

Rotation,Nutation,andPrecessionModel

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:42:27

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	21
8.1.2.2	~NutationJ2000	21
8.1.2.3	NutationJ2000	21
8.1.3	Member Function Documentation	21
8.1.3.1	initialize	21
8.1.3.2	operator=	21
8.1.3.3	update_rotation	21
8.1.4	Friends And Related Function Documentation	21
8.1.4.1	init_attrjeod__NutationJ2000	21
8.1.4.2	InputProcessor	22
8.1.5	Field Documentation	22
8.1.5.1	D	22
8.1.5.2	D_coeffs	22
8.1.5.3	epsilon_bar	22
8.1.5.4	equa_of_equi	22
8.1.5.5	F	22
8.1.5.6	F_coeffs	22
8.1.5.7	L	23
8.1.5.8	L_coeffs	23
8.1.5.9	long_coeffs	23
8.1.5.10	long_t_coeffs	23
8.1.5.11	M	23
8.1.5.12	M_coeffs	23
8.1.5.13	num_coeffs	24
8.1.5.14	nutration_in_longitude	24
8.1.5.15	nutration_in_obliquity	24
8.1.5.16	obliq_coeffs	24
8.1.5.17	obliq_t_coeffs	24
8.1.5.18	omega	24
8.1.5.19	omega_coeffs	24
8.2	jeod::NutationJ2000Init Class Reference	25
8.2.1	Detailed Description	26
8.2.2	Constructor & Destructor Documentation	26
8.2.2.1	NutationJ2000Init	26
8.2.2.2	~NutationJ2000Init	26
8.2.2.3	NutationJ2000Init	26

8.2.3	Member Function Documentation	26
8.2.3.1	operator=	26
8.2.4	Friends And Related Function Documentation	26
8.2.4.1	init_attrjeod__NutationJ2000Init	26
8.2.4.2	InputProcessor	26
8.2.5	Field Documentation	26
8.2.5.1	D_coeffs	26
8.2.5.2	F_coeffs	27
8.2.5.3	L_coeffs	27
8.2.5.4	long_coeffs	27
8.2.5.5	long_t_coeffs	27
8.2.5.6	M_coeffs	27
8.2.5.7	num_coeffs	28
8.2.5.8	obliq_coeffs	28
8.2.5.9	obliq_t_coeffs	28
8.2.5.10	omega_coeffs	28
8.3	jeod::NutationJ2000Init_nutation_j2000_default_data Class Reference	28
8.3.1	Detailed Description	28
8.3.2	Member Function Documentation	29
8.3.2.1	initialize	29
8.4	jeod::NutationMars Class Reference	29
8.4.1	Detailed Description	30
8.4.2	Constructor & Destructor Documentation	30
8.4.2.1	NutationMars	30
8.4.2.2	~NutationMars	30
8.4.2.3	NutationMars	30
8.4.3	Member Function Documentation	30
8.4.3.1	operator=	30
8.4.3.2	update_rotation	31
8.4.4	Friends And Related Function Documentation	31
8.4.4.1	init_attrjeod__NutationMars	31
8.4.4.2	InputProcessor	31
8.4.5	Field Documentation	31
8.4.5.1	I_at_j2000	31
8.4.5.2	I_dot	31
8.4.5.3	I_m_orig	31
8.4.5.4	int_to_double	31
8.4.5.5	mean_anomaly_j2000	32
8.4.5.6	mean_motion	32
8.4.5.7	nutration_in_longitude	32

8.4.5.8	nutation_in_obliquity	32
8.4.5.9	obliquity_angle	32
8.4.5.10	psi_m_orig	32
8.4.5.11	q_angle_j2000	33
8.5	jeod::PlanetOrientation Class Reference	33
8.5.1	Detailed Description	34
8.5.2	Constructor & Destructor Documentation	34
8.5.2.1	PlanetOrientation	34
8.5.2.2	~PlanetOrientation	34
8.5.2.3	PlanetOrientation	35
8.5.3	Member Function Documentation	35
8.5.3.1	activate	35
8.5.3.2	deactivate	35
8.5.3.3	ephem_activate	35
8.5.3.4	ephem_build_tree	35
8.5.3.5	ephem_initialize	35
8.5.3.6	get_name	36
8.5.3.7	initialize	36
8.5.3.8	operator=	36
8.5.3.9	set_name	36
8.5.4	Friends And Related Function Documentation	36
8.5.4.1	init_attrjeod__PlanetOrientation	36
8.5.4.2	InputProcessor	36
8.5.5	Field Documentation	36
8.5.5.1	active	36
8.5.5.2	name	37
8.5.5.3	orient_interface	37
8.5.5.4	planet	37
8.5.5.5	planet_omega	37
8.5.5.6	planet_rot_state	37
8.6	jeod::PlanetRNP Class Reference	38
8.6.1	Detailed Description	39
8.6.2	Member Enumeration Documentation	39
8.6.2.1	RNPFidelity	39
8.6.3	Constructor & Destructor Documentation	39
8.6.3.1	PlanetRNP	39
8.6.3.2	~PlanetRNP	40
8.6.3.3	PlanetRNP	40
8.6.4	Member Function Documentation	40
8.6.4.1	get_name	40

8.6.4.2	<code>operator=</code>	40
8.6.4.3	<code>propagate_rnp</code>	40
8.6.4.4	<code>update_axial_rotation</code>	40
8.6.4.5	<code>update_rnp</code>	40
8.6.5	Friends And Related Function Documentation	41
8.6.5.1	<code>init_attrjeod__PlanetRNP</code>	41
8.6.5.2	<code>InputProcessor</code>	41
8.6.6	Field Documentation	41
8.6.6.1	<code>enable_polar</code>	41
8.6.6.2	<code>NP_matrix</code>	41
8.6.6.3	<code>nutaton</code>	41
8.6.6.4	<code>polar_motion</code>	41
8.6.6.5	<code>precession</code>	42
8.6.6.6	<code>rnp_type</code>	42
8.6.6.7	<code>rotation</code>	42
8.6.6.8	<code>scratch_matrix</code>	42
8.7	<code>jeod::PlanetRotation</code> Class Reference	42
8.7.1	Detailed Description	44
8.7.2	Constructor & Destructor Documentation	44
8.7.2.1	<code>PlanetRotation</code>	44
8.7.2.2	<code>~PlanetRotation</code>	44
8.7.2.3	<code>PlanetRotation</code>	44
8.7.3	Member Function Documentation	44
8.7.3.1	<code>get_rotation</code>	44
8.7.3.2	<code>get_rotation_transpose</code>	44
8.7.3.3	<code>initialize</code>	44
8.7.3.4	<code>operator=</code>	45
8.7.3.5	<code>update_rotation</code>	45
8.7.3.6	<code>update_time</code>	45
8.7.4	Friends And Related Function Documentation	45
8.7.4.1	<code>init_attrjeod__PlanetRotation</code>	45
8.7.4.2	<code>InputProcessor</code>	45
8.7.5	Field Documentation	45
8.7.5.1	<code>current_time</code>	45
8.7.5.2	<code>DAYTOJULIANCENT</code>	46
8.7.5.3	<code>DEGTORAD</code>	46
8.7.5.4	<code>DEGTOSEC</code>	46
8.7.5.5	<code>JULIANCENTTODAY</code>	46
8.7.5.6	<code>JULIANDAYTOSEC</code>	46
8.7.5.7	<code>RADTODEG</code>	46

8.7.5.8	rotation	47
8.7.5.9	SECTODEG	47
8.7.5.10	SECTOJULIANDAY	47
8.8	jeod::PlanetRotationInit Class Reference	47
8.8.1	Detailed Description	48
8.8.2	Constructor & Destructor Documentation	48
8.8.2.1	PlanetRotationInit	48
8.8.2.2	~PlanetRotationInit	48
8.8.2.3	PlanetRotationInit	48
8.8.3	Member Function Documentation	48
8.8.3.1	operator=	48
8.8.4	Friends And Related Function Documentation	48
8.8.4.1	init_attrjeod__PlanetRotationInit	48
8.8.4.2	InputProcessor	48
8.9	jeod::PolarMotionJ2000 Class Reference	48
8.9.1	Detailed Description	50
8.9.2	Constructor & Destructor Documentation	50
8.9.2.1	PolarMotionJ2000	50
8.9.2.2	~PolarMotionJ2000	50
8.9.2.3	PolarMotionJ2000	50
8.9.3	Member Function Documentation	50
8.9.3.1	initialize	50
8.9.3.2	operator=	50
8.9.3.3	update_rotation	50
8.9.4	Friends And Related Function Documentation	51
8.9.4.1	init_attrjeod__PolarMotionJ2000	51
8.9.4.2	InputProcessor	51
8.9.5	Field Documentation	51
8.9.5.1	last_table_index	51
8.9.5.2	override_table	51
8.9.5.3	polar_mjd	51
8.9.5.4	warn_table	51
8.9.5.5	xp	51
8.9.5.6	xp_tbl	52
8.9.5.7	yp	52
8.9.5.8	yp_tbl	52
8.10	jeod::PolarMotionJ2000Init Class Reference	52
8.10.1	Detailed Description	53
8.10.2	Constructor & Destructor Documentation	53
8.10.2.1	PolarMotionJ2000Init	53

8.10.2.2	~PolarMotionJ2000Init	53
8.10.2.3	PolarMotionJ2000Init	54
8.10.3	Member Function Documentation	54
8.10.3.1	operator=	54
8.10.4	Friends And Related Function Documentation	54
8.10.4.1	init_attrjeod__PolarMotionJ2000Init	54
8.10.4.2	InputProcessor	54
8.10.5	Field Documentation	54
8.10.5.1	last_table_index	54
8.10.5.2	override_table	54
8.10.5.3	polar_mjd	54
8.10.5.4	xp	54
8.10.5.5	xp_tbl	55
8.10.5.6	yp	55
8.10.5.7	yp_tbl	55
8.11	jeod::PolarMotionJ2000Init_xpyp_daily_default_data Class Reference	55
8.11.1	Detailed Description	55
8.11.2	Member Function Documentation	55
8.11.2.1	initialize	55
8.12	jeod::PolarMotionJ2000Init_xpyp_monthly_default_data Class Reference	56
8.12.1	Detailed Description	56
8.12.2	Member Function Documentation	56
8.12.2.1	initialize	56
8.13	jeod::PrecessionJ2000 Class Reference	56
8.13.1	Detailed Description	57
8.13.2	Constructor & Destructor Documentation	57
8.13.2.1	PrecessionJ2000	57
8.13.2.2	~PrecessionJ2000	57
8.13.2.3	PrecessionJ2000	57
8.13.3	Member Function Documentation	57
8.13.3.1	operator=	57
8.13.3.2	update_rotation	57
8.13.4	Friends And Related Function Documentation	58
8.13.4.1	init_attrjeod__PrecessionJ2000	58
8.13.4.2	InputProcessor	58
8.14	jeod::PrecessionMars Class Reference	58
8.14.1	Detailed Description	59
8.14.2	Constructor & Destructor Documentation	59
8.14.2.1	PrecessionMars	59
8.14.2.2	~PrecessionMars	59

8.14.2.3	PrecessionMars	59
8.14.3	Member Function Documentation	59
8.14.3.1	compute_fixed_matrices	59
8.14.3.2	operator=	60
8.14.3.3	update_rotation	60
8.14.4	Friends And Related Function Documentation	60
8.14.4.1	init_attrjeod__PrecessionMars	60
8.14.4.2	InputProcessor	60
8.14.5	Field Documentation	60
8.14.5.1	J	60
8.14.5.2	N	60
8.14.5.3	NJ_matrix	60
8.14.5.4	nutaton	60
8.14.5.5	psi_at_j2000	61
8.14.5.6	psi_dot	61
8.14.5.7	psi_precess	61
8.15	jeod::RNPJ2000 Class Reference	61
8.15.1	Detailed Description	63
8.15.2	Constructor & Destructor Documentation	63
8.15.2.1	RNPJ2000	63
8.15.2.2	~RNPJ2000	63
8.15.2.3	RNPJ2000	63
8.15.3	Member Function Documentation	63
8.15.3.1	ephem_update	63
8.15.3.2	get_dyn_time_ptr	63
8.15.3.3	get_name	63
8.15.3.4	initialize	64
8.15.3.5	operator=	64
8.15.3.6	timestamp	64
8.15.3.7	update_axial_rotation	64
8.15.3.8	update_rnp	64
8.15.4	Friends And Related Function Documentation	65
8.15.4.1	init_attrjeod__RNPJ2000	65
8.15.4.2	InputProcessor	65
8.15.5	Field Documentation	65
8.15.5.1	gmst_ptr	65
8.15.5.2	internal_name	65
8.15.5.3	last_updated_time_full	65
8.15.5.4	last_updated_time_rotational	65
8.15.5.5	never_updated_full	66

8.15.5.6	never_updated_rotational	66
8.15.5.7	NJ2000	66
8.15.5.8	PJ2000	66
8.15.5.9	PMJ2000	66
8.15.5.10	RJ2000	66
8.15.5.11	time_dyn_ptr	67
8.16	jeod::RNPJ2000_rnp_j2000_default_data Class Reference	67
8.16.1	Detailed Description	67
8.16.2	Member Function Documentation	67
8.16.2.1	initialize	67
8.17	jeod::RNPMars Class Reference	67
8.17.1	Detailed Description	69
8.17.2	Constructor & Destructor Documentation	69
8.17.2.1	RNPMars	69
8.17.2.2	~RNPMars	69
8.17.2.3	RNPMars	69
8.17.3	Member Function Documentation	69
8.17.3.1	ephem_update	69
8.17.3.2	get_dyn_time_ptr	70
8.17.3.3	get_name	70
8.17.3.4	initialize	70
8.17.3.5	operator=	70
8.17.3.6	timestamp	70
8.17.3.7	update_axial_rotation	70
8.17.3.8	update_rnp	71
8.17.4	Friends And Related Function Documentation	71
8.17.4.1	init_attrjeod__RNPMars	71
8.17.4.2	InputProcessor	71
8.17.5	Field Documentation	71
8.17.5.1	internal_name	71
8.17.5.2	last_updated_time_full	71
8.17.5.3	last_updated_time_rotational	71
8.17.5.4	never_updated_full	72
8.17.5.5	never_updated_rotational	72
8.17.5.6	NMars	72
8.17.5.7	PMars	72
8.17.5.8	RMars	72
8.17.5.9	time_dyn_ptr	72
8.17.5.10	tt_ptr	73
8.18	jeod::RNPMars_rnp_mars_default_data Class Reference	73

8.18.1 Detailed Description	73
8.18.2 Member Function Documentation	73
8.18.2.1 initialize	73
8.19 jeod::RNPMessages Class Reference	73
8.19.1 Detailed Description	74
8.19.2 Constructor & Destructor Documentation	74
8.19.2.1 RNPMessages	74
8.19.2.2 RNPMessages	74
8.19.3 Member Function Documentation	74
8.19.3.1 operator=	74
8.19.4 Friends And Related Function Documentation	74
8.19.4.1 init_attrjeod__RNPMessages	74
8.19.4.2 InputProcessor	74
8.19.5 Field Documentation	74
8.19.5.1 fidelity_error	74
8.19.5.2 initialization_error	75
8.19.5.3 polar_motion_table_warning	75
8.19.5.4 setup_error	75
8.20 jeod::RotationJ2000 Class Reference	76
8.20.1 Detailed Description	76
8.20.2 Constructor & Destructor Documentation	77
8.20.2.1 RotationJ2000	77
8.20.2.2 ~RotationJ2000	77
8.20.2.3 RotationJ2000	77
8.20.3 Member Function Documentation	77
8.20.3.1 operator=	77
8.20.3.2 update_rotation	77
8.20.4 Friends And Related Function Documentation	77
8.20.4.1 init_attrjeod__RotationJ2000	77
8.20.4.2 InputProcessor	77
8.20.5 Field Documentation	77
8.20.5.1 GMST	77
8.20.5.2 nutation	77
8.20.5.3 planet_rotational_velocity	78
8.20.5.4 theta_gast	78
8.20.5.5 use_full_rnp	78
8.21 jeod::RotationMars Class Reference	78
8.21.1 Detailed Description	79
8.21.2 Constructor & Destructor Documentation	79
8.21.2.1 RotationMars	79

8.21.2.2	~RotationMars	79
8.21.2.3	RotationMars	80
8.21.3	Member Function Documentation	80
8.21.3.1	operator=	80
8.21.3.2	update_rotation	80
8.21.4	Friends And Related Function Documentation	80
8.21.4.1	init_attrjeod__RotationMars	80
8.21.4.2	InputProcessor	80
8.21.5	Field Documentation	80
8.21.5.1	nutation	80
8.21.5.2	phi_at_j2000	80
8.21.5.3	phi_spin	80
8.21.5.4	planet_rotational_velocity	81
8.21.5.5	use_full_rnp	81
9	File Documentation	83
9.1	class_declarations.hh File Reference	83
9.1.1	Detailed Description	83
9.2	class_declarations.hh File Reference	83
9.2.1	Detailed Description	83
9.3	data_nutation_j2000.cc File Reference	83
9.3.1	Macro Definition Documentation	84
9.3.1.1	JEOD_FRIEND_CLASS	84
9.4	data_rnp_j2000.cc File Reference	84
9.4.1	Macro Definition Documentation	84
9.4.1.1	JEOD_FRIEND_CLASS	84
9.5	data_rnp_mars.cc File Reference	85
9.5.1	Macro Definition Documentation	85
9.5.1.1	JEOD_FRIEND_CLASS	85
9.6	nutation_j2000.cc File Reference	85
9.6.1	Detailed Description	85
9.7	nutation_j2000.hh File Reference	86
9.8	nutation_j2000.hh File Reference	86
9.8.1	Detailed Description	86
9.9	nutation_j2000_init.cc File Reference	86
9.9.1	Detailed Description	87
9.10	nutation_j2000_init.hh File Reference	87
9.10.1	Detailed Description	87
9.11	nutation_mars.cc File Reference	87
9.11.1	Detailed Description	87

9.12	nutaton_mars.hh File Reference	88
9.12.1	Detailed Description	88
9.13	planet_orientation.cc File Reference	88
9.13.1	Detailed Description	88
9.14	planet_orientation.hh File Reference	88
9.14.1	Detailed Description	89
9.15	planet_rnp.cc File Reference	89
9.15.1	Detailed Description	89
9.16	planet_rnp.hh File Reference	90
9.16.1	Detailed Description	90
9.17	planet_rotation.cc File Reference	90
9.17.1	Detailed Description	90
9.18	planet_rotation.hh File Reference	90
9.18.1	Detailed Description	91
9.19	planet_rotation_init.cc File Reference	91
9.19.1	Detailed Description	91
9.20	planet_rotation_init.hh File Reference	91
9.20.1	Detailed Description	92
9.21	polar_motion_j2000.cc File Reference	92
9.21.1	Detailed Description	92
9.22	polar_motion_j2000.hh File Reference	92
9.22.1	Detailed Description	92
9.23	polar_motion_j2000_init.cc File Reference	93
9.23.1	Detailed Description	93
9.24	polar_motion_j2000_init.hh File Reference	93
9.24.1	Detailed Description	93
9.25	precession_j2000.cc File Reference	93
9.25.1	Detailed Description	94
9.26	precession_j2000.hh File Reference	94
9.26.1	Detailed Description	94
9.27	precession_mars.cc File Reference	94
9.27.1	Detailed Description	95
9.28	precession_mars.hh File Reference	95
9.28.1	Detailed Description	95
9.29	rnp_j2000.cc File Reference	95
9.29.1	Detailed Description	96
9.30	rnp_j2000.hh File Reference	96
9.31	rnp_j2000.hh File Reference	96
9.31.1	Detailed Description	96
9.32	rnp_mars.cc File Reference	96

9.32.1 Detailed Description	97
9.33 rnp_mars.hh File Reference	97
9.34 rnp_mars.hh File Reference	97
9.34.1 Detailed Description	98
9.35 RNP_messages.cc File Reference	98
9.35.1 Detailed Description	98
9.36 RNP_messages.hh File Reference	98
9.36.1 Detailed Description	99
9.37 rotation_j2000.cc File Reference	99
9.37.1 Detailed Description	99
9.38 rotation_j2000.hh File Reference	99
9.38.1 Detailed Description	99
9.39 rotation_mars.cc File Reference	100
9.39.1 Detailed Description	100
9.40 rotation_mars.hh File Reference	100
9.40.1 Detailed Description	100
9.41 xpyp_daily.cc File Reference	100
9.41.1 Macro Definition Documentation	101
9.41.1.1 JEOD_FRIEND_CLASS	101
9.42 xpyp_daily.hh File Reference	101
9.43 xpyp_monthly.cc File Reference	101
9.43.1 Macro Definition Documentation	102
9.43.1.1 JEOD_FRIEND_CLASS	102
9.44 xpyp_monthly.hh File Reference	102

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	33
jeod::PlanetRNP	38
jeod::RNPJ2000	61
jeod::RNPMars	67
jeod::NutationJ2000Init_nutation_j2000_default_data	28
jeod::PlanetRotation	42
jeod::NutationJ2000	19
jeod::NutationMars	29
jeod::PolarMotionJ2000	48
jeod::PrecessionJ2000	56
jeod::PrecessionMars	58
jeod::RotationJ2000	76
jeod::RotationMars	78
jeod::PlanetRotationInit	47
jeod::NutationJ2000Init	25
jeod::PolarMotionJ2000Init	52
jeod::PolarMotionJ2000Init_xpyp_daily_default_data	55
jeod::PolarMotionJ2000Init_xpyp_monthly_default_data	56
RefFrameOwner	
jeod::PlanetOrientation	33
jeod::RNPJ2000_rnp_j2000_default_data	67
jeod::RNPMars_rnp_mars_default_data	73
jeod::RNPMessages	73

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	25
jeod::NutationJ2000Init_nutation_j2000_default_data		28
jeod::NutationMars	Implements the nutation portion of the "Pathfinder" Mars RNP model	29
jeod::PlanetOrientation	The generic framework for orientation models to interact with a DynManager object	33
jeod::PlanetRNP	The generic framework for orientation models based on the RNP paradigm	38
jeod::PlanetRotation	The generic base class for all planet transformations	42
jeod::PlanetRotationInit	The generic base class for all initializing classes for PlanetRotation derived classes	47
jeod::PolarMotionJ2000	Implements the polar motion portion of the J2000 RNP	48
jeod::PolarMotionJ2000Init	Initializes the PolarMotionJ2000 object	52
jeod::PolarMotionJ2000Init_xpyp_daily_default_data		55
jeod::PolarMotionJ2000Init_xpyp_monthly_default_data		56
jeod::PrecessionJ2000	Implements the precession portion of the J2000 RNP	56
jeod::PrecessionMars	Implements the axial rotation portion of the "Pathfinder" Mars RNP model	58
jeod::RNPJ2000	Implements the J2000 RNP model using the generic RNP framework	61
jeod::RNPJ2000_rnp_j2000_default_data		67
jeod::RNPMars	Implements the "Pathfinder" Mars RNP model using the generic RNP framework	67
jeod::RNPMars_rnp_mars_default_data		73
jeod::RNPMessages	Describes messages used in the RNP model	73
jeod::RotationJ2000	Implements the axial rotation portion of J2000 RNP	76
jeod::RotationMars	Implements the axial rotation portion of the "Pathfinder" Mars RNP model	78

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	83
RNPJ2000/include/class_declarations.hh	Forward declarations of classes defined for JEOD 2.0 J2000 RNP	83
data_nutation_j2000.cc		83
data_rnp_j2000.cc		84
data_rnp_mars.cc		85
nututation_j2000.cc	Implementation for the NutationJ2000 class	85
data/include/nutation_j2000.hh		86
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	86
nututation_j2000_init.cc	Implementation of NutationJ2000Init	86
nututation_j2000_init.hh	Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize . . .	87
nututation_mars.cc	Implementation for the NutationMars class	87
nututation_mars.hh	Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model . . .	88
planet_orientation.cc	Implement PlanetOrientation	88
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 .	88
planet_rnp.cc	Implement PlanetRNP	89
planet_rnp.hh	Establish a framework for rotation-nutation-precession style planet attitude models	90
planet_rotation.cc	Implemenation for PlanetRotation	90
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	90
planet_rotation_init.cc	Implementation for PlanetRotationInit	91

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	91
polar_motion_j2000.cc	Implementation of PolarMotionJ2000	92
polar_motion_j2000.hh	Model the polar motion portion of the RNP routine for the Standard Epoch J2000	92
polar_motion_j2000_init.cc	Implementation for PolarMotionJ2000Init	93
polar_motion_j2000_init.hh	Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize	93
precession_j2000.cc	Implementation for PrecessionJ2000	93
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	94
precession_mars.cc	Implementation of PrecessionMars	94
precession_mars.hh	Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model . .	95
rnp_j2000.cc	Implementation for RNPJ2000	95
data/include/rnp_j2000.hh		96
include/rnp_j2000.hh	A specific implementation of PlanetRNP, for Earth	96
rnp_mars.cc	Implementation of RNPMars	96
data/include/rnp_mars.hh		97
include/rnp_mars.hh	A specific implementation of PlanetRNP, for Mars	97
RNP_messages.cc	Implement RNP_messages	98
RNP_messages.hh	Implement RNP_messages	98
rotation_j2000.cc	Implementation of RotationJ2000	99
rotation_j2000.hh	Model the axial rotation portion of the RNP routine for the Standard Epoch J2000	99
rotation_mars.cc	Implementation of RotationMars	100
rotation_mars.hh	Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model .	100
xpyp_daily.cc		100
xpyp_daily.hh		101
xpyp_monthly.cc		101
xpyp_monthly.hh		102

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](#)
Implemenation 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 `#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 [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 [RNPMessages](#)
Describes messages used in the RNP model.
- class [NutationJ2000Init_nutation_j2000_default_data](#)
- class [RNPJ2000_rnp_j2000_default_data](#)
- class [PolarMotionJ2000Init_xpyp_daily_default_data](#)
- class [PolarMotionJ2000Init_xpyp_monthly_default_data](#)
- 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 [PolarMotionJ2000](#)
Implements the polar motion portion of the J2000 RNP.
- class [PolarMotionJ2000Init](#)
Initializes the [PolarMotionJ2000](#) object.
- class [PrecessionJ2000](#)
Implements the precession portion of the J2000 RNP.
- class [RNPJ2000](#)
Implements the J2000 RNP model using the generic RNP framework.
- class [RotationJ2000](#)
Implements the axial rotation portion of J2000 RNP.
- class [RNPMars_rnp_mars_default_data](#)
- class [NutationMars](#)

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

- class [PrecessionMars](#)

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

- class [RNPMars](#)

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

- class [RotationMars](#)

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

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

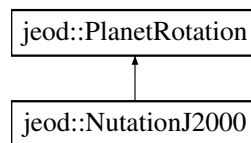
Data Structure Documentation

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

Data Fields

- unsigned 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.
- 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 jeod::NutationJ2000::NutationJ2000 (void)

Constructor.

Initialize all in class data

Definition at line 66 of file nutation_j2000.cc.

8.1.2.2 jeod::NutationJ2000::~~NutationJ2000 (void) [override]

destructor

Definition at line 96 of file nutation_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 jeod::NutationJ2000::NutationJ2000 (const NutationJ2000 & rhs) [private]

8.1.3 Member Function Documentation

8.1.3.1 void jeod::NutationJ2000::initialize (PlanetRotationInit * init) [override],[virtual]

Initialize the various coefficients needed for the calculation of nutation.

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

Parameters

in	init	NutationJ2000Init object with needed coefficients
----	------	-------------------------------------------------------------------

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

Definition at line 267 of file nutation_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 NutationJ2000& jeod::NutationJ2000::operator= (const NutationJ2000 & rhs) [private]

8.1.3.3 void jeod::NutationJ2000::update_rotation (void) [override],[virtual]

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

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

Definition at line 144 of file nutation_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`, `nutation_in_longitude`, `nutation_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 void init_attrjeod__NutationJ2000 () [friend]

8.1.4.2 `friend class InputProcessor` `[friend]`

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

8.1.5 Field Documentation

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

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

trick_units(count)

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

Referenced by initialize(), and update_rotation().

8.1.5.14 double jeod::NutationJ2000::nututation_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 double jeod::NutationJ2000::nututation_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 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 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 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 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:

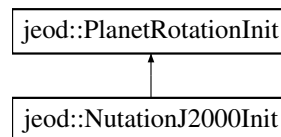
- [include/nutation_j2000.hh](#)
- [nututation_j2000.cc](#)

8.2 jeod::NutationJ2000Init Class Reference

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

```
#include <nututation_j2000_init.hh>
```

Inheritance diagram for jeod::NutationJ2000Init:



Public Member Functions

- [NutationJ2000Init](#) ()
constructor.
- [~NutationJ2000Init](#) () override
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 coefficnets 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 `jeod::NutationJ2000Init::NutationJ2000Init (void)`

constructor.

initialize low level data

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

8.2.2.2 `jeod::NutationJ2000Init::~~NutationJ2000Init (void)` `[override]`

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

8.2.3 Member Function Documentation

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

8.2.4 Friends And Related Function Documentation

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

8.2.4.2 `friend class InputProcessor` `[friend]`

Definition at line 94 of file `nutations_j2000_init.hh`.

8.2.5 Field Documentation

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

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

`trick_units(count)`

Definition at line 102 of file `nutations_j2000_init.hh`.

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

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

Zero order coefficients for calculating the nutation in obliquity.

`trick_units(-)`

Definition at line 141 of file `nutations_j2000_init.hh`.

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

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

First order coefficients for calculating the nutation in obliquity.

`trick_units(-)`

Definition at line 146 of file `nutations_j2000_init.hh`.

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

8.2.5.10 `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 `nutations_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:

- [nutations_j2000_init.hh](#)
- [nutations_j2000_init.cc](#)

8.3 `jeod::NutationJ2000Init_nutation_j2000_default_data` Class Reference

```
#include <nutations_j2000.hh>
```

Public Member Functions

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

8.3.1 Detailed Description

Definition at line 54 of file `data/include/nutations_j2000.hh`.

8.3.2 Member Function Documentation

8.3.2.1 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:

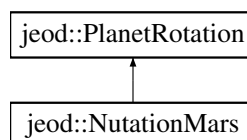
- [data/include/nutation_j2000.hh](#)
- [data_nutation_j2000.cc](#)

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.
- [~NutationMars](#) () override
Destructor.
- void [update_rotation](#) () override
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 `jeod::NutationMars::NutationMars (void)`

Default constructor.

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

8.4.2.2 `jeod::NutationMars::~~NutationMars (void)` `[override]`

Destructor.

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

References `l_m_orig`, `int_to_double`, and `psi_m_orig`.

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

8.4.3 Member Function Documentation

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

8.4.3.2 `void jeod::NutationMars::update_rotation (void) [override],[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 `void init_attrjeod__NutationMars () [friend]`

8.4.4.2 `friend class InputProcessor [friend]`

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

8.4.5 Field Documentation

8.4.5.1 `double jeod::NutationMars::l_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 `double jeod::NutationMars::l_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 `double* jeod::NutationMars::l_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 `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 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 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 double jeod::NutationMars::nutaton_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 double jeod::NutationMars::nutaton_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 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 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 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 `nutaton_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:

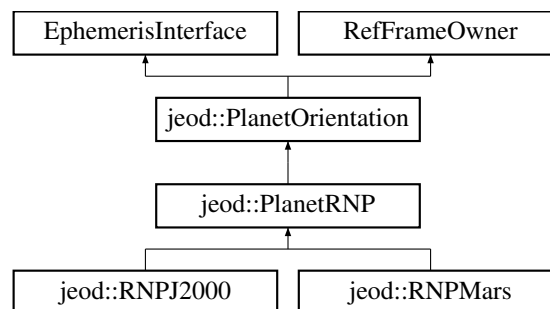
- [nutaton_mars.hh](#)
- [nutaton_mars.cc](#)

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.
- [~PlanetOrientation](#) () override
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.
- void [activate](#) () override
Activates the [PlanetOrientation](#) model.
- void [deactivate](#) () override
Deactivates the [PlanetOrientation](#) model.
- void [ephem_initialize](#) (EphemeridesManager &manager) override
*Implements the *EphemInterface* pure virtual function *ephem_initialize*.*
- void [ephem_activate](#) (EphemeridesManager &manager) override
Mark the model as being activate or inactive.
- void [ephem_build_tree](#) (EphemeridesManager &manager) override
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.

- `const char * get_name () const override=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 `jeod::PlanetOrientation::PlanetOrientation (void)`

Default constructor; constructs a `PlanetOrientation` object.

Definition at line 59 of file planet_orientation.cc.

8.5.2.2 `jeod::PlanetOrientation::~~PlanetOrientation (void) [override]`

Class destructor.

Definition at line 74 of file planet_orientation.cc.

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

Not implemented.

8.5.3 Member Function Documentation

8.5.3.1 void jeod::PlanetOrientation::activate () [override]

Activates the [PlanetOrientation](#) model.

Definition at line 141 of file planet_orientation.cc.

References [active](#).

8.5.3.2 void jeod::PlanetOrientation::deactivate () [override]

Deactivates the [PlanetOrientation](#) model.

Definition at line 150 of file planet_orientation.cc.

References [active](#).

8.5.3.3 void jeod::PlanetOrientation::ephem_activate (EphemeridesManager & manager) [override]

Mark the model as being activate or inactive.

Parameters

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

Definition at line 176 of file planet_orientation.cc.

8.5.3.4 void jeod::PlanetOrientation::ephem_build_tree (EphemeridesManager & manager) [override]

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

This implementation, by default, does nothing

Parameters

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

Definition at line 190 of file planet_orientation.cc.

8.5.3.5 void jeod::PlanetOrientation::ephem_initialize (EphemeridesManager & manager) [override]

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

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

Definition at line 165 of file planet_orientation.cc.

8.5.3.6 `const char* jeod::PlanetOrientation::get_name () const` `[override],[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 `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 `PlanetOrientation& jeod::PlanetOrientation::operator= (const PlanetOrientation & rhs)` `[private]`

Not implemented.

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

Setter for the name.

Definition at line 171 of file planet_orientation.hh.

8.5.4 Friends And Related Function Documentation

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

8.5.4.2 `friend class InputProcessor` `[friend]`

Definition at line 101 of file planet_orientation.hh.

8.5.5 Field Documentation

8.5.5.1 `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 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(), and initialize().

8.5.5.3 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* 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 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 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:

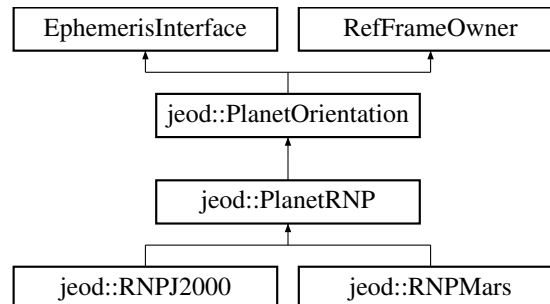
- [planet_orientation.hh](#)
- [planet_orientation.cc](#)

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.
- [~PlanetRNP](#) () override
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.
- const char * [get_name](#) () const override=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 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 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 `jeod::PlanetRNP::~~PlanetRNP (void) [override]`

Class destructor.

Definition at line 88 of file `planet_rnp.cc`.

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

Not implemented.

8.6.4 Member Function Documentation

8.6.4.1 `const char* jeod::PlanetRNP::get_name () const [override],[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 `PlanetRNP& jeod::PlanetRNP::operator= (const PlanetRNP & rhs) [private]`

Not implemented.

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

8.6.5.2 friend class InputProcessor [friend]

Definition at line 106 of file planet_rnp.hh.

8.6.6 Field Documentation

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

Rotation Matrix representing: transpose(nutation->rotation) * transpose(precession->rotation)

trick_units(-)

Definition at line 159 of file planet_rnp.hh.

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

8.6.6.3 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 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 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 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(), 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 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 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:

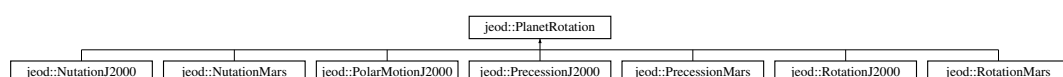
- [planet_rnp.hh](#)
- [planet_rnp.cc](#)

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

Destructor.

Definition at line 76 of file planet_rotation.cc.

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

8.7.3 Member Function Documentation

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

Copy the last calculated rotation to 'rot'.

Parameters

<code>out</code>	<code>rot</code>	Where the rotation matrix will be stored
------------------	------------------	------------------------------------------

Definition at line 98 of file planet_rotation.cc.

References rotation.

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

Same as `get_rotation`, but returns the transpose.

Parameters

<code>out</code>	<code>rot</code>	Where the transpose matrix will be stored
------------------	------------------	-------------------------------------------

Definition at line 110 of file planet_rotation.cc.

References rotation.

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

Initializes the invoking object from an initialization object.

Parameters

<i>in</i>	<i>init</i>	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 PlanetRotation& jeod::PlanetRotation::operator= (const PlanetRotation & rhs) [private]

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

Update the time that the next update_rotation call will use.

Parameters

<i>in</i>	<i>time</i>	Time. units and UTC/UT1/TAI etc are determined by the individual Planet-Rotation 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 void init_attrjeod__PlanetRotation () [friend]

8.7.4.2 friend class InputProcessor [friend]

Definition at line 94 of file planet_rotation.hh.

8.7.5 Field Documentation

8.7.5.1 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 modeltrick_units(–)

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 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 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 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 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 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 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 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 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 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:

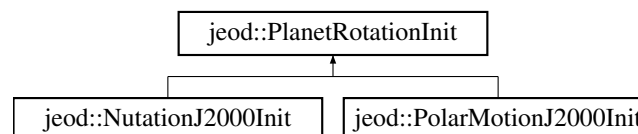
- [planet_rotation.hh](#)
- [planet_rotation.cc](#)

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

constructor

Definition at line 46 of file [planet_rotation_init.cc](#).

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

destructor

Definition at line 55 of file [planet_rotation_init.cc](#).

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

8.8.3 Member Function Documentation

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

8.8.4 Friends And Related Function Documentation

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

8.8.4.2 `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:

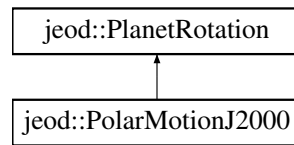
- [planet_rotation_init.hh](#)
- [planet_rotation_init.cc](#)

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.
- [~PolarMotionJ2000](#) () override
Destructor.
- void [update_rotation](#) () override
[PolarMotionJ2000](#) specific implementation of [update_rotation](#) from [PlanetRotation](#).
- void [initialize](#) ([PlanetRotationInit](#) *init) override
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 jeod::PolarMotionJ2000::PolarMotionJ2000 (void)

Default constructor.

Initializes all data.

Definition at line 63 of file polar_motion_j2000.cc.

8.9.2.2 jeod::PolarMotionJ2000::~~PolarMotionJ2000 (void) [override]

Destructor.

Definition at line 81 of file polar_motion_j2000.cc.

References polar_mjd, xp_tbl, and yp_tbl.

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

8.9.3 Member Function Documentation

8.9.3.1 void jeod::PolarMotionJ2000::initialize (PlanetRotationInit * init) [override],[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 PolarMotionJ2000& jeod::PolarMotionJ2000::operator= (const PolarMotionJ2000 & rhs) [private]

8.9.3.3 void jeod::PolarMotionJ2000::update_rotation (void) [override],[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 void init_attrjeod__PolarMotionJ2000 () [friend]

8.9.4.2 friend class InputProcessor [friend]

Definition at line 92 of file polar_motion_j2000.hh.

8.9.5 Field Documentation

8.9.5.1 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 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 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 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 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 `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 `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 `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:

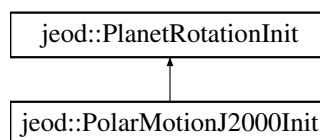
- [polar_motion_j2000.hh](#)
- [polar_motion_j2000.cc](#)

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.
- [~PolarMotionJ2000Init](#) () *override*
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 jeod::PolarMotionJ2000Init::PolarMotionJ2000Init (void)

constructor.

Initializes all data.

Definition at line 54 of file polar_motion_j2000_init.cc.

8.10.2.2 jeod::PolarMotionJ2000Init::~~PolarMotionJ2000Init (void) [override]

Destructor.

Definition at line 71 of file polar_motion_j2000_init.cc.

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

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

8.10.3 Member Function Documentation

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

8.10.4 Friends And Related Function Documentation

8.10.4.1 `void init_attrjeod__PolarMotionJ2000Init () [friend]`

8.10.4.2 `friend class InputProcessor [friend]`

Definition at line 92 of file `polar_motion_j2000_init.hh`.

8.10.5 Field Documentation

8.10.5.1 `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 `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 `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 `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 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 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 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 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 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:

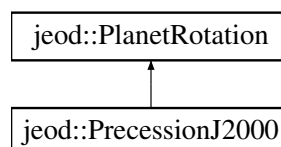
- [xpyp_monthly.hh](#)
- [xpyp_monthly.cc](#)

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
- [~PrecessionJ2000](#) () *override*
destructor

- void [update_rotation](#) () override

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

constructor

Definition at line 47 of file `precession_j2000.cc`.

8.13.2.2 `jeod::PrecessionJ2000::~~PrecessionJ2000 (void) [override]`

destructor

Definition at line 55 of file `precession_j2000.cc`.

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

8.13.3 Member Function Documentation

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

8.13.3.2 `void jeod::PrecessionJ2000::update_rotation (void) [override],[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::DEGTOSEC`, and `jeod::PlanetRotation::rotation`.

8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod__PrecessionJ2000 () [friend]`

8.13.4.2 `friend class InputProcessor [friend]`

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

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

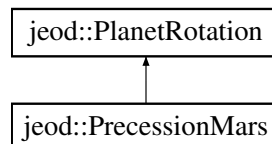
- [precession_j2000.hh](#)
- [precession_j2000.cc](#)

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.
- [~PrecessionMars](#) () override
Destructor.
- void [update_rotation](#) () override
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](#) * [nutaton](#)
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 `jeod::PrecessionMars::PrecessionMars (void)`

Default constructor.

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

References `NJ_matrix`.

8.14.2.2 `jeod::PrecessionMars::~~PrecessionMars (void)` `[override]`

Destructor.

Definition at line 78 of file `precession_mars.cc`.

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

8.14.3 Member Function Documentation

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

8.14.3.3 `void jeod::PrecessionMars::update_rotation (void)` `[override],[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 `void init_attrjeod__PrecessionMars ()` `[friend]`

8.14.4.2 `friend class InputProcessor` `[friend]`

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

8.14.5 Field Documentation

8.14.5.1 `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 `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 `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 `NutationMars* jeod::PrecessionMars::nutaton`

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 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 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 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:

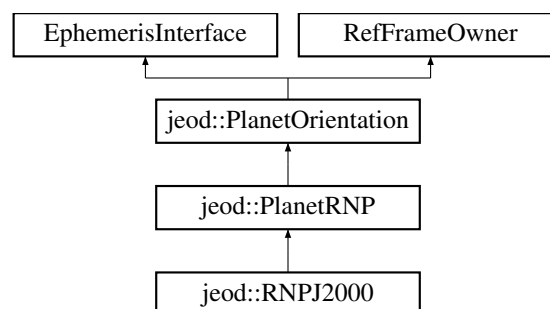
- [precession_mars.hh](#)
- [precession_mars.cc](#)

8.15 jeod::RNPJ2000 Class Reference

Implements the J2000 RNP model using the generic RNP framework.

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

Inheritance diagram for jeod::RNPJ2000:



Public Member Functions

- [RNPJ2000](#) ()
default constructor.
- [~RNPJ2000](#) () override

Destructor.

- void `initialize` (DynManager &manager) override
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` (const TimeTT &time_tt, TimeGMST &time_gmst, const 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.
- double `timestamp` () const override
- const char * `get_name` () const override
A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.
- void `ephem_update` () override

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 `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, 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 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 jeod::RNPJ2000::~~RNPJ2000 (void) [override]

Destructor.

Definition at line 95 of file rnp_j2000.cc.

8.15.2.3 jeod::RNPJ2000::RNPJ2000 (const RNPJ2000 & rhs) [private]

8.15.3 Member Function Documentation

8.15.3.1 void jeod::RNPJ2000::ephem_update () [override]

Definition at line 325 of file rnp_j2000.cc.

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

8.15.3.2 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 const char * jeod::RNPJ2000::get_name () const [override], [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 `void jeod::RNPJ2000::initialize (DynManager & dyn_manager) [override], [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 `RNPJ2000& jeod::RNPJ2000::operator= (const RNPJ2000 & rhs) [private]`

8.15.3.6 `double jeod::RNPJ2000::timestamp () const [override]`

Definition at line 308 of file rnp_j2000.cc.

References `last_updated_time_rotational`.

8.15.3.7 `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

<code>in</code>	<code>time_gmst</code>	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 `void jeod::RNPJ2000::update_rnp (const TimeTT & time_tt, TimeGMST & time_gmst, const 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, jeod::PlanetRNP::nutaton, 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 void init_attrjeod_RNPJ2000 () [friend]

8.15.4.2 friend class InputProcessor [friend]

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

8.15.5 Field Documentation

8.15.5.1 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 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 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 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 `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 `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 `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 `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 `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 `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 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 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:

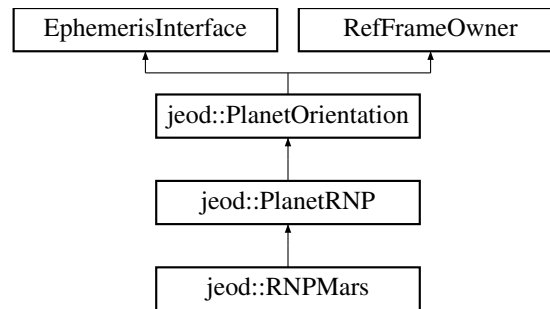
- [data/include/rnp_j2000.hh](#)
- [data_rnp_j2000.cc](#)

8.17 jeod::RNPMars Class Reference

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

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

Inheritance diagram for jeod::RNPMars:



Public Member Functions

- [RNPMars](#) ()
Default constructor.
- [~RNPMars](#) () override
Destructor.
- void [initialize](#) (DynManager &manager) override
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.
- double [timestamp](#) () const override
Return the last time at which the RNP was updated.
- const char * [get_name](#) () const override
Return the internal name of the object.
- void [ephem_update](#) () override
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 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 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 jeod::RNPMars::RNPMars (void)

Default constructor.

Definition at line 68 of file rnp_mars.cc.

References [jeod::PlanetRNP::enable_polar](#), [NMars](#), [jeod::PlanetRNP::nutration](#), [PMars](#), [jeod::PlanetRNP::polar_motion](#), [jeod::PlanetRNP::precession](#), [RMars](#), and [jeod::PlanetRNP::rotation](#).

8.17.2.2 jeod::RNPMars::~~RNPMars () [override]

Destructor.

Definition at line 98 of file rnp_mars.cc.

8.17.2.3 jeod::RNPMars::RNPMars (const RNPMars & rhs) [private]

8.17.3 Member Function Documentation

8.17.3.1 void jeod::RNPMars::ephemeris_update () [override]

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

Get simulation time via a two-step pointer trail.

Parameters

<code>in</code>	<code><i>time_tt</i></code>	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 `const char * jeod::RNPMars::get_name () const` `[override],[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 `void jeod::RNPMars::initialize (DynManager & dyn_manager)` `[override],[virtual]`

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

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></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::nutation`, `jeod::RotationMars::nutation`, `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 `RNPMars& jeod::RNPMars::operator= (const RNPMars & rhs)` `[private]`

8.17.3.6 `double jeod::RNPMars::timestamp () const` `[override]`

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

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

Parameters

<i>in</i>	<i>time_tt</i>	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::rotation, time_dyn_ptr, jeod::PlanetRNP::update_axial_rotation(), and jeod::PlanetRotation::update_time().

8.17.3.8 void jeod::RNPMars::update_rnp (TimeTT & *time_tt*)

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

Parameters

<i>in</i>	<i>time_tt</i>	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 void init_attrjeod__RNPMars () [friend]

8.17.4.2 friend class InputProcessor [friend]

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

8.17.5 Field Documentation

8.17.5.1 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 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 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.

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

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

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

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

trick_units(-)

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

Referenced by ephemeris_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 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 `jeod::RNPMessages::RNPMessages (void)` [private]

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

8.19.3 Member Function Documentation

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

8.19.4 Friends And Related Function Documentation

8.19.4.1 `void init_attrjeod__RNPMessages ()` [friend]

8.19.4.2 `friend class InputProcessor` [friend]

Definition at line 86 of file RNP_messages.hh.

8.19.5 Field Documentation

8.19.5.1 `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 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 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 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::RNPMessages::ephem_update(), jeod::RNPMars::ephem_update(), jeod::RNPMessages::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:

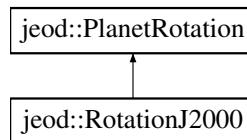
- [RNP_messages.hh](#)
- [RNP_messages.cc](#)

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
- [~RotationJ2000](#) () override
destructor
- void [update_rotation](#) () override
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 jeod::RotationJ2000::RotationJ2000 (void)

default constructor, initialize low level data

Definition at line 60 of file rotation_j2000.cc.

8.20.2.2 jeod::RotationJ2000::~~RotationJ2000 (void) [override]

destructor

Definition at line 75 of file rotation_j2000.cc.

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

8.20.3 Member Function Documentation

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

8.20.3.2 void jeod::RotationJ2000::update_rotation (void) [override],[virtual]

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

For axial rotation

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

Definition at line 86 of file rotation_j2000.cc.

References [jeod::PlanetRotation::current_time](#), [jeod::PlanetRotation::DEGTORAD](#), [jeod::NutationJ2000::equa_of_equi](#), [nutration](#), [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 void init_attrjeod__RotationJ2000 () [friend]

8.20.4.2 friend class InputProcessor [friend]

Definition at line 94 of file rotation_j2000.hh.

8.20.5 Field Documentation

8.20.5.1 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 NutationJ2000* jeod::RotationJ2000::nutration

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](#) object [trick_units\(–\)](#)

Definition at line 108 of file rotation_j2000.hh.

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

8.20.5.3 `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 `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 `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 class `trick_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:

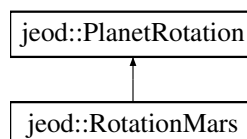
- [rotation_j2000.hh](#)
- [rotation_j2000.cc](#)

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.
- [~RotationMars](#) () override
Destructor.

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

Data Fields

- double [planet_rotational_velocity](#)
The Mars average axial rotational velocity.
- [NutationMars](#) * [nututation](#)
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 jeod::RotationMars::RotationMars (void)

Default constructor, initialize low level data.
Definition at line 61 of file `rotation_mars.cc`.

8.21.2.2 jeod::RotationMars::~~RotationMars (void) [override]

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

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

8.21.3 Member Function Documentation

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

8.21.3.2 `void jeod::RotationMars::update_rotation (void) [override],[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 `void init_attrjeod__RotationMars () [friend]`

8.21.4.2 `friend class InputProcessor [friend]`

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

8.21.5 Field Documentation

8.21.5.1 `NutationMars* jeod::RotationMars::nutaton`

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 `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 `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 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 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.

Definition in file [GenericRNP/include/class_declarations.hh](#).

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.

Definition in file [RNPJ2000/include/class_declarations.hh](#).

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 `#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 `#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 #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.

Definition in file [nutation_j2000.cc](#).

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.

Definition in file [include/nutation_j2000.hh](#).

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.

Definition in file [nutation_j2000_init.cc](#).

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.

Definition in file [nutation_j2000_init.hh](#).

9.11 nutation_mars.cc File Reference

Implementation for the NutationMars class.

```
#include <cstdlib>
#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.

Definition in file [nutation_mars.cc](#).

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.

Definition in file [nutation_mars.hh](#).

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.

Definition in file [planet_orientation.cc](#).

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.

Definition in file [planet_orientation.hh](#).

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.

Definition in file [planet_rnp.cc](#).

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.

Definition in file [planet_rnp.hh](#).

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.

Definition in file [planet_rotation.cc](#).

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.

Definition in file [planet_rotation.hh](#).

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.

Definition in file [planet_rotation_init.cc](#).

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.

Definition in file [planet_rotation_init.hh](#).

9.21 polar_motion_j2000.cc File Reference

Implementation of PolarMotionJ2000.

```
#include <cstddef>
#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.

Definition in file [polar_motion_j2000.cc](#).

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

Definition in file [polar_motion_j2000.hh](#).

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.

Definition in file [polar_motion_j2000_init.cc](#).

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.

Definition in file [polar_motion_j2000_init.hh](#).

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.

Definition in file [precession_j2000.cc](#).

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.

Definition in file [precession_j2000.hh](#).

9.27 precession_mars.cc File Reference

Implementation of PrecessionMars.

```
#include <cstdlib>
#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.

Definition in file [precession_mars.cc](#).

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 "nututation_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.

Definition in file [precession_mars.hh](#).

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.

Definition in file [rnp_j2000.cc](#).

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 "nutation_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

Definition in file [include/rnp_j2000.hh](#).

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.

Definition in file [rnp_mars.cc](#).

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.

Definition in file [include/rnp_mars.hh](#).

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.

Definition in file [RNP_messages.cc](#).

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.

Definition in file [RNP_messages.hh](#).

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.

Definition in file [rotation_j2000.cc](#).

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

Definition in file [rotation_j2000.hh](#).

9.39 rotation_mars.cc File Reference

Implementation of RotationMars.

```
#include <cstdint>
#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.

Definition in file [rotation_mars.cc](#).

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.

Definition in file [rotation_mars.hh](#).

9.41 xpyd_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 `#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 `#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, [26](#)
- ~NutationMars
 - jeod::NutationMars, [30](#)
- ~PlanetOrientation
 - jeod::PlanetOrientation, [34](#)
- ~PlanetRNP
 - jeod::PlanetRNP, [39](#)
- ~PlanetRotation
 - jeod::PlanetRotation, [44](#)
- ~PlanetRotationInit
 - jeod::PlanetRotationInit, [48](#)
- ~PolarMotionJ2000
 - jeod::PolarMotionJ2000, [50](#)
- ~PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, [53](#)
- ~PrecessionJ2000
 - jeod::PrecessionJ2000, [57](#)
- ~PrecessionMars
 - jeod::PrecessionMars, [59](#)
- ~RNPJ2000
 - jeod::RNPJ2000, [63](#)
- ~RNPMars
 - jeod::RNPMars, [69](#)
- ~RotationJ2000
 - jeod::RotationJ2000, [77](#)
- ~RotationMars
 - jeod::RotationMars, [79](#)
- activate
 - jeod::PlanetOrientation, [35](#)
- active
 - jeod::PlanetOrientation, [36](#)
- class_declarations.hh, [83](#)
- compute_fixed_matrices
 - jeod::PrecessionMars, [59](#)
- ConstantNP
 - jeod::PlanetRNP, [39](#)
- current_time
 - jeod::PlanetRotation, [45](#)
- D
 - jeod::NutationJ2000, [22](#)
- D_coeffs
 - jeod::NutationJ2000, [22](#)
 - jeod::NutationJ2000Init, [26](#)
- DAYTOJULIANCENT
 - jeod::PlanetRotation, [45](#)
- DEGTORAD
 - jeod::PlanetRotation, [46](#)
- DEGTOTSEC
 - jeod::PlanetRotation, [46](#)
- data_nutation_j2000.cc, [83](#)
- data_rnp_j2000.cc, [84](#)
- data_rnp_mars.cc, [85](#)
- deactivate
 - jeod::PlanetOrientation, [35](#)
- enable_polar
 - jeod::PlanetRNP, [41](#)
- Environment, [12](#)
- ephem_activate
 - jeod::PlanetOrientation, [35](#)
- ephem_build_tree
 - jeod::PlanetOrientation, [35](#)
- ephem_initialize
 - jeod::PlanetOrientation, [35](#)
- ephem_update
 - jeod::RNPJ2000, [63](#)
 - jeod::RNPMars, [69](#)
- epsilon_bar
 - jeod::NutationJ2000, [22](#)
- equa_of_equi
 - jeod::NutationJ2000, [22](#)
- F
 - jeod::NutationJ2000, [22](#)
- F_coeffs
 - jeod::NutationJ2000, [22](#)
 - jeod::NutationJ2000Init, [27](#)
- fidelity_error
 - jeod::RNPMessages, [74](#)
- FullRNP
 - jeod::PlanetRNP, [39](#)
- GMST
 - jeod::RotationJ2000, [77](#)
- GenericRNP, [14](#)
 - PATH, [14](#)
- get_dyn_time_ptr
 - jeod::RNPJ2000, [63](#)
 - jeod::RNPMars, [70](#)
- get_name
 - jeod::PlanetOrientation, [35](#)
 - jeod::PlanetRNP, [40](#)
 - jeod::RNPJ2000, [63](#)
 - jeod::RNPMars, [70](#)

- get_rotation
 - jeod::PlanetRotation, [44](#)
- get_rotation_transpose
 - jeod::PlanetRotation, [44](#)
- gmst_ptr
 - jeod::RNPJ2000, [65](#)
- I_at_j2000
 - jeod::NutationMars, [31](#)
- I_dot
 - jeod::NutationMars, [31](#)
- I_m_orig
 - jeod::NutationMars, [31](#)
- init_attrjeod__NutationJ2000
 - jeod::NutationJ2000, [21](#)
- init_attrjeod__NutationJ2000Init
 - jeod::NutationJ2000Init, [26](#)
- init_attrjeod__NutationMars
 - jeod::NutationMars, [31](#)
- init_attrjeod__PlanetOrientation
 - jeod::PlanetOrientation, [36](#)
- init_attrjeod__PlanetRNP
 - jeod::PlanetRNP, [41](#)
- init_attrjeod__PlanetRotation
 - jeod::PlanetRotation, [45](#)
- init_attrjeod__PlanetRotationInit
 - jeod::PlanetRotationInit, [48](#)
- init_attrjeod__PolarMotionJ2000
 - jeod::PolarMotionJ2000, [51](#)
- init_attrjeod__PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, [54](#)
- init_attrjeod__PrecessionJ2000
 - jeod::PrecessionJ2000, [58](#)
- init_attrjeod__PrecessionMars
 - jeod::PrecessionMars, [60](#)
- init_attrjeod__RNPJ2000
 - jeod::RNPJ2000, [65](#)
- init_attrjeod__RNPMars
 - jeod::RNPMars, [71](#)
- init_attrjeod__RNPMessages
 - jeod::RNPMessages, [74](#)
- init_attrjeod__RotationJ2000
 - jeod::RotationJ2000, [77](#)
- init_attrjeod__RotationMars
 - jeod::RotationMars, [80](#)
- initialization_error
 - jeod::RNPMessages, [75](#)
- initialize
 - jeod::NutationJ2000, [21](#)
 - jeod::NutationJ2000Init_nutation_j2000_default_data, [29](#)
 - jeod::PlanetOrientation, [36](#)
 - jeod::PlanetRotation, [44](#)
 - jeod::PolarMotionJ2000, [50](#)
 - jeod::PolarMotionJ2000Init_xpyp_daily_default_data, [55](#)
 - jeod::PolarMotionJ2000Init_xpyp_monthly_default_data, [56](#)
 - jeod::RNPJ2000, [64](#)
 - jeod::RNPJ2000_rnp_j2000_default_data, [67](#)
 - jeod::RNPMars, [70](#)
 - jeod::RNPMars_rnp_mars_default_data, [73](#)
- InputProcessor
 - jeod::NutationJ2000, [21](#)
 - jeod::NutationJ2000Init, [26](#)
 - jeod::NutationMars, [31](#)
 - jeod::PlanetOrientation, [36](#)
 - jeod::PlanetRNP, [41](#)
 - jeod::PlanetRotation, [45](#)
 - jeod::PlanetRotationInit, [48](#)
 - jeod::PolarMotionJ2000, [51](#)
 - jeod::PolarMotionJ2000Init, [54](#)
 - jeod::PrecessionJ2000, [58](#)
 - jeod::PrecessionMars, [60](#)
 - jeod::RNPJ2000, [65](#)
 - jeod::RNPMars, [71](#)
 - jeod::RNPMessages, [74](#)
 - jeod::RotationJ2000, [77](#)
 - jeod::RotationMars, [80](#)
- int_to_double
 - jeod::NutationMars, [31](#)
- internal_name
 - jeod::RNPJ2000, [65](#)
 - jeod::RNPMars, [71](#)
- J
 - jeod::PrecessionMars, [60](#)
- JEOD_FRIEND_CLASS
 - data_nutation_j2000.cc, [84](#)
 - data_rnp_j2000.cc, [84](#)
 - data_rnp_mars.cc, [85](#)
 - xpyp_daily.cc, [101](#)
 - xpyp_monthly.cc, [102](#)
- JULIANCENTTODAY
 - jeod::PlanetRotation, [46](#)
- JULIANDAYTOSEC
 - jeod::PlanetRotation, [46](#)
- jeod, [17](#)
 - jeod::PlanetRNP
 - ConstantNP, [39](#)
 - FullRNP, [39](#)
 - RotationOnly, [39](#)
 - jeod::NutationJ2000, [19](#)
 - ~NutationJ2000, [21](#)
 - D, [22](#)
 - D_coeffs, [22](#)
 - epsilon_bar, [22](#)
 - equa_of_equi, [22](#)
 - F, [22](#)
 - F_coeffs, [22](#)
 - init_attrjeod__NutationJ2000, [21](#)
 - initialize, [21](#)
 - InputProcessor, [21](#)
 - L, [23](#)
 - L_coeffs, [23](#)
 - long_coeffs, [23](#)
 - long_t_coeffs, [23](#)
 - M, [23](#)

- M_coeffs, 23
- num_coeffs, 23
- nutation_in_longitude, 24
- nutation_in_obliquity, 24
- NutationJ2000, 21
- obliq_coeffs, 24
- obliq_t_coeffs, 24
- omega, 24
- omega_coeffs, 24
- operator=, 21
- update_rotation, 21
- jeod::NutationJ2000Init, 25
 - ~NutationJ2000Init, 26
 - D_coeffs, 26
 - F_coeffs, 27
 - init_attrjeod__NutationJ2000Init, 26
 - InputProcessor, 26
 - L_coeffs, 27
 - long_coeffs, 27
 - long_t_coeffs, 27
 - M_coeffs, 27
 - num_coeffs, 27
 - NutationJ2000Init, 26
 - obliq_coeffs, 28
 - obliq_t_coeffs, 28
 - omega_coeffs, 28
 - operator=, 26
- jeod::NutationJ2000Init_nutation_j2000_default_data, 28
 - initialize, 29
- jeod::NutationMars, 29
 - ~NutationMars, 30
 - I_at_j2000, 31
 - I_dot, 31
 - I_m_orig, 31
 - init_attrjeod__NutationMars, 31
 - InputProcessor, 31
 - int_to_double, 31
 - mean_anomaly_j2000, 31
 - mean_motion, 32
 - nutation_in_longitude, 32
 - nutation_in_obliquity, 32
 - NutationMars, 30
 - obliquity_angle, 32
 - operator=, 30
 - psi_m_orig, 32
 - q_angle_j2000, 32
 - update_rotation, 30
- jeod::PlanetOrientation, 33
 - ~PlanetOrientation, 34
 - activate, 35
 - active, 36
 - deactivate, 35
 - ephem_activate, 35
 - ephem_build_tree, 35
 - ephem_initialize, 35
 - get_name, 35
 - init_attrjeod__PlanetOrientation, 36
 - initialize, 36
 - InputProcessor, 36
 - name, 37
 - operator=, 36
 - orient_interface, 37
 - planet, 37
 - planet_omega, 37
 - planet_rot_state, 37
 - PlanetOrientation, 34
 - set_name, 36
- jeod::PlanetRNP, 38
 - ~PlanetRNP, 39
 - enable_polar, 41
 - get_name, 40
 - init_attrjeod__PlanetRNP, 41
 - InputProcessor, 41
 - NP_matrix, 41
 - nutation, 41
 - operator=, 40
 - PlanetRNP, 39, 40
 - polar_motion, 41
 - precession, 41
 - propagate_rnp, 40
 - RNPFidelity, 39
 - rnp_type, 42
 - rotation, 42
 - scratch_matrix, 42
 - update_axial_rotation, 40
 - update_rnp, 40
- jeod::PlanetRotation, 42
 - ~PlanetRotation, 44
 - current_time, 45
 - DAYTOJULIANCENT, 45
 - DEGTORAD, 46
 - DEGTOSEC, 46
 - get_rotation, 44
 - get_rotation_transpose, 44
 - init_attrjeod__PlanetRotation, 45
 - initialize, 44
 - InputProcessor, 45
 - JULIANCENTTODAY, 46
 - JULIANDAYTOSEC, 46
 - operator=, 45
 - PlanetRotation, 44
 - RADTODEG, 46
 - rotation, 46
 - SECTODEG, 47
 - SECTOJULIANDAY, 47
 - update_rotation, 45
 - update_time, 45
- jeod::PlanetRotationInit, 47
 - ~PlanetRotationInit, 48
 - init_attrjeod__PlanetRotationInit, 48
 - InputProcessor, 48
 - operator=, 48
 - PlanetRotationInit, 48
- jeod::PolarMotionJ2000, 48
 - ~PolarMotionJ2000, 50

- init_attrjeod__PolarMotionJ2000, 51
- initialize, 50
- InputProcessor, 51
- last_table_index, 51
- operator=, 50
- override_table, 51
- polar_mjd, 51
- PolarMotionJ2000, 50
- update_rotation, 50
- warn_table, 51
- xp, 51
- xp_tbl, 51
- yp, 52
- yp_tbl, 52
- jeod::PolarMotionJ2000Init, 52
 - ~PolarMotionJ2000Init, 53
 - init_attrjeod__PolarMotionJ2000Init, 54
 - InputProcessor, 54
 - last_table_index, 54
 - operator=, 54
 - override_table, 54
 - polar_mjd, 54
 - PolarMotionJ2000Init, 53
 - xp, 54
 - xp_tbl, 54
 - yp, 55
 - yp_tbl, 55
- jeod::PolarMotionJ2000Init_xpyp_daily_default_data, 55
 - initialize, 55
- jeod::PolarMotionJ2000Init_xpyp_monthly_default_data, 56
 - initialize, 56
- jeod::PrecessionJ2000, 56
 - ~PrecessionJ2000, 57
 - init_attrjeod__PrecessionJ2000, 58
 - InputProcessor, 58
 - operator=, 57
 - PrecessionJ2000, 57
 - update_rotation, 57
- jeod::PrecessionMars, 58
 - ~PrecessionMars, 59
 - compute_fixed_matrices, 59
 - init_attrjeod__PrecessionMars, 60
 - InputProcessor, 60
 - J, 60
 - N, 60
 - NJ_matrix, 60
 - nutaton, 60
 - operator=, 59
 - PrecessionMars, 59
 - psi_at_j2000, 60
 - psi_dot, 61
 - psi_precess, 61
 - update_rotation, 60
- jeod::RNPNJ2000, 61
 - ~RNPNJ2000, 63
 - ephem_update, 63
 - get_dyn_time_ptr, 63
 - get_name, 63
 - gmst_ptr, 65
 - init_attrjeod__RNPNJ2000, 65
 - initialize, 64
 - InputProcessor, 65
 - internal_name, 65
 - last_updated_time_full, 65
 - last_updated_time_rotational, 65
 - NJ2000, 66
 - never_updated_full, 66
 - never_updated_rotational, 66
 - operator=, 64
 - PJ2000, 66
 - PMJ2000, 66
 - RJ2000, 66
 - RNPNJ2000, 63
 - time_dyn_ptr, 67
 - timestamp, 64
 - update_axial_rotation, 64
 - update_rnp, 64
- jeod::RNPNJ2000_rnp_j2000_default_data, 67
 - initialize, 67
- jeod::RNPMars, 67
 - ~RNPMars, 69
 - ephem_update, 69
 - get_dyn_time_ptr, 70
 - get_name, 70
 - init_attrjeod__RNPMars, 71
 - initialize, 70
 - InputProcessor, 71
 - internal_name, 71
 - last_updated_time_full, 71
 - last_updated_time_rotational, 71
 - NMars, 72
 - never_updated_full, 72
 - never_updated_rotational, 72
 - operator=, 70
 - PMars, 72
 - RMars, 72
 - RNPMars, 69
 - time_dyn_ptr, 72
 - timestamp, 70
 - tt_ptr, 73
 - update_axial_rotation, 70
 - update_rnp, 71
- jeod::RNPMars_rnp_mars_default_data, 73
 - initialize, 73
- jeod::RNPMessages, 73
 - fidelity_error, 74
 - init_attrjeod__RNPMessages, 74
 - initialization_error, 75
 - InputProcessor, 74
 - operator=, 74
 - polar_motion_table_warning, 75
 - RNPMessages, 74
 - setup_error, 75
- jeod::RotationJ2000, 76
 - ~RotationJ2000, 77

- GMST, [77](#)
- init_attrjeod__RotationJ2000, [77](#)
- InputProcessor, [77](#)
- nutaton, [77](#)
- operator=, [77](#)
- planet_rotational_velocity, [78](#)
- RotationJ2000, [77](#)
- theta_gast, [78](#)
- update_rotation, [77](#)
- use_full_rnp, [78](#)
- jeod::RotationMars, [78](#)
- ~RotationMars, [79](#)
- init_attrjeod__RotationMars, [80](#)
- InputProcessor, [80](#)
- nutaton, [80](#)
- operator=, [80](#)
- phi_at_j2000, [80](#)
- phi_spin, [80](#)
- planet_rotational_velocity, [80](#)
- RotationMars, [79](#)
- update_rotation, [80](#)
- use_full_rnp, [81](#)
- L
 - jeod::NutationJ2000, [23](#)
- L_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [27](#)
- last_table_index
 - jeod::PolarMotionJ2000, [51](#)
 - jeod::PolarMotionJ2000Init, [54](#)
- last_updated_time_full
 - jeod::RNPJ2000, [65](#)
 - jeod::RNPMars, [71](#)
- last_updated_time_rotational
 - jeod::RNPJ2000, [65](#)
 - jeod::RNPMars, [71](#)
- long_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [27](#)
- long_t_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [27](#)
- M
 - jeod::NutationJ2000, [23](#)
- M_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [27](#)
- mean_anomaly_j2000
 - jeod::NutationMars, [31](#)
- mean_motion
 - jeod::NutationMars, [32](#)
- Models, [11](#)
- N
 - jeod::PrecessionMars, [60](#)
- NJ2000
 - jeod::RNPJ2000, [66](#)
- NJ_matrix
 - jeod::PrecessionMars, [60](#)
- NMars
 - jeod::RNPMars, [72](#)
- NP_matrix
 - jeod::PlanetRNP, [41](#)
- name
 - jeod::PlanetOrientation, [37](#)
- never_updated_full
 - jeod::RNPJ2000, [66](#)
 - jeod::RNPMars, [72](#)
- never_updated_rotational
 - jeod::RNPJ2000, [66](#)
 - jeod::RNPMars, [72](#)
- num_coeffs
 - jeod::NutationJ2000, [23](#)
 - jeod::NutationJ2000Init, [27](#)
- nutaton
 - jeod::PlanetRNP, [41](#)
 - jeod::PrecessionMars, [60](#)
 - jeod::RotationJ2000, [77](#)
 - jeod::RotationMars, [80](#)
- nutaton_in_longitude
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationMars, [32](#)
- nutaton_in_obliquity
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationMars, [32](#)
- nutaton_j2000.cc, [85](#)
- nutaton_j2000.hh, [86](#)
- nutaton_j2000_init.cc, [86](#)
- nutaton_j2000_init.hh, [87](#)
- nutaton_mars.cc, [87](#)
- nutaton_mars.hh, [88](#)
- NutationJ2000
 - jeod::NutationJ2000, [21](#)
- NutationJ2000Init
 - jeod::NutationJ2000Init, [26](#)
- NutationMars
 - jeod::NutationMars, [30](#)
- obliq_coeffs
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationJ2000Init, [28](#)
- obliq_t_coeffs
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationJ2000Init, [28](#)
- obliquity_angle
 - jeod::NutationMars, [32](#)
- omega
 - jeod::NutationJ2000, [24](#)
- omega_coeffs
 - jeod::NutationJ2000, [24](#)
 - jeod::NutationJ2000Init, [28](#)
- operator=
 - jeod::NutationJ2000, [21](#)
 - jeod::NutationJ2000Init, [26](#)
 - jeod::NutationMars, [30](#)
 - jeod::PlanetOrientation, [36](#)

- jeod::PlanetRNP, 40
- jeod::PlanetRotation, 45
- jeod::PlanetRotationInit, 48
- jeod::PolarMotionJ2000, 50
- jeod::PolarMotionJ2000Init, 54
- jeod::PrecessionJ2000, 57
- jeod::PrecessionMars, 59
- jeod::RNPJ2000, 64
- jeod::RNPMars, 70
- jeod::RNPMessages, 74
- jeod::RotationJ2000, 77
- jeod::RotationMars, 80
- orient_interface
 - jeod::PlanetOrientation, 37
- override_table
 - jeod::PolarMotionJ2000, 51
 - jeod::PolarMotionJ2000Init, 54
- PATH
 - GenericRNP, 14
- PJ2000
 - jeod::RNPJ2000, 66
- PMJ2000
 - jeod::RNPJ2000, 66
- PMars
 - jeod::RNPMars, 72
- phi_at_j2000
 - jeod::RotationMars, 80
- phi_spin
 - jeod::RotationMars, 80
- planet
 - jeod::PlanetOrientation, 37
- planet_omega
 - jeod::PlanetOrientation, 37
- planet_orientation.cc, 88
- planet_orientation.hh, 88
- planet_rnp.cc, 89
- planet_rnp.hh, 90
- planet_rot_state
 - jeod::PlanetOrientation, 37
- planet_rotation.cc, 90
- planet_rotation.hh, 90
- planet_rotation_init.cc, 91
- planet_rotation_init.hh, 91
- planet_rotational_velocity
 - jeod::RotationJ2000, 78
 - jeod::RotationMars, 80
- PlanetOrientation
 - jeod::PlanetOrientation, 34
- PlanetRNP
 - jeod::PlanetRNP, 39, 40
- PlanetRotation
 - jeod::PlanetRotation, 44
- PlanetRotationInit
 - jeod::PlanetRotationInit, 48
- polar_mjd
 - jeod::PolarMotionJ2000, 51
 - jeod::PolarMotionJ2000Init, 54
- polar_motion
 - jeod::PlanetRNP, 41
 - polar_motion_j2000.cc, 92
 - polar_motion_j2000.hh, 92
 - polar_motion_j2000_init.cc, 93
 - polar_motion_j2000_init.hh, 93
 - polar_motion_table_warning
 - jeod::RNPMessages, 75
 - PolarMotionJ2000
 - jeod::PolarMotionJ2000, 50
 - PolarMotionJ2000Init
 - jeod::PolarMotionJ2000Init, 53
 - precession
 - jeod::PlanetRNP, 41
 - precession_j2000.cc, 93
 - precession_j2000.hh, 94
 - precession_mars.cc, 94
 - precession_mars.hh, 95
 - PrecessionJ2000
 - jeod::PrecessionJ2000, 57
 - PrecessionMars
 - jeod::PrecessionMars, 59
 - propagate_rnp
 - jeod::PlanetRNP, 40
 - psi_at_j2000
 - jeod::PrecessionMars, 60
 - psi_dot
 - jeod::PrecessionMars, 61
 - psi_m_orig
 - jeod::NutationMars, 32
 - psi_precess
 - jeod::PrecessionMars, 61
 - q_angle_j2000
 - jeod::NutationMars, 32
- RADTODEG
 - jeod::PlanetRotation, 46
- RJ2000
 - jeod::RNPJ2000, 66
- RMars
 - jeod::RNPMars, 72
- RNP, 13
- RNP_messages.cc, 98
- RNP_messages.hh, 98
- RNPFidelity
 - jeod::PlanetRNP, 39
- RNPJ2000, 15
 - jeod::RNPJ2000, 63
- RNPMars, 16
 - jeod::RNPMars, 69
- RNPMessages
 - jeod::RNPMessages, 74
- rnp_j2000.cc, 95
- rnp_j2000.hh, 96
- rnp_mars.cc, 96
- rnp_mars.hh, 97
- rnp_type
 - jeod::PlanetRNP, 42
- rotation

- jeod::PlanetRNP, [42](#)
 - jeod::PlanetRotation, [46](#)
- RotationOnly
 - jeod::PlanetRNP, [39](#)
- rotation_j2000.cc, [99](#)
- rotation_j2000.hh, [99](#)
- rotation_mars.cc, [100](#)
- rotation_mars.hh, [100](#)
- RotationJ2000
 - jeod::RotationJ2000, [77](#)
- RotationMars
 - jeod::RotationMars, [79](#)
- SECTODEG
 - jeod::PlanetRotation, [47](#)
- SECTOJULIANDAY
 - jeod::PlanetRotation, [47](#)
- scratch_matrix
 - jeod::PlanetRNP, [42](#)
- set_name
 - jeod::PlanetOrientation, [36](#)
- setup_error
 - jeod::RNPMessages, [75](#)
- theta_gast
 - jeod::RotationJ2000, [78](#)
- time_dyn_ptr
 - jeod::RNPJ2000, [67](#)
 - jeod::RNPMars, [72](#)
- timestamp
 - jeod::RNPJ2000, [64](#)
 - jeod::RNPMars, [70](#)
- tt_ptr
 - jeod::RNPMars, [73](#)
- update_axial_rotation
 - jeod::PlanetRNP, [40](#)
 - jeod::RNPJ2000, [64](#)
 - jeod::RNPMars, [70](#)
- update_rnp
 - jeod::PlanetRNP, [40](#)
 - jeod::RNPJ2000, [64](#)
 - jeod::RNPMars, [71](#)
- update_rotation
 - jeod::NutationJ2000, [21](#)
 - jeod::NutationMars, [30](#)
 - jeod::PlanetRotation, [45](#)
 - jeod::PolarMotionJ2000, [50](#)
 - jeod::PrecessionJ2000, [57](#)
 - jeod::PrecessionMars, [60](#)
 - jeod::RotationJ2000, [77](#)
 - jeod::RotationMars, [80](#)
- update_time
 - jeod::PlanetRotation, [45](#)
- use_full_rnp
 - jeod::RotationJ2000, [78](#)
 - jeod::RotationMars, [81](#)
- warn_table
 - jeod::PolarMotionJ2000, [51](#)
- xp
 - jeod::PolarMotionJ2000, [51](#)
 - jeod::PolarMotionJ2000Init, [54](#)
- xp_tbl
 - jeod::PolarMotionJ2000, [51](#)
 - jeod::PolarMotionJ2000Init, [54](#)
- xpyp_daily.cc, [100](#)
 - JEOD_FRIEND_CLASS, [101](#)
- xpyp_daily.hh, [101](#)
- xpyp_monthly.cc, [101](#)
 - JEOD_FRIEND_CLASS, [102](#)
- xpyp_monthly.hh, [102](#)
- yp
 - jeod::PolarMotionJ2000, [52](#)
 - jeod::PolarMotionJ2000Init, [55](#)
- yp_tbl
 - jeod::PolarMotionJ2000, [52](#)
 - jeod::PolarMotionJ2000Init, [55](#)