)

Definition at line 185 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.4 `equa_of_equi`

```
double jeod::NutationJ2000::equa_of_equi
```

Last calculated equations of the equinox.

`trick_units(-)`

Definition at line 190 of file `include/nutation_j2000.hh`.

Referenced by `jeod::RotationJ2000::update_rotation()`, and `update_rotation()`.

8.1.5.5 `F`

```
double jeod::NutationJ2000::F
```

Last calculated mean argument of latitude of the moon.

`trick_units(-)`

Definition at line 171 of file `include/nutation_j2000.hh`.

Referenced by `update_rotation()`.

8.1.5.6 `F_coeffs`

```
double* jeod::NutationJ2000::F_coeffs
```

The coefficients to calculate mean argument of latitude of the moon.

`trick_units(-)`

Definition at line 116 of file `include/nutation_j2000.hh`.

Referenced by `initialize()`, `update_rotation()`, and `~NutationJ2000()`.

8.1.5.7 `L`

```
double jeod::NutationJ2000::L
```

Last calculated mean anomaly of the moon.

`trick_units(-)`

Definition at line 163 of file `include/nutation_j2000.hh`.

Referenced by `update_rotation()`.

8.1.5.8 L_coeffs

```
double* jeod::NutationJ2000::L_coeffs
```

The coefficients to calculate mean anomaly of the moon.

trick_units(—)

Definition at line 108 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.9 long_coeffs

```
double* jeod::NutationJ2000::long_coeffs
```

Zero order (in time) coefficients for calculating the nutation in longitude.

trick_units(—)

Definition at line 132 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.10 long_t_coeffs

```
double* jeod::NutationJ2000::long_t_coeffs
```

The first order (in time) coefficients for calculating the nutation in longitude.

trick_units(—)

Definition at line 137 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.11 M

```
double jeod::NutationJ2000::M
```

Last calculated mean anomaly of the sun.

trick_units(—)

Definition at line 167 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.12 M_coeffs

```
double* jeod::NutationJ2000::M_coeffs
```

The coefficients to calculate mean anomaly of the sun.

trick_units(-)

Definition at line 112 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.13 num_coeffs

```
unsigned int jeod::NutationJ2000::num_coeffs
```

The number of coefficients in the 9 arrays directly following this decleration.

trick_units(count)

Definition at line 103 of file include/nutation_j2000.hh.

Referenced by initialize(), and update_rotation().

8.1.5.14 nutation_in_longitude

```
double jeod::NutationJ2000::nutation_in_longitude
```

The nutation in longitude from the last call of update.

trick_units(-)

Definition at line 153 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.15 nutation_in_obliquity

```
double jeod::NutationJ2000::nutation_in_obliquity
```

The nutation in obliquity from the last call of update.

trick_units(-)

Definition at line 158 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.16 obliq_coeffs

```
double* jeod::NutationJ2000::obliq_coeffs
```

Zero order coefficients for calculating the nutation in obliquity.

trick_units(-)

Definition at line 142 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.17 obliq_t_coeffs

```
double* jeod::NutationJ2000::obliq_t_coeffs
```

First order coefficients for calculating the nutation in obliquity.

trick_units(-)

Definition at line 147 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

8.1.5.18 omega

```
double jeod::NutationJ2000::omega
```

Last calculated ascension of the ascending node of the mean lunar orbit.

trick_units(-)

Definition at line 180 of file include/nutation_j2000.hh.

Referenced by update_rotation().

8.1.5.19 omega_coeffs

```
double* jeod::NutationJ2000::omega_coeffs
```

The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.

trick_units(-)

Definition at line 126 of file include/nutation_j2000.hh.

Referenced by initialize(), update_rotation(), and ~NutationJ2000().

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

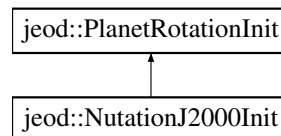
- [include/nutation_j2000.hh](#)
- [nututation_j2000.cc](#)

8.2 jeod::NutationJ2000Init Class Reference

The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.

```
#include <nutatation_j2000_init.hh>
```

Inheritance diagram for jeod::NutationJ2000Init:



Public Member Functions

- [NutationJ2000Init](#) ()
constructor.
- virtual [~NutationJ2000Init](#) ()
Destructor.

Data Fields

- int [num_coeffs](#)
The number of coefficients in the 9 arrays directly following this declaration.
- double * [L_coeffs](#)
The coefficients to calculate mean anomaly of the moon.
- double * [M_coeffs](#)
The coefficients to calculate mean anomaly of the sun.
- double * [F_coeffs](#)
The coefficients to calculate mean argument of latitude of the moon.
- double * [D_coeffs](#)
The coefficients to calculate mean elongation from the sun.
- double * [omega_coeffs](#)
The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.
- double * [long_coeffs](#)
Zero order (in time) coefficients for calculating the nutation in longitude.
- double * [long_t_coeffs](#)
The first order (in time) coefficients for calculating the nutation in longitude.
- double * [obliq_coeffs](#)
Zero order coefficients for calculating the nutation in obliquity.
- double * [obliq_t_coeffs](#)
First order coefficients for calculating the nutation in obliquity.

Private Member Functions

- [NutationJ2000Init](#) & [operator=](#) (const [NutationJ2000Init](#) &rhs)
- [NutationJ2000Init](#) (const [NutationJ2000Init](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__NutationJ2000Init](#) ()

8.2.1 Detailed Description

The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.

Definition at line 92 of file `nutations_j2000_init.hh`.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 NutationJ2000Init() [1/2]

```
jeod::NutationJ2000Init::NutationJ2000Init (
    void )
```

constructor.

initialize low level data

Definition at line 56 of file `nutations_j2000_init.cc`.

8.2.2.2 ~NutationJ2000Init()

```
jeod::NutationJ2000Init::~~NutationJ2000Init (
    void ) [virtual]
```

Destructor.

Definition at line 76 of file `nutations_j2000_init.cc`.

References `D_coeffs`, `F_coeffs`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M_coeffs`, `obliq_coeffs`, `obliq_t_coeffs`, and `omega_coeffs`.

8.2.2.3 NutationJ2000Init() [2/2]

```
jeod::NutationJ2000Init::NutationJ2000Init (
    const NutationJ2000Init & rhs ) [private]
```

8.2.3 Member Function Documentation

8.2.3.1 operator=()

```
NutationJ2000Init& jeod::NutationJ2000Init::operator= (
    const NutationJ2000Init & rhs ) [private]
```

8.2.4 Friends And Related Function Documentation

8.2.4.1 init_attrjeod__NutationJ2000Init

```
void init_attrjeod__NutationJ2000Init ( ) [friend]
```

8.2.4.2 InputProcessor

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

Definition at line 94 of file nutation_j2000_init.hh.

8.2.5 Field Documentation

8.2.5.1 D_coeffs

```
double* jeod::NutationJ2000Init::D_coeffs
```

The coefficients to calculate mean elongation from the sun.

trick_units(-)

Definition at line 119 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.2 F_coeffs

```
double* jeod::NutationJ2000Init::F_coeffs
```

The coefficients to calculate mean argument of latitude of the moon.

trick_units(-)

Definition at line 115 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.3 L_coeffs

```
double* jeod::NutationJ2000Init::L_coeffs
```

The coefficients to calculate mean anomaly of the moon.

trick_units(-)

Definition at line 107 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.4 long_coeffs

```
double* jeod::NutationJ2000Init::long_coeffs
```

Zero order (in time) coefficients for calculating the nutation in longitude.

trick_units(-)

Definition at line 131 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.5 long_t_coeffs

```
double* jeod::NutationJ2000Init::long_t_coeffs
```

The first order (in time) coefficients for calculating the nutation in longitude.

trick_units(-)

Definition at line 136 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.6 M_coeffs

```
double* jeod::NutationJ2000Init::M_coeffs
```

The coefficients to calculate mean anomaly of the sun.

trick_units(-)

Definition at line 111 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.7 num_coeffs

```
int jeod::NutationJ2000Init::num_coeffs
```

The number of coefficients in the 9 arrays directly following this decleration.

trick_units(count)

Definition at line 102 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), and jeod::NutationJ2000::initialize().

8.2.5.8 obliq_coeffs

```
double* jeod::NutationJ2000Init::obliq_coeffs
```

Zero order coefficients for calculating the nutation in obliquity.

trick_units(-)

Definition at line 141 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.9 obliq_t_coeffs

```
double* jeod::NutationJ2000Init::obliq_t_coeffs
```

First order coefficnets for calculating the nutation in obliquity.

trick_units(-)

Definition at line 146 of file nutation_j2000_init.hh.

Referenced by jeod::NutationJ2000Init_nutation_j2000_default_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

8.2.5.10 omega_coeffs

```
double* jeod::NutationJ2000Init::omega_coeffs
```

The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.

trick_units(-)

Definition at line 125 of file nutation_j2000_init.hh.

Referenced by `jeod::NutationJ2000Init_nutation_j2000_default_data::initialize()`, `jeod::NutationJ2000::initialize()`, and `~NutationJ2000Init()`.

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

- [nutation_j2000_init.hh](#)
- [nutation_j2000_init.cc](#)

8.3 jeod::NutationJ2000Init_nutation_j2000_default_data Class Reference

```
#include <nutation_j2000.hh>
```

Public Member Functions

- void [initialize](#) ([NutationJ2000Init](#) *)

8.3.1 Detailed Description

Definition at line 54 of file data/include/nutation_j2000.hh.

8.3.2 Member Function Documentation

8.3.2.1 initialize()

```
void jeod::NutationJ2000Init_nutation_j2000_default_data::initialize (
    NutationJ2000Init * NutationJ2000Init_ptr )
```

Definition at line 37 of file data_nutation_j2000.cc.

References `jeod::NutationJ2000Init::D_coeffs`, `jeod::NutationJ2000Init::F_coeffs`, `jeod::NutationJ2000Init::L_coeffs`, `jeod::NutationJ2000Init::long_coeffs`, `jeod::NutationJ2000Init::long_t_coeffs`, `jeod::NutationJ2000Init::M_coeffs`, `jeod::NutationJ2000Init::num_coeffs`, `jeod::NutationJ2000Init::obliq_coeffs`, `jeod::NutationJ2000Init::obliq_t_coeffs`, and `jeod::NutationJ2000Init::omega_coeffs`.

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

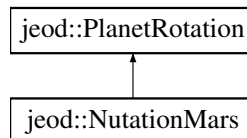
- [data/include/nutation_j2000.hh](#)
- [data_nutation_j2000.cc](#)

8.4 jeod::NutationMars Class Reference

Implements the nutation portion of the "Pathfinder" Mars RNP model.

```
#include <nutation_mars.hh>
```

Inheritance diagram for jeod::NutationMars:



Public Member Functions

- [NutationMars](#) ()
Default constructor.
- virtual [~NutationMars](#) ()
Destructor.
- virtual void [update_rotation](#) ()
[NutationMars](#) specific implementation of update_rotation, used here to describe nutation effects in longitude and obliquity.

Data Fields

- double [nutation_in_longitude](#)
The latest calculated nutation correction term in longitude.
- double [nutation_in_obliquity](#)
The latest calculated nutation correction term in obliquity.
- double [l_at_j2000](#)
The (constant) obliquity angle of Mars at the J2000 epoch.
- double [l_dot](#)
The (constant) simple secular change in Mars obliquity relative to the Mars mean orbit.
- double [obliquity_angle](#)
The latest calculated angle of obliquity, that is, the current orbit inclination angle as measured relative to the Mars mean orbit, measured since the J2000 epoch.
- double [mean_motion](#)
The mean motion of Mars, that is, the average orbit arc traversed.
- double [mean_anomaly_j2000](#)
The Mars mean anomaly at the J2000 epoch.
- double [q_angle_j2000](#)
Parameter q is a slowly varying angle defined in Konopliv 2006 as 2 times w , where w is the argument of perihelion of the Mars orbit relative to the node of the Mars equator and Mars mean orbit.
- double * [l_m_orig](#)
The obliquity nutation amplitude parameters.
- double * [psi_m_orig](#)
The longitude nutation amplitude parameters.
- double * [int_to_double](#)
0 to 9 cast as doubles for use in summation

Private Member Functions

- [NutationMars](#) & [operator=](#) (const [NutationMars](#) &rhs)
- [NutationMars](#) (const [NutationMars](#) &rhs)

Friends

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

Additional Inherited Members

8.4.1 Detailed Description

Implements the nutation portion of the "Pathfinder" Mars RNP model.

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

8.4.2 Constructor & Destructor Documentation

8.4.2.1 [NutationMars\(\)](#) [1/2]

```
jeod::NutationMars::NutationMars (  
    void )
```

Default constructor.

Definition at line 64 of file `nutation_mars.cc`.

8.4.2.2 [~NutationMars\(\)](#)

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

Destructor.

Definition at line 85 of file `nutation_mars.cc`.

References `I_m_orig`, `int_to_double`, and `psi_m_orig`.

8.4.2.3 NutationMars() [2/2]

```
jeod::NutationMars::NutationMars (
    const NutationMars & rhs ) [private]
```

8.4.3 Member Function Documentation

8.4.3.1 operator=()

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

8.4.3.2 update_rotation()

```
void jeod::NutationMars::update_rotation (
    void ) [virtual]
```

[NutationMars](#) specific implementation of update_rotation, used here to describe nutation effects in longitude and obliquity.

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

Definition at line 110 of file nutation_mars.cc.

References [jeod::PlanetRotation::current_time](#), [l_at_j2000](#), [l_dot](#), [l_m_orig](#), [int_to_double](#), [mean_anomaly_j2000](#), [mean_motion](#), [nutation_in_longitude](#), [nutation_in_obliquity](#), [obliquity_angle](#), [psi_m_orig](#), [q_angle_j2000](#), and [jeod::PlanetRotation::rotation](#).

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__NutationMars

```
void init_attrjeod__NutationMars ( ) [friend]
```

8.4.4.2 InputProcessor

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

Definition at line 97 of file nutation_mars.hh.

8.4.5 Field Documentation

8.4.5.1 I_at_j2000

```
double jeod::NutationMars::I_at_j2000
```

The (constant) obliquity angle of Mars at the J2000 epoch.

trick_units(rad)

Definition at line 114 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.4.5.2 I_dot

```
double jeod::NutationMars::I_dot
```

The (constant) simple secular change in Mars obliquity relative to the Mars mean orbit.

trick_units(rad/s)

Definition at line 120 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.4.5.3 I_m_orig

```
double* jeod::NutationMars::I_m_orig
```

The obliquity nutation amplitude parameters.

trick_units(rad)

Definition at line 150 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), update_rotation(), and ~NutationMars().

8.4.5.4 int_to_double

```
double* jeod::NutationMars::int_to_double
```

0 to 9 cast as doubles for use in summation

trick_units(-)

Definition at line 160 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), update_rotation(), and ~NutationMars().

8.4.5.5 mean_anomaly_j2000

```
double jeod::NutationMars::mean_anomaly_j2000
```

The Mars mean anomaly at the J2000 epoch.

trick_units(rad)

Definition at line 137 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.4.5.6 mean_motion

```
double jeod::NutationMars::mean_motion
```

The mean motion of Mars, that is, the average orbit arc traversed.

trick_units(rad/s)

Definition at line 132 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.4.5.7 nutation_in_longitude

```
double jeod::NutationMars::nutation_in_longitude
```

The latest calculated nutation correction term in longitude.

trick_units(rad)

Definition at line 104 of file nutation_mars.hh.

Referenced by jeod::RotationMars::update_rotation(), jeod::PrecessionMars::update_rotation(), and update_rotation().

8.4.5.8 nutation_in_obliquity

```
double jeod::NutationMars::nutation_in_obliquity
```

The latest calculated nutation correction term in obliquity.

trick_units(rad)

Definition at line 109 of file nutation_mars.hh.

Referenced by update_rotation().

8.4.5.9 obliquity_angle

```
double jeod::NutationMars::obliquity_angle
```

The latest calculated angle of obliquity, that is, the current orbit inclination angle as measured relative to the Mars mean orbit, measured since the J2000 epoch.

trick_units(rad)

Definition at line 127 of file nutation_mars.hh.

Referenced by jeod::RotationMars::update_rotation(), and update_rotation().

8.4.5.10 psi_m_orig

```
double* jeod::NutationMars::psi_m_orig
```

The longitude nutation amplitude parameters.

trick_units(rad)

Definition at line 155 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), update_rotation(), and ~NutationMars().

8.4.5.11 q_angle_j2000

```
double jeod::NutationMars::q_angle_j2000
```

Parameter q is a slowly varying angle defined in Konopliv 2006 as 2 times w, where w is the argument of perihelion of the Mars orbit relative to the node of the Mars equator and Mars mean orbit.

This data member is the value of q at the J2000 epoch.trick_units(rad)

Definition at line 145 of file nutation_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

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

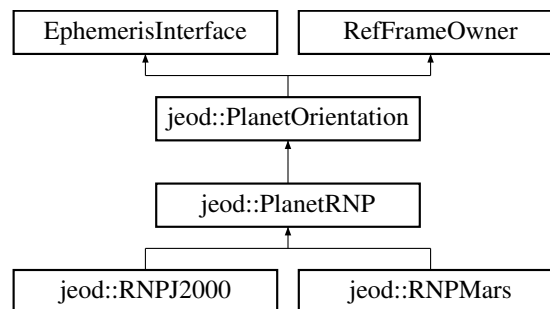
- [nutation_mars.hh](#)
- [nutation_mars.cc](#)

8.5 jeod::PlanetOrientation Class Reference

The generic framework for orientation models to interact with a DynManager object.

```
#include <planet_orientation.hh>
```

Inheritance diagram for jeod::PlanetOrientation:



Public Member Functions

- [PlanetOrientation](#) ()
Default constructor; constructs a [PlanetOrientation](#) object.
- virtual [~PlanetOrientation](#) ()
Class destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Goes to the dyn manager given and searches for the planet indicated by the user inputted name, in the given dyn manager.
- virtual void [activate](#) ()
Activates the [PlanetOrientation](#) model.
- virtual void [deactivate](#) ()
Deactivates the [PlanetOrientation](#) model.
- virtual void [ephem_initialize](#) (EphemeridesManager &manager)
Implements the EphemerisInterface pure virtual function `ephem_initialize`.
- virtual void [ephem_activate](#) (EphemeridesManager &manager)
Mark the model as being activate or inactive.
- virtual void [ephem_build_tree](#) (EphemeridesManager &manager)
Function, for a particular ephemeris model, nominally builds the particular models portions of the reference frame tree.
- void [set_name](#) (std::string name_in)
Setter for the name.
- virtual const char * [get_name](#) () const =0
A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

Data Fields

- bool [active](#)
Is the orientation model actively updating? Defaults to true.
- Planet * [planet](#)
The planet the attitude model will be working on.
- std::string [name](#)
Name of the planet the attitude model will be working on.
- RefFrameRot * [planet_rot_state](#)
The current rotational state of the planet.
- double [planet_omega](#)
Nominal axial velocity of the earth.
- EphemerisOrientation [orient_interface](#)
The ephemeris interface to the in question orientation.

Private Member Functions

- [PlanetOrientation](#) & [operator=](#) (const [PlanetOrientation](#) &rhs)
Not implemented.
- [PlanetOrientation](#) (const [PlanetOrientation](#) &rhs)
Not implemented.

Friends

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

8.5.1 Detailed Description

The generic framework for orientation models to interact with a DynManager object.

Definition at line 98 of file planet_orientation.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 PlanetOrientation() [1/2]

```
jeod::PlanetOrientation::PlanetOrientation (
    void )
```

Default constructor; constructs a [PlanetOrientation](#) object.

Definition at line 59 of file planet_orientation.cc.

8.5.2.2 ~PlanetOrientation()

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

Class destructor.

Definition at line 74 of file planet_orientation.cc.

8.5.2.3 PlanetOrientation() [2/2]

```
jeod::PlanetOrientation::PlanetOrientation (
    const PlanetOrientation & rhs ) [private]
```

Not implemented.

8.5.3 Member Function Documentation

8.5.3.1 activate()

```
void jeod::PlanetOrientation::activate ( ) [virtual]
```

Activates the [PlanetOrientation](#) model.

Definition at line 140 of file planet_orientation.cc.

References active.

8.5.3.2 deactivate()

```
void jeod::PlanetOrientation::deactivate ( ) [virtual]
```

Deactivates the [PlanetOrientation](#) model.

Definition at line 149 of file planet_orientation.cc.

References active.

8.5.3.3 ephem_activate()

```
void jeod::PlanetOrientation::ephem_activate (
    EphemeridesManager & manager ) [virtual]
```

Mark the model as being activate or inactive.

Parameters

in, out	<i>manager</i>	ephemerides manager
---------	----------------	---------------------

Definition at line 175 of file planet_orientation.cc.

8.5.3.4 ephem_build_tree()

```
void jeod::PlanetOrientation::ephem_build_tree (
    EphemeridesManager & manager ) [virtual]
```

Function, for a particular ephemeris model, nominally builds the particular models portions of the reference frame tree.

This implementation, by default, does nothing

Parameters

in, out	<i>manager</i>	ephemerides manager
---------	----------------	---------------------

Definition at line 189 of file planet_orientation.cc.

8.5.3.5 ephem_initialize()

```
void jeod::PlanetOrientation::ephem_initialize (
    EphemeridesManager & manager ) [virtual]
```

Implements the EphemInterface pure virtual function ephem_initialize.

Note that the stand alone "initialize" function MUST be called before this function. Normal operating procedure is the DynamicsManager calls this function, not the user.

Parameters

in, out	<i>manager</i>	ephemerides manager
---------	----------------	---------------------

Definition at line 164 of file planet_orientation.cc.

8.5.3.6 get_name()

```
virtual const char* jeod::PlanetOrientation::get_name ( ) const [pure virtual]
```

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

Returns

Planet name.

Implemented in [jeod::PlanetRNP](#), [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

8.5.3.7 initialize()

```
void jeod::PlanetOrientation::initialize (
    DynManager & dyn_manager ) [virtual]
```

Goes to the dyn manager given and searches for the planet indicated by the user inputted name, in the given dyn manager.

Will send a fail message if the named planet is not found. Additionally, this will register this ephemeris interface with the given DynManager, and will set up the contained EphemerisOrientation to control the named planet.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	DynManager where the attitude will be applied
----------------------	--------------------------	---

Reimplemented in [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

Definition at line 90 of file planet_orientation.cc.

References [jeod::RNPMessages::initialization_error](#), [name](#), [orient_interface](#), [planet](#), [planet_omega](#), and [planet_rot_state](#).

Referenced by [jeod::RNPJ2000::initialize\(\)](#), and [jeod::RNPMars::initialize\(\)](#).

8.5.3.8 operator=()

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

Not implemented.

8.5.3.9 set_name()

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

Setter for the name.

Definition at line 171 of file planet_orientation.hh.

References [name](#).

8.5.4 Friends And Related Function Documentation

8.5.4.1 init_attrjeod__PlanetOrientation

```
void init_attrjeod__PlanetOrientation ( ) [friend]
```

8.5.4.2 InputProcessor

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

Definition at line 101 of file planet_orientation.hh.

8.5.5 Field Documentation

8.5.5.1 active

```
bool jeod::PlanetOrientation::active
```

Is the orientation model actively updating? Defaults to true.

trick_units(—)

Definition at line 121 of file planet_orientation.hh.

Referenced by activate(), deactivate(), jeod::RNPJ2000::ephem_update(), jeod::RNPMars::ephem_update(), jeod::RNPJ2000::update_axial_rotation(), jeod::RNPMars::update_axial_rotation(), jeod::RNPJ2000::update_↔rnp(), and jeod::RNPMars::update_rnp().

8.5.5.2 name

```
std::string jeod::PlanetOrientation::name
```

Name of the planet the attitude model will be working on.

Planet must be found in the DynManager sent in at initializationtrick_units(—)

Definition at line 132 of file planet_orientation.hh.

Referenced by jeod::RNPJ2000_rnp_j2000_default_data::initialize(), jeod::RNPMars_rnp_mars_default_data↔::initialize(), initialize(), and set_name().

8.5.5.3 orient_interface

`EphemerisOrientation jeod::PlanetOrientation::orient_interface`

The ephemeris interface to the in question orientation.

`trick_units(-)`

Definition at line 165 of file `planet_orientation.hh`.

Referenced by `jeod::RNPJ2000::ephem_update()`, `jeod::RNPMars::ephem_update()`, and `initialize()`.

8.5.5.4 planet

`Planet* jeod::PlanetOrientation::planet`

The planet the attitude model will be working on.

`trick_units(-)`

Definition at line 126 of file `planet_orientation.hh`.

Referenced by `initialize()`, `jeod::RNPJ2000::update_axial_rotation()`, `jeod::RNPMars::update_axial_rotation()`, `jeod::RNPJ2000::update_rnp()`, and `jeod::RNPMars::update_rnp()`.

8.5.5.5 planet_omega

`double jeod::PlanetOrientation::planet_omega`

Nominal axial velocity of the earth.

`trick_units(rad/s)`

Definition at line 142 of file `planet_orientation.hh`.

Referenced by `jeod::RNPJ2000_rnp_j2000_default_data::initialize()`, `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, `jeod::RNPJ2000::initialize()`, `jeod::RNPMars::initialize()`, and `jeod::PlanetRNP::propagate_rnp()`.

8.5.5.6 planet_rot_state

`RefFrameRot* jeod::PlanetOrientation::planet_rot_state`

The current rotational state of the planet.

`trick_units(-)`

Definition at line 137 of file `planet_orientation.hh`.

Referenced by `initialize()`, and `jeod::PlanetRNP::propagate_rnp()`.

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

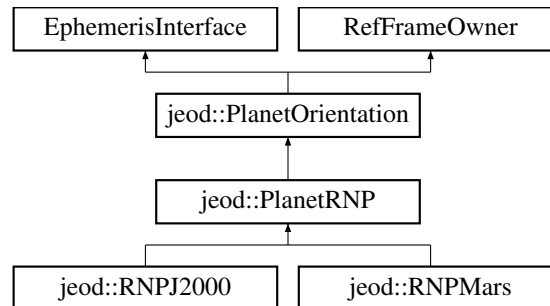
- [planet_orientation.hh](#)
- [planet_orientation.cc](#)

8.6 jeod::PlanetRNP Class Reference

The generic framework for orientation models based on the RNP paradigm.

```
#include <planet_rnp.hh>
```

Inheritance diagram for jeod::PlanetRNP:



Public Types

- enum [RNPFidelity](#) { [FullRNP](#) = 0, [RotationOnly](#) = 1, [ConstantNP](#) = 2 }
- Specifies the initialization fidelity of the RNP model.*

Public Member Functions

- [PlanetRNP](#) ()
Default constructor; constructs a [PlanetRNP](#) object.
- virtual [~PlanetRNP](#) ()
Class destructor.
- void [update_rnp](#) ()
Invokes the calculation for all rotation models contained in the RNP, used on the last time set in each model through [PlanetRotation::set_time](#).
- void [update_axial_rotation](#) ()
Same as [update_rnp](#), but only the [axial_rotation](#) will be updated.
- void [propagate_rnp](#) ()
Multiplies out the (up to) four planet rotation models (nutation, precession, polar motion and rotation), leaving out models whose pointers are NULL, and feeds it to the planet attitude found in the dyn manager given in initialize.
- virtual const char * [get_name](#) () const =0
A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

Data Fields

- [PlanetRotation](#) * [nutation](#)
Pointer to the nutation model.
- [PlanetRotation](#) * [precession](#)
Pointer to the precession model.
- [PlanetRotation](#) * [polar_motion](#)
Pointer to the polar_motion model.
- [PlanetRotation](#) * [rotation](#)

- *Pointer to the rotation model.*
- [RNPFidelity rnp_type](#)
The fidelity of the RNP model.
- `bool enable_polar`
Gives the option of turning on or off polar motion.
- `double NP_matrix [3][3]`
*Rotation Matrix representing: $\text{transpose}(\text{nutaton} \rightarrow \text{rotation}) * \text{transpose}(\text{precession} \rightarrow \text{rotation})$*

Protected Attributes

- `double scratch_matrix [3][3]`
A transformation matrix used for intermediate math steps.

Private Member Functions

- `PlanetRNP & operator= (const PlanetRNP &rhs)`
Not implemented.
- `PlanetRNP (const PlanetRNP &rhs)`
Not implemented.

Friends

- `class InputProcessor`
- `void init_attrjeod__PlanetRNP ()`

8.6.1 Detailed Description

The generic framework for orientation models based on the RNP paradigm.

Definition at line 104 of file planet_rnp.hh.

8.6.2 Member Enumeration Documentation

8.6.2.1 RNPFidelity

```
enum jeod::PlanetRNP::RNPFidelity
```

Specifies the initialization fidelity of the RNP model.

Enumerator

FullRNP	Full fidelity RNP matrix. Formerly Full_Term_RNP.
RotationOnly	Identity NP matrix, then rotation calculates linearly.
ConstantNP	Once calculated NP matrix (at the start), then rotation.

Definition at line 115 of file planet_rnp.hh.

8.6.3 Constructor & Destructor Documentation

8.6.3.1 PlanetRNP() [1/2]

```
jeod::PlanetRNP::PlanetRNP (  
    void )
```

Default constructor; constructs a [PlanetRNP](#) object.

Definition at line 72 of file planet_rnp.cc.

References [NP_matrix](#).

8.6.3.2 ~PlanetRNP()

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

Class destructor.

Definition at line 88 of file planet_rnp.cc.

8.6.3.3 PlanetRNP() [2/2]

```
jeod::PlanetRNP::PlanetRNP (  
    const PlanetRNP & rhs ) [private]
```

Not implemented.

8.6.4 Member Function Documentation

8.6.4.1 `get_name()`

```
virtual const char* jeod::PlanetRNP::get_name ( ) const [pure virtual]
```

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

Returns

Planet name.

Implements [jeod::PlanetOrientation](#).

Implemented in [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

8.6.4.2 `operator=()`

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

Not implemented.

8.6.4.3 `propagate_rnp()`

```
void jeod::PlanetRNP::propagate_rnp (
    void )
```

Multiples out the (up to) four planet rotation models (nutation, precession, polar motion and rotation), leaving out models whose pointers are NULL, and feeds it to the planet attitude found in the dyn manager given in initialize.

Definition at line 201 of file planet_rnp.cc.

References ConstantNP, enable_polar, jeod::RNPMessages::fidelity_error, FullRNP, NP_matrix, jeod::PlanetOrientation::planet_omega, jeod::PlanetOrientation::planet_rot_state, polar_motion, rnp_type, jeod::PlanetRotation::rotation, rotation, RotationOnly, scratch_matrix, and jeod::RNPMessages::setup_error.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

8.6.4.4 `update_axial_rotation()`

```
void jeod::PlanetRNP::update_axial_rotation (
    void )
```

Same as `update_rnp`, but only the `axial_rotation` will be updated.

Definition at line 173 of file planet_rnp.cc.

References `propagate_rnp()`, `rotation`, jeod::RNPMessages::setup_error, and jeod::PlanetRotation::update_axial_rotation().

Referenced by jeod::RNPJ2000::ephem_update(), jeod::RNPMars::ephem_update(), jeod::RNPJ2000::update_axial_rotation(), and jeod::RNPMars::update_axial_rotation().

8.6.4.5 update_rnp()

```
void jeod::PlanetRNP::update_rnp (
    void )
```

Invokes the calculation for all rotation models contained in the RNP, used on the last time set in each model through PlanetRotation::set_time.

Then multiplies out and updates the attitude of the planet found during initialization.

Definition at line 102 of file planet_rnp.cc.

References enable_polar, jeod::RNPMessages::fidelity_error, FullRNP, NP_matrix, nutation, polar_motion, precession, propagate_rnp(), rnp_type, jeod::PlanetRotation::rotation, rotation, jeod::RNPMessages::setup_error, and jeod::PlanetRotation::update_rotation().

Referenced by jeod::RNPJ2000::update_rnp(), and jeod::RNPMars::update_rnp().

8.6.5 Friends And Related Function Documentation

8.6.5.1 init_attrjeod__PlanetRNP

```
void init_attrjeod__PlanetRNP ( ) [friend]
```

8.6.5.2 InputProcessor

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

Definition at line 106 of file planet_rnp.hh.

8.6.6 Field Documentation

8.6.6.1 enable_polar

```
bool jeod::PlanetRNP::enable_polar
```

Gives the option of turning on or off polar motion.

```
trick_units(-)
```

Definition at line 153 of file planet_rnp.hh.

Referenced by jeod::RNPJ2000_rnp_j2000_default_data::initialize(), jeod::RNPMars_rnp_mars_default_data::initialize(), propagate_rnp(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update_rnp(), and update_rnp().

8.6.6.2 NP_matrix

```
double jeod::PlanetRNP::NP_matrix[3][3]
```

Rotation Matrix representing: $\text{transpose}(\text{nutution} \rightarrow \text{rotation}) * \text{transpose}(\text{precession} \rightarrow \text{rotation})$

trick_units(-)

Definition at line 159 of file planet_rnp.hh.

Referenced by PlanetRNP(), propagate_rnp(), and update_rnp().

8.6.6.3 nutation

```
PlanetRotation* jeod::PlanetRNP::nutation
```

Pointer to the nutation model.

trick_units(-)

Definition at line 132 of file planet_rnp.hh.

Referenced by jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update_rnp(), update_rnp(), and jeod::RNPMars::update_rnp().

8.6.6.4 polar_motion

```
PlanetRotation* jeod::PlanetRNP::polar_motion
```

Pointer to the polar_motion model.

trick_units(-)

Definition at line 140 of file planet_rnp.hh.

Referenced by propagate_rnp(), jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update_rnp(), and update_rnp().

8.6.6.5 precession

```
PlanetRotation* jeod::PlanetRNP::precession
```

Pointer to the precession model.

trick_units(-)

Definition at line 136 of file planet_rnp.hh.

Referenced by jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update_rnp(), update_rnp(), and jeod::RNPMars::update_rnp().

8.6.6.6 rnp_type

`RNPFidelity` jeod::PlanetRNP::rnp_type

The fidelity of the RNP model.

trick_units(-)

Definition at line 149 of file planet_rnp.hh.

Referenced by jeod::RNPJ2000_rnp_j2000_default_data::initialize(), jeod::RNPMars_rnp_mars_default_data::initialize(), jeod::RNPJ2000::initialize(), jeod::RNPMars::initialize(), propagate_rnp(), jeod::RNPJ2000::update_axial_rotation(), jeod::RNPMars::update_axial_rotation(), jeod::RNPJ2000::update_rnp(), update_rnp(), and jeod::RNPMars::update_rnp().

8.6.6.7 rotation

`PlanetRotation*` jeod::PlanetRNP::rotation

Pointer to the rotation model.

trick_units(-)

Definition at line 144 of file planet_rnp.hh.

Referenced by propagate_rnp(), jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update_axial_rotation(), update_axial_rotation(), jeod::RNPMars::update_axial_rotation(), jeod::RNPJ2000::update_rnp(), update_rnp(), and jeod::RNPMars::update_rnp().

8.6.6.8 scratch_matrix

`double` jeod::PlanetRNP::scratch_matrix[3][3] [protected]

A transformation matrix used for intermediate math steps.

trick_units(-)

Definition at line 166 of file planet_rnp.hh.

Referenced by propagate_rnp().

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

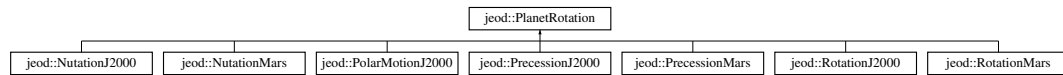
- [planet_rnp.hh](#)
- [planet_rnp.cc](#)

8.7 jeod::PlanetRotation Class Reference

The generic base class for all planet transformations.

```
#include <planet_rotation.hh>
```

Inheritance diagram for jeod::PlanetRotation:



Public Member Functions

- [PlanetRotation](#) ()
Constructor, initialize base level data.
- virtual [~PlanetRotation](#) ()
Destructor.
- virtual void [update_rotation](#) (void)

Updates the rotation of this particular object, based on the last time set in "set_time".
- virtual void [initialize](#) ([PlanetRotationInit](#) *init)

Initializes the invoking object from an initialization object.
- virtual void [update_time](#) (double time)
Update the time that the next update_rotation call will use.
- void [get_rotation](#) (double rot[3][3])
Copy the last calculated rotation to 'rot'.
- void [get_rotation_transpose](#) (double rot[3][3])
Same as get_rotation, but returns the transpose.

Data Fields

- double [rotation](#) [3][3]
The transformation matrix outputted by the model after update is called.
- double [current_time](#)
The current time the transformation matrix will be calculated from.

Protected Attributes

- double [RADTODEG](#)
Convert from radians to degrees.
- double [DEGTORAD](#)
Inverse of RADTODEG.
- double [DEGTOSEC](#)
Convert from degrees to arcseconds.
- double [SECTODEG](#)
Inverse of DEGTOSEC.
- double [JULIANDAYTOSEC](#)
Convert from julian day to seconds.
- double [SECTOJULIANDAY](#)
Inverse of JULIANDAYTOSEC.
- double [JULIANCENTTODAY](#)
Convert from julian centuries to julian days.
- double [DAYTOJULIANCENT](#)
Inverse of JULIANCENTTODAY.

Private Member Functions

- [PlanetRotation](#) & [operator=](#) (const [PlanetRotation](#) &rhs)
- [PlanetRotation](#) (const [PlanetRotation](#) &rhs)

Friends

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

8.7.1 Detailed Description

The generic base class for all planet transformations.

Definition at line 92 of file planet_rotation.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 PlanetRotation() [1/2]

```
jeod::PlanetRotation::PlanetRotation (
    void )
```

Constructor, initialize base level data.

Definition at line 53 of file planet_rotation.cc.

References [DAYTOJULIANCENT](#), [DEGTORAD](#), [DEGTOSEC](#), [JULIANCENTTODAY](#), [JULIANDAYTOSEC](#), [RADTODEG](#), [rotation](#), [SECTODEG](#), and [SECTOJULIANDAY](#).

8.7.2.2 ~PlanetRotation()

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

Destructor.

Definition at line 76 of file planet_rotation.cc.

8.7.2.3 PlanetRotation() [2/2]

```
jeod::PlanetRotation::PlanetRotation (
    const PlanetRotation & rhs ) [private]
```

8.7.3 Member Function Documentation

8.7.3.1 get_rotation()

```
void jeod::PlanetRotation::get_rotation (
    double rot[3][3] )
```

Copy the last calculated rotation to 'rot'.

Parameters

out	rot	Where the rotation matrix will be stored
-----	-----	--

Definition at line 98 of file planet_rotation.cc.

References rotation.

8.7.3.2 get_rotation_transpose()

```
void jeod::PlanetRotation::get_rotation_transpose (
    double rot[3][3] )
```

Same as get_rotation, but returns the transpose.

Parameters

out	rot	Where the transpose matrix will be stored
-----	-----	---

Definition at line 110 of file planet_rotation.cc.

References rotation.

8.7.3.3 initialize()

```
virtual void jeod::PlanetRotation::initialize (
    PlanetRotationInit * init ) [inline], [virtual]
```

Initializes the invoking object from an initialization object.

Parameters

in	init	The object that will initialize the invoking object.
----	------	--

Reimplemented in [jeod::NutationJ2000](#), and [jeod::PolarMotionJ2000](#).

Definition at line 190 of file planet_rotation.hh.

8.7.3.4 operator=()

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

8.7.3.5 update_rotation()

```
virtual void jeod::PlanetRotation::update_rotation (
    void ) [inline], [virtual]
```

Updates the rotation of this particular object, based on the last time set in "set_time".

Reimplemented in [jeod::NutationJ2000](#), [jeod::NutationMars](#), [jeod::PrecessionMars](#), [jeod::RotationMars](#), [jeod::RotationJ2000](#), [jeod::PolarMotionJ2000](#), and [jeod::PrecessionJ2000](#).

Definition at line 166 of file planet_rotation.hh.

Referenced by [jeod::PlanetRNP::update_axial_rotation\(\)](#), and [jeod::PlanetRNP::update_rnp\(\)](#).

8.7.3.6 update_time()

```
void jeod::PlanetRotation::update_time (
    double time ) [virtual]
```

Update the time that the next update_rotation call will use.

Parameters

in	<i>time</i>	Time. units and UTC/UT1/TAI etc are determined by the individual PlanetRotation model
----	-------------	---

Definition at line 86 of file planet_rotation.cc.

References [current_time](#).

Referenced by [jeod::RNPJ2000::update_axial_rotation\(\)](#), [jeod::RNPMars::update_axial_rotation\(\)](#), [jeod::RNPJ2000::update_rnp\(\)](#), and [jeod::RNPMars::update_rnp\(\)](#).

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__PlanetRotation

```
void init_attrjeod__PlanetRotation ( ) [friend]
```

8.7.4.2 InputProcessor

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

Definition at line 94 of file planet_rotation.hh.

8.7.5 Field Documentation

8.7.5.1 current_time

```
double jeod::PlanetRotation::current_time
```

The current time the transformation matrix will be calculated from.

Unitless so that models with different time scales can be used with one underlying model

Definition at line 109 of file planet_rotation.hh.

Referenced by jeod::PrecessionJ2000::update_rotation(), jeod::PolarMotionJ2000::update_rotation(), jeod::RotationJ2000::update_rotation(), jeod::RotationMars::update_rotation(), jeod::PrecessionMars::update_rotation(), jeod::NutationMars::update_rotation(), jeod::NutationJ2000::update_rotation(), and update_time().

8.7.5.2 DAYTOJULIANCENT

```
double jeod::PlanetRotation::DAYTOJULIANCENT [protected]
```

Inverse of JULIANCENTTODAY.

trick_units(-)

Definition at line 147 of file planet_rotation.hh.

Referenced by PlanetRotation().

8.7.5.3 DEGTORAD

```
double jeod::PlanetRotation::DEGTORAD [protected]
```

Inverse of RADTODEG.

trick_units(rad/degree)

Definition at line 123 of file planet_rotation.hh.

Referenced by PlanetRotation(), jeod::PrecessionJ2000::update_rotation(), jeod::RotationJ2000::update_rotation(), and jeod::NutationJ2000::update_rotation().

8.7.5.4 DEGTOSEC

```
double jeod::PlanetRotation::DEGTOSEC [protected]
```

Convert from degrees to arcseconds.

trick_units(arcsecond/degree)

Definition at line 127 of file planet_rotation.hh.

Referenced by PlanetRotation(), and jeod::PrecessionJ2000::update_rotation().

8.7.5.5 JULIANCENTTODAY

```
double jeod::PlanetRotation::JULIANCENTTODAY [protected]
```

Convert from julian centuries to julian days.

trick_units(-)

Definition at line 143 of file planet_rotation.hh.

Referenced by PlanetRotation().

8.7.5.6 JULIANDAYTOSEC

```
double jeod::PlanetRotation::JULIANDAYTOSEC [protected]
```

Convert from julian day to seconds.

trick_units(s/day)

Definition at line 135 of file planet_rotation.hh.

Referenced by PlanetRotation().

8.7.5.7 RADTODEG

```
double jeod::PlanetRotation::RADTODEG [protected]
```

Convert from radians to degrees.

trick_units(degree/rad)

Definition at line 119 of file planet_rotation.hh.

Referenced by PlanetRotation().

8.7.5.8 rotation

```
double jeod::PlanetRotation::rotation[3][3]
```

The transformation matrix outputted by the model after update is called.

trick_units(-)

Definition at line 102 of file planet_rotation.hh.

Referenced by `get_rotation()`, `get_rotation_transpose()`, `PlanetRotation()`, `jeod::PlanetRNP::propagate_rnp()`, `jeod::PlanetRNP::update_rnp()`, `jeod::PrecessionJ2000::update_rotation()`, `jeod::PolarMotionJ2000::update_rotation()`, `jeod::RotationJ2000::update_rotation()`, `jeod::RotationMars::update_rotation()`, `jeod::PrecessionMars::update_rotation()`, `jeod::NutationMars::update_rotation()`, and `jeod::NutationJ2000::update_rotation()`.

8.7.5.9 SECTODEG

```
double jeod::PlanetRotation::SECTODEG [protected]
```

Inverse of DEGTOSEC.

trick_units(degree/arcsecond)

Definition at line 131 of file planet_rotation.hh.

Referenced by `PlanetRotation()`.

8.7.5.10 SECTOJULIANDAY

```
double jeod::PlanetRotation::SECTOJULIANDAY [protected]
```

Inverse of JULIANDAYTOSEC.

trick_units(day/s)

Definition at line 139 of file planet_rotation.hh.

Referenced by `PlanetRotation()`.

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

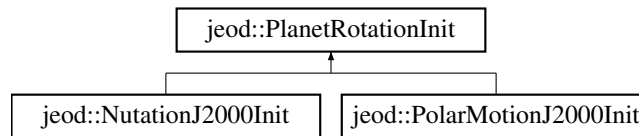
- [planet_rotation.hh](#)
- [planet_rotation.cc](#)

8.8 jeod::PlanetRotationInit Class Reference

The generic base class for all initializing classes for [PlanetRotation](#) derived classes.

```
#include <planet_rotation_init.hh>
```

Inheritance diagram for jeod::PlanetRotationInit:



Public Member Functions

- [PlanetRotationInit](#) ()
constructor
- virtual [~PlanetRotationInit](#) ()
destructor

Private Member Functions

- [PlanetRotationInit](#) & [operator=](#) (const [PlanetRotationInit](#) &rhs)
- [PlanetRotationInit](#) (const [PlanetRotationInit](#) &rhs)

Friends

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

8.8.1 Detailed Description

The generic base class for all initializing classes for [PlanetRotation](#) derived classes.

Definition at line 87 of file planet_rotation_init.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 PlanetRotationInit() [1/2]

```
jeod::PlanetRotationInit::PlanetRotationInit (
    void )
```

constructor

Definition at line 46 of file planet_rotation_init.cc.

8.8.2.2 ~PlanetRotationInit()

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

destructor

Definition at line 55 of file planet_rotation_init.cc.

8.8.2.3 PlanetRotationInit() [2/2]

```
jeod::PlanetRotationInit::PlanetRotationInit (
    const PlanetRotationInit & rhs ) [private]
```

8.8.3 Member Function Documentation

8.8.3.1 operator=()

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

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__PlanetRotationInit

```
void init_attrjeod__PlanetRotationInit ( ) [friend]
```

8.8.4.2 InputProcessor

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

Definition at line 89 of file planet_rotation_init.hh.

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

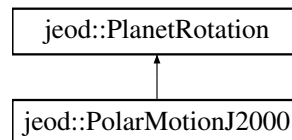
- [planet_rotation_init.hh](#)
- [planet_rotation_init.cc](#)

8.9 jeod::PolarMotionJ2000 Class Reference

Implements the polar motion portion of the J2000 RNP.

```
#include <polar_motion_j2000.hh>
```

Inheritance diagram for jeod::PolarMotionJ2000:



Public Member Functions

- [PolarMotionJ2000](#) ()
Default constructor.
- virtual [~PolarMotionJ2000](#) ()
Destructor.
- virtual void [update_rotation](#) ()
[PolarMotionJ2000](#) specific implementaiton of [update_rotation](#) from [PlanetRotation](#).
- virtual void [initialize](#) ([PlanetRotationInit](#) *init)
Initialize the coefficients of [PolarMotionJ2000](#).

Data Fields

- double [xp](#)
Current X Polar coordinate.
- double [yp](#)
Current Y Polar coordinate.
- double * [xp_tbl](#)
X Polar coordinate table.
- double * [yp_tbl](#)
Y Polar coordinate table.
- double * [polar_mjd](#)
Independent variable for the XY coordinate table.
- bool [override_table](#)
If true, do no table lookup and use the currently set xp and yp.
- unsigned int [last_table_index](#)
Size - 1 of [xp_tbl](#), [yp_tbl](#) and [polar_mjd](#) (last index)
- bool [warn_table](#)
Have we warned about being off the table end?

Private Member Functions

- [PolarMotionJ2000](#) & [operator=](#) (const [PolarMotionJ2000](#) &rhs)
- [PolarMotionJ2000](#) (const [PolarMotionJ2000](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__PolarMotionJ2000](#) ()

Additional Inherited Members

8.9.1 Detailed Description

Implements the polar motion portion of the J2000 RNP.

Definition at line 90 of file `polar_motion_j2000.hh`.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 `PolarMotionJ2000()` [1/2]

```
jeod::PolarMotionJ2000::PolarMotionJ2000 (
    void )
```

Default constructor.

Initializes all data.

Definition at line 63 of file `polar_motion_j2000.cc`.

8.9.2.2 `~PolarMotionJ2000()`

```
jeod::PolarMotionJ2000::~~PolarMotionJ2000 (
    void ) [virtual]
```

Destructor.

Definition at line 81 of file `polar_motion_j2000.cc`.

References `polar_mjd`, `xp_tbl`, and `yp_tbl`.

8.9.2.3 `PolarMotionJ2000()` [2/2]

```
jeod::PolarMotionJ2000::PolarMotionJ2000 (
    const PolarMotionJ2000 & rhs ) [private]
```

8.9.3 Member Function Documentation

8.9.3.1 `initialize()`

```
void jeod::PolarMotionJ2000::initialize (
    PlanetRotationInit * init ) [virtual]
```

Initialize the coefficients of [PolarMotionJ2000](#).

init must be of type [PolarMotionJ2000Init](#) or a fail message will occur.

Parameters

in	init	PolarMotionJ2000Init with needed coefficients
----	------	---

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

Definition at line 209 of file polar_motion_j2000.cc.

References [jeod::RNPMessages::initialization_error](#), [last_table_index](#), [jeod::PolarMotionJ2000Init::last_table_index](#), [override_table](#), [jeod::PolarMotionJ2000Init::override_table](#), [polar_mjd](#), [jeod::PolarMotionJ2000Init::polar_mjd](#), [xp](#), [jeod::PolarMotionJ2000Init::xp](#), [xp_tbl](#), [jeod::PolarMotionJ2000Init::xp_tbl](#), [yp](#), [jeod::PolarMotionJ2000Init::yp](#), [yp_tbl](#), and [jeod::PolarMotionJ2000Init::yp_tbl](#).

8.9.3.2 operator=()

```
PolarMotionJ2000& jeod::PolarMotionJ2000::operator= (
    const PolarMotionJ2000 & rhs ) [private]
```

8.9.3.3 update_rotation()

```
void jeod::PolarMotionJ2000::update_rotation (
    void ) [virtual]
```

[PolarMotionJ2000](#) specific implementaiton of [update_rotation](#) from [PlanetRotation](#).

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

Definition at line 104 of file polar_motion_j2000.cc.

References [jeod::PlanetRotation::current_time](#), [last_table_index](#), [override_table](#), [polar_mjd](#), [jeod::RNPMessages::polar_motion_table_warning](#), [jeod::PlanetRotation::rotation](#), [warn_table](#), [xp](#), [xp_tbl](#), [yp](#), and [yp_tbl](#).

8.9.4 Friends And Related Function Documentation

8.9.4.1 init_attrjeod__PolarMotionJ2000

```
void init_attrjeod__PolarMotionJ2000 ( ) [friend]
```

8.9.4.2 InputProcessor

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

Definition at line 92 of file polar_motion_j2000.hh.

8.9.5 Field Documentation

8.9.5.1 last_table_index

```
unsigned int jeod::PolarMotionJ2000::last_table_index
```

Size - 1 of xp_tbl, yp_tbl and polar_mjd (last index)

trick_units(count)

Definition at line 124 of file polar_motion_j2000.hh.

Referenced by initialize(), and update_rotation().

8.9.5.2 override_table

```
bool jeod::PolarMotionJ2000::override_table
```

If true, do no table lookup and use the currently set xp and yp.

trick_units(-)

Definition at line 120 of file polar_motion_j2000.hh.

Referenced by initialize(), and update_rotation().

8.9.5.3 polar_mjd

```
double* jeod::PolarMotionJ2000::polar_mjd
```

Independent variable for the XY coordinate table.

trick_units(-)

Definition at line 115 of file polar_motion_j2000.hh.

Referenced by initialize(), update_rotation(), and ~PolarMotionJ2000().

8.9.5.4 warn_table

```
bool jeod::PolarMotionJ2000::warn_table
```

Have we warned about being off the table end?

trick_units(—)

Definition at line 128 of file polar_motion_j2000.hh.

Referenced by update_rotation().

8.9.5.5 xp

```
double jeod::PolarMotionJ2000::xp
```

Current X Polar coordinate.

trick_units(rad)

Definition at line 99 of file polar_motion_j2000.hh.

Referenced by initialize(), and update_rotation().

8.9.5.6 xp_tbl

```
double* jeod::PolarMotionJ2000::xp_tbl
```

X Polar coordinate table.

trick_units(rad)

Definition at line 107 of file polar_motion_j2000.hh.

Referenced by initialize(), update_rotation(), and ~PolarMotionJ2000().

8.9.5.7 yp

```
double jeod::PolarMotionJ2000::yp
```

Current Y Polar coordinate.

trick_units(rad)

Definition at line 103 of file polar_motion_j2000.hh.

Referenced by initialize(), and update_rotation().

8.9.5.8 yp_tbl

```
double* jeod::PolarMotionJ2000::yp_tbl
```

Y Polar coordinate table.

trick_units(rad)

Definition at line 111 of file polar_motion_j2000.hh.

Referenced by initialize(), update_rotation(), and ~PolarMotionJ2000().

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

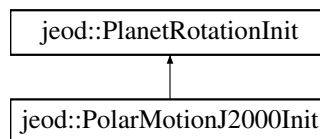
- [polar_motion_j2000.hh](#)
- [polar_motion_j2000.cc](#)

8.10 jeod::PolarMotionJ2000Init Class Reference

Initializes the [PolarMotionJ2000](#) object.

```
#include <polar_motion_j2000_init.hh>
```

Inheritance diagram for jeod::PolarMotionJ2000Init:



Public Member Functions

- [PolarMotionJ2000Init](#) ()
constructor.
- virtual [~PolarMotionJ2000Init](#) ()
Destructor.

Data Fields

- double [xp](#)
Current X Polar coordinate.
- double [yp](#)
Current Y Polar coordinate.
- double * [xp_tbl](#)
X Polar coordinate table.
- double * [yp_tbl](#)
Y Polar coordinate table.
- double * [polar_mjd](#)
Independent variable for the XY coordinate table.
- bool [override_table](#)
If true, do no table lookup and use the currently set xp and yp.
- unsigned int [last_table_index](#)
Size - 1 of xp_tbl, yp_tbl and polar_mjd (last index)

Private Member Functions

- [PolarMotionJ2000Init](#) & operator= (const [PolarMotionJ2000Init](#) &rhs)
- [PolarMotionJ2000Init](#) (const [PolarMotionJ2000Init](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__PolarMotionJ2000Init](#) ()

8.10.1 Detailed Description

Initializes the [PolarMotionJ2000](#) object.

Definition at line 90 of file polar_motion_j2000_init.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 PolarMotionJ2000Init() [1/2]

```
jeod::PolarMotionJ2000Init::PolarMotionJ2000Init (  
    void )
```

constructor.

Initializes all data.

Definition at line 54 of file polar_motion_j2000_init.cc.

8.10.2.2 ~PolarMotionJ2000Init()

```
jeod::PolarMotionJ2000Init::~~PolarMotionJ2000Init (  
    void ) [virtual]
```

Destructor.

Definition at line 71 of file polar_motion_j2000_init.cc.

References [polar_mjd](#), [xp_tbl](#), and [yp_tbl](#).

8.10.2.3 PolarMotionJ2000Init() [2/2]

```
jeod::PolarMotionJ2000Init::PolarMotionJ2000Init (
    const PolarMotionJ2000Init & rhs ) [private]
```

8.10.3 Member Function Documentation

8.10.3.1 operator=()

```
PolarMotionJ2000Init& jeod::PolarMotionJ2000Init::operator= (
    const PolarMotionJ2000Init & rhs ) [private]
```

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__PolarMotionJ2000Init

```
void init_attrjeod__PolarMotionJ2000Init ( ) [friend]
```

8.10.4.2 InputProcessor

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

Definition at line 92 of file polar_motion_j2000_init.hh.

8.10.5 Field Documentation

8.10.5.1 last_table_index

```
unsigned int jeod::PolarMotionJ2000Init::last_table_index
```

Size - 1 of xp_tbl, yp_tbl and polar_mjd (last index)

trick_units(count)

Definition at line 126 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize(), jeod::PolarMotionJ2000Init_xpyp↔_monthly_default_data::initialize(), and jeod::PolarMotionJ2000::initialize().

8.10.5.2 override_table

```
bool jeod::PolarMotionJ2000Init::override_table
```

If true, do no table lookup and use the currently set xp and yp.

trick_units(-)

Definition at line 122 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize(), jeod::PolarMotionJ2000Init_xpyp↔_monthly_default_data::initialize(), and jeod::PolarMotionJ2000::initialize().

8.10.5.3 polar_mjd

```
double* jeod::PolarMotionJ2000Init::polar_mjd
```

Independent variable for the XY coordinate table.

trick_units(-)

Definition at line 117 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize(), jeod::PolarMotionJ2000Init_xpyp↔_monthly_default_data::initialize(), jeod::PolarMotionJ2000::initialize(), and ~PolarMotionJ2000Init().

8.10.5.4 xp

```
double jeod::PolarMotionJ2000Init::xp
```

Current X Polar coordinate.

trick_units(rad)

Definition at line 101 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000::initialize().

8.10.5.5 xp_tbl

```
double* jeod::PolarMotionJ2000Init::xp_tbl
```

X Polar coordinate table.

trick_units(rad)

Definition at line 109 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize(), jeod::PolarMotionJ2000Init_xpyp↔_monthly_default_data::initialize(), jeod::PolarMotionJ2000::initialize(), and ~PolarMotionJ2000Init().

8.10.5.6 yp

```
double jeod::PolarMotionJ2000Init::yp
```

Current Y Polar coordinate.

trick_units(rad)

Definition at line 105 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000::initialize().

8.10.5.7 yp_tbl

```
double* jeod::PolarMotionJ2000Init::yp_tbl
```

Y Polar coordinate table.

trick_units(rad)

Definition at line 113 of file polar_motion_j2000_init.hh.

Referenced by jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize(), jeod::PolarMotionJ2000Init_xpyp↔_monthly_default_data::initialize(), jeod::PolarMotionJ2000::initialize(), and ~PolarMotionJ2000Init().

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

- [polar_motion_j2000_init.hh](#)
- [polar_motion_j2000_init.cc](#)

8.11 jeod::PolarMotionJ2000Init_xpyp_daily_default_data Class Reference

```
#include <xpyp_daily.hh>
```

Public Member Functions

- void [initialize](#) ([PolarMotionJ2000Init](#) *)

8.11.1 Detailed Description

Definition at line 54 of file xpyp_daily.hh.

8.11.2 Member Function Documentation

8.11.2.1 initialize()

```
void jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize (
    PolarMotionJ2000Init * PolarMotionJ2000Init_ptr )
```

Definition at line 38 of file xpyp_daily.cc.

References [jeod::PolarMotionJ2000Init::last_table_index](#), [jeod::PolarMotionJ2000Init::override_table](#), [jeod::PolarMotionJ2000Init::polar_mjd](#), [jeod::PolarMotionJ2000Init::xp_tbl](#), and [jeod::PolarMotionJ2000Init::yp_tbl](#).

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

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

8.12 jeod::PolarMotionJ2000Init_xpyp_monthly_default_data Class Reference

```
#include <xpyp_monthly.hh>
```

Public Member Functions

- void [initialize](#) ([PolarMotionJ2000Init](#) *)

8.12.1 Detailed Description

Definition at line 55 of file xpyp_monthly.hh.

8.12.2 Member Function Documentation

8.12.2.1 initialize()

```
void jeod::PolarMotionJ2000Init_xpyp_monthly_default_data::initialize (
    PolarMotionJ2000Init * PolarMotionJ2000Init_ptr )
```

Definition at line 38 of file xpyp_monthly.cc.

References [jeod::PolarMotionJ2000Init::last_table_index](#), [jeod::PolarMotionJ2000Init::override_table](#), [jeod::PolarMotionJ2000Init::polar_mjd](#), [jeod::PolarMotionJ2000Init::xp_tbl](#), and [jeod::PolarMotionJ2000Init::yp_tbl](#).

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

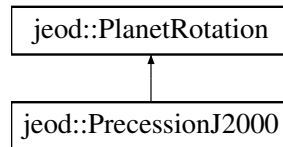
- [xpyp_monthly.hh](#)
- [xpyp_monthly.cc](#)

8.13 jeod::PrecessionJ2000 Class Reference

Implements the precession portion of the J2000 RNP.

```
#include <precession_j2000.hh>
```

Inheritance diagram for jeod::PrecessionJ2000:



Public Member Functions

- [PrecessionJ2000](#) ()
constructor
- virtual [~PrecessionJ2000](#) ()
destructor
- virtual void [update_rotation](#) ()
Precession J2000 specific implementation of update_rotation, inherited from Planetrotation.

Private Member Functions

- [PrecessionJ2000](#) & [operator=](#) (const [PrecessionJ2000](#) &rhs)
- [PrecessionJ2000](#) (const [PrecessionJ2000](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__PrecessionJ2000](#) ()

Additional Inherited Members

8.13.1 Detailed Description

Implements the precession portion of the J2000 RNP.

Definition at line 89 of file `precession_j2000.hh`.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 PrecessionJ2000() [1/2]

```
jeod::PrecessionJ2000::PrecessionJ2000 (
    void )
```

constructor

Definition at line 47 of file precession_j2000.cc.

8.13.2.2 ~PrecessionJ2000()

```
jeod::PrecessionJ2000::~~PrecessionJ2000 (
    void ) [virtual]
```

destructor

Definition at line 55 of file precession_j2000.cc.

8.13.2.3 PrecessionJ2000() [2/2]

```
jeod::PrecessionJ2000::PrecessionJ2000 (
    const PrecessionJ2000 & rhs ) [private]
```

8.13.3 Member Function Documentation**8.13.3.1 operator=()**

```
PrecessionJ2000& jeod::PrecessionJ2000::operator= (
    const PrecessionJ2000 & rhs ) [private]
```

8.13.3.2 update_rotation()

```
void jeod::PrecessionJ2000::update_rotation (
    void ) [virtual]
```

Precession J2000 specific implementation of update_rotation, inherited from Planetrotation.

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

Definition at line 65 of file precession_j2000.cc.

References [jeod::PlanetRotation::current_time](#), [jeod::PlanetRotation::DEGTORAD](#), [jeod::PlanetRotation::DEGT↔OSEC](#), and [jeod::PlanetRotation::rotation](#).

8.13.4 Friends And Related Function Documentation

8.13.4.1 init_attrjeod__PrecessionJ2000

```
void init_attrjeod__PrecessionJ2000 ( ) [friend]
```

8.13.4.2 InputProcessor

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

Definition at line 91 of file `precession_j2000.hh`.

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

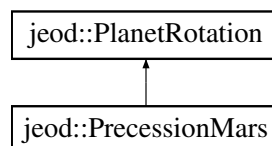
- [precession_j2000.hh](#)
- [precession_j2000.cc](#)

8.14 jeod::PrecessionMars Class Reference

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

```
#include <precession_mars.hh>
```

Inheritance diagram for `jeod::PrecessionMars`:



Public Member Functions

- [PrecessionMars](#) ()
Default constructor.
- virtual [~PrecessionMars](#) ()
Destructor.
- virtual void [update_rotation](#) ()
[PrecessionMars](#) specific implementation of `update_rotation`, to calculate precession.
- void [compute_fixed_matrices](#) ()
Calculate constant rotation matrices resulting from N and J .

Data Fields

- [NutationMars](#) * [nututation](#)
Pointer to the Mars nutation object, used to access nutation in longitude information.
- double [psi_at_j2000](#)
The (constant) precession angle of Mars at the J2000 epoch.
- double [psi_dot](#)
The (constant) simple precession rate of Mars.
- double [psi_precess](#)
The most recent calculated value of the precession angle for Mars, measured since J2000 epoch.
- double [N](#)
The (constant) angle from the J2000 vernal equinox to the node of the Mars mean orbit and ICRF x-y plane.
- double [J](#)
The (constant) inclination of the Mars mean orbit relative to the ICRF x-y plane.

Private Member Functions

- [PrecessionMars](#) & [operator=](#) (const [PrecessionMars](#) &rhs)
- [PrecessionMars](#) (const [PrecessionMars](#) &rhs)

Private Attributes

- double [NJ_matrix](#) [3][3]
The (constant) rotation matrix calculated from N and J.

Friends

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

Additional Inherited Members

8.14.1 Detailed Description

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

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

8.14.2 Constructor & Destructor Documentation

8.14.2.1 PrecessionMars() [1/2]

```
jeod::PrecessionMars::PrecessionMars (  
    void )
```

Default constructor.

Definition at line 61 of file precession_mars.cc.

References NJ_matrix.

8.14.2.2 ~PrecessionMars()

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

Destructor.

Definition at line 78 of file precession_mars.cc.

8.14.2.3 PrecessionMars() [2/2]

```
jeod::PrecessionMars::PrecessionMars (  
    const PrecessionMars & rhs ) [private]
```

8.14.3 Member Function Documentation

8.14.3.1 compute_fixed_matrices()

```
void jeod::PrecessionMars::compute_fixed_matrices (  
    void )
```

Calculate constant rotation matrices resulting from N and J.

Definition at line 140 of file precession_mars.cc.

References J, N, and NJ_matrix.

Referenced by jeod::RNPMars::initialize().

8.14.3.2 operator=()

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

8.14.3.3 update_rotation()

```
void jeod::PrecessionMars::update_rotation (
    void ) [virtual]
```

[PrecessionMars](#) specific implementation of update_rotation, to calculate precession.

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

Definition at line 90 of file precession_mars.cc.

References [jeod::PlanetRotation::current_time](#), [NJ_matrix](#), [nutaton](#), [jeod::NutationMars::nutaton_in_longitude](#), [psi_at_j2000](#), [psi_dot](#), [psi_precess](#), [jeod::PlanetRotation::rotation](#), and [jeod::RNPMessages::setup_error](#).

8.14.4 Friends And Related Function Documentation

8.14.4.1 init_attrjeod__PrecessionMars

```
void init_attrjeod__PrecessionMars ( ) [friend]
```

8.14.4.2 InputProcessor

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

Definition at line 98 of file precession_mars.hh.

8.14.5 Field Documentation

8.14.5.1 J

```
double jeod::PrecessionMars::J
```

The (constant) inclination of the Mars mean orbit relative to the ICRF x-y plane.

trick_units(rad)

Definition at line 135 of file precession_mars.hh.

Referenced by compute_fixed_matrices(), and jeod::RNPMars_rnp_mars_default_data::initialize().

8.14.5.2 N

```
double jeod::PrecessionMars::N
```

The (constant) angle from the J2000 vernal equinox to the node of the Mars mean orbit and ICRF x-y plane.

trick_units(rad)

Definition at line 129 of file precession_mars.hh.

Referenced by compute_fixed_matrices(), and jeod::RNPMars_rnp_mars_default_data::initialize().

8.14.5.3 NJ_matrix

```
double jeod::PrecessionMars::NJ_matrix[3][3] [private]
```

The (constant) rotation matrix calculated from N and J.

trick_units(-)

Definition at line 144 of file precession_mars.hh.

Referenced by compute_fixed_matrices(), PrecessionMars(), and update_rotation().

8.14.5.4 nutation

```
NutationMars* jeod::PrecessionMars::nutation
```

Pointer to the Mars nutation object, used to access nutation in longitude information.

Will be NULL (automatically) if anything but Full_Term_RNP is set in the [RNPMars](#) object.trick_units(-)

Definition at line 107 of file precession_mars.hh.

Referenced by jeod::RNPMars::initialize(), and update_rotation().

8.14.5.5 psi_at_j2000

```
double jeod::PrecessionMars::psi_at_j2000
```

The (constant) precession angle of Mars at the J2000 epoch.

trick_units(rad)

Definition at line 112 of file precession_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.14.5.6 psi_dot

```
double jeod::PrecessionMars::psi_dot
```

The (constant) simple precession rate of Mars.

trick_units(rad/s)

Definition at line 117 of file precession_mars.hh.

Referenced by jeod::RNPMars_rnp_mars_default_data::initialize(), and update_rotation().

8.14.5.7 psi_precess

```
double jeod::PrecessionMars::psi_precess
```

The most recent calculated value of the precession angle for Mars, measured since J2000 epoch.

trick_units(rad)

Definition at line 123 of file precession_mars.hh.

Referenced by update_rotation().

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

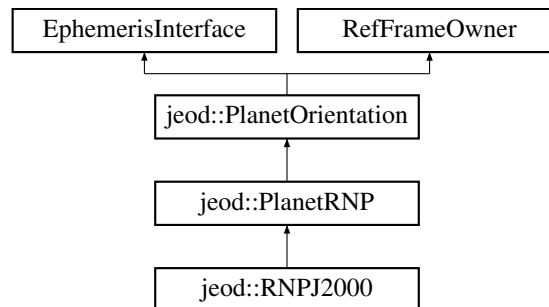
- [precession_mars.hh](#)
- [precession_mars.cc](#)

8.15 jeod::RNPJ2000 Class Reference

Implements the J2000 RNP model using the generic RNP framework.

```
#include <rnj2000.hh>
```

Inheritance diagram for jeod::RNPJ2000:



Public Member Functions

- [RNPJ2000](#) ()
default constructor.
- virtual [~RNPJ2000](#) ()
Destructor.
- void [initialize](#) (DynManager &manager)
Apply the various options, including fidelity of RNP and if polar motion is enabled or not, then call the base class initializer.
- void [update_rnp](#) (TimeTT &time_tt, TimeGMST &time_gmst, TimeUT1 &time_ut1)
Updates the complete RNP and supplies it to the Dynamics Manager.
- void [update_axial_rotation](#) (TimeGMST &time_gmst)
Updates the axial rotation portion of RNP and supplies the entire RNP to the Dynamics Manager.
- virtual double [timestamp](#) () const
- virtual const char * [get_name](#) () const
A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.
- virtual void [ephem_update](#) ()

Data Fields

- [RotationJ2000 RJ2000](#)
Earth J2000 rotation model.
- [NutationJ2000 NJ2000](#)
Earth J2000 nutation model.
- [PrecessionJ2000 PJ2000](#)
Earth J2000 precession model.
- [PolarMotionJ2000 PMJ2000](#)
Earth J2000 polar motion model.
- std::string [internal_name](#)
The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get_name".

Private Member Functions

- void [get_dyn_time_ptr](#) (TimeGMST &gmst)
- [RNPJ2000](#) & [operator=](#) (const [RNPJ2000](#) &rhs)
- [RNPJ2000](#) (const [RNPJ2000](#) &rhs)

Private Attributes

- TimeGMST * [gmst_ptr](#)
Pointer to the TimeGMST used to update this object when ephemeris_update is invoked.
- TimeDyn * [time_dyn_ptr](#)
Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.
- double [last_updated_time_full](#)
The last update time, when updated through update_rnp, for the RNP, referencing TimeDyn.seconds .
- bool [never_updated_full](#)
Indicates that last_updated_time_full has never been populated, and that the update must be done regardless of given time.
- double [last_updated_time_rotational](#)
The last update time, when updated through update_axial_rotation, referencing TimeDyn.seconds .
- bool [never_updated_rotational](#)
Indicates that last_updated_time_rotational has never been populated, and that the update must be done regardless of given time.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__RNPJ2000](#) ()

Additional Inherited Members

8.15.1 Detailed Description

Implements the J2000 RNP model using the generic RNP framework.

Definition at line 101 of file include/rnp_j2000.hh.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 RNPJ2000() [1/2]

```
jeod::RNPJ2000::RNPJ2000 (
    void )
```

default constructor.

Initialize all data

Definition at line 73 of file rnp_j2000.cc.

References [NJ2000](#), [jeod::PlanetRNP::nutation](#), [PJ2000](#), [PMJ2000](#), [jeod::PlanetRNP::polar_motion](#), [jeod::PlanetRNP::precession](#), [RJ2000](#), and [jeod::PlanetRNP::rotation](#).

8.15.2.2 ~RNPJ2000()

```
jeod::RNPJ2000::~~RNPJ2000 (
    void ) [virtual]
```

Destructor.

Definition at line 95 of file rnp_j2000.cc.

8.15.2.3 RNPJ2000() [2/2]

```
jeod::RNPJ2000::~RNPJ2000 (
    const RNPJ2000 & rhs ) [private]
```

8.15.3 Member Function Documentation

8.15.3.1 ephem_update()

```
void jeod::RNPJ2000::ephem_update ( ) [virtual]
```

Definition at line 325 of file rnp_j2000.cc.

References [jeod::PlanetOrientation::active](#), [gmst_ptr](#), [jeod::PlanetOrientation::orient_interface](#), [jeod::RNPJ2000::Messages::setup_error](#), and [jeod::PlanetRNP::update_axial_rotation\(\)](#).

8.15.3.2 get_dyn_time_ptr()

```
void jeod::RNPJ2000::get_dyn_time_ptr (
    TimeGMST & gmst ) [private]
```

Definition at line 346 of file rnp_j2000.cc.

References [jeod::RNPMessages::setup_error](#), and [time_dyn_ptr](#).

Referenced by [update_axial_rotation\(\)](#), and [update_rnp\(\)](#).

8.15.3.3 get_name()

```
const char * jeod::RNPJ2000::get_name ( ) const [virtual]
```

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

Returns

Planet name.

Implements [jeod::PlanetRNP](#).

Definition at line 312 of file rnp_j2000.cc.

References [internal_name](#).

8.15.3.4 initialize()

```
void jeod::RNPJ2000::initialize (
    DynManager & dyn_manager ) [virtual]
```

Apply the various options, including fidelity of RNP and if polar motion is enabled or not, then call the base class initializer.

Note that this function will still run even if the active flag is set to false. Additionally, note that this function does not update the rotation matrix between inertial and planet fixed. It only sets up the object to work with the DynManager. `update_rnp` must be called to update the RNP matrix. This function will also cache of TimeGMST from the given TimeManager

Parameters

<code>in, out</code>	<code>dyn_manager</code>	DynManager where the planet attitude to be updated is contained
----------------------	--------------------------	---

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

Definition at line 114 of file rnp_j2000.cc.

References [jeod::PlanetRNP::ConstantNP](#), [jeod::PlanetRNP::FullRNP](#), [jeod::PlanetOrientation::initialize\(\)](#), [NJ2000](#), [jeod::RotationJ2000::nutation](#), [jeod::PlanetOrientation::planet_omega](#), [jeod::RotationJ2000::planet_rotational_↵](#) velocity, [RJ2000](#), [jeod::PlanetRNP::rnp_type](#), [jeod::PlanetRNP::RotationOnly](#), and [jeod::RotationJ2000::use_full↵](#) _rnp.

8.15.3.5 operator=()

```
RNPJ2000& jeod::RNPJ2000::operator= (
    const RNPJ2000 & rhs ) [private]
```

8.15.3.6 timestamp()

```
double jeod::RNPJ2000::timestamp ( ) const [virtual]
```

Definition at line 308 of file rnp_j2000.cc.

References `last_updated_time_rotational`.

8.15.3.7 update_axial_rotation()

```
void jeod::RNPJ2000::update_axial_rotation (
    TimeGMST & time_gmst )
```

Updates the axial rotation portion of RNP and supplies the entire RNP to the Dynamics Manager.

Parameters

in	<i>time_gmst</i>	current time in the GMST time standard Units: The
----	------------------	--

Definition at line 256 of file rnp_j2000.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `last_updated_time_rotational`, `never_updated_rotational`, `jeod::PlanetOrientation::planet`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::rotation`, `time_dyn_ptr`, `jeod::PlanetRNP::update_axial_rotation()`, and `jeod::PlanetRotation::update_time()`.

8.15.3.8 update_rnp()

```
void jeod::RNPJ2000::update_rnp (
    TimeTT & time_tt,
    TimeGMST & time_gmst,
    TimeUT1 & time_ut1 )
```

Updates the complete RNP and supplies it to the Dynamics Manager.

Parameters

in	<i>time_tt</i>	The current time in the TT time standard
in	<i>time_gmst</i>	current time in the GMST time standard Units: The
in	<i>time_ut1</i>	current time in the UT1 time standard Units: The

Definition at line 157 of file rnp_j2000.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::enable_polar`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `gmst_ptr`, `last_updated_time_full`, `last_updated_time_rotational`, `never_updated_full`, `never_updated_rotational`, and `update_rnp()`.

_rotational, jeod::PlanetRNP::nututation, jeod::PlanetOrientation::planet, jeod::PlanetRNP::polar_motion, jeod::PlanetRNP::precession, jeod::PlanetRNP::rnp_type, jeod::PlanetRNP::rotation, time_dyn_ptr, jeod::PlanetRNP::update_rnp(), and jeod::PlanetRotation::update_time().

8.15.4 Friends And Related Function Documentation

8.15.4.1 init_attrjeod__RNPJ2000

```
void init_attrjeod__RNPJ2000 ( ) [friend]
```

8.15.4.2 InputProcessor

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

Definition at line 103 of file include/rnp_j2000.hh.

8.15.5 Field Documentation

8.15.5.1 gmst_ptr

```
TimeGMST* jeod::RNPJ2000::gmst_ptr [private]
```

Pointer to the TimeGMST used to update this object when ephemeris_update is invoked.

trick_units(-)

Definition at line 182 of file include/rnp_j2000.hh.

Referenced by ephemeris_update(), and update_rnp().

8.15.5.2 internal_name

```
std::string jeod::RNPJ2000::internal_name
```

The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get_name".

trick_units(-)

Definition at line 167 of file include/rnp_j2000.hh.

Referenced by get_name().

8.15.5.3 last_updated_time_full

```
double jeod::RNPJ2000::last_updated_time_full [private]
```

The last update time, when updated through `update_rnp`, for the RNP, referencing `TimeDyn.seconds` .

If the time from `time_dyn_ptr` is the same as this update time, then the RNP will not be updated. This is to prevent unnecessary updating.`trick_units(s)`

Definition at line 197 of file `include/rnp_j2000.hh`.

Referenced by `update_rnp()`.

8.15.5.4 last_updated_time_rotational

```
double jeod::RNPJ2000::last_updated_time_rotational [private]
```

The last update time, when updated through `update_axial_rotation`, referencing `TimeDyn.seconds` .

If the time from `time_dyn_ptr` is the same as this update time, then the R component of RNP will not be updated. This is to prevent unnecessary updating.`trick_units(s)`

Definition at line 212 of file `include/rnp_j2000.hh`.

Referenced by `timestamp()`, `update_axial_rotation()`, and `update_rnp()`.

8.15.5.5 never_updated_full

```
bool jeod::RNPJ2000::never_updated_full [private]
```

Indicates that `last_updated_time_full` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 203 of file `include/rnp_j2000.hh`.

Referenced by `update_rnp()`.

8.15.5.6 never_updated_rotational

```
bool jeod::RNPJ2000::never_updated_rotational [private]
```

Indicates that `last_updated_time_rotational` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 218 of file `include/rnp_j2000.hh`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

8.15.5.7 NJ2000

`NutationJ2000` `jeod::RNPJ2000::NJ2000`

Earth J2000 nutation model.

`trick_units(-)`

Definition at line 116 of file `include/rnp_j2000.hh`.

Referenced by `initialize()`, and `RNPJ2000()`.

8.15.5.8 PJ2000

`PrecessionJ2000` `jeod::RNPJ2000::PJ2000`

Earth J2000 precession model.

`trick_units(-)`

Definition at line 120 of file `include/rnp_j2000.hh`.

Referenced by `RNPJ2000()`.

8.15.5.9 PMJ2000

`PolarMotionJ2000` `jeod::RNPJ2000::PMJ2000`

Earth J2000 polar motion model.

`trick_units(-)`

Definition at line 124 of file `include/rnp_j2000.hh`.

Referenced by `RNPJ2000()`.

8.15.5.10 RJ2000

`RotationJ2000` `jeod::RNPJ2000::RJ2000`

Earth J2000 rotation model.

`trick_units(-)`

Definition at line 112 of file `include/rnp_j2000.hh`.

Referenced by `initialize()`, and `RNPJ2000()`.

8.15.5.11 time_dyn_ptr

```
TimeDyn* jeod::RNPJ2000::time_dyn_ptr [private]
```

Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.

trick_units(-)

Definition at line 188 of file include/rnp_j2000.hh.

Referenced by get_dyn_time_ptr(), update_axial_rotation(), and update_rnp().

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

- [include/rnp_j2000.hh](#)
- [rnp_j2000.cc](#)

8.16 jeod::RNPJ2000_rnp_j2000_default_data Class Reference

```
#include <rnp_j2000.hh>
```

Public Member Functions

- void [initialize](#) (RNPJ2000 *)

8.16.1 Detailed Description

Definition at line 54 of file data/include/rnp_j2000.hh.

8.16.2 Member Function Documentation

8.16.2.1 initialize()

```
void jeod::RNPJ2000_rnp_j2000_default_data::initialize (
    RNPJ2000 * RNPJ2000_ptr )
```

Definition at line 38 of file data_rnp_j2000.cc.

References [jeod::PlanetRNP::enable_polar](#), [jeod::PlanetRNP::FullRNP](#), [jeod::PlanetOrientation::name](#), [jeod::PlanetOrientation::planet_omega](#), and [jeod::PlanetRNP::rnp_type](#).

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

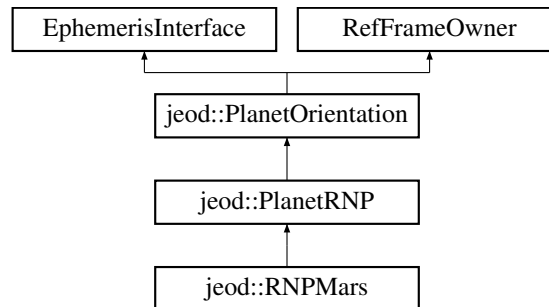
- [data/include/rnp_j2000.hh](#)
- [data_rnp_j2000.cc](#)

8.17 jeod::RNPMars Class Reference

Implements the "Pathfinder" Mars RNP model using the generic RNP framework.

```
#include <rnpmars.hh>
```

Inheritance diagram for jeod::RNPMars:



Public Member Functions

- [RNPMars \(\)](#)
Default constructor.
- virtual [~RNPMars \(\)](#)
Destructor.
- void [initialize](#) (DynManager &manager)
Perform setup of the object, then call the base class initializer.
- void [update_rnp](#) (TimeTT &time_tt)
Update the complete RNP and supply it to the Dynamics Manager.
- void [update_axial_rotation](#) (TimeTT &time_tt)
Update rotation portion of RNP and supply RNP to dynamics manager.
- virtual double [timestamp](#) () const
Return the last time at which the RNP was updated.
- virtual const char * [get_name](#) () const
Return the internal name of the object.
- virtual void [ephem_update](#) ()
Define 'alias' to call axial update function, for polymorphism.

Data Fields

- [RotationMars RMars](#)
"Pathfinder" Mars rotation model.
- [NutationMars NMars](#)
"Pathfinder" Mars nutation model.
- [PrecessionMars PMars](#)
"Pathfinder" Mars precession model.
- std::string [internal_name](#)
The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get_name".

Private Member Functions

- void [get_dyn_time_ptr](#) (TimeTT &time_tt)
Get simulation time via a two-step pointer trail.
- [RNPMars](#) & [operator=](#) (const [RNPMars](#) &rhs)
- [RNPMars](#) (const [RNPMars](#) &rhs)

Private Attributes

- TimeTT * [tt_ptr](#)
Pointer to the TimeTT used to update this object when `ephem_update` is invoked.
- TimeDyn * [time_dyn_ptr](#)
Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.
- double [last_updated_time_full](#)
The last update time for the RNP, when updated through `update_rnp`.
- bool [never_updated_full](#)
Indicates that `last_updated_time_full` has never been populated, and that the update must be done regardless of given time.
- double [last_updated_time_rotational](#)
The last rotational update time, when updated through `update_axial_rotation`, referencing `TimeDyn.seconds`.
- bool [never_updated_rotational](#)
Indicates that `last_updated_time_rotational` has never been populated, and that the update must be done regardless of given time.

Friends

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

Additional Inherited Members

8.17.1 Detailed Description

Implements the "Pathfinder" Mars RNP model using the generic RNP framework.

Definition at line 101 of file `include/rnp_mars.hh`.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 [RNPMars\(\)](#) [1/2]

```
jeod::RNPMars::RNPMars (
    void )
```

Default constructor.

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

References `jeod::PlanetRNP::enable_polar`, `NMars`, `jeod::PlanetRNP::nututation`, `PMars`, `jeod::PlanetRNP::polar_↔motion`, `jeod::PlanetRNP::precession`, `RMars`, and `jeod::PlanetRNP::rotation`.

8.17.2.2 ~RNPMars()

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

Destructor.

Definition at line 98 of file rnp_mars.cc.

8.17.2.3 RNPMars() [2/2]

```
jeod::RNPMars::~RNPMars (
    const RNPMars & rhs ) [private]
```

8.17.3 Member Function Documentation

8.17.3.1 ephem_update()

```
void jeod::RNPMars::ephem_update ( ) [virtual]
```

Define 'alias' to call axial update function, for polymorphism.

Definition at line 277 of file rnp_mars.cc.

References [jeod::PlanetOrientation::active](#), [jeod::PlanetOrientation::orient_interface](#), [jeod::RNPMessages::setup_error](#), [tt_ptr](#), and [jeod::PlanetRNP::update_axial_rotation\(\)](#).

8.17.3.2 get_dyn_time_ptr()

```
void jeod::RNPMars::get_dyn_time_ptr (
    TimeTT & time_tt ) [private]
```

Get simulation time via a two-step pointer trail.

Parameters

in	<i>time_tt</i>	Current TT time
----	----------------	-----------------

Definition at line 302 of file rnp_mars.cc.

References [jeod::RNPMessages::setup_error](#), and [time_dyn_ptr](#).

Referenced by `update_axial_rotation()`, and `update_rnp()`.

8.17.3.3 `get_name()`

```
const char * jeod::RNPMars::get_name ( ) const [virtual]
```

Return the internal name of the object.

Implements [jeod::PlanetRNP](#).

Definition at line 268 of file `rnp_mars.cc`.

References `internal_name`.

8.17.3.4 `initialize()`

```
void jeod::RNPMars::initialize (
    DynManager & dyn_manager ) [virtual]
```

Perform setup of the object, then call the base class initializer.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Ref to dynamics manager
----------------------	--------------------------	-------------------------

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

Definition at line 107 of file `rnp_mars.cc`.

References `jeod::PrecessionMars::compute_fixed_matrices()`, `jeod::PlanetRNP::ConstantNP`, `jeod::PlanetRNP::FullRNP`, `jeod::PlanetOrientation::initialize()`, `NMars`, `jeod::PrecessionMars::nutaton`, `jeod::RotationMars::nutaton`, `jeod::PlanetOrientation::planet_omega`, `jeod::RotationMars::planet_rotational_velocity`, `PMars`, `RMars`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::RotationOnly`, and `jeod::RotationMars::use_full_rnp`.

8.17.3.5 `operator=()`

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

8.17.3.6 timestamp()

```
double jeod::RNPMars::timestamp ( ) const [virtual]
```

Return the last time at which the RNP was updated.

Definition at line 259 of file rnp_mars.cc.

References `last_updated_time_rotational`.

8.17.3.7 update_axial_rotation()

```
void jeod::RNPMars::update_axial_rotation (
    TimeTT & time_tt )
```

Update rotation portion of RNP and supply RNP to dynamics manager.

Parameters

in	<i>time</i> ↔ _tt	Current TT time
----	----------------------	-----------------

Definition at line 205 of file rnp_mars.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `last_updated_time`↔
`rotational`, `never_updated_rotational`, `jeod::PlanetOrientation::planet`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP`↔
`P::rotation`, `time_dyn_ptr`, `jeod::PlanetRNP::update_axial_rotation()`, and `jeod::PlanetRotation::update_time()`.

8.17.3.8 update_rnp()

```
void jeod::RNPMars::update_rnp (
    TimeTT & time_tt )
```

Update the complete RNP and supply it to the Dynamics Manager.

Parameters

in	<i>time</i> ↔ _tt	Current Terrestrial Time
----	----------------------	--------------------------

Definition at line 142 of file rnp_mars.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `last_updated_time_full`,
`last_updated_time_rotational`, `never_updated_full`, `never_updated_rotational`, `jeod::PlanetRNP::nutation`, `jeod`↔
`::PlanetOrientation::planet`, `jeod::PlanetRNP::precession`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::rotation`,
`time_dyn_ptr`, `tt_ptr`, `jeod::PlanetRNP::update_rnp()`, and `jeod::PlanetRotation::update_time()`.

8.17.4 Friends And Related Function Documentation

8.17.4.1 init_attrjeod__RNPMars

```
void init_attrjeod__RNPMars ( ) [friend]
```

8.17.4.2 InputProcessor

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

Definition at line 103 of file include/rnp_mars.hh.

8.17.5 Field Documentation

8.17.5.1 internal_name

```
std::string jeod::RNPMars::internal_name
```

The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get_name".

trick_units(-)

Definition at line 126 of file include/rnp_mars.hh.

Referenced by get_name().

8.17.5.2 last_updated_time_full

```
double jeod::RNPMars::last_updated_time_full [private]
```

The last update time for the RNP, when updated through update_rnp.

References TimeDyn.seconds. If the time from time_dyn_ptr is the same as this update time, then the RNP will not be updated; prevents unnecessary updating.trick_units(s)

Definition at line 147 of file include/rnp_mars.hh.

Referenced by update_rnp().

8.17.5.3 last_updated_time_rotational

```
double jeod::RNPMars::last_updated_time_rotational [private]
```

The last rotational update time, when updated through `update_axial_rotation`, referencing `TimeDyn.seconds`.

If the time from `time_dyn_ptr` matches this update time, then the R component of the RNP will not be updated; prevents unnecessary updating. `trick_units(s)`

Definition at line 161 of file `include/rnp_mars.hh`.

Referenced by `timestamp()`, `update_axial_rotation()`, and `update_rnp()`.

8.17.5.4 never_updated_full

```
bool jeod::RNPMars::never_updated_full [private]
```

Indicates that `last_updated_time_full` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 153 of file `include/rnp_mars.hh`.

Referenced by `update_rnp()`.

8.17.5.5 never_updated_rotational

```
bool jeod::RNPMars::never_updated_rotational [private]
```

Indicates that `last_updated_time_rotational` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 167 of file `include/rnp_mars.hh`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

8.17.5.6 NMars

```
NutationMars jeod::RNPMars::NMars
```

"Pathfinder" Mars nutation model.

`trick_units(-)`

Definition at line 115 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

8.17.5.7 PMars

[PrecessionMars](#) `jeod::RNPMars::PMars`

"Pathfinder" Mars precession model.

`trick_units(-)`

Definition at line 120 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

8.17.5.8 RMars

[RotationMars](#) `jeod::RNPMars::RMars`

"Pathfinder" Mars rotation model.

`trick_units(-)`

Definition at line 110 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

8.17.5.9 time_dyn_ptr

`TimeDyn* jeod::RNPMars::time_dyn_ptr [private]`

Pointer to the `TimeDyn` object, used to time stamp the reference frame when it is being updated.

`trick_units(-)`

Definition at line 139 of file `include/rnp_mars.hh`.

Referenced by `get_dyn_time_ptr()`, `update_axial_rotation()`, and `update_rnp()`.

8.17.5.10 tt_ptr

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

Pointer to the `TimeTT` used to update this object when `ephem_update` is invoked.

`trick_units(-)`

Definition at line 133 of file `include/rnp_mars.hh`.

Referenced by `ephem_update()`, and `update_rnp()`.

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

- [include/rnp_mars.hh](#)
- [rnp_mars.cc](#)

8.18 jeod::RNPMars_rnp_mars_default_data Class Reference

```
#include <rnp_mars.hh>
```

Public Member Functions

- void [initialize](#) (RNPMars *)

8.18.1 Detailed Description

Definition at line 54 of file data/include/rnp_mars.hh.

8.18.2 Member Function Documentation

8.18.2.1 initialize()

```
void jeod::RNPMars_rnp_mars_default_data::initialize (
    RNPMars * RNPmars_ptr )
```

Definition at line 44 of file data_rnp_mars.cc.

References [jeod::PlanetRNP::enable_polar](#), [jeod::PlanetRNP::FullRNP](#), [jeod::NutationMars::l_at_j2000](#), [jeod::NutationMars::l_dot](#), [jeod::NutationMars::l_m_orig](#), [jeod::NutationMars::int_to_double](#), [jeod::PrecessionMars::J](#), [jeod::NutationMars::mean_anomaly_j2000](#), [jeod::NutationMars::mean_motion](#), [jeod::PrecessionMars::N](#), [jeod::PlanetOrientation::name](#), [jeod::RNPMars::NMars](#), [jeod::RotationMars::phi_at_j2000](#), [jeod::PlanetOrientation::planet_omega](#), [jeod::RNPMars::PMars](#), [jeod::PrecessionMars::psi_at_j2000](#), [jeod::PrecessionMars::psi_dot](#), [jeod::NutationMars::psi_m_orig](#), [jeod::NutationMars::q_angle_j2000](#), [jeod::RNPMars::RMars](#), and [jeod::PlanetRNP::rnp_type](#).

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

- [data/include/rnp_mars.hh](#)
- [data_rnp_mars.cc](#)

8.19 jeod::RNPMessages Class Reference

Describes messages used in the RNP model.

```
#include <RNP_messages.hh>
```

Static Public Attributes

- static char const * [initialization_error](#)
Indicates an error during initialization.
- static char const * [fidelity_error](#)
Indicates a mismatch between the requested fidelity and what is available to the model.
- static char const * [setup_error](#)
Indicates an error during setup of the RNP model.
- static char const * [polar_motion_table_warning](#)
Indicates a problem with the interpolation table commonly found in polar motion implementations.

Private Member Functions

- [RNPMessages](#) (void)
- [RNPMessages](#) (const [RNPMessages](#) &rhs)
- [RNPMessages](#) & operator= (const [RNPMessages](#) &rhs)

Friends

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

8.19.1 Detailed Description

Describes messages used in the RNP model.

Definition at line 84 of file RNP_messages.hh.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 RNPMessages() [1/2]

```
jeod::RNPMessages::RNPMessages (
    void ) [private]
```

8.19.2.2 RNPMessages() [2/2]

```
jeod::RNPMessages::RNPMessages (
    const RNPMessages & rhs ) [private]
```


8.19.3 Member Function Documentation

8.19.3.1 operator=()

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

8.19.4 Friends And Related Function Documentation

8.19.4.1 init_attrjeod__RNPMessages

```
void init_attrjeod__RNPMessages ( ) [friend]
```

8.19.4.2 InputProcessor

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

Definition at line 86 of file RNP_messages.hh.

8.19.5 Field Documentation

8.19.5.1 fidelity_error

```
char const * jeod::RNPMessages::fidelity_error [static]
```

Initial value:

```
=
    "environment/RNP/" "fidelity_error"
```

Indicates a mismatch between the requested fidelity and what is available to the model.

trick_units(-)

Definition at line 103 of file RNP_messages.hh.

Referenced by jeod::PlanetRNP::propagate_rnp(), and jeod::PlanetRNP::update_rnp().

8.19.5.2 initialization_error

```
char const * jeod::RNPMessages::initialization_error [static]
```

Initial value:

```
=
    "environment/RNP/" "initialization_error"
```

Indicates an error during initialization.

trick_units(-)

Definition at line 97 of file RNP_messages.hh.

Referenced by jeod::PlanetOrientation::initialize(), jeod::PolarMotionJ2000::initialize(), and jeod::NutationJ2000::initialize().

8.19.5.3 polar_motion_table_warning

```
char const * jeod::RNPMessages::polar_motion_table_warning [static]
```

Initial value:

```
=
    "environment/RNP/" "polar_motion_table_warning"
```

Indicates a problem with the interpolation table commonly found in polar motion implementations.

trick_units(-)

Definition at line 116 of file RNP_messages.hh.

Referenced by jeod::PolarMotionJ2000::update_rotation().

8.19.5.4 setup_error

```
char const * jeod::RNPMessages::setup_error [static]
```

Initial value:

```
=
    "environment/RNP/" "setup_error"
```

Indicates an error during setup of the RNP model.

trick_units(-)

Definition at line 108 of file RNP_messages.hh.

Referenced by jeod::RNPJ2000::ephem_update(), jeod::RNPMars::ephem_update(), jeod::RNPJ2000::get_dyn_time_ptr(), jeod::RNPMars::get_dyn_time_ptr(), jeod::PlanetRNP::propagate_rnp(), jeod::PlanetRNP::update_axial_rotation(), jeod::PlanetRNP::update_rnp(), jeod::RotationJ2000::update_rotation(), jeod::RotationMars::update_rotation(), and jeod::PrecessionMars::update_rotation().

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

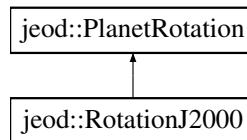
- [RNP_messages.hh](#)
- [RNP_messages.cc](#)

8.20 jeod::RotationJ2000 Class Reference

Implements the axial rotation portion of J2000 RNP.

```
#include <rotation_j2000.hh>
```

Inheritance diagram for jeod::RotationJ2000:



Public Member Functions

- [RotationJ2000](#) ()
default constructor, initialize low level data
- virtual [~RotationJ2000](#) ()
destructor
- virtual void [update_rotation](#) ()
J2000 specific implementation of update_rotation, from [PlanetRotation](#).

Data Fields

- double [planet_rotational_velocity](#)
The nominal axial rotational velocity of the earth.
- [NutationJ2000](#) * [nutation](#)
Pointer to the J2000 nutation object, used for get obliquity information out.
- bool [use_full_rnp](#)
Tells the rotation object if it should use a full blown rotation formulation, or just use the time passed multiplied by the rotational velocity.
- double [theta_gast](#)
The last theta_gast (angle the earth had axially rotated) calculated.
- double [GMST](#)
GMST, currently saved for logging purposes.

Private Member Functions

- [RotationJ2000](#) & [operator=](#) (const [RotationJ2000](#) &rhs)
- [RotationJ2000](#) (const [RotationJ2000](#) &rhs)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__RotationJ2000](#) ()

Additional Inherited Members

8.20.1 Detailed Description

Implements the axial rotation portion of J2000 RNP.

Definition at line 92 of file rotation_j2000.hh.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 RotationJ2000() [1/2]

```
jeod::RotationJ2000::RotationJ2000 (
    void )
```

default constructor, initialize low level data

Definition at line 60 of file rotation_j2000.cc.

8.20.2.2 ~RotationJ2000()

```
jeod::RotationJ2000::~~RotationJ2000 (
    void ) [virtual]
```

destructor

Definition at line 73 of file rotation_j2000.cc.

8.20.2.3 RotationJ2000() [2/2]

```
jeod::RotationJ2000::RotationJ2000 (
    const RotationJ2000 & rhs ) [private]
```

8.20.3 Member Function Documentation

8.20.3.1 operator=()

```
RotationJ2000& jeod::RotationJ2000::operator= (
    const RotationJ2000 & rhs ) [private]
```

8.20.3.2 update_rotation()

```
void jeod::RotationJ2000::update_rotation (
    void ) [virtual]
```

J2000 specific implementation of update_rotation, from [PlanetRotation](#).

For axial rotation

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

Definition at line 84 of file rotation_j2000.cc.

References [jeod::PlanetRotation::current_time](#), [jeod::PlanetRotation::DEGTORAD](#), [jeod::NutationJ2000::equinox_of_equi](#), [nutaton](#), [planet_rotational_velocity](#), [jeod::PlanetRotation::rotation](#), [jeod::RNPMessages::setup_error](#), [theta_gast](#), and [use_full_rnp](#).

8.20.4 Friends And Related Function Documentation

8.20.4.1 init_attrjeod__RotationJ2000

```
void init_attrjeod__RotationJ2000 ( ) [friend]
```

8.20.4.2 InputProcessor

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

Definition at line 94 of file rotation_j2000.hh.

8.20.5 Field Documentation

8.20.5.1 GMST

```
double jeod::RotationJ2000::GMST
```

GMST, currently saved for logging purposes.

trick_units(-)

Definition at line 128 of file rotation_j2000.hh.

8.20.5.2 nutation

```
NutationJ2000* jeod::RotationJ2000::nutation
```

Pointer to the J2000 nutation object, used for get obliquity information out.

Will be NULL (automatically) if anything but Full_Term_RNP is set in the [RNPJ2000](#) objecttrick_units(-)

Definition at line 108 of file rotation_j2000.hh.

Referenced by jeod::RNPJ2000::initialize(), and update_rotation().

8.20.5.3 planet_rotational_velocity

```
double jeod::RotationJ2000::planet_rotational_velocity
```

The nominal axial rotational velocity of the earth.

trick_units(rad/s)

Definition at line 101 of file rotation_j2000.hh.

Referenced by jeod::RNPJ2000::initialize(), and update_rotation().

8.20.5.4 theta_gast

```
double jeod::RotationJ2000::theta_gast
```

The last theta_gast (angle the earth had axially rotated) calculated.

trick_units(rad)

Definition at line 123 of file rotation_j2000.hh.

Referenced by update_rotation().

8.20.5.5 use_full_rnp

```
bool jeod::RotationJ2000::use_full_rnp
```

Tells the rotation object if it should use a full blown rotation formulation, or just use the time passed multiplied by the rotational velocity.

Used with the different initialization options for the main RNP classtrick_units(-)

Definition at line 117 of file rotation_j2000.hh.

Referenced by jeod::RNPJ2000::initialize(), and update_rotation().

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

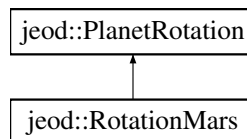
- [rotation_j2000.hh](#)
- [rotation_j2000.cc](#)

8.21 jeod::RotationMars Class Reference

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

```
#include <rotation_mars.hh>
```

Inheritance diagram for jeod::RotationMars:



Public Member Functions

- [RotationMars](#) ()
Default constructor, initialize low level data.
- virtual [~RotationMars](#) ()
Destructor.
- virtual void [update_rotation](#) ()
RotationMars specific implementation of update_rotation, for axial rotation.

Data Fields

- double [planet_rotational_velocity](#)
The Mars average axial rotational velocity.
- [NutationMars](#) * [nutation](#)
Pointer to the Mars nutation object, used to access nutation in longitude information.
- bool [use_full_rnp](#)
Tells the rotation object if it should use a full blown rotation formulation, or if it should just use the time passed multiplied by the rotational velocity.
- double [phi_at_j2000](#)
The (constant) rotated angle of Mars at the J2000 epoch.
- double [phi_spin](#)
The most recent calculated value of the rotation angle for Mars, measured since J2000 epoch.

Private Member Functions

- [RotationMars](#) & [operator=](#) (const [RotationMars](#) &rhs)
- [RotationMars](#) (const [RotationMars](#) &rhs)

Friends

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

Additional Inherited Members

8.21.1 Detailed Description

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

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

8.21.2 Constructor & Destructor Documentation

8.21.2.1 [RotationMars\(\)](#) [1/2]

```
jeod::RotationMars::RotationMars (  
    void )
```

Default constructor, initialize low level data.

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

8.21.2.2 [~RotationMars\(\)](#)

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

Destructor.

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

8.21.2.3 [RotationMars\(\)](#) [2/2]

```
jeod::RotationMars::RotationMars (  
    const RotationMars & rhs ) [private]
```


8.21.3 Member Function Documentation

8.21.3.1 operator=()

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

8.21.3.2 update_rotation()

```
void jeod::RotationMars::update_rotation (
    void ) [virtual]
```

[RotationMars](#) specific implementation of update_rotation, for axial rotation.

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

Definition at line 87 of file rotation_mars.cc.

References [jeod::PlanetRotation::current_time](#), [nutaton](#), [jeod::NutationMars::nutaton_in_longitude](#), [jeod::NutationMars::obliquity_angle](#), [phi_at_j2000](#), [phi_spin](#), [planet_rotational_velocity](#), [jeod::PlanetRotation::rotation](#), [jeod::RNPMessages::setup_error](#), and [use_full_rnp](#).

8.21.4 Friends And Related Function Documentation

8.21.4.1 init_attrjeod__RotationMars

```
void init_attrjeod__RotationMars ( ) [friend]
```

8.21.4.2 InputProcessor

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

Definition at line 99 of file rotation_mars.hh.

8.21.5 Field Documentation

8.21.5.1 nutation

```
NutationMars* jeod::RotationMars::nutation
```

Pointer to the Mars nutation object, used to access nutation in longitude information.

Will be NULL (automatically) if anything but Full_Term_RNP is set in the [RNPMars](#) object.`trick_units(-)`

Definition at line 113 of file `rotation_mars.hh`.

Referenced by `jeod::RNPMars::initialize()`, and `update_rotation()`.

8.21.5.2 phi_at_j2000

```
double jeod::RotationMars::phi_at_j2000
```

The (constant) rotated angle of Mars at the J2000 epoch.

`trick_units(rad)`

Definition at line 126 of file `rotation_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, and `update_rotation()`.

8.21.5.3 phi_spin

```
double jeod::RotationMars::phi_spin
```

The most recent calculated value of the rotation angle for Mars, measured since J2000 epoch.

`trick_units(rad)`

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

Referenced by `update_rotation()`.

8.21.5.4 planet_rotational_velocity

```
double jeod::RotationMars::planet_rotational_velocity
```

The Mars average axial rotational velocity.

`trick_units(rad/s)`

Definition at line 106 of file `rotation_mars.hh`.

Referenced by `jeod::RNPMars::initialize()`, and `update_rotation()`.

8.21.5.5 use_full_rnp

```
bool jeod::RotationMars::use_full_rnp
```

Tells the rotation object if it should use a full blown rotation formulation, or if it should just use the time passed multiplied by the rotational velocity.

Used with the different initialization options for the main RNP class.`trick_units(-)`

Definition at line 121 of file `rotation_mars.hh`.

Referenced by `jeod::RNPMars::initialize()`, and `update_rotation()`.

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

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

Chapter 9

File Documentation

9.1 `class_declarations.hh` File Reference

Forward declarations of classes defined for JEOD 2.0 Generic RNP.

Namespaces

- [jeod](#)
Namespace jeod.

9.1.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 Generic RNP.

9.2 `class_declarations.hh` File Reference

Forward declarations of classes defined for JEOD 2.0 J2000 RNP.

Namespaces

- [jeod](#)
Namespace jeod.

9.2.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 J2000 RNP.

9.3 data_nutation_j2000.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "environment/RNP/RNPJ2000/include/nutation_j2000_init.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/nutation_j2000.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- `#define` [JEOD_FRIEND_CLASS](#) NutationJ2000Init_nutation_j2000_default_data

9.3.1 Macro Definition Documentation

9.3.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS NutationJ2000Init_nutation_j2000_default_data
```

Definition at line 21 of file data_nutation_j2000.cc.

9.4 data_rnp_j2000.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "environment/RNP/RNPJ2000/include/rnp_j2000.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/ref_frames/include/ref_frame_interface.hh"
#include "../include/rnp_j2000.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- `#define` [JEOD_FRIEND_CLASS](#) RNPJ2000_rnp_j2000_default_data

9.4.1 Macro Definition Documentation

9.4.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS RNPJ2000_rnp_j2000_default_data
```

Definition at line 21 of file data_rnp_j2000.cc.

9.5 data_rnp_mars.cc File Reference

```
#include <cmath>
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "environment/RNP/RNPMars/include/rnp_mars.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/rnp_mars.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- `#define` [JEOD_FRIEND_CLASS](#) RNPMars_rnp_mars_default_data

9.5.1 Macro Definition Documentation

9.5.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS RNPMars_rnp_mars_default_data
```

Definition at line 23 of file data_rnp_mars.cc.

9.6 nutation_j2000.cc File Reference

Implementation for the NutationJ2000 class.

```
#include <cstdint>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/nutation_j2000.hh"
#include "../include/nutation_j2000_init.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.6.1 Detailed Description

Implementation for the NutationJ2000 class.

9.7 nutation_j2000.hh File Reference

Data Structures

- class [jeod::NutationJ2000Init_nutation_j2000_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.8 nutation_j2000.hh File Reference

Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
```

Data Structures

- class [jeod::NutationJ2000](#)
Implements the nutation portion of the J2000 RNP.

Namespaces

- [jeod](#)

Namespace jeod.

9.8.1 Detailed Description

Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

9.9 nutation_j2000_init.cc File Reference

Implementation of NutationJ2000Init.

```
#include <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/nutation_j2000_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.9.1 Detailed Description

Implementation of NutationJ2000Init.

9.10 nutation_j2000_init.hh File Reference

Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::NutationJ2000Init](#)

The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.

Namespaces

- [jeod](#)

Namespace jeod.

9.10.1 Detailed Description

Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.

9.11 nutation_mars.cc File Reference

Implementation for the NutationMars class.

```
#include <cstdint>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/nutation_mars.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.11.1 Detailed Description

Implementation for the NutationMars class.

9.12 nutation_mars.hh File Reference

Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::NutationMars](#)

Implements the nutation portion of the "Pathfinder" Mars RNP model.

Namespaces

- [jeod](#)

Namespace jeod.

9.12.1 Detailed Description

Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.

9.13 planet_orientation.cc File Reference

Implement PlanetOrientation.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/planet_orientation.hh"
#include "../include/RNP_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.13.1 Detailed Description

Implement PlanetOrientation.

9.14 planet_orientation.hh File Reference

Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.

```
#include <string>
#include <utility>
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/ref_frame_interface.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_interface.↵
hh"
#include "environment/ephemerides/ephem_item/include/ephem_orient.hh"
```

Data Structures

- class [jeod::PlanetOrientation](#)

The generic framework for orientation models to interact with a DynManager object.

Namespaces

- [jeod](#)

Namespace jeod.

9.14.1 Detailed Description

Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.

9.15 planet_rnp.cc File Reference

Implement PlanetRNP.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/planet_rnp.hh"
#include "../include/planet_rotation.hh"
#include "../include/RNP_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.15.1 Detailed Description

Implement PlanetRNP.

9.16 planet_rnp.hh File Reference

Establish a framework for rotation-nutation-precession style planet attitude models.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "planet_orientation.hh"
```

Data Structures

- class [jeod::PlanetRNP](#)

The generic framework for orientation models based on the RNP paradigm.

Namespaces

- [jeod](#)

Namespace jeod.

9.16.1 Detailed Description

Establish a framework for rotation-nutation-precession style planet attitude models.

9.17 planet_rotation.cc File Reference

Implementation for PlanetRotation.

```
#include <cmath>
#include "utils/math/include/matrix3x3.hh"
#include "../include/planet_rotation.hh"
#include "../include/planet_rotation_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.17.1 Detailed Description

Implementation for PlanetRotation.

9.18 planet_rotation.hh File Reference

Pure virtual polymorphic base class for all forms of planet rotation in the RNP model including precession, nutation, polar motion and axial rotation.

```
#include <math.h>
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::PlanetRotation](#)

The generic base class for all planet transformations.

Namespaces

- [jeod](#)

Namespace jeod.

9.18.1 Detailed Description

Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation.

9.19 planet_rotation_init.cc File Reference

Implementation for PlanetRotationInit.

```
#include "../include/planet_rotation_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.19.1 Detailed Description

Implementation for PlanetRotationInit.

9.20 planet_rotation_init.hh File Reference

Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::PlanetRotationInit](#)

The generic base class for all initializing classes for [PlanetRotation](#) derived classes.

Namespaces

- [jeod](#)

Namespace jeod.

9.20.1 Detailed Description

Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.

9.21 polar_motion_j2000.cc File Reference

Implementation of PolarMotionJ2000.

```
#include <cstdint>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/polar_motion_j2000.hh"
#include "../include/polar_motion_j2000_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.21.1 Detailed Description

Implementation of PolarMotionJ2000.

9.22 polar_motion_j2000.hh File Reference

Model the polar motion portion of the RNP routine for the Standard Epoch J2000.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::PolarMotionJ2000](#)

Implements the polar motion portion of the J2000 RNP.

Namespaces

- [jeod](#)

Namespace jeod.

9.22.1 Detailed Description

Model the polar motion portion of the RNP routine for the Standard Epoch J2000.

This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

9.23 polar_motion_j2000_init.cc File Reference

Implementation for PolarMotionJ2000Init.

```
#include <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/polar_motion_j2000_init.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.23.1 Detailed Description

Implementation for PolarMotionJ2000Init.

9.24 polar_motion_j2000_init.hh File Reference

Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::PolarMotionJ2000Init](#)
Initializes the [PolarMotionJ2000](#) object.

Namespaces

- [jeod](#)
Namespace jeod.

9.24.1 Detailed Description

Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.

9.25 precession_j2000.cc File Reference

Implementation for PrecessionJ2000.

```
#include "../include/precession_j2000.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.25.1 Detailed Description

Implementation for PrecessionJ2000.

9.26 precession_j2000.hh File Reference

Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"  
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::PrecessionJ2000](#)
Implements the precession portion of the J2000 RNP.

Namespaces

- [jeod](#)
Namespace jeod.

9.26.1 Detailed Description

Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

9.27 precession_mars.cc File Reference

Implementation of PrecessionMars.

```
#include <cstdint>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/precession_mars.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.27.1 Detailed Description

Implementation of PrecessionMars.

9.28 precession_mars.hh File Reference

Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nutation_mars.hh"
```

Data Structures

- class [jeod::PrecessionMars](#)

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

Namespaces

- [jeod](#)

Namespace jeod.

9.28.1 Detailed Description

Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.

9.29 rnp_j2000.cc File Reference

Implementation for RNPJ2000.

```
#include <cstdint>
#include "environment/time/include/time_manager.hh"
#include "environment/time/include/time_tt.hh"
#include "environment/time/include/time_ut1.hh"
#include "environment/time/include/time_gmst.hh"
#include "environment/time/include/time_dyn.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/math/include/numerical.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "../include/rnp_j2000.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.29.1 Detailed Description

Implementation for RNPJ2000.

9.30 rnp_j2000.hh File Reference

Data Structures

- class [jeod::RNPJ2000_rnp_j2000_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.31 rnp_j2000.hh File Reference

A specific implementation of PlanetRNP, for Earth.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "nutration_j2000.hh"
#include "polar_motion_j2000.hh"
#include "rotation_j2000.hh"
#include "precession_j2000.hh"
```

Data Structures

- class [jeod::RNPJ2000](#)

Implements the J2000 RNP model using the generic RNP framework.

Namespaces

- [jeod](#)

Namespace jeod.

9.31.1 Detailed Description

A specific implementation of PlanetRNP, for Earth.

This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

9.32 rnp_mars.cc File Reference

Implementation of RNPMars.

```
#include <cstdint>
#include "environment/time/include/time_manager.hh"
#include "environment/time/include/time_tt.hh"
#include "environment/time/include/time_dyn.hh"
#include "environment/planet/include/planet.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/rnp_mars.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.32.1 Detailed Description

Implementation of RNPMars.

9.33 rnp_mars.hh File Reference

Data Structures

- class [jeod::RNPMars_rnp_mars_default_data](#)

Namespaces

- [jeod](#)

Namespace jeod.

9.34 rnp_mars.hh File Reference

A specific implementation of PlanetRNP, for Mars.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "nutation_mars.hh"
#include "rotation_mars.hh"
#include "precession_mars.hh"
```

Data Structures

- class [jeod::RNPMars](#)

Implements the "Pathfinder" Mars RNP model using the generic RNP framework.

Namespaces

- [jeod](#)

Namespace jeod.

9.34.1 Detailed Description

A specific implementation of PlanetRNP, for Mars.

This is a form of the JPL-developed "Pathfinder" Mars orientation model.

9.35 RNP_messages.cc File Reference

Implement RNP_messages.

```
#include "../include/RNP_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

Macros

- `#define` [PATH](#) "environment/RNP/"

9.35.1 Detailed Description

Implement RNP_messages.

9.36 RNP_messages.hh File Reference

Implement RNP_messages.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::RNPMessages](#)
Describes messages used in the RNP model.

Namespaces

- [jeod](#)
Namespace jeod.

9.36.1 Detailed Description

Implement RNP_messages.

9.37 rotation_j2000.cc File Reference

Implementation of RotationJ2000.

```
#include <cstdint>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "../include/rotation_j2000.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.37.1 Detailed Description

Implementation of RotationJ2000.

9.38 rotation_j2000.hh File Reference

Model the axial rotation portion of the RNP routine for the Standard Epoch J2000.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nutation_j2000.hh"
```

Data Structures

- class [jeod::RotationJ2000](#)
Implements the axial rotation portion of J2000 RNP.

Namespaces

- [jeod](#)
Namespace jeod.

9.38.1 Detailed Description

Model the axial rotation portion of the RNP routine for the Standard Epoch J2000.

This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

9.39 rotation_mars.cc File Reference

Implementation of RotationMars.

```
#include <cstdlib>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/rotation_mars.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.39.1 Detailed Description

Implementation of RotationMars.

9.40 rotation_mars.hh File Reference

Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nutation_mars.hh"
```

Data Structures

- class [jeod::RotationMars](#)
Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

Namespaces

- [jeod](#)
Namespace jeod.

9.40.1 Detailed Description

Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.

9.41 xpyp_daily.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "environment/RNP/RNPJ2000/include/polar_motion_j2000_init.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/xpyp_daily.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- `#define` [JEOD_FRIEND_CLASS](#) PolarMotionJ2000Init_xpyp_daily_default_data

9.41.1 Macro Definition Documentation

9.41.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS PolarMotionJ2000Init_xpyp_daily_default_data
```

Definition at line 21 of file xpyp_daily.cc.

9.42 xpyp_daily.hh File Reference

Data Structures

- class [jeod::PolarMotionJ2000Init_xpyp_daily_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.43 xpyp_monthly.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"  
#include "environment/RNP/RNPJ2000/include/polar_motion_j2000_init.hh"  
#include "utils/memory/include/jeod_alloc.hh"  
#include "../include/xpyp_monthly.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- #define [JEOD_FRIEND_CLASS](#) PolarMotionJ2000Init_xpyp_monthly_default_data

9.43.1 Macro Definition Documentation

9.43.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS PolarMotionJ2000Init_xpyp_monthly_default_data
```

Definition at line 21 of file xpyp_monthly.cc.

9.44 xpyp_monthly.hh File Reference

Data Structures

- class [jeod::PolarMotionJ2000Init_xpyp_monthly_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

Index

- ~NutationJ2000
 - jeod::NutationJ2000, [21](#)
- ~NutationJ2000Init
 - jeod::NutationJ2000Init, [29](#)
- ~NutationMars
 - jeod::NutationMars, [35](#)
- ~PlanetOrientation
 - jeod::PlanetOrientation, [41](#)
- ~PlanetRNP
 - jeod::PlanetRNP, [49](#)
- ~PlanetRotation
 - jeod::PlanetRotation, [55](#)
- ~PlanetRotationInit
 - jeod::PlanetRotationInit, [61](#)
- ~PolarMotionJ2000
 - jeod::PolarMotionJ2000, [64](#)
- ~PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, [69](#)
- ~PrecessionJ2000
 - jeod::PrecessionJ2000, [75](#)
- ~PrecessionMars
 - jeod::PrecessionMars, [78](#)
- ~RNPJ2000
 - jeod::RNPJ2000, [83](#)
- ~RNPMars
 - jeod::RNPMars, [92](#)
- ~RotationJ2000
 - jeod::RotationJ2000, [104](#)
- ~RotationMars
 - jeod::RotationMars, [108](#)
- activate
 - jeod::PlanetOrientation, [42](#)
- active
 - jeod::PlanetOrientation, [45](#)
- class_declarations.hh, [113](#)
- compute_fixed_matrices
 - jeod::PrecessionMars, [78](#)
- current_time
 - jeod::PlanetRotation, [58](#)
- D
 - jeod::NutationJ2000, [23](#)
- D_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [30](#)
- DAYTOJULIANCENT
 - jeod::PlanetRotation, [58](#)
- DEGTORAD
 - jeod::PlanetRotation, [58](#)
- DEGTOSEC
 - jeod::PlanetRotation, [58](#)
- data_nutation_j2000.cc, [114](#)
 - JEOD_FRIEND_CLASS, [114](#)
- data_rnp_j2000.cc, [114](#)
 - JEOD_FRIEND_CLASS, [115](#)
- data_rnp_mars.cc, [115](#)
 - JEOD_FRIEND_CLASS, [115](#)
- deactivate
 - jeod::PlanetOrientation, [42](#)
- enable_polar
 - jeod::PlanetRNP, [51](#)
- Environment, [12](#)
- ephem_activate
 - jeod::PlanetOrientation, [42](#)
- ephem_build_tree
 - jeod::PlanetOrientation, [43](#)
- ephem_initialize
 - jeod::PlanetOrientation, [43](#)
- ephem_update
 - jeod::RNPJ2000, [84](#)
 - jeod::RNPMars, [93](#)
- epsilon_bar
 - jeod::NutationJ2000, [23](#)
- equa_of_equi
 - jeod::NutationJ2000, [23](#)
- F
 - jeod::NutationJ2000, [24](#)
- F_coeffs
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationJ2000Init, [30](#)
- fidelity_error
 - jeod::RNPMessages, [101](#)
- GMST
 - jeod::RotationJ2000, [105](#)
- GenericRNP, [14](#)
 - PATH, [14](#)
- get_dyn_time_ptr
 - jeod::RNPJ2000, [84](#)
 - jeod::RNPMars, [93](#)
- get_name
 - jeod::PlanetOrientation, [43](#)
 - jeod::PlanetRNP, [49](#)
 - jeod::RNPJ2000, [84](#)
 - jeod::RNPMars, [94](#)
- get_rotation

- jeod::PlanetRotation, 55
- get_rotation_transpose
 - jeod::PlanetRotation, 56
- gmst_ptr
 - jeod::RNPJ2000, 87
- I_at_j2000
 - jeod::NutationMars, 37
- I_dot
 - jeod::NutationMars, 37
- I_m_orig
 - jeod::NutationMars, 37
- init_attrjeod__NutationJ2000
 - jeod::NutationJ2000, 22
- init_attrjeod__NutationJ2000Init
 - jeod::NutationJ2000Init, 30
- init_attrjeod__NutationMars
 - jeod::NutationMars, 36
- init_attrjeod__PlanetOrientation
 - jeod::PlanetOrientation, 45
- init_attrjeod__PlanetRNP
 - jeod::PlanetRNP, 51
- init_attrjeod__PlanetRotation
 - jeod::PlanetRotation, 57
- init_attrjeod__PlanetRotationInit
 - jeod::PlanetRotationInit, 62
- init_attrjeod__PolarMotionJ2000
 - jeod::PolarMotionJ2000, 65
- init_attrjeod__PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, 70
- init_attrjeod__PrecessionJ2000
 - jeod::PrecessionJ2000, 76
- init_attrjeod__PrecessionMars
 - jeod::PrecessionMars, 79
- init_attrjeod__RNPJ2000
 - jeod::RNPJ2000, 87
- init_attrjeod__RNPMars
 - jeod::RNPMars, 96
- init_attrjeod__RNPMessages
 - jeod::RNPMessages, 101
- init_attrjeod__RotationJ2000
 - jeod::RotationJ2000, 105
- init_attrjeod__RotationMars
 - jeod::RotationMars, 109
- initialization_error
 - jeod::RNPMessages, 101
- initialize
 - jeod::NutationJ2000, 21
 - jeod::NutationJ2000Init_nutation_j2000_default_↔
data, 33
 - jeod::PlanetOrientation, 44
 - jeod::PlanetRotation, 56
 - jeod::PolarMotionJ2000, 64
 - jeod::PolarMotionJ2000Init_xpyp_daily_default_↔
data, 72
 - jeod::PolarMotionJ2000Init_xpyp_monthly_↔
default_data, 73
 - jeod::RNPJ2000, 85
 - jeod::RNPJ2000_rnp_j2000_default_data, 90
 - jeod::RNPMars, 94
 - jeod::RNPMars_rnp_mars_default_data, 99
- InputProcessor
 - jeod::NutationJ2000, 22
 - jeod::NutationJ2000Init, 30
 - jeod::NutationMars, 36
 - jeod::PlanetOrientation, 45
 - jeod::PlanetRNP, 51
 - jeod::PlanetRotation, 57
 - jeod::PlanetRotationInit, 62
 - jeod::PolarMotionJ2000, 65
 - jeod::PolarMotionJ2000Init, 70
 - jeod::PrecessionJ2000, 76
 - jeod::PrecessionMars, 79
 - jeod::RNPJ2000, 87
 - jeod::RNPMars, 96
 - jeod::RNPMessages, 101
 - jeod::RotationJ2000, 105
 - jeod::RotationMars, 109
- int_to_double
 - jeod::NutationMars, 37
- internal_name
 - jeod::RNPJ2000, 87
 - jeod::RNPMars, 96
- J
 - jeod::PrecessionMars, 79
- JEOD_FRIEND_CLASS
 - data_nutation_j2000.cc, 114
 - data_rnp_j2000.cc, 115
 - data_rnp_mars.cc, 115
 - xpyp_daily.cc, 133
 - xpyp_monthly.cc, 133
- JULIANCENTTODAY
 - jeod::PlanetRotation, 59
- JULIANDAYTOSEC
 - jeod::PlanetRotation, 59
- jeod, 17
- jeod::NutationJ2000, 19
 - ~NutationJ2000, 21
- D, 23
- D_coeffs, 23
- epsilon_bar, 23
- equa_of_equi, 23
- F, 24
- F_coeffs, 24
- init_attrjeod__NutationJ2000, 22
- initialize, 21
- InputProcessor, 22
- L, 24
- L_coeffs, 24
- long_coeffs, 25
- long_t_coeffs, 25
- M, 25
- M_coeffs, 25
- num_coeffs, 26
- nutation_in_longitude, 26
- nutation_in_obliquity, 26
- NutationJ2000, 21

- obliq_coefs, 26
- obliq_t_coefs, 27
- omega, 27
- omega_coefs, 27
- operator=, 22
- update_rotation, 22
- jeod::NutationJ2000Init, 28
 - ~NutationJ2000Init, 29
 - D_coefs, 30
 - F_coefs, 30
 - init_attrjeod__NutationJ2000Init, 30
 - InputProcessor, 30
 - L_coefs, 31
 - long_coefs, 31
 - long_t_coefs, 31
 - M_coefs, 31
 - num_coefs, 32
 - NutationJ2000Init, 29
 - obliq_coefs, 32
 - obliq_t_coefs, 32
 - omega_coefs, 32
 - operator=, 30
- jeod::NutationJ2000Init_nutation_j2000_default_data, 33
 - initialize, 33
- jeod::NutationMars, 34
 - ~NutationMars, 35
 - I_at_j2000, 37
 - I_dot, 37
 - I_m_orig, 37
 - init_attrjeod__NutationMars, 36
 - InputProcessor, 36
 - int_to_double, 37
 - mean_anomaly_j2000, 38
 - mean_motion, 38
 - nutration_in_longitude, 38
 - nutration_in_obliquity, 38
 - NutationMars, 35
 - obliquity_angle, 39
 - operator=, 36
 - psi_m_orig, 39
 - q_angle_j2000, 39
 - update_rotation, 36
- jeod::PlanetOrientation, 40
 - ~PlanetOrientation, 41
 - activate, 42
 - active, 45
 - deactivate, 42
 - ephem_activate, 42
 - ephem_build_tree, 43
 - ephem_initialize, 43
 - get_name, 43
 - init_attrjeod__PlanetOrientation, 45
 - initialize, 44
 - InputProcessor, 45
 - name, 45
 - operator=, 44
 - orient_interface, 45
 - planet, 46
 - planet_omega, 46
 - planet_rot_state, 46
 - PlanetOrientation, 41, 42
 - set_name, 44
- jeod::PlanetRNP, 47
 - ~PlanetRNP, 49
 - enable_polar, 51
 - get_name, 49
 - init_attrjeod__PlanetRNP, 51
 - InputProcessor, 51
 - NP_matrix, 51
 - nutration, 52
 - operator=, 50
 - PlanetRNP, 49
 - polar_motion, 52
 - precession, 52
 - propagate_rnp, 50
 - RNPFidelity, 48
 - rnp_type, 52
 - rotation, 53
 - scratch_matrix, 53
 - update_axial_rotation, 50
 - update_rnp, 50
- jeod::PlanetRotation, 54
 - ~PlanetRotation, 55
 - current_time, 58
 - DAYTOJULIANCENT, 58
 - DEGTORAD, 58
 - DEGTOSEC, 58
 - get_rotation, 55
 - get_rotation_transpose, 56
 - init_attrjeod__PlanetRotation, 57
 - initialize, 56
 - InputProcessor, 57
 - JULIANCENTTODAY, 59
 - JULIANDAYTOSEC, 59
 - operator=, 56
 - PlanetRotation, 55
 - RADTODEG, 59
 - rotation, 59
 - SECTODEG, 60
 - SECTOJULIANDAY, 60
 - update_rotation, 56
 - update_time, 57
- jeod::PlanetRotationInit, 61
 - ~PlanetRotationInit, 61
 - init_attrjeod__PlanetRotationInit, 62
 - InputProcessor, 62
 - operator=, 62
 - PlanetRotationInit, 61, 62
- jeod::PolarMotionJ2000, 63
 - ~PolarMotionJ2000, 64
 - init_attrjeod__PolarMotionJ2000, 65
 - initialize, 64
 - InputProcessor, 65
 - last_table_index, 66
 - operator=, 65

- override_table, 66
- polar_mjd, 66
- PolarMotionJ2000, 64
- update_rotation, 65
- warn_table, 66
- xp, 67
- xp_tbl, 67
- yp, 67
- yp_tbl, 67
- jeod::PolarMotionJ2000Init, 68
 - ~PolarMotionJ2000Init, 69
 - init_attrjeod__PolarMotionJ2000Init, 70
 - InputProcessor, 70
 - last_table_index, 70
 - operator=, 70
 - override_table, 70
 - polar_mjd, 71
 - PolarMotionJ2000Init, 69
 - xp, 71
 - xp_tbl, 71
 - yp, 71
 - yp_tbl, 72
- jeod::PolarMotionJ2000Init_xpyp_daily_default_data, 72
 - initialize, 72
- jeod::PolarMotionJ2000Init_xpyp_monthly_default_data, 73
 - data, 73
 - initialize, 73
- jeod::PrecessionJ2000, 74
 - ~PrecessionJ2000, 75
 - init_attrjeod__PrecessionJ2000, 76
 - InputProcessor, 76
 - operator=, 75
 - PrecessionJ2000, 74, 75
 - update_rotation, 75
- jeod::PrecessionMars, 76
 - ~PrecessionMars, 78
 - compute_fixed_matrices, 78
 - init_attrjeod__PrecessionMars, 79
 - InputProcessor, 79
 - J, 79
 - N, 80
 - NJ_matrix, 80
 - nutation, 80
 - operator=, 78
 - PrecessionMars, 77, 78
 - psi_at_j2000, 80
 - psi_dot, 81
 - psi_precess, 81
 - update_rotation, 79
- jeod::RNPIJ2000, 82
 - ~RNPIJ2000, 83
 - ephem_update, 84
 - get_dyn_time_ptr, 84
 - get_name, 84
 - gmst_ptr, 87
 - init_attrjeod__RNPIJ2000, 87
 - initialize, 85
 - InputProcessor, 87
 - internal_name, 87
 - last_updated_time_full, 87
 - last_updated_time_rotational, 88
 - NJ2000, 88
 - never_updated_full, 88
 - never_updated_rotational, 88
 - operator=, 85
 - PJ2000, 89
 - PMJ2000, 89
 - RJ2000, 89
 - RNPIJ2000, 83, 84
 - time_dyn_ptr, 89
 - timestamp, 85
 - update_axial_rotation, 86
 - update_rnp, 86
- jeod::RNPIJ2000_rnp_j2000_default_data, 90
 - initialize, 90
- jeod::RNPMars, 91
 - ~RNPMars, 92
 - ephem_update, 93
 - get_dyn_time_ptr, 93
 - get_name, 94
 - init_attrjeod__RNPMars, 96
 - initialize, 94
 - InputProcessor, 96
 - internal_name, 96
 - last_updated_time_full, 96
 - last_updated_time_rotational, 96
 - NMars, 97
 - never_updated_full, 97
 - never_updated_rotational, 97
 - operator=, 94
 - PMars, 97
 - RMars, 98
 - RNPMars, 92, 93
 - time_dyn_ptr, 98
 - timestamp, 94
 - tt_ptr, 98
 - update_axial_rotation, 95
 - update_rnp, 95
- jeod::RNPMars_rnp_mars_default_data, 99
 - initialize, 99
- jeod::RNPMessages, 99
 - fidelity_error, 101
 - init_attrjeod__RNPMessages, 101
 - initialization_error, 101
 - InputProcessor, 101
 - operator=, 101
 - polar_motion_table_warning, 102
 - RNPMessages, 100
 - setup_error, 102
- jeod::RotationJ2000, 103
 - ~RotationJ2000, 104
 - GMST, 105
 - init_attrjeod__RotationJ2000, 105
 - InputProcessor, 105
 - nutation, 106
 - operator=, 104

- planet_rotational_velocity, 106
- RotationJ2000, 104
- theta_gast, 106
- update_rotation, 105
- use_full_rnp, 106
- jeod::RotationMars, 107
 - ~RotationMars, 108
 - init_attrjeod__RotationMars, 109
 - InputProcessor, 109
 - nutration, 109
 - operator=, 109
 - phi_at_j2000, 110
 - phi_spin, 110
 - planet_rotational_velocity, 110
 - RotationMars, 108
 - update_rotation, 109
 - use_full_rnp, 110
- L
 - jeod::NutationJ2000, 24
- L_coeffs
 - jeod::NutationJ2000, 24
 - jeod::NutationJ2000Init, 31
- last_table_index
 - jeod::PolarMotionJ2000, 66
 - jeod::PolarMotionJ2000Init, 70
- last_updated_time_full
 - jeod::RNPJ2000, 87
 - jeod::RNPMars, 96
- last_updated_time_rotational
 - jeod::RNPJ2000, 88
 - jeod::RNPMars, 96
- long_coeffs
 - jeod::NutationJ2000, 25
 - jeod::NutationJ2000Init, 31
- long_t_coeffs
 - jeod::NutationJ2000, 25
 - jeod::NutationJ2000Init, 31
- M
 - jeod::NutationJ2000, 25
- M_coeffs
 - jeod::NutationJ2000, 25
 - jeod::NutationJ2000Init, 31
- mean_anomaly_j2000
 - jeod::NutationMars, 38
- mean_motion
 - jeod::NutationMars, 38
- Models, 11
- N
 - jeod::PrecessionMars, 80
- NJ2000
 - jeod::RNPJ2000, 88
- NJ_matrix
 - jeod::PrecessionMars, 80
- NMars
 - jeod::RNPMars, 97
- NP_matrix
 - jeod::PlanetRNP, 51
- name
 - jeod::PlanetOrientation, 45
- never_updated_full
 - jeod::RNPJ2000, 88
 - jeod::RNPMars, 97
- never_updated_rotational
 - jeod::RNPJ2000, 88
 - jeod::RNPMars, 97
- num_coeffs
 - jeod::NutationJ2000, 26
 - jeod::NutationJ2000Init, 32
- nutration
 - jeod::PlanetRNP, 52
 - jeod::PrecessionMars, 80
 - jeod::RotationJ2000, 106
 - jeod::RotationMars, 109
- nutration_in_longitude
 - jeod::NutationJ2000, 26
 - jeod::NutationMars, 38
- nutration_in_obliquity
 - jeod::NutationJ2000, 26
 - jeod::NutationMars, 38
- nutration_j2000.cc, 116
- nutration_j2000.hh, 116
- nutration_j2000_init.cc, 117
- nutration_j2000_init.hh, 117
- nutration_mars.cc, 118
- nutration_mars.hh, 118
- NutationJ2000
 - jeod::NutationJ2000, 21
- NutationJ2000Init
 - jeod::NutationJ2000Init, 29
- NutationMars
 - jeod::NutationMars, 35
- obliq_coeffs
 - jeod::NutationJ2000, 26
 - jeod::NutationJ2000Init, 32
- obliq_t_coeffs
 - jeod::NutationJ2000, 27
 - jeod::NutationJ2000Init, 32
- obliquity_angle
 - jeod::NutationMars, 39
- omega
 - jeod::NutationJ2000, 27
- omega_coeffs
 - jeod::NutationJ2000, 27
 - jeod::NutationJ2000Init, 32
- operator=
 - jeod::NutationJ2000, 22
 - jeod::NutationJ2000Init, 30
 - jeod::NutationMars, 36
 - jeod::PlanetOrientation, 44
 - jeod::PlanetRNP, 50
 - jeod::PlanetRotation, 56
 - jeod::PlanetRotationInit, 62
 - jeod::PolarMotionJ2000, 65
 - jeod::PolarMotionJ2000Init, 70

- jeod::PrecessionJ2000, 75
- jeod::PrecessionMars, 78
- jeod::RNPJ2000, 85
- jeod::RNPMars, 94
- jeod::RNPMessages, 101
- jeod::RotationJ2000, 104
- jeod::RotationMars, 109
- orient_interface
 - jeod::PlanetOrientation, 45
- override_table
 - jeod::PolarMotionJ2000, 66
 - jeod::PolarMotionJ2000Init, 70
- PATH
 - GenericRNP, 14
- PJ2000
 - jeod::RNPJ2000, 89
- PMJ2000
 - jeod::RNPJ2000, 89
- PMars
 - jeod::RNPMars, 97
- phi_at_j2000
 - jeod::RotationMars, 110
- phi_spin
 - jeod::RotationMars, 110
- planet
 - jeod::PlanetOrientation, 46
- planet_omega
 - jeod::PlanetOrientation, 46
- planet_orientation.cc, 119
- planet_orientation.hh, 119
- planet_rnp.cc, 120
- planet_rnp.hh, 120
- planet_rot_state
 - jeod::PlanetOrientation, 46
- planet_rotation.cc, 121
- planet_rotation.hh, 121
- planet_rotation_init.cc, 122
- planet_rotation_init.hh, 122
- planet_rotational_velocity
 - jeod::RotationJ2000, 106
 - jeod::RotationMars, 110
- PlanetOrientation
 - jeod::PlanetOrientation, 41, 42
- PlanetRNP
 - jeod::PlanetRNP, 49
- PlanetRotation
 - jeod::PlanetRotation, 55
- PlanetRotationInit
 - jeod::PlanetRotationInit, 61, 62
- polar_mjd
 - jeod::PolarMotionJ2000, 66
 - jeod::PolarMotionJ2000Init, 71
- polar_motion
 - jeod::PlanetRNP, 52
- polar_motion_j2000.cc, 123
- polar_motion_j2000.hh, 123
- polar_motion_j2000_init.cc, 124
- polar_motion_j2000_init.hh, 124
- polar_motion_table_warning
 - jeod::RNPMessages, 102
- PolarMotionJ2000
 - jeod::PolarMotionJ2000, 64
- PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, 69
- precession
 - jeod::PlanetRNP, 52
- precession_j2000.cc, 125
- precession_j2000.hh, 125
- precession_mars.cc, 126
- precession_mars.hh, 126
- PrecessionJ2000
 - jeod::PrecessionJ2000, 74, 75
- PrecessionMars
 - jeod::PrecessionMars, 77, 78
- propagate_rnp
 - jeod::PlanetRNP, 50
- psi_at_j2000
 - jeod::PrecessionMars, 80
- psi_dot
 - jeod::PrecessionMars, 81
- psi_m_orig
 - jeod::NutationMars, 39
- psi_precess
 - jeod::PrecessionMars, 81
- q_angle_j2000
 - jeod::NutationMars, 39
- RADTODEG
 - jeod::PlanetRotation, 59
- RJ2000
 - jeod::RNPJ2000, 89
- RMars
 - jeod::RNPMars, 98
- RNP_messages.cc, 129
- RNP_messages.hh, 130
- RNPFidelity
 - jeod::PlanetRNP, 48
- RNPJ2000, 15
 - jeod::RNPJ2000, 83, 84
- RNPMars, 16
 - jeod::RNPMars, 92, 93
- RNPMessages
 - jeod::RNPMessages, 100
- RNP, 13
 - rnp_j2000.cc, 127
 - rnp_j2000.hh, 127
 - rnp_mars.cc, 128
 - rnp_mars.hh, 128, 129
- rnp_type
 - jeod::PlanetRNP, 52
- rotation
 - jeod::PlanetRNP, 53
 - jeod::PlanetRotation, 59
- rotation_j2000.cc, 130
- rotation_j2000.hh, 131
- rotation_mars.cc, 131

- rotation_mars.hh, [132](#)
- RotationJ2000
 - jeod::RotationJ2000, [104](#)
- RotationMars
 - jeod::RotationMars, [108](#)
- SECTODEG
 - jeod::PlanetRotation, [60](#)
- SECTOJULIANDAY
 - jeod::PlanetRotation, [60](#)
- scratch_matrix
 - jeod::PlanetRNP, [53](#)
- set_name
 - jeod::PlanetOrientation, [44](#)
- setup_error
 - jeod::RNPMessages, [102](#)
- theta_gast
 - jeod::RotationJ2000, [106](#)
- time_dyn_ptr
 - jeod::RNPJ2000, [89](#)
 - jeod::RNPMars, [98](#)
- timestamp
 - jeod::RNPJ2000, [85](#)
 - jeod::RNPMars, [94](#)
- tt_ptr
 - jeod::RNPMars, [98](#)
- update_axial_rotation
 - jeod::PlanetRNP, [50](#)
 - jeod::RNPJ2000, [86](#)
 - jeod::RNPMars, [95](#)
- update_rnp
 - jeod::PlanetRNP, [50](#)
 - jeod::RNPJ2000, [86](#)
 - jeod::RNPMars, [95](#)
- update_rotation
 - jeod::NutationJ2000, [22](#)
 - jeod::NutationMars, [36](#)
 - jeod::PlanetRotation, [56](#)
 - jeod::PolarMotionJ2000, [65](#)
 - jeod::PrecessionJ2000, [75](#)
 - jeod::PrecessionMars, [79](#)
 - jeod::RotationJ2000, [105](#)
 - jeod::RotationMars, [109](#)
- update_time
 - jeod::PlanetRotation, [57](#)
- use_full_rnp
 - jeod::RotationJ2000, [106](#)
 - jeod::RotationMars, [110](#)
- warn_table
 - jeod::PolarMotionJ2000, [66](#)
- xp
 - jeod::PolarMotionJ2000, [67](#)
 - jeod::PolarMotionJ2000Init, [71](#)
- xp_tbl
 - jeod::PolarMotionJ2000, [67](#)
- jeod::PolarMotionJ2000Init, [71](#)
- xpyp_daily.cc, [132](#)
 - JEOD_FRIEND_CLASS, [133](#)
- xpyp_daily.hh, [133](#)
- xpyp_monthly.cc, [133](#)
 - JEOD_FRIEND_CLASS, [133](#)
- xpyp_monthly.hh, [134](#)
- yp
 - jeod::PolarMotionJ2000, [67](#)
 - jeod::PolarMotionJ2000Init, [71](#)
- yp_tbl
 - jeod::PolarMotionJ2000, [67](#)
 - jeod::PolarMotionJ2000Init, [72](#)