

GEOS-Chem Reference

2. Core Modules

GEOS-CHEM SUPPORT TEAM

10 Jul 2018

Contents

1	GEOS-Chem header modules	18
1.1	Fortran: Module Interface charpak_mod.F90	18
1.1.1	CntMat	19
1.1.2	CopyTxt	20
1.1.3	Cstrip	20
1.1.4	IsDigit	21
1.1.5	StrRepl	21
1.1.6	StrSplit	22
1.1.7	StrSqueeze	22
1.1.8	TranLc	22
1.1.9	TranUc	23
1.1.10	TxtExt	23
1.1.11	Str2Hash14	24
1.1.12	Str2Hash	25
1.1.13	To_Uppercase	26
1.1.14	ReadOneLine	26
1.1.15	CleanText	27
1.1.16	WordWrapPrint	27
1.2	Fortran: Module Interface state_chm_mod.F90	28
1.2.1	Init_State_Chm	32
1.2.2	Cleanup_State_Chm	33
1.2.3	Get_Metadata_State_Chm	34
1.2.4	Register_ChmField_R4_3D	34
1.2.5	Register_ChmField_Rfp_3D	35
1.2.6	Register_ChmField_Rfp_4D	36
1.2.7	Ind_	36
1.2.8	GetNumProdLossSpecies	37
1.2.9	MapProdLossSpecies	38
1.3	Fortran: Module Interface CMN_FJX_mod.F	38
1.3.1	Cleanup_Cmn_Fjx	45
1.4	Fortran: Module Interface registry_params_mod.F90	46
1.5	Fortran: Module Interface CMN_SIZE_mod.F	47
1.5.1	Init_Cmn_Size	57

1.5.2	Cleanup_Cmn_Size	58
1.6	Fortran: Module Interface state_met_mod.F90	59
1.6.1	Init_State_Met	65
1.6.2	Cleanup_State_Met	66
1.6.3	Get_Metadata_State_Met	67
1.6.4	Register_MetField_Rfp_2D	68
1.6.5	Register_MetField_Rfp_3D	68
1.6.6	Register_MetField_Int_2D	69
1.6.7	Register_MetField_Int_3D	69
1.7	Fortran: Module Interface registry_mod.F90	70
1.7.1	Registry_AddField	72
1.7.2	Registry_Lookup	74
1.7.3	Registry_Print	76
1.7.4	Registry_Destroy	77
1.7.5	MetaRegItem_AddNew	77
1.7.6	MetaRegItem_Create	78
1.7.7	MetaRegItem_Insert	79
1.7.8	MetaRegItem_Destroy	79
1.8	Fortran: Module Interface state_diag_mod.F90	80
1.8.1	Cleanup_State_Diag	86
1.8.2	Get_Metadata_State_Diag	86
1.8.3	Get_TagInfo	87
1.8.4	Get_NameInfo	88
1.8.5	Register_DiagField_R4_2D	88
1.8.6	Register_DiagField_R4_3D	89
1.8.7	Register_DiagField_R4_4D	90
1.8.8	Register_DiagField_Rfp_3D	90
1.8.9	Register_DiagField_R8_4D	91
1.9	Fortran: Module Interface species_mod.F90	92
1.9.1	SpcData_Init	96
1.9.2	Spc_GetIndx	97
1.9.3	Spc_Create	99
1.9.4	Spc_Print	101
1.9.5	Spc_GetNumSpecies	102
1.10	Fortran: Module Interface CMN_O3_mod.F	103
1.10.1	Init_Cmn_O3	104
1.10.2	Cleanup_Cmn_O3	104
1.11	Fortran: Module Interface inquireMod	105
1.11.1	findFreeLUN	106
1.11.2	I_Am_UnOPENed	106
1.12	Fortran: Module Interface CMN_DIAG_mod.F	107
1.12.1	Init_Cmn_Diag	111
1.13	Fortran: Module Interface errcode_mod.F90	111
1.13.1	GC_Error	112
1.13.2	GC_Warning	113
1.13.3	GC_CheckVar	113
1.14	Fortran: Module Interface precision_mod.F	114
1.15	Fortran: Module Interface diaglist_mod.F90	115

1.15.1	Init_DiagList	117
1.15.2	Init_DiagItem	118
1.15.3	Init_ColItem	119
1.15.4	Set_ColItem	119
1.15.5	InsertBeginning_DiagList	120
1.15.6	InsertBeginning_ColList	120
1.15.7	Search_DiagList	121
1.15.8	Search_ColList	121
1.15.9	Check_DiagList	122
1.15.10	Print_DiagList	122
1.15.11	Print_ColList	123
1.15.12	Cleanup_DiagList	123
1.15.13	Cleanup_ColList	124
1.16	Fortran: Module Interface species_database_mod.F90	124
1.16.1	Init_Species_Database	125
1.16.2	Cleanup_Species_Database	127
1.16.3	TranUc	127
1.16.4	Unique_Species_Names	128
1.16.5	Cleanup_Work_Arrays	129
1.17	Fortran: Module Interface physconstants.F	129
1.18	Fortran: Module Interface input_opt_mod.F90	131
1.18.1	Set_Input_Opt	144
1.18.2	Set_Input_Opt_Advect	145
1.18.3	Set_Input_Opt_Passive	146
1.18.4	Cleanup_Input_Opt	147
2	GEOS-Chem driver and other operations	147
2.1	Fortran: Module Interface main.F	147
2.1.1	display_grid_and_model	153
2.1.2	ctm_flush	154
2.1.3	display_end_time	154
2.1.4	read_initial_met_fields	155
2.1.5	read_met_fields	155
2.1.6	get_overhead_o3_for_fastj	156
2.2	Fortran: Module Interface input_mod.F	157
2.2.1	read_input_file	160
2.2.2	split_one_line	161
2.2.3	read_simulation_menu	162
2.2.4	read_timestep_menu	163
2.2.5	read_advected_species_menu	164
2.2.6	read_aerosol_menu	165
2.2.7	read_co_sim_menu	175
2.2.8	read_co2_sim_menu	176
2.2.9	read_chemistry_menu	177
2.2.10	read_radiation_menu	178
2.2.11	read_transport_menu	179
2.2.12	read_convection_menu	180
2.2.13	read_deposition_menu	180

2.2.14	read_gamap_menu	182
2.2.15	read_output_menu	182
2.2.16	read_diagnostic_menu	183
2.2.17	set_tindex	185
2.2.18	read_planeflight_menu	186
2.2.19	read_nd48_menu	186
2.2.20	read_nd49_menu	187
2.2.21	read_nd50_menu	188
2.2.22	read_nd51_menu	189
2.2.23	read_nd51b_menu	189
2.2.24	read_nd63_menu	190
2.2.25	read_prod_loss_menu	191
2.2.26	read_nested_grid_menu	192
2.2.27	read_benchmark_menu	193
2.2.28	read_mercury_menu	193
2.2.29	read_ch4_menu	194
2.2.30	read_pops_menu	195
2.2.31	read_passive_species_menu	196
2.2.32	validate_directories	197
2.2.33	check_directory	198
2.2.34	check_time_steps	198
2.2.35	is_last_day_good	199
2.2.36	do_error_checks	200
2.3	Fortran: Module Interface gc_environment_mod.F90	201
2.3.1	gc_allocate_all	202
2.3.2	gc_init_stateobj	203
2.3.3	gc_init_grid	204
2.3.4	gc_init_extra	206
2.3.5	gc_init_regridding	208
2.3.6	init_tomas_microphysics	209
2.4	Fortran: Module Interface restart_mod.F	211
2.4.1	init_gc_restart	212
2.4.2	read_gc_restart	213
2.4.3	write_gc_restart	215
2.4.4	check_rst_dimensions	216
2.4.5	compare_trc_spc	217
2.5	Fortran: Module Interface cleanup.F	217
3	Transport modules	222
3.1	Fortran: Module Interface transport_mod.F	223
3.1.1	do_transport	224
3.1.2	do_global_adv	225
3.1.3	do_window_transport	227
3.1.4	init_transport	229
3.1.5	init_window_transport	230
3.1.6	cleanup_transport	231
3.2	Fortran: Module Interface pjc_pfix_mod.F	231
3.2.1	Do_Pjc_Pfix	232

3.2.2	Calc_Pressure	234
3.2.3	Calc_Advection_Factors	234
3.2.4	Adjust_Press	235
3.2.5	Init_Press_Fix	237
3.2.6	Do_Press_Fix_Llnl	239
3.2.7	Average_Press_Poles	240
3.2.8	Convert_Winds	241
3.2.9	Calc_Horiz_Mass_Flux	242
3.2.10	Calc_Divergence	243
3.2.11	Set_Press_Terms	244
3.2.12	Do_Divergence_Pole_Sum	244
3.2.13	Xpavg	245
3.2.14	Init_Pjc_Pfix	246
3.2.15	Cleanup_Pjc_Pfix	246
3.3	Fortran: Module Interface tpcore_bc_mod.F	247
3.3.1	set_clean_bc	250
3.3.2	open_bc_file	250
3.3.3	save_global_tpcore_bc	251
3.3.4	do_window_tpcore_bc	252
3.3.5	clean_window_tpcore_bc	253
3.3.6	read_window_tpcore_bc	254
3.3.7	get_4x5_bc_init	255
3.3.8	get_4x5_bc	255
3.3.9	get_2x25_bc_init	256
3.3.10	get_2x25_bc	257
3.3.11	its_time_for_bc	258
3.3.12	init_tpcore_bc	258
3.3.13	cleanup_tpcore_bc	259
3.4	Fortran: Module Interface tpcore_fvdas_mod.F90	260
3.4.1	Init_Tpcore	263
3.4.2	Exit_Tpcore	264
3.4.3	Tpcore_FvDas	265
3.4.4	Average_Const_Poles	268
3.4.5	Set_Cross_Terms	269
3.4.6	Calc_Vert_Mass_Flux	270
3.4.7	Set_Jn_Js	271
3.4.8	Calc_Advec_Cross_Terms	273
3.4.9	Qckxyz	274
3.4.10	Set_Lmts	275
3.4.11	Set_Press_Terms	276
3.4.12	Calc_Courant	278
3.4.13	Calc_Divergence	279
3.4.14	Do_Divergence_Pole_Sum	280
3.4.15	Do_Cross_Terms_Pole_I2d2	282
3.4.16	Xadv_Dao2	283
3.4.17	Yadv_Dao2	284
3.4.18	Do_Yadv_Pole_I2d2	286
3.4.19	Do_Yadv_Pole_Sum	287

3.4.20	Xtp	288
3.4.21	Xmist	289
3.4.22	Fxppm	291
3.4.23	Lmtpm	292
3.4.24	Ytp	293
3.4.25	Ymist	295
3.4.26	Do_Ymist_Pole1_I2d2	296
3.4.27	Do_Ymist_Pole2_I2d2	297
3.4.28	Fyppm	298
3.4.29	Do_Fyppm_Pole_I2d2	299
3.4.30	Do_Ytp_Pole_Sum	300
3.4.31	Fzppm	302
3.4.32	Average_Press_Poles	303
4	Convection and scavenging modules	304
4.1	Fortran: Module Interface convection_mod.F90	304
4.1.1	do_convection	306
4.1.2	do_cloud_convection	308
4.2	Fortran: Module Interface wetscav_mod.F	312
4.2.1	do_wetdep	315
4.2.2	make_qq	316
4.2.3	e_ice	319
4.2.4	compute_l2g	319
4.2.5	compute_f	320
4.2.6	compute_ki	323
4.2.7	f_aerosol	323
4.2.8	rainout	324
4.2.9	apply_rainout_eff	327
4.2.10	get_rainfrac	327
4.2.11	washout	328
4.2.12	washfrac_fine_aerosol	331
4.2.13	washfrac_coarse_aerosol	332
4.2.14	washfrac_size_aerosol	332
4.2.15	washfrac_hno3	333
4.2.16	washfrac_liq_gas	334
4.2.17	wetdep	335
4.2.18	ls_k_rain	339
4.2.19	ls_f_prime	340
4.2.20	conv_f_prime	340
4.2.21	do_rainout_only	341
4.2.22	do_washout_only	342
4.2.23	do_complete_reevap	346
4.2.24	do_washout_at_sfc	347
4.2.25	safety	349
4.2.26	get_vud	350
4.2.27	get_f	351
4.2.28	init_wetscav	351
4.2.29	setup_wetscav	352

4.2.30	cleanup_wetscav	354
5	Boundary layer mixing modules	354
5.1	Fortran: Module Interface mixing_mod.F90	354
5.1.1	init_mixing	355
5.1.2	do_mixing	356
5.1.3	do_tend	357
5.2	Fortran: Module Interface pbl_mix_mod.F	359
5.2.1	do_pbl_mix	360
5.2.2	compute_pbl_height	361
5.2.3	turbday	362
5.2.4	get_frac_of_pbl	364
5.2.5	get_frac_under_pbltop	364
5.2.6	get_pbl_max_l	365
5.2.7	get_pbl_top_hpa	365
5.2.8	get_pbl_top_l	366
5.2.9	get_pbl_top_m	366
5.2.10		367
5.2.11	init_pbl_mix	367
5.2.12	cleanup_pbl_mix	368
5.3	Fortran: Module Interface vdiff_mod.F90	368
5.3.1	pbinti	371
5.3.2	vdiff	371
5.3.3	pbldif	373
5.3.4	qvdiff	375
5.3.5	vdiffar	376
5.3.6	pbldifar	376
5.3.7	vdinti	377
5.3.8	vdifldr	378
5.3.9	do_pbl_mix_2	380
5.4	Fortran: Module Interface vdiff_pre_mod.F90	382
5.4.1	init_vdiff_pre	383
5.4.2	cleanup_vdiff_pre	383
5.4.3	set_vdiff_values	384
6	Dry deposition modules	384
6.1	Fortran: Module Interface drydep_mod.F	385
6.1.1	do_drydep	389
6.1.2	metero	391
6.1.3	depvel	393
6.1.4	diffg	398
6.1.5	read_drydep_inputs	398
6.1.6	aero_sfcrsii	400
6.1.7	init_weightss	401
6.1.8	dust_sfcrsi	402
6.1.9	adust_sfcrsii	402
6.1.10	dust_sfcrsii	403
6.1.11	init_drydep	404

6.1.12	cleanup_drydep	406
6.2	Fortran: Module Interface modis_lai_mod.F90	406
6.2.1	compute_modis_lai	408
6.2.2	compute_xlai_gchp	409
6.2.3	compute_modis	410
6.2.4	read_modis_lai	411
6.2.5	read_modis	411
6.2.6	find_lai_month	412
6.2.7	RoundOff	413
6.2.8	init_modis	414
6.2.9	cleanup_modis_lai	414
6.3	Fortran: Module Interface olson_landmap_mod.F90	415
6.3.1	compute_olson_landmap_gchp	419
6.3.2	init_olson_landmap	419
6.3.3	compute_olson_landmap	420
6.3.4	cleanup_olson_landmap	421
6.4	Fortran: Module Interface mapping_mod.F90	422
6.4.1	init_mapping	423
6.4.2	get_map_wt	424
6.4.3	cleanup_mapping	425
6.5	Fortran: Module Interface get_ndep_mod.F	425
6.5.1	soil_drydep	426
6.5.2	soil_wetdep	427
6.5.3	reset_dep_N	427
6.5.4	Init_Get_Ndep	427
6.5.5	cleanup_commsoil	428
7	Emissions and chemistry modules	428
7.1	Fortran: Module Interface emissions_mod.F90	429
7.1.1	emissions_init	429
7.1.2	emissions_run	430
7.1.3	emissions_final	431
7.2	Fortran: Module Interface chemistry_mod.F90	432
7.2.1	do_chemistry	433
7.2.2	recompute_od	436
7.2.3	chem_passive_species	437
7.2.4	init_chemistry	438
7.3	Fortran: Module Interface fast_jx_mod.F	439
7.3.1	fast_jx	440
7.3.2	blkslv	443
7.3.3	gen_id	444
7.3.4	jratet	444
7.3.5	x_interp	445
7.3.6	sphere2	445
7.3.7	extral	446
7.3.8	exitc	447
7.3.9	int_fjx	448
7.3.10	rd_xxx	448

7.3.11	rd_mie	449
7.3.12	rd_aod	450
7.3.13	calc_aod	451
7.3.14	rd_js_jx	453
7.3.15	solar_jx	454
7.3.16	photo_jx	454
7.3.17	opmie	455
7.3.18	miesct	456
7.3.19	legend0	457
7.3.20	set_prof	457
7.3.21	set_aer	458
7.3.22	rd_prof_nc	459
7.3.23	phorate_adj	460
7.4	Fortran: Module Interface uvalbedo_mod.F90	461
7.4.1	get_uvalbedo	462
7.5	Fortran: Module Interface set_global_ch4_mod.F90	462
7.5.1	set_ch4	463
7.5.2	cleanup_set_global_ch4	464
7.5.3	set_prof_o3	464
7.6	Fortran: Module Interface toms_mod.F	465
7.6.1	read_toms	467
7.6.2	compute_overhead_o3	469
7.6.3	get_overhead_o3	471
7.6.4	init_toms	472
7.6.5	cleanup_toms	472
7.7	Fortran: Module Interface flexchem_mod.F90	473
7.7.1	do_flexchem	474
7.7.2	Diag_OH_HO2_O1D_O3P	476
7.7.3	init_flexchem	477
7.7.4	cleanup_flexchem	478
7.8	Fortran: Module Interface gckpp_HetRates	479
7.8.1	Set_Het	483
7.8.2	kiir1ltd	484
7.8.3	kiir1r2l	485
7.8.4	HetIXCycleSSA	486
7.8.5	HetIUptake	486
7.8.6	HetNO3	487
7.8.7	HetNO2	487
7.8.8	HetHO2	488
7.8.9	HetHBr	488
7.8.10	HetN2O5	489
7.8.11	HetGLYX	489
7.8.12	HetMGLY	490
7.8.13	HetIEPOX	490
7.8.14	HetIMAE	491
7.8.15	HetLVOC	491
7.8.16	HetISN1OG	492
7.8.17	HetISOPND	492

7.8.18	HetISOPNB	493
7.8.19	HetMACRN	493
7.8.20	HetMVKN	494
7.8.21	HetR4N2	494
7.8.22	HetDHDN	495
7.8.23	HetMONITS	495
7.8.24	HetMONITU	496
7.8.25	HetHONIT	496
7.8.26	HetIONITA	497
7.8.27	HetMONITA	497
7.8.28	HetN2O5_SS	498
7.8.29	HetN2O5_HCl	498
7.8.30	HetHXUptake_js	499
7.8.31	HetO3_SS_JS	499
7.8.32	HetClNO3_SS_JS	500
7.8.33	HetHOBr_SS_JS	501
7.8.34	HetO3_HBr_JS	501
7.8.35	Gamma_O3_Br	502
7.8.36	HetClNO3_HBr_JS	503
7.8.37	HetClNO3_JS	503
7.8.38	HetHOBr_HCl_JS	504
7.8.39	HetHOBr_HBr_JS	505
7.8.40	HetHOBr_HSO3	506
7.8.41	HetHOBr_SO3	506
7.8.42	Gamma_HX_Uptake	507
7.8.43	Gamma_HOBr_X	508
7.8.44	gamma_HOBr_cld	508
7.8.45	gamma_HOBr_aer	509
7.8.46	Gamma_ClNO3_Br	510
7.8.47	COTH	510
7.8.48	REACTODIFF_CORR	510
7.8.49	HetBrNO3_JS	511
7.8.50	HetClNO3_HCl	511
7.8.51	HetBrNO3_HCl	512
7.8.52	HetHOCl_HCl	513
7.8.53	HetHOCl_HBr	513
7.8.54	N2O5	514
7.8.55	HO2	514
7.8.56	EPOXUPTK	516
7.8.57	Cld_Params	517
7.8.58	Get_Halide_CldConc	518
7.8.59	Get_Halide_SSAConc	518
7.8.60	Compute_L2G_Local	519
7.8.61	Cld1k_XNO3	520
7.8.62	FcrO2HO2	521
7.8.63	FyHORO	521
7.8.64	FyrNO3	522
7.8.65	Arsl1k	522

7.8.66	Check_Nat	523
7.9	Fortran: Module Interface ucx_mod.F	524
7.9.1	montreal_scalefac	526
7.9.2	ucx_nox	527
7.9.3	get_noxcoeff	528
7.9.4	emiss_basic	529
7.9.5	read_sfc	530
7.9.6	set_montreal_ncdf	530
7.9.7	get_montreal_ncdf	531
7.9.8	set_montreal	531
7.9.9	settle_strat_aer	532
7.9.10	calc_h2so4_gas	533
7.9.11	so4_photfrac	533
7.9.12	calc_strat_aer	534
7.9.13	kg_strat_aer	535
7.9.14	rho_strat_aer	535
7.9.15	get_strat_opt	536
7.9.16	ternary	536
7.9.17	carslaw_density	537
7.9.18	calc_fallvel	538
7.9.19	cacl_sla_gamma	538
7.9.20	molec_speed	539
7.9.21	read_psc_file	540
7.9.22	set_clock_trac	540
7.9.23	set_h2o_trac	541
7.9.24	ucx_h2so4phot	542
7.9.25	sfcmr_init	543
7.9.26	sfcmr_read	544
7.9.27	sfcmr_get	544
7.9.28	noxcoeff_init	545
7.9.29	get_jjnox	545
7.9.30	init_ucx	546
7.9.31	diaginit_ucx	546
7.9.32	write_state_psc	547
7.9.33	cleanup_ucx	548
7.10	Fortran: Module Interface bromocarb_mod.F	548
7.10.1	set_ch3br	549
7.10.2	set_bro	550
7.11	Fortran: Module Interface cldice_hbrhobr_rxn.F	551
7.12	Fortran: Module Interface strat_chem_mod.F90	553
7.12.1	Do_Strat_Chem	554
7.12.2	Set_BryPointers	556
7.12.3	Set_Plvec	557
7.12.4	Calc_Ste	558
7.12.5	Init_Strat_Chem	559
7.12.6	Set_Minit	560
7.12.7	Cleanup_Strat_Chem	561
7.12.8	Do_Synoz	561

7.13	Fortran: Module Interface linoz_mod.F	564
7.13.1	do_linoz	566
7.13.2	linoz_chem3	567
7.13.3	linoz_stratl	568
7.13.4	linoz_strt2m	569
7.13.5	linoz_somlfq	571
7.13.6	linoz_read	572
7.13.7	linoz_intpl	573
7.13.8	init_linoz	573
7.13.9	cleanup_linoz	574
8	Aerosol modules	574
8.1	Fortran: Module Interface aerosol_mod.F	574
8.1.1	aerosol_conc	577
8.1.2	rdaer	579
8.1.3	init_aerosol	581
8.1.4	cleanup_aerosol	582
8.2	Fortran: Module Interface carbon_mod.F	582
8.2.1	chemcarbon	587
8.2.2	chem_bcpo	589
8.2.3	chem_bcp_i	590
8.2.4	chem_ocpo	591
8.2.5	chem_ocp_i	592
8.2.6	aging_carb	593
8.2.7	soa_chemistry	593
8.2.8	soa_equil	597
8.2.9	zeroin	598
8.2.10	rtbis	599
8.2.11	soa_para	600
8.2.12	soa_para_init	602
8.2.13	chem_nvoc	602
8.2.14	soa_partition	604
8.2.15	soa_lump	605
8.2.16	emitsgc	606
8.2.17	scalecarb	607
8.2.18	emisscarbon	608
8.2.19	emisscarbontomas	609
8.2.20	emithigh2	610
8.2.21	ohno3time	611
8.2.22	get_oh	612
8.2.23	get_no3	613
8.2.24	get_o3	614
8.2.25	get_daro2	615
8.2.26	get_doh	616
8.2.27	check_eqlb	617
8.2.28	save_oaginit	618
8.2.29	check_mb	618
8.2.30	get_no	619

8.2.31	get_ho2	620
8.2.32	get_isopno3	621
8.2.33	init_carbon	621
8.2.34	cleanup_carbon	623
8.3	Fortran: Module Interface dust_mod.F	623
8.3.1	chemdust	624
8.3.2	settledust	626
8.3.3	emissdust	627
8.3.4	dry_settling	628
8.3.5	rdust_online	629
8.3.6	rdust_offline	631
8.3.7	get_dust_alk	634
8.3.8	init_dust	635
8.3.9	cleanup_dust	636
8.4	Fortran: Module Interface seasalt_mod.F	636
8.4.1	chemseasalt	638
8.4.2	wet_settling	639
8.4.3	chem_mopo	641
8.4.4	chem_mopi	642
8.4.5	init_seasalt	643
8.4.6	cleanup_seasalt	644
8.4.7	srcsalt30	644
8.5	Fortran: Module Interface sulfate_mod.F	645
8.5.1	chemsulfate	648
8.5.2	emissulfatetomas	650
8.5.3	chem_dms	666
8.5.4	chem_h2o2	668
8.5.5	seasalt_chem	675
8.5.6	dust_chem	677
8.5.7	get_hplus	679
8.5.8	kCO21	680
8.5.9	kCO22	680
8.5.10	kSO21	681
8.5.11	kSO22	682
8.5.12	kHNO3	682
8.5.13	kHCl	683
8.5.14	kNH3	683
8.5.15	cubic	684
8.5.16	aqchem_so2	684
8.5.17	het_drop_chem	686
8.5.18	chem_so4	687
8.5.19	chem_so4_aq	689
8.5.20	chem_msa	690
8.5.21	chem_nit	691
8.5.22	sulfate_pbl_mix	693
8.5.23	get_oh	694
8.5.24	get_no3	695
8.5.25	get_o3	696

8.5.26	ohno3time	696
8.5.27	get_alk	697
8.5.28	init_sulfate	698
8.5.29	cleanup_sulfate	700
8.6	Fortran: Module Interface isoropiaii_mod.F	700
8.6.1	do_isoropiaii	702
8.6.2	safelog10	703
8.6.3	get_hno3	704
8.6.4	set_hno3	704
8.6.5	get_gno3	705
8.6.6	init_isoropiaII	705
8.6.7	cleanup_isoropiaII	706
9	Meteorological field modules	706
9.1	Fortran: Module Interface dao_mod.F	706
9.1.1	avgpole	709
9.1.2	airqnt	710
9.1.3	interp	712
9.1.4	gigc_cap_tropopause_prs	714
9.1.5	set_dry_surface_pressure	715
9.1.6	is_land	716
9.1.7	is_water	717
9.1.8	is_ice	718
9.1.9	is_near	719
9.1.10	get_obk	720
9.1.11	get_cosine_sza	721
9.1.12	cossza	721
9.1.13	copy_i3_i6_fields	723
9.2	Fortran: Module Interface merra2_read_mod.F90	724
9.2.1	Get_Resolution_String	725
9.2.2	Check_Dimensions	726
9.2.3	Merra2_Read_cn	726
9.2.4	Merra2_Read_a1	727
9.2.5	Merra2_Read_a3	728
9.2.6	Merra2_Read_a3cld	729
9.2.7	Merra2_Read_a3dyn	729
9.2.8	Merra2_Read_a3mstc	730
9.2.9	Merra2_Read_a3mste	731
9.2.10	Merra2_Read_I3_1	732
9.2.11	Merra2_Read_I3_2	732
9.2.12	Cleanup_Merra2_Read	733
9.3	Fortran: Module Interface geosfp_read_mod.F90	733
9.3.1	Get_Resolution_String	735
9.3.2	Check_Dimensions	735
9.3.3	GeosFp_Read_cn	736
9.3.4	GeosFp_Read_a1	737
9.3.5	GeosFp_Read_a3	738
9.3.6	GeosFp_Read_a3cld	739

9.3.7	GeosFp_Read_a3dyn	740
9.3.8	GeosFp_Read_a3mstc	741
9.3.9	GeosFp_Read_a3mste	742
9.3.10	GeosFp_Read_I3_1	743
9.3.11	GeosFp_Read_I3_2	744
9.3.12	Cleanup_GeosFp_Read	745
10	Specialty simulation modules	745
10.1	Fortran: Module Interface c2h6_mod.F	745
10.1.1	chemc2h6	746
10.1.2	get_c2h6_anthro	748
10.1.3	init_c2h6	748
10.1.4	cleanup_c2h6	749
10.2	Fortran: Module Interface co2_mod.F	749
10.2.1	emissco2	751
10.2.2	def_biosph_co2_regions.f	753
10.2.3	def_ocean_co2_regions.f	753
10.2.4	def_fossil_co2_regions.f	754
10.2.5	init_co2	755
10.2.6	cleanup_co2	755
10.3	Fortran: Module Interface exchange_mod.F90	756
10.3.1	exchange_global_init	757
10.3.2	exchange_nested_init	757
10.3.3	exchange_global_post	758
10.3.4	exchange_nested_pre	758
10.3.5	exchange_nested_post	759
10.3.6	its_time_for_exchange	759
10.3.7	init_exchange	760
10.3.8	cleanup_exchange	760
10.3.9	ex_dump_for_global	761
10.3.10	ex_read_and_apply_feedback	761
10.3.11	ex_regrid_up	762
10.3.12	ex_convert_units_cspec	762
10.3.13	wait_for_unlock	763
10.3.14	done_lock_me_up	763
10.3.15	no_more_lock	764
10.3.16	alert_for_kill_command	764
10.3.17	touch_file	764
10.3.18	open_bin_file_for_read	765
10.3.19	open_bin_file_for_write	765
10.3.20	open_bin_file_for_append	765
10.3.21	close_bin_file	766
10.3.22	write_3d_real8_array	766
10.3.23	read_3d_real8_array	766
10.3.24	gen_regrid_mat	767
10.4	Fortran: Module Interface global_ch4_mod.F	767
10.4.1	chemch4	772
10.4.2	ch4_decay	774

10.4.3	ch4_ohsave	775
10.4.4	ch4_strat	776
10.4.5	ch4_distrib	777
10.4.6	init_global_ch4	778
10.4.7	cleanup_global_ch4	779
10.5	Fortran: Module Interface mercury_mod.F	779
10.5.1	chemmercury	783
10.5.2	get_hgaq_rate	787
10.5.3	get_hgbr_rate	788
10.5.4	get_hgcl_rate	789
10.5.5	get_hgbry_rate	791
10.5.6	emissmercury	792
10.5.7	emithg	794
10.5.8	srcHg0	794
10.5.9	srcHg2	795
10.5.10	srcHgp	796
10.5.11	mercury_readyr	797
10.5.12	get_o3	798
10.5.13	get_oh	799
10.5.14	get_ho2	800
10.5.15	get_clo	801
10.5.16	get_cl	801
10.5.17	get_HOCl	802
10.5.18	get_NO2	803
10.5.19	get_Br	803
10.5.20	ohno3time	804
10.5.21	calc_hg2_seasalt_lossrate	805
10.5.22	get_jno2	806
10.5.23	partitionhg2	807
10.5.24	red_inplume_grid	807
10.5.25	do_red_inplume	808
10.5.26	offlineocean_readmo	809
10.5.27	set_options_from_hemco	816
10.5.28	cleanup_mercury	817
10.6	Fortran: Module Interface depo_mercury_mod.F	817
10.6.1	add_Hg2_dd	819
10.6.2	add_Hg2_wd	819
10.6.3	add_HgP_dd	820
10.6.4	add_HgP_wd	820
10.6.5	add_hg2_snowpack	821
10.6.6	reset_hg_dep_arrays	822
10.6.7	make_gtmm_restart	822
10.6.8	read_gtmm_restart	823
10.6.9	update_dep	824
10.6.10	check_dimensions	825
10.6.11	init_depo_mercury	825
10.6.12	cleanup_depo_mercury	826
10.7	Fortran: Module Interface land_mercury_mod.F	826

10.7.1	land_mercury_flux	827
10.7.2	biomasshg	828
10.7.3	vegemis	829
10.7.4	soilemis	830
10.7.5	snowpack_mercury_flux	832
10.7.6	gtmm_dr	833
10.7.7	init_land_mercury	834
10.7.8	cleanup_land_mercury	834
10.8	Fortran: Module Interface pops_mod.F	835
10.8.1	chempops	835
10.8.2	chem_popgp	836
10.8.3	rxn_ox_nodep	838
10.8.4	rxn_ox_withdep	838
10.8.5	no_rxn_withdep	839
10.8.6	get_oh	839
10.8.7	ohno3time	840
10.8.8	init_pops	841
10.8.9	cleanup_pops	841
10.9	Fortran: Module Interface RnPbBe_mod.F	842
10.9.1	chemRnPbBe	843
10.10	Fortran: Module Interface rrtmg_rad_transfer_mod.F	845
10.10.1	do_rrtmg_rad_transfer	846
10.10.2	set_specmask	848
10.10.3	read_surface_rad	849
10.10.4	read_strat_clim	850
10.10.5	init_surface_rad	851
10.10.6	init_strat_clim	851
10.10.7	init_mcica_clouds	851
10.10.8	cleanup_surface_rad	852
10.10.9	cleanup_strat_clim	852
10.10.10	cleanup_mcica_clouds	852
10.11	Fortran: Module Interface tagged_co_mod.F	853
10.11.1	chem_tagged_co	855
10.11.2	calc_diurnal	858
10.11.3	get_alpha_isop	858
10.11.4	init_tagged_co	859
10.11.5	cleanup_tagged_co	860
10.12	Fortran: Module Interface tagged_o3_mod.F	860
10.12.1	get_regional_po3	863
10.12.2	chem_tagged_o3	864
10.12.3	init_tagged_o3	866
10.12.4	cleanup_tagged_o3	867

1 GEOS-Chem header modules

Here follows a list modules that are used to define various global parameters and variables used by GEOS-Chem. These were formerly COMMON-block include files (*.h), but were converted to Fortran-90 modules as part of the GEOS-Chem HP (high-performance) project.

1.1 Fortran: Module Interface charpak_mod.F90

Module CHARPAK_MOD contains routines from the CHARPAK string and character manipulation package used by GEOS-Chem.

INTERFACE:

```
MODULE Charpak_Mod
```

USES:

```
IMPLICIT NONE  
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: CleanText  
PUBLIC  :: CntMat  
PUBLIC  :: CopyTxt  
PUBLIC  :: CStrip  
PUBLIC  :: IsDigit  
PUBLIC  :: ReadOneLine  
PUBLIC  :: Str2Hash14  
PUBLIC  :: Str2Hash31  
PUBLIC  :: StrRepl  
PUBLIC  :: StrSplit  
PUBLIC  :: StrSqueeze  
PUBLIC  :: To_UpperCase  
PUBLIC  :: TranLc  
PUBLIC  :: TranUc  
PUBLIC  :: Txttext  
PUBLIC  :: WordWrapPrint  
!PRIVATE MEMBER FUNCTIONS
```

REMARKS:

CHARPAK routines by Robert D. Stewart, 1992. Subsequent modifications made for GEOS-CHEM by Bob Yantosca (1998, 2002, 2004).

REVISION HISTORY:

- (1) Moved "cntmat.f", "copytxt.f", "cstrip.f", "fillstr.f", "txt2inum.f", "txttext.f", into this F90 module for easier bookkeeping
(bmy, 10/15/01)
- (2) Moved "tranuc.f" into this F90 module (bmy, 11/15/01)

- (3) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
- (4) Wrote a new file "strrepl.f", which replaces a character pattern within a string with replacement text. Moved "tranlc.f" into this module. Replaced calls to function LENTRIM with F90 intrinsic function LEN_TRIM. Removed function FILLSTR and replaced it w/ F90 intrinsic REPEAT. (bmy, 6/25/02)
- (5) Added routine STRSPLIT as a wrapper for TXTEXT. Also added routines STRREPL and STRSQUEEZE. (bmy, 7/30/02)
- (6) Added function ISDIGIT. Also replace LEN_TRIM with LEN in routine STRREPL, to allow us to replace tabs w/ spaces. (bmy, 7/20/04)
- 20 Nov 2009 - R. Yantosca - Added ProTeX header
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 07 Aug 2017 - R. Yantosca - Add ProTeX headers for subroutines
- 31 Oct 2017 - R. Yantosca - Converted module to F90 free-format
- 01 Nov 2017 - R. Yantosca - Added ReadOneLine and CleanText
- 01 Nov 2017 - R. Yantosca - Added MaxStrLen parameter for ReadOneLine

1.1.1 CntMat

Counts the number of characters in str1 that match a character in str2.

INTERFACE:

```
SUBROUTINE CntMat( Str1, Str2, Imat, Locations )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN) :: Str1           ! Text to scan
CHARACTER(LEN=*), INTENT(IN) :: Str2           ! Character to match
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: imat           ! Number of matches
INTEGER,          OPTIONAL    :: Locations(255) ! Positions of matches
```

REVISION HISTORY:

- DATE: JAN. 6, 1995
- AUTHOR: R.D. STEWART
- COMMENTS: Revised slightly (2-5-1996) so that trailing blanks in str1 are ignored. Revised again on 3-6-1996.
- 7 Aug 2017 - R. Yantosca - Added ProTeX header
- 20 Dec 2017 - R. Yantosca - Now returns the positions where matches occur

1.1.2 CopyTxt

Write all of the characters in str1 into variable str2 beginning at column, col. If the length of str1 + col is longer than the number of characters str2 can store, some characters will not be transferred to str2. Any characters already existing in str2 will be overwritten.

INTERFACE:

```
SUBROUTINE CopyTxt( col, str1, str2 )
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)      :: col
CHARACTER(LEN=*), INTENT(IN)      :: str1
```

OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(INOUT)  :: str2
```

REVISION HISTORY:

```
DATE:   DEC. 24, 1993
AUTHOR: R.D. STEWART
7 Aug 2017 - R. Yantosca - Added ProTeX header
```

1.1.3 Cstrip

Strip blanks and null characters for the variable TEXT.

INTERFACE:

```
SUBROUTINE CStrip( text, KeepSpaces )
```

INPUT PARAMETERS:

```
LOGICAL,          OPTIONAL        :: KeepSpaces ! If =T, then keep spaces
                                                    ! but skip all other
                                                    ! non-printing chars
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(INOUT)  :: TEXT      ! Text to be modified
```

REMARKS:

The original "text" is destroyed upon exit.

REVISION HISTORY:

```
AUTHOR: Robert D. Stewart
DATE:   May 19, 1992
07 Aug 2017 - R. Yantosca - Added ProTeX header
30 Jan 2018 - R. Yantosca - Added KEEPSPACES optional argument
```

1.1.4 IsDigit

Returned as true if ch is a numeric character (i.e., one of the numbers from 0 to 9).

INTERFACE:

```
FUNCTION IsDigit( ch ) RESULT( lnum )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=1), INTENT(IN) :: ch
```

RETURN VALUE:

```
LOGICAL :: lnum
```

REMARKS:

NOTE: Changed name from ISNUM to ISDIGIT (bmy, 7/15/04)

REVISION HISTORY:

DATE: NOV. 11, 1993

AUTHOR: R.D. STEWART

7 Aug 2017 - R. Yantosca - Added ProTeX header

1.1.5 StrRepl

Subroutine StrRepl replaces all instances of PATTERN within a string STR with replacement text REPLTXT.

INTERFACE:

```
SUBROUTINE StrRepl( Str, Pattern, ReplTxt )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN) :: Pattern ! Pattern to search for
```

```
CHARACTER(LEN=*), INTENT(IN) :: ReplTxt ! Text to replace
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(INOUT) :: Str ! String to be manipulated
```

REMARKS:

PATTERN and REPLTXT can now have a different number of characters.

REVISION HISTORY:

25 Jun 2002 - R. Yantosca - Initial version

07 Aug 2017 - R. Yantosca - Updated algorithm to allow PATTERN and

REPLTXT to have different numbers of characters

7 Aug 2017 - R. Yantosca - Added ProTeX header

1.1.6 StrSplit

Subroutine STRSPLIT returns substrings in a string, separated by a separator character (similar to IDL's StrSplit function). This is mainly a convenience wrapper for CHARPAK routine TxtExt.

INTERFACE:

```
SUBROUTINE StrSplit( Str, Sep, Result, N_SubStrs )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)  :: STR      ! String to be searched
CHARACTER(LEN=1), INTENT(IN)  :: SEP      ! Separator character
```

OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(OUT) :: Result(255) ! Returned substrings
INTEGER,          OPTIONAL    :: N_SubStrs   ! # of substrings
```

REVISION HISTORY:

```
11 Jul 2002 - R. Yantosca - Initial version
30 Sep 2014 - R. Yantosca - WORD now has 2047 chars for extra-long strings
```

1.1.7 StrSqueeze

Subroutine STRSQUEEZE strips white space from both ends of a string. White space in the middle of the string (i.e. between characters) will be preserved as-is. Somewhat similar (though not exactly) to IDL's STRCOMPRESS function.

INTERFACE:

```
SUBROUTINE StrSqueeze( Str )
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(INOUT) :: Str ! String to be squeezed
```

REVISION HISTORY:

```
11 Jul 2002 - R. Yantosca - Initial version
07 Aug 2017 - R. Yantosca - Added ProTeX header
```

1.1.8 TranLc

Tranlate a character variable to all lowercase letters. Non-alphabetic characters are not affected.

INTERFACE:

```
SUBROUTINE TranLc( text )
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*) :: text
```

REMARKS:

The original "text" is destroyed.

REVISION HISTORY:

```
AUTHOR: Robert D. Stewart
DATE: May 19, 1992
7 Aug 2017 - R. Yantosca - Added ProTeX header
```

1.1.9 TranUc

Tranlate a character variable to all upper case letters. Non-alphabetic characters are not affected.

INTERFACE:

```
SUBROUTINE TranUc( text )
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*) :: text
```

REMARKS:

The original "text" is destroyed.

REVISION HISTORY:

```
AUTHOR: Robert D. Stewart
DATE: May 19, 1992
7 Aug 2017 - R. Yantosca - Added ProTeX header
```

1.1.10 TxtExt

TxtExt extracts a sequence of characters from text and transfers them to word. The extraction procedure uses a set of character "delimiters" to denote the desired sequence of characters. For example if ch=' ', the first character sequence bracketed by blank spaces will be returned in word. The extraction procedure begins in column, col, of TEXT. If text(col:col) = ch (any character in the character string), the text is returned beginning with col+1 in text (i.e., the first match with ch is ignored).

After completing the extraction, col is incremented to the location of the first character following the end of the extracted text.

A status flag is also returned with the following meaning(s)

IF iflg = -1, found a text block, but no more characters are available in TEXT iflg = 0, task completed successfully (normal term) iflg = 1, ran out of text before finding a block of text.

INTERFACE:

```
SUBROUTINE TxtExt(ch,text,col,word,iflg)
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)      :: ch,text
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT)  :: col
```

OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(OUT)    :: word
INTEGER                    :: iflg
```

REMARKS:

TxtExt is short for Text Extraction. This routine provides a set of powerful line-by-line text search and extraction capabilities in standard FORTRAN.

REVISION HISTORY:

```
AUTHOR: Robert D. Stewart
DATE: Jan. 1st, 1995
REVISIONS: FEB 22, 1996. Slight bug fix (introduced by a
(recent = FLIB 1.04) change in the CntMat routine)
so that TxtExt correctly handles groups of characters
delimited by blanks).
MODIFICATIONS by Bob Yantosca (6/25/02)
(1) Replace call to FILLSTR with F90 intrinsic REPEAT
7 Aug 2017 - R. Yantosca - Added ProTeX header
```

1.1.11 Str2Hash14

Returns a unique integer hash for a given character string. This allows us to implement a fast name lookup algorithm.

INTERFACE:

```
FUNCTION Str2Hash14( Str ) RESULT( Hash )
```

INPUT PARAMETERS:


```
CHARACTER(LEN=14), INTENT(IN) :: Str    ! String (14 chars long)
```

RETURN VALUE:

```
INTEGER                                :: Hash    ! Hash value from string
```

REMARKS:

- (1) Algorithm taken from this web page:
<https://fortrandev.wordpress.com/2013/07/06/fortran-hashing-algorithm/>
- (2) For now, we only use the first 14 characters of the character string to compute the hash value. Most GEOS-Chem species names only use at most 14 unique characters. We can change this later if need be.

REVISION HISTORY:

- 04 May 2016 - R. Yantosca - Initial version
05 May 2016 - R. Yantosca - Now make the input string 14 chars long
-

1.1.12 Str2Hash

Returns a unique integer hash for a given character string. This allows us to implement a fast name lookup algorithm.

INTERFACE:

```
FUNCTION Str2Hash31( Str ) RESULT( Hash )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=31), INTENT(IN) :: Str    ! String (31 chars long)
```

RETURN VALUE:

```
INTEGER                                :: Hash    ! Hash value from string
```

REMARKS:

- (1) Algorithm taken from this web page:
<https://fortrandev.wordpress.com/2013/07/06/fortran-hashing-algorithm/>
- (2) For now, we only use the first 31 characters of the character string to compute the hash value. Most GEOS-Chem variable names only use up to 31 unique characters. We can change this later if need be.

REVISION HISTORY:

- 26 Jun 2017 - R. Yantosca - Initial version
-

1.1.13 To_Uppercase

Converts a string to uppercase, so that we can reliably do string matching.

INTERFACE:

```
FUNCTION To_UpperCase( Text ) RESULT( UpCaseText )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN) :: Text          ! Input test
```

RETURN VALUE:

```
CHARACTER(LEN=255)           :: UpCaseText    ! Output text, uppercase
```

REMARKS:

Code originally from routine TRANUC (Author: R. D. Stewart, 19 May 1992)

REVISION HISTORY:

26 Jun 2017 - R. Yantosca - Initial version

1.1.14 ReadOneLine

Subroutine READ_ONE_LINE reads a line from the input file. If the global variable VERBOSE is set, the line will be printed to stdout. READ_ONE_LINE can trap an unexpected EOF if LOCATION is passed. Otherwise, it will pass a logical flag back to the calling routine, where the error trapping will be done.

INTERFACE:

```
FUNCTION ReadOneLine( fId, EndOfFile, IoStatus, Squeeze ) RESULT( Line )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN)          :: fId          ! File unit number
LOGICAL, OPTIONAL           :: Squeeze      ! Call Strsqueeze?
```

OUTPUT PARAMETERS:

```
LOGICAL, INTENT(OUT)        :: EndOfFile    ! Denotes EOF condition
INTEGER, INTENT(OUT)        :: IoStatus     ! I/O status code
```

RETURN VALUE:

```
CHARACTER(LEN=MAXSTRLEN) :: Line          ! Single line from the input file
```

REMARKS:

Mostly used by routines in the History/ folder.

REVISION HISTORY:

16 Jun 2017 - R. Yantosca - Initial version, based on GEOS-Chem
 01 Nov 2017 - R. Yantosca - Moved to charpak_mod.F90

1.1.15 CleanText

Strips commas, apostrophes, spaces, and tabs from a string.

INTERFACE:

```
FUNCTION CleanText( Str ) RESULT( CleanStr )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN) :: Str          ! Original string
!RETURN VALUE
CHARACTER(LEN=255)           :: CleanStr     ! Cleaned-up string
```

REMARKS:

Mostly used by routines in the History/ folder.

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Initial version
21 Jun 2017 - R. Yantosca - Now call CSTRIP to remove tabs etc.
01 Nov 2017 - R. Yantosca - Moved to charpak_mod.F90
```

1.1.16 WordWrapPrint

Prints a text string wrapped to a specified line width. Useful for displaying error and warning messages.

INTERFACE:

```
SUBROUTINE WordWrapPrint( Text, LineWidth, Delimiter )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN) :: Text          ! Text to print
INTEGER,           INTENT(IN) :: LineWidth    ! Width (characters) of lines
CHARACTER(LEN=1), OPTIONAL    :: Delimiter    ! Delimiter between words
```

REMARKS:

The default DELIMITER is the space (" ") character.

REVISION HISTORY:

```
20 Dec 2015 - R. Yantosca - Initial version
```

1.2 Fortran: Module Interface state_chm_mod.F90

Module STATE_CHM_MOD contains the derived type used to define the Chemistry State object for GEOS-Chem.

This module also contains the routines that allocate and deallocate memory to the Chemistry State object. The chemistry state object is not defined in this module. It must be declared as variable in the top-level driver routine, and then passed to lower-level routines as an argument.

INTERFACE:

```

MODULE State_Chm_Mod
  USES:
  USE ErrorCode_Mod           ! Error handling
  USE PhysConstants          ! Physical constants
  USE Precision_Mod          ! GEOS-Chem precision types
  USE Registry_Mod           ! Registry module
  USE Species_Mod            ! For species database object

  IMPLICIT NONE
  PRIVATE

```

PUBLIC MEMBER FUNCTIONS:

```

PUBLIC :: Init_State_Chm
PUBLIC :: Cleanup_State_Chm
PUBLIC :: Get_Metadata_State_Chm
PUBLIC :: Ind_
!PRIVATE MEMBER FUNCTIONS
PRIVATE :: Register_ChmField
!PRIVATE DATA MEMBERS:
TYPE(SpcPtr), PRIVATE, POINTER :: SpcDataLocal(:) ! Local pointer to
                                                    ! StateChm%SpcData for
                                                    ! availability to IND_

```

PUBLIC DATA MEMBERS:

```

!=====
! Derived type for Chemistry State
!=====
TYPE, PUBLIC :: ChmState

!-----
! Count of each type of species
!-----
INTEGER          :: nSpecies          ! # species (all)
INTEGER          :: nAdvect            ! # advected species
INTEGER          :: nAero              ! # of Aerosol Types
INTEGER          :: nDryDep            ! # drydep species

```

```

INTEGER          :: nGasSpc           ! # gas phase species
INTEGER          :: nHygGrth         ! # hygroscopic growth
INTEGER          :: nKppVar          ! # KPP variable species
INTEGER          :: nKppFix          ! # KPP fixed species
INTEGER          :: nKppSpc          ! # KPP chem species
INTEGER          :: nLoss             ! # of loss species
INTEGER          :: nPhotol          ! # photolysis species
INTEGER          :: nProd             ! # of prod species
INTEGER          :: nWetDep          ! # wetdep species

```

```
!-----
```

```
! Mapping vectors to subset types of species
```

```
!-----
```

```

INTEGER,          POINTER :: Map_Advect (:      ) ! Advected species IDs
INTEGER,          POINTER :: Map_Aero   (:      ) ! Aerosol species IDs
INTEGER,          POINTER :: Map_DryDep (:      ) ! Drydep species IDs
INTEGER,          POINTER :: Map_GasSpc (:      ) ! Gas species IDs
INTEGER,          POINTER :: Map_HygGrth(:      ) ! HygGrth species IDs
INTEGER,          POINTER :: Map_KppVar (:      ) ! Kpp variable spc IDs
INTEGER,          POINTER :: Map_KppFix (:      ) ! KPP fixed species IDs
INTEGER,          POINTER :: Map_KppSpc (:      ) ! KPP chem species IDs
INTEGER,          POINTER :: Map_Loss   (:      ) ! Loss diag species
CHARACTER(LEN=36), POINTER :: Name_Loss (:      ) ! ID's and names
INTEGER,          POINTER :: Map_Photol (:      ) ! Photolysis species IDs
INTEGER,          POINTER :: Map_Prod   (:      ) ! Prod diag species
CHARACTER(LEN=36), POINTER :: Name_Prod (:      ) ! ID and names
INTEGER,          POINTER :: Map_WetDep (:      ) ! Wetdep species IDs

```

```
!-----
```

```
! Physical properties & indices for each species
```

```
!-----
```

```
TYPE(SpcPtr),    POINTER :: SpcData   (:      ) ! GC Species database
```

```
!-----
```

```
! Chemical species
```

```
!-----
```

```

REAL(fp),        POINTER :: Species   (:,:,,:) ! Species [molec/cm3]
CHARACTER(LEN=20) :: Spc_Units        ! Species units

```

```
!-----
```

```
! Aerosol quantities
```

```
!-----
```

```

REAL(fp),        POINTER :: AeroArea  (:,:,,:) ! Aerosol Area [cm2/cm3]
REAL(fp),        POINTER :: AeroRadi  (:,:,,:) ! Aerosol Radius [cm]
REAL(fp),        POINTER :: WetAeroArea(:,:,,:) ! Aerosol Area [cm2/cm3]
REAL(fp),        POINTER :: WetAeroRadi(:,:,,:) ! Aerosol Radius [cm]
REAL(fp),        POINTER :: SSAlk     (:,:,,:) ! Sea-salt alkalinity[-]

```

```

!-----
! Cloud quantities
!-----
REAL(fp),          POINTER :: pHCloud    (:,:, ) ! Cloud pH [-]

!-----
! Fields for UCX mechanism
!-----
REAL(f4),          POINTER :: STATE_PSC  (:,:, ) ! PSC type (see Kirner
! et al. 2011, GMD)
REAL(fp),          POINTER :: KHETI_SLA  (:,:,,:) ! Strat. liquid aerosol
! reaction cofactors

!-----
! For isoprene SOA via ISORROPIA
!-----
REAL(fp),          POINTER :: pHsav      (:,:, ) ! ISORROPIA aerosol pH
REAL(fp),          POINTER :: HplusSav   (:,:, ) ! H+ concentration [M]
REAL(fp),          POINTER :: WaterSav   (:,:, ) ! ISORROPIA aerosol H2O
REAL(fp),          POINTER :: SulRatSav  (:,:, ) ! Sulfate conc [M]
REAL(fp),          POINTER :: NaRatSav   (:,:, ) ! Nitrate conc [M]
REAL(fp),          POINTER :: AcidPurSav (:,:, ) !
REAL(fp),          POINTER :: BiSulSav   (:,:, ) ! Bisulfate conc [M]

!-----
! For the tagged Hg simulation
!-----
INTEGER            :: N_HG_CATS          ! # of Hg categories
INTEGER,           POINTER :: Hg0_Id_List(: ) ! Hg0 cat <-> tracer #
INTEGER,           POINTER :: Hg2_Id_List(: ) ! Hg2 cat <-> tracer #
INTEGER,           POINTER :: HgP_Id_List(: ) ! HgP cat <-> tracer #
CHARACTER(LEN=4),  POINTER :: Hg_Cat_Name(: ) ! Category names

!-----
! For HOBr + S(IV) heterogeneous chemistry
!-----
REAL(fp),          POINTER :: HSO3_AQ    (:,:, ) ! Cloud bisulfite[mol/l]
REAL(fp),          POINTER :: SO3_AQ     (:,:, ) ! Cloud sulfite [mol/l]
REAL(fp),          POINTER :: fupdateHOBr(:,:, ) ! Correction factor for
! HOBr removal by SO2
! [unitless]

!-----
! Registry of variables contained within State_Chm
!-----
CHARACTER(LEN=4)   :: State              = 'CHEM'  ! Name of this state
TYPE(MetaRegItem), POINTER :: Registry  => NULL() ! Registry object

```

END TYPE ChmState

REMARKS:

REVISION HISTORY:

- 19 Oct 2012 - R. Yantosca - Initial version, based on "gc_type2_mod.F90"
 - 26 Oct 2012 - R. Yantosca - Add fields for stratospheric chemistry
 - 26 Feb 2013 - M. Long - Add DEPSAV to derived type ChmState
 - 07 Mar 2013 - R. Yantosca - Add Register_Tracer subroutine
 - 07 Mar 2013 - R. Yantosca - Now make POSITION a locally SAVED variable
 - 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 - 19 May 2014 - C. Keller - Removed Trac_Btend. DepSav array covers now all species.
 - 03 Dec 2014 - M. Yannetti - Added PRECISION_MOD
 - 11 Dec 2014 - R. Yantosca - Keep JLOP and JLOP_PREV for ESMF runs only
 - 17 Feb 2015 - E. Lundgren - New tracer units kg/kg dry air (previously kg)
 - 13 Aug 2015 - E. Lundgren - Add tracer units string to ChmState derived type
 - 28 Aug 2015 - R. Yantosca - Remove strat chemistry fields, these are now handled by the HEMCO component
 - 05 Jan 2016 - E. Lundgren - Use global physical constants
 - 28 Jan 2016 - M. Sulprizio- Add STATE_PSC, KHETI_SLA. These were previously local arrays in ucx_mod.F, but now need to be accessed in gckpp_HetRates.F90.
 - 12 May 2016 - M. Sulprizio- Add WetAeroArea, WetAeroRadi to replace 1D arrays WTARE, WERADIUS previously in comode_mod.F
 - 18 May 2016 - R. Yantosca - Add mapping vectors for subsetting species
 - 07 Jun 2016 - M. Sulprizio- Remove routines Get_Indx, Register_Species, and Register_Tracer made obsolete by the species database.
 - 22 Jun 2016 - R. Yantosca - Rename Id_Hg0 to Hg0_Id_List, Id_Hg2 to Hg2_Id_List, and Id_HgP to HgP_Id_List
 - 16 Aug 2016 - M. Sulprizio- Rename from gigc_state_chm_mod.F90 to state_chm_mod.F90. The "gigc" nomenclature is no longer used.
 - 23 Aug 2016 - M. Sulprizio- Remove tracer fields from State_Chm. These are now entirely replaced with the species fields.
 - 08 Jun 2017 - M. Sulprizio- Add fields for isoprene SOA updates from E.Marais
 - 29 Jun 2017 - R. Yantosca - Add fields of State_Chm to the registry
 - 29 Jun 2017 - R. Yantosca - Remove Spec_Id, it's no longer used
 - 30 Jun 2017 - R. Yantosca - Now register variables of State_Chm
 - 31 Jul 2017 - R. Yantosca - Add fixes in registering ISORROPIA *_SAV fields
 - 26 Sep 2017 - E. Lundgren - Remove Lookup_State_Chm and Print_State_Chm
 - 02 Oct 2017 - E. Lundgren - Abstract metadata and routine to add to Registry
 - 27 Nov 2017 - E. Lundgren - Add # and ID mapping for more species categories
 - 31 Jan 2018 - E. Lundgren - Remove underscores from diagnostic names
-

1.2.1 Init_State_Chm

Routine INIT_STATE_CHM allocates and initializes the pointer fields of the chemistry state object.

INTERFACE:

```
SUBROUTINE Init_State_Chm( am_I_Root, Input_Opt, State_Chm, RC )
```

USES:

```
USE CMN_Size_Mod,          ONLY : IIPAR, JJPAR, LLPAR, NDUST, NAER
USE GCKPP_Parameters,     ONLY : NSPEC
USE Input_Opt_Mod,        ONLY : OptInput
USE Species_Database_Mod, ONLY : Init_Species_Database
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input Options object
TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Return code
```

REMARKS:

In the near future we will put some error trapping on the allocations so that we can stop the simulation if the allocations cannot be made.

REVISION HISTORY:

```
19 Oct 2012 - R. Yantosca - Renamed from gc_type2_mod.F90
19 Oct 2012 - R. Yantosca - Now pass all dimensions as arguments
26 Oct 2012 - R. Yantosca - Now allocate Strat_P, Strat_k fields
26 Oct 2012 - R. Yantosca - Add nSchem, nSchemBry as arguments
01 Nov 2012 - R. Yantosca - Don't allocate strat chem fields if nSchm=0
                        and nSchmBry=0 (i.e. strat chem is turned off)
26 Feb 2013 - M. Long     - Now pass Input_Opt via the argument list
26 Feb 2013 - M. Long     - Now allocate the State_Chm%DEPSAV field
11 Dec 2014 - R. Yantosca - Remove TRAC_TEND and DEPSAV fields
13 Aug 2015 - E. Lundgren - Initialize trac_units to ''
28 Aug 2015 - R. Yantosca - Remove stratospheric chemistry fields;
                        these are all now read in via HEMCO
28 Aug 2015 - R. Yantosca - Also initialize the species database object
09 Oct 2015 - R. Yantosca - Bug fix: set State_Chm%SpcData to NULL
16 Dec 2015 - R. Yantosca - Now overwrite the Input_Opt%TRACER_MW_G and
                        related fields w/ info from species database
```


29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 02 May 2016 - R. Yantosca - Nullify Hg index fields for safety's sake
 18 May 2016 - R. Yantosca - Now determine the # of each species first,
 then allocate fields of State_Chm
 18 May 2016 - R. Yantosca - Now populate the species mapping vectors
 30 Jun 2016 - M. Sulprizio- Remove nSpecies as an input argument. This is now
 initialized as the size of SpcData.
 22 Jul 2016 - E. Lundgren - Initialize spc_units to ''
 28 Nov 2016 - R. Yantosca - Only allocate STATE_PSC and KHETI_SLA for UCX
 simulations; set to NULL otherwise
 28 Nov 2016 - R. Yantosca - Only allocate State_Chm%*Aero* fields for
 fullchem and/or aerosol-only simulations
 16 Nov 2017 - E. Lundgren - Get grid params and # aerosls from CMN_Size_Mod
 rather than arguments list

1.2.2 Cleanup_State_Chm

Routine CLEANUP_STATE_CHM deallocates all fields of the chemistry state object.

INTERFACE:

```
SUBROUTINE Cleanup_State_Chm( am_I_Root, State_Chm, RC )
```

USES:

```
USE Species_Database_Mod, ONLY : Cleanup_Species_Database
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Return code
```

REMARKS:

For now the am_I_Root and RC arguments are not used. We include these for consistency and also to facilitate future expansion. (bmy, 10/16/12)

REVISION HISTORY:

15 Oct 2012 - R. Yantosca - Initial version
 26 Oct 2012 - R. Yantosca - Now deallocate Strat_P, Strat_k fields
 26 Feb 2013 - M. Long - Now deallocate State_Chm%DEPSAV
 11 Dec 2014 - R. Yantosca - Remove TRAC_TEND and DEPSAV fields
 28 Aug 2015 - R. Yantosca - Remove stratospheric chemistry fields;
 these are all now read in via HEMCO
 28 Aug 2015 - R. Yantosca - Also initialize the species database object
 29 Jun 2017 - R. Yantosca - Add error checks for deallocations. Also
 destroy the registry of State_Chm fields.

1.2.3 Get_Metadata_State_Chm

Subroutine GET_METADATA_STATE_CHM retrieves basic information about each State_Chm field.

INTERFACE:

```

SUBROUTINE Get_Metadata_State_Chm( am_I_Root,  metadataID, Found,      &
                                   RC,          Desc,        Units,      &
                                   PerSpecies, Rank,        Type,       &
                                   VLoc )

```

USES:

```

USE Charpak_Mod,          ONLY : To_UpperCase
USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,                  INTENT(IN)  :: am_I_Root   ! Is this the root CPU?
CHARACTER(LEN=*),        INTENT(IN)  :: metadataID  ! State_Chm field name

```

OUTPUT PARAMETERS:

```

LOGICAL,                  INTENT(OUT)      :: Found      ! Item found?
INTEGER,                  INTENT(OUT)      :: RC          ! Return code
CHARACTER(LEN=255),       INTENT(OUT), OPTIONAL :: Desc      ! Long name string
CHARACTER(LEN=255),       INTENT(OUT), OPTIONAL :: Units     ! Units string
CHARACTER(LEN=255),       INTENT(OUT), OPTIONAL :: PerSpecies ! Max spc wildcard
INTEGER,                  INTENT(OUT), OPTIONAL :: Rank      ! # of dimensions
INTEGER,                  INTENT(OUT), OPTIONAL :: Type      ! Desc of data type
INTEGER,                  INTENT(OUT), OPTIONAL :: VLoc      ! Vert placement

```

REMARKS:

REVISION HISTORY:

```

02 Oct 2017 - E. Lundgren - Initial version
20 Oct 2017 - R. Yantosca - Update metadata to better match array names,
                           and to remove special characters like "+"

```

1.2.4 Register_ChmField_R4_3D

Registers a 3-dimensional, 4-byte floating point array field of the State_Chm object. This allows the diagnostic modules get a pointer to the field by searching on the field name.

INTERFACE:

```

SUBROUTINE Register_ChmField_R4_3D( am_I_Root,  metadataID, Ptr2Data, &
                                     State_Chm, RC )

```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,           INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*),  INTENT(IN)      :: metadataID     ! Name
REAL(f4),          POINTER         :: Ptr2Data(:, :, :) ! pointer to data
TYPE(ChmState),    INTENT(IN)      :: State_Chm      ! Obj for chem state
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,           INTENT(OUT)     :: RC              ! Success/failure
```

REMARKS:**REVISION HISTORY:**

```
20 Sep 2017 - E. Lundgren - Initial version
```

1.2.5 Register_ChmField_Rfp_3D

Registers a 3-dimensional, flexible-precision array field of the State_Chm object. This allows the diagnostic modules get a pointer to the field by searching on the field name.

INTERFACE:

```
SUBROUTINE Register_ChmField_Rfp_3D( am_I_Root, metadataID, Ptr2Data, &
                                       State_Chm, RC )
```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,           INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*),  INTENT(IN)      :: metadataID     ! State_Chm field ID
REAL(fp),          POINTER         :: Ptr2Data(:, :, :) ! pointer to data
TYPE(ChmState),    INTENT(IN)      :: State_Chm      ! Obj for chem state
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,           INTENT(OUT)     :: RC              ! Success/failure
```

REMARKS:**REVISION HISTORY:**

```
20 Sep 2017 - E. Lundgren - Initial version
```

1.2.6 Register_ChmField_Rfp_4D

Registers a 4-dimensional, flexible-precision array field of the State_Chm object. This allows the diagnostic modules get a pointer to the field by searching on the field name.

INTERFACE:

```

SUBROUTINE Register_ChmField_Rfp_4D( am_I_Root,  metadataID, Ptr2Data,      &
                                   State_Chm,  RC,          Ncat          )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root          ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID         ! State_Chm field id
REAL(fp),         POINTER         :: Ptr2Data(:, :, :, :) ! pointer to data
TYPE(ChmState),   INTENT(IN)      :: State_Chm         ! Obj for chem state
INTEGER,          OPTIONAL        :: Ncat              ! category index

```

INPUT/OUTPUT PARAMETERS:

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC                ! Success/failure

```

REMARKS:

REVISION HISTORY:

```

20 Sep 2017 - E. Lundgren - Initial version

```

1.2.7 Ind_

Function IND_ returns the index of an advected species or chemical species contained in the chemistry state object by name.

INTERFACE:

```

FUNCTION Ind_( name, flag ) RESULT( Indx )

```

USES:

```

USE CharPak_Mod, ONLY : Str2Hash14, To_UpperCase

```

INPUT PARAMETERS:

```

CHARACTER(LEN=*),          INTENT(IN) :: name ! Species name
CHARACTER(LEN=*), OPTIONAL, INTENT(IN) :: flag ! Species type

```

RETURN VALUE:

```

      INTEGER                               :: Indx  ! Index of this species
!REMARKS

```

REVISION HISTORY:

```

07 Oct 2016 - M. Long      - Initial version
15 Jun 2016 - M. Sulprizio- Make species name uppercase before computing hash
17 Aug 2016 - M. Sulprizio- Tracer flag 'T' is now advected species flag 'A'
01 Nov 2017 - R. Yantosca - Now use Str2Hash14 from charpak_mod.F90, which
                           computes a hash from an input string of 14 chars
27 Nov 2017 - E. Lundgren - Add flags for additional species categories

```

1.2.8 GetNumProdLossSpecies

Saves the number of production and loss diagnostic species in the State_Chm%nProdLoss variable. This will be used to set up the State_Chm%Map_ProdLoss species index vector.

INTERFACE:

```

SUBROUTINE GetNumProdLossSpecies( am_I_Root, Input_Opt, State_Chm, RC )

```

USES:

```

      USE GcKpp_Monitor,    ONLY : Fam_Names
      USE GcKpp_Parameters, ONLY : nFam
      USE Input_Opt_Mod,    ONLY : OptInput

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input Options object
      TYPE(ChmState),  INTENT(INOUT) :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)   :: RC          ! Return code

```

REMARKS:**REVISION HISTORY:**

```

06 Jan 2015 - R. Yantosca - Initial version

```

1.2.9 MapProdLossSpecies

Stores the ModelId (from the GEOS-Chem Species Database) of each prod/loss diagnostic species in the State_Chm%Map_ProdLoss vector.

INTERFACE:

```
SUBROUTINE MapProdLossSpecies( am_I_Root, Input_Opt, State_Chm, RC )
```

USES:

```
USE GcKpp_Monitor,    ONLY : Fam_Names
USE GcKpp_Parameters, ONLY : nFam
USE Input_Opt_Mod,    ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)  :: Input_Opt    ! Input Options object
TYPE(ChmState),  INTENT(INOUT)  :: State_Chm     ! Chemistry State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC           ! Return code
```

REMARKS:

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Initial version
```

1.3 Fortran: Module Interface CMN_FJX_mod.F

Module CMN_FJX_MOD contains parameters and global variables used to interface between Harvard chemistry and UC-Irvine photolysis programs (Fast-J/Fast-JX), along with all Fast-J(X) global variables and some physical constants for the GEOS-Chem chemistry code.

INTERFACE:

```
MODULE CMN_FJX_MOD
```

USES:

```
USE CMN_SIZE_MOD,    ONLY : IIPAR, JJPAR, LLPAR, LLCHEM
USE CMN_SIZE_MOD,    ONLY : NDUST, NAER, NRHAER, NSTRATAER, NRH
```

```
USE PRECISION_MOD    ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PUBLIC
```

DEFINED PARAMETERS:

```

! New (SDE 03/28/13)
! Index in RAA & QAA of 999 nm wavelength
INTEGER, PARAMETER :: IND999 = 5

! Required size of aerosol arrays
INTEGER          :: L_          ! Number of CTM layers

INTEGER          :: L1_         ! Number of CTM layer edges

INTEGER          :: L2_         ! Number of levels in FJX grid that
                               ! inc. both edges and mid-points

INTEGER          :: JVL_        ! Vertical levels for J-values

INTEGER, PARAMETER :: JVN_ = 130 ! Max number of J-values

INTEGER          :: AN_         ! # of separate aerosols per layer
                               ! Now set in Init_CMN_FJX below

! Variables used to interface GEOS-Chem and Fast-JX at runtime
! Branches for photolysis species
INTEGER          :: BRANCH(JVN_)

! Names of photolysis species
CHARACTER (LEN=10) :: RNames(JVN_)

! Mapping array from Harvard species names to UCI species names
INTEGER          :: RINDEX(JVN_)

! GEOS-Chem "ModelId" corresponding to each photolysis species
INTEGER          :: GC_Photo_Id(JVN_)

! Output J values
REAL(fp), ALLOCATABLE :: ZPJ(:,:,:, :)

!-----
! variables used to map fast-JX J's onto CTM J's
!-----

! Multiplication factor for fast-JX calculated J
REAL(fp)          :: JFACTA(JVN_)

! Index arrays that map Jvalue(j) onto rates
INTEGER          :: JIND(JVN_)

! Number of Photolysis reactions in CTM chemistry, derived here NRATJ

```

```

! must be .le. JVN_
INTEGER          :: NRATJ

! Label of J-value used in the main chem model
CHARACTER*50     :: JLABEL(JVN_)

! JXL_: vertical(levels) dim for J-values computed within fast-JX
INTEGER          :: JXL_
INTEGER          :: JXL1_

! JXL2_: 2*JXL_ + 2 = mx no. levels in the basic Fast-JX grid (mid-level)
INTEGER          :: JXL2_

! WX_ = dim = no. of wavelengths in input file
INTEGER, PARAMETER :: WX_ = 18

! X_ = dim = max no. of X-section data sets (input data)
! SDE 2016-11-04: Increased from 75 to 123 for new halogen
! chemistry (iodine cross-sections)
INTEGER, PARAMETER :: X_ = 123

! A_ = dim = no. of Aerosol/cloud Mie sets (input data)
INTEGER, PARAMETER :: A_ = 56

! W_ = dim = no. of Wavelength bins: =18 std, =12 trop only
! Now set below in Init_CMN_FJX (mps, 1/3/18)
INTEGER          :: W_

! N_ = no. of levels in Mie scattering arrays
!      = 2*NC+1 = 4*(L_+1) + 1 + 2*sum(JADDLV)
INTEGER, PARAMETER :: N_ = 601

! M_ = no. of Gauss points used, must = 4 in fast_JX (no option)
INTEGER, PARAMETER :: M_ = 4

! M2_ = 2*M_ = 8, replaces MFIT
INTEGER, PARAMETER :: M2_ = 2*M_

!-----
! 4 Gauss pts = 8-stream
REAL(fp), DIMENSION(M_), PARAMETER ::
&          EMU = [.06943184420297e+0_fp, .33000947820757e+0_fp,
&                .66999052179243e+0_fp, .93056815579703e+0_fp]
REAL(fp), DIMENSION(M_), PARAMETER ::
&          WT = [.17392742256873e+0_fp, .32607257743127e+0_fp,
&                .32607257743127e+0_fp, .17392742256873e+0_fp]
!-----

```



```
! ZZHT: scale height (cm)
REAL(fp), PARAMETER  :: ZZHT   = 5.e+5_fp

! RAD: Radius of Earth (cm)
REAL(fp), PARAMETER  :: RAD     = 6375.e+5_fp

! ATAU: heating rate (factor increase from one layer to the next)
REAL(fp), PARAMETER  :: ATAU    = 1.120e+0_fp

! ATAU0: minimum heating rate
REAL(fp), PARAMETER  :: ATAU0   = 0.010e+0_fp

! JTAUMX = maximum number of divisions (i.e., may not get to ATAUMN)
INTEGER              :: JTAUMX

! Physical constants
REAL(fp), PARAMETER  :: UVXPLANCK = 6.62606957e-34
REAL(fp), PARAMETER  :: UVXCONST  = 2.99792458e8

! Conversion factors from photons/cm2s to W/m2
REAL(fp), DIMENSION(WX_) :: UVXFACTOR

!-----
! Variables in file 'FJX_spec.dat' (RD_XXX)
!-----

! WBIN: Boundaries of wavelength bins
REAL(fp)              :: WBIN(WX_+1)

! WL: Centres of wavelength bins - 'effective wavelength'
REAL(fp)              :: WL(WX_)

! FL: Solar flux incident on top of atmosphere (cm-2.s-1)
REAL(fp)              :: FL(WX_)

REAL(fp)              :: Q02(WX_,3) ! Q02: O2 cross-sections
REAL(fp)              :: Q03(WX_,3) ! Q03: O3 cross-sections
REAL(fp)              :: Q1D(WX_,3) ! Q1D: O3 => O(1D) quantum yield

! QQQ: Supplied cross sections in each wavelength bin (cm2)
REAL(fp)              :: QQQ(WX_,3,X_)

! QRAYL: Rayleigh parameters (effective cross-section) (cm2)
REAL(fp)              :: QRAYL(WX_+1)

! TQQ: Temperature for supplied cross sections
REAL(fp)              :: TQQ(3,X_)
```

```
! LQQ = 1, 2, or 3 to determine interpolation with T or P
INTEGER          :: LQQ(X_)

! TITLEJX: Title for supplied cross sections, from 'FJX_spec.dat'
CHARACTER*6      :: TITLEJX(X_)

! SQQ: Flag for supplied cross sections, from 'FJX_spec.dat'
CHARACTER*1      :: SQQ(X_)

!-----
! Variables in file 'jv_spec_mie.dat' (RD_MIE)
!-----

! QAA: Aerosol scattering phase functions
REAL(fp)         :: QAA(5,A_)

! WAA: 5 Wavelengths for the supplied phase functions
REAL(fp)         :: WAA(5,A_)

! PAA: Phase function: first 8 terms of expansion
REAL(fp)         :: PAA(8,5,A_)

! RAA: Effective radius associated with aerosol type
REAL(fp)         :: RAA(5,A_)

! SAA: Single scattering albedo
REAL(fp)         :: SAA(5,A_)

! NAA: Number of categories for scattering phase functions
INTEGER          :: NAA

!-----
! Variables in file 'jv_spec_aod.dat' (RD_AOD)
!-----

! QAA_AOD: Aerosol scattering phase functions
REAL(fp)         :: QAA_AOD(A_)

! WAA: 5 Wavelengths for the supplied phase functions
REAL(fp)         :: WAA_AOD(A_)

! PAA: Phase function: first 8 terms of expansion
REAL(fp)         :: PAA_AOD(8,A_)

! RAA: Effective radius associated with aerosol type
REAL(fp)         :: RAA_AOD(A_)

! SAA: Single scattering albedo
```

```

REAL(fp)          :: SAA_AOD(A_)

!-----
! Variables in file 'atmos_std.dat' (RD_PROF)
!-----

! T and O3 reference profiles
REAL(fp), DIMENSION(51,18,12) :: TREF, OREF

! Interfacing indices for GC and FJX aerosols
INTEGER, ALLOCATABLE :: MIEDX(:)

! TITLEAA: Title for scattering data
CHARACTER*80, DIMENSION(A_) :: TITLEAA

! Dust and aerosol optical depths
REAL(fp), ALLOCATABLE :: ODMDUST(:,:,:,,:)
REAL(fp), ALLOCATABLE :: ODAER(:,:,:,,:)
REAL(fp), ALLOCATABLE :: ISOPOD(:,:,:,:) ! eam, 2014

INTEGER NJX,NW1,NW2

!-----
! Variables added for RRTMG (dar, mps, 12/5/14)
!-----

INTEGER, PARAMETER :: NWVAA = 41 !number of wavelengths in LUT
INTEGER, PARAMETER :: NWVAAO = 11 !number of non-RRTMG wavelengths
INTEGER, PARAMETER :: NWVAART = NWVAA-NWVAAO !number of RRTMG wvs
INTEGER, PARAMETER :: NRAA = 7 !number of aer sizes in LUT
INTEGER, PARAMETER :: NALBD = 2
INTEGER, PARAMETER :: NEMISS = 16
INTEGER, PARAMETER :: NSPECRADMENU = 11!number of choices for RT flux

! Now set the following in Init_CMN_FJX below (mps, 1/3/18)
INTEGER :: NSPAA !number of species in LUT
INTEGER :: NASPECRAD !aerosol species in RT
INTEGER :: NSPECRAD !aerosol+gas species in RT

! New optical arrays
REAL*8, ALLOCATABLE :: WVAA(:,:)
REAL*8, ALLOCATABLE :: RHAA(:,:)
REAL*8, ALLOCATABLE :: NRLAA(:,:,:)
REAL*8, ALLOCATABLE :: NCMAA(:,:,:)
REAL*8, ALLOCATABLE :: RDAA(:,:)
REAL*8, ALLOCATABLE :: RWAA(:,:)
REAL*8, ALLOCATABLE :: SGAA(:,:)
REAL*8, ALLOCATABLE :: QQAA(:,:,:)

```

```

REAL*8, ALLOCATABLE :: ALPHA(:,:,:)
REAL*8, ALLOCATABLE :: REAA(:,:)
REAL*8, ALLOCATABLE :: SSAA(:,:,:)
REAL*8, ALLOCATABLE :: ASYMAA(:,:,:)
REAL*8, ALLOCATABLE :: PHAA(:,:,:)
INTEGER :: IWVSELECT(2,3) !index of requested wavelengths
INTEGER :: IRTWVSELECT(2,3) !index of requested RT wavelengths

! max of 3 but need 2 per wavelength if interpolating
INTEGER :: NWVREQUIRED !number of wvs required for interpolation
INTEGER :: IWVREQUIRED(6) !index of wavelengths for interpo.
INTEGER :: NRTWVREQUIRED !number of wvs required for RT interpolation
INTEGER :: IRTWVREQUIRED(6) !index of wavelengths for RT interpo.
! list of required wavelengths, up to max of 3 x 2

INTEGER :: IWV1000 !Store the wavelength index for 1000nm for Fast-J

!coefficients for interpolation of wavelength (and for RT too)
REAL*8      :: ACOEF_WV(3),BCOEF_WV(3),CCOEF_WV(3)
REAL*8      :: ACOEF_RTWV(3),BCOEF_RTWV(3),CCOEF_RTWV(3)
INTEGER     :: LSPECRADMENU(NSPECRADMENU) !indices for number of species
                                           ! on input menu
INTEGER, ALLOCATABLE :: SPECMASK(:) !list of binary switches for different
                                     !species flux output

! RH indices
INTEGER, ALLOCATABLE :: IRHARR(:,:,:)

#if defined( RRTMG )
!to pass to RT code
!one for each hydrophilic/hydrophobic aerosol and optical dust bin
!and also sulfate, nitrate and ammonia are separate too
REAL*8, ALLOCATABLE :: RTODAER(:,:,:,,:)
REAL*8, ALLOCATABLE :: RTSSAER(:,:,:,,:)
REAL*8, ALLOCATABLE :: RTASYMAER(:,:,:,,:)
#endif

CONTAINS
EOC
-----
GEOS-Chem Global Chemical Transport Model !
-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\mbox{}\hrulefill\

\subsection [Init\_Cmn\_Fjx] {Init\_Cmn\_Fjx}

```

Routine INIT\CMN\FJX initializes quantities based on the grid-independent size parameters.

```

\\
\\{\bf INTERFACE:}
\begin{verbatim}
      SUBROUTINE Init_CMN_FJX( am_I_Root, Input_Opt, RC )

```

USES:

```

      USE ErrCode_Mod
      USE Input_Opt_Mod, ONLY : OptInput

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
      TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT) :: RC           ! Success or failure?

```

REVISION HISTORY:

- 21 Feb 2014 - M. Sulprizio- Renamed from Set_CMN_FJX_MOD to Init_CMN_FJX.
We now set several variables here that depend on LLPAR, since LLPAR is no longer a parameter and can't be used to define a parameter.
- 29 Dec 2017 - C. Keller - Initialize RNames to empty strings.
- 03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input_Opt%LUCX
- 23 Mar 2018 - E. Lundgren - Move NWVSELECT, WVSELECT, and STRWVSELECT to Input_Opt since read in from input.geos

1.3.1 Cleanup_Cmn_Fjx

Subroutine CLEANUP_CMN_FJX deallocates all module arrays.

INTERFACE:

```

      SUBROUTINE Cleanup_CMN_FJX( am_I_Root, RC )

```

USES:

```

      USE ErrCode_Mod

```

INPUT PARAMETERS:

```

      LOGICAL, INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?

```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

21 Feb 2014 - M. Sulprizio- Initial version

1.4 Fortran: Module Interface registry_params_mod.F90

Contains parameters that are used to denote the types of pointers arrays (e.g. REAL(fp), REAL(f4), INTEGER) used in the GEOS-Chem Registry and History routines, as well as the vertical location.

INTERFACE:

```
MODULE Registry_Params_Mod
```

USES:

```
USE Precision_Mod
IMPLICIT NONE
```

DEFINED PARAMETERS:

```
! Denotes REAL(fp), aka "flexible precision
INTEGER, PUBLIC, PARAMETER :: KINDVAL_FP = 1

! Denotes REAL(f8), aka REAL*8 or 8-byte floating point
INTEGER, PUBLIC, PARAMETER :: KINDVAL_F8 = 2

! Denotes REAL(f4), aka REAL*4, or 4-byte floating point
INTEGER, PUBLIC, PARAMETER :: KINDVAL_F4 = 3

! Denotes INTEGER
INTEGER, PUBLIC, PARAMETER :: KINDVAL_I4 = 4

! Vertical locations
INTEGER, PUBLIC, PARAMETER :: VLocationNone = 0
INTEGER, PUBLIC, PARAMETER :: VLocationEdge = 1
INTEGER, PUBLIC, PARAMETER :: VLocationCenter = 2
```

REVISION HISTORY:

14 Jul 2017 - R. Yantosca - Initial version
25 Aug 2017 - R. Yantosca - Add KINDVAL_F8 parameter for REAL*8 data
07 Sep 2017 - E. Lundgren - Add vertical location parameters

1.5 Fortran: Module Interface CMN_SIZE_mod.F

CMN_SIZE contains size parameters for GEOS-Chem arrays.

INTERFACE:

```
MODULE CMN_SIZE_MOD
```

USES:

```
USE PRECISION_MOD, ONLY : fpp => fp      ! For GEOS-Chem Precision (fpp)
```

```
IMPLICIT NONE
```

```
PUBLIC
```

DEFINED PARAMETERS:

```

=====
! DISIZE = size (in degrees) of a longitude grid box
! DJSIZE = size (in degrees) of a latitude grid box
! NOTE: DISIZE and DJSIZE are only needed for "GEOS-Chem Classic",
! as these are used to define the Delta-Lon and Delta-Lat values
! that need to get saved to the binary punch files.  If you are
! using "GEOS-Chem HP", then you can set this to a dummy value.
! (mlong, bmy, 5/22/15)
=====
#if defined( GRID4x5 )
REAL(fpp), PARAMETER :: DISIZE = 5.0e+0_fpp
REAL(fpp), PARAMETER :: DJSIZE = 4.0e+0_fpp
#elif defined( GRID2x25 )
REAL(fpp), PARAMETER :: DISIZE = 2.5e+0_fpp
REAL(fpp), PARAMETER :: DJSIZE = 2.0e+0_fpp
#elif defined( GRID05x0625 )
REAL(fpp), PARAMETER :: DISIZE = 0.625e+0_fpp
REAL(fpp), PARAMETER :: DJSIZE = 0.5e+0_fpp
#elif defined( GRID025x03125 )
REAL(fpp), PARAMETER :: DISIZE = 0.3125e+0_fpp
REAL(fpp), PARAMETER :: DJSIZE = 0.25e+0_fpp
#elif defined( EXTERNAL_GRID )
REAL(fpp)           :: DISIZE
REAL(fpp)           :: DJSIZE
#endif

=====
! GRID SETTINGS #1: Necessary for the grid-independent GEOS-Chem
=====
INTEGER           :: I_LO      ! Minimum lon index on this CPU
INTEGER           :: J_LO      ! Minimum lat index on this CPU
INTEGER           :: I_HI      ! Maximum lon index on this CPU
INTEGER           :: J_HI      ! Maximum lat index on this CPU

```

```

INTEGER          :: IM_WORLD      ! # of lons in the global grid
INTEGER          :: JM_WORLD      ! # of lats in the global grid
INTEGER          :: LM_WORLD      ! # of levls in the global grid
REAL(fpp), ALLOCATABLE :: DLON(:, :, :) ! Array of delta-lon [degrees]
REAL(fpp), ALLOCATABLE :: DLAT(:, :, :) ! Array of delta-lat [degrees]

!=====
! GRID SETTINGS #2: Mostly historical declarations (keep for now)
! IGLOB          = global longitude dimension
! JGLOB          = global latitude dimension
! LGLOB          = max number of sigma levels
! IIPAR          = window longitude dimension
! JJPARG          = window latitude dimension
! LLPARG          = window vertical dimension
! LLTROP         = maximum number of tropospheric levels for variable
!                 tropopause
! LLTROP_FIX    = number of tropospheric levels for offline simulations
! LLSTRAT        = maximum number of levels below stratopause
! LLCHEM         = maximum number of levels included in chemistry grid
! LLCHEM_FIX    = number of chemgrid levels for offline simulations
! PTOPT          = model top pressure (mb)
! Most of the time, GEOS-CHEM is used for global simulations.
! In this case, then IIPAR=IGLOB, JJPARG=JGLOB, LLPARG=LGLOB.
! For nested grids, then IIPAR<IGLOB, JJPARG<JGLOB, LLPARG<LGLOB.
!=====

! IIPAR and JJPARG are always set at runtime
! for both GEOS-Chem HP and GEOS-Chem Classic
INTEGER          :: IIPARG
INTEGER          :: JJPARG

#if defined ( EXTERNAL_GRID ) || defined( EXTERNAL_FORCING ) || defined( ESMF_ )
!-----
!           %%%%%%%%% GEOS-Chem HP (with ESMF & MPI) %%%%%%%%%
! We need to make all of these be variables rather than
! parameters, so that they can be defined directly from the
! ESMF interface. (bmy, 10/11/12)
!-----
INTEGER          :: IGLOB
INTEGER          :: JGLOB
INTEGER          :: LGLOB
INTEGER          :: LLPARG
INTEGER          :: LLTROP_FIX
INTEGER          :: LLTROP
INTEGER          :: LLSTRAT
REAL*8,         PARAMETER :: PTOPT          = 0.01d0
#else

```



```

!-----
!           %%%%%%%%% GEOS-Chem CLASSIC (with OpenMP) %%%%%%%%%
! For GEOS-Chem "Classic", we can preset the values of IGLOB,
! JGLOB, LGLOB, LLTROP, LLTROP_FIX, LLSTRAT, (bmy, 5/22/15)
!-----

#if defined( GEOS_FP ) && defined( GRID025x03125 ) && defined( NESTED_CH )

!-----
! GEOS-FP: Nested China Grid
!-----
INTEGER           :: IGLOB       = 225
INTEGER           :: JGLOB       = 161
INTEGER           :: LGLOB       = 72
#if defined( GRIDREDUCED )
INTEGER           :: LLPAR       = 47           ! Reduced vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 38           ! -- 47 levels
INTEGER, PARAMETER :: LLTROP    = 38
INTEGER, PARAMETER :: LLSTRAT   = 44
#else
INTEGER           :: LLPAR       = 47           ! Full vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 40           ! -- 72 levels
INTEGER, PARAMETER :: LLTROP    = 40
INTEGER, PARAMETER :: LLSTRAT   = 59
#endif
REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( GEOS_FP ) && defined( GRID025x03125 ) && defined( NESTED_NA)

!-----
! GEOS-FP Nested NA Grid
!-----
INTEGER           :: IGLOB       = 225
INTEGER           :: JGLOB       = 202
INTEGER           :: LGLOB       = 72
#if defined( GRIDREDUCED )
INTEGER           :: LLPAR       = 47           ! Reduced vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 38           ! -- 47 levels
INTEGER, PARAMETER :: LLTROP    = 38
INTEGER, PARAMETER :: LLSTRAT   = 44
#else
INTEGER           :: LLPAR       = 47           ! Full vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 40           ! -- 72 levels
INTEGER, PARAMETER :: LLTROP    = 40
INTEGER, PARAMETER :: LLSTRAT   = 59
#endif
REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

```

```

!Anna Prot added the following sizes for nesting EU, 7 May 2015
#elif defined( GEOS_FP ) && defined( GRID025x03125 ) && defined( NESTED_EU)

!-----
! GEOS-FP Nested EU Grid
!-----
INTEGER          :: IGLOB      = 177
INTEGER          :: JGLOB      = 115
INTEGER          :: LGLOB      = 72
#if defined( GRIDREDUCED )
INTEGER          :: LLPAR      = 47      ! Reduced vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 38      ! -- 47 levels
INTEGER, PARAMETER :: LLTROP    = 38
INTEGER, PARAMETER :: LLSTRAT   = 44
#else
INTEGER          :: LLPAR      = 47      ! Full vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 40      ! -- 72 levels
INTEGER, PARAMETER :: LLTROP    = 40
INTEGER, PARAMETER :: LLSTRAT   = 59
#endif
REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( GEOS_FP ) && defined( GRID2x25 )

!-----
! GEOS-FP: 2 x 2.5
!-----
INTEGER          :: IGLOB      = 144
INTEGER          :: JGLOB      = 91
INTEGER          :: LGLOB      = 72
#if defined( GRIDREDUCED )
INTEGER          :: LLPAR      = 47      ! Reduced vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 38      ! -- 47 levels
INTEGER, PARAMETER :: LLTROP    = 38
INTEGER, PARAMETER :: LLSTRAT   = 44
#else
INTEGER          :: LLPAR      = 47      ! Full vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 40      ! -- 72 levels
INTEGER, PARAMETER :: LLTROP    = 40
INTEGER, PARAMETER :: LLSTRAT   = 59
#endif
REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( GEOS_FP ) && defined( GRID4x5 )

!-----
! GEOS-FP: 4 x 5
!-----

```

```

        INTEGER          :: IGLOBAL = 72
        INTEGER          :: JGLOBAL = 46
        INTEGER          :: LGLOBAL = 72
#if defined( GRIDREDUCED )
        INTEGER          :: LLPAR    = 47      ! Reduced vertical grid
        INTEGER, PARAMETER :: LLTROP_FIX = 38    ! -- 47 levels
        INTEGER, PARAMETER :: LLTROP   = 38
        INTEGER, PARAMETER :: LLSTRAT  = 44
#else
        INTEGER          :: LLPAR    = 72      ! Full vertical grid
        INTEGER, PARAMETER :: LLTROP_FIX = 40    ! -- 72 levels
        INTEGER, PARAMETER :: LLTROP   = 40
        INTEGER, PARAMETER :: LLSTRAT  = 59
#endif
        REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( MERRA2 ) && defined( GRID05x0625 ) && defined( NESTED_AS )

        !-----
        ! MERRA2: Nested China Grid
        !-----
        INTEGER          :: IGLOBAL = 145
        INTEGER          :: JGLOBAL = 133
        INTEGER          :: LGLOBAL = 72
#if defined( GRIDREDUCED )
        INTEGER          :: LLPAR    = 47      ! Reduced vertical grid
        INTEGER, PARAMETER :: LLTROP_FIX = 38    ! -- 47 levels
        INTEGER, PARAMETER :: LLTROP   = 38
        INTEGER, PARAMETER :: LLSTRAT  = 44
#else
        INTEGER          :: LLPAR    = 72      ! Full vertical grid
        INTEGER, PARAMETER :: LLTROP_FIX = 40    ! -- 72 levels
        INTEGER, PARAMETER :: LLTROP   = 40
        INTEGER, PARAMETER :: LLSTRAT  = 59
#endif
        REAL(fpp), PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( MERRA2 ) && defined( GRID05x0625 ) && defined( NESTED_NA )

        !-----
        ! MERRA2: Nested NA Grid
        !-----
        INTEGER          :: IGLOBAL = 161
        INTEGER          :: JGLOBAL = 121
        INTEGER          :: LGLOBAL = 72
#if defined( GRIDREDUCED )
        INTEGER          :: LLPAR    = 47      ! Reduced vertical grid
        INTEGER, PARAMETER :: LLTROP_FIX = 38    ! -- 47 levels

```

```

        INTEGER,    PARAMETER :: LLTROP    = 38
        INTEGER,    PARAMETER :: LLSTRAT   = 44
#else
        INTEGER          :: LLPAR          ! Full vertical grid
        INTEGER,    PARAMETER :: LLTROP_FIX = 40    ! -- 72 levels
        INTEGER,    PARAMETER :: LLTROP    = 40
        INTEGER,    PARAMETER :: LLSTRAT   = 59
#endif
        REAL(fpp),    PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( MERRA2 ) && defined( GRID05x0625 ) && defined( NESTED_EU )

        !-----
        ! MERRA2: Nested EU Grid
        !-----
        INTEGER          :: IGLOB          = 129
        INTEGER          :: JGLOB          = 81
        INTEGER          :: LGLOB          = 72
#if defined( GRIDREDUCED )
        INTEGER          :: LLPAR          = 47    ! Reduced vertical grid
        INTEGER,    PARAMETER :: LLTROP_FIX = 38    ! -- 47 levels
        INTEGER,    PARAMETER :: LLTROP    = 38
        INTEGER,    PARAMETER :: LLSTRAT   = 44
#else
        INTEGER          :: LLPAR          ! Full vertical grid
        INTEGER,    PARAMETER :: LLTROP_FIX = 40    ! -- 72 levels
        INTEGER,    PARAMETER :: LLTROP    = 40
        INTEGER,    PARAMETER :: LLSTRAT   = 59
#endif
        REAL(fpp),    PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( MERRA2 ) && defined( GRID2x25 )

        !-----
        ! MERRA2: 2 x 2.5
        !-----
        INTEGER          :: IGLOB          = 144
        INTEGER          :: JGLOB          = 91
        INTEGER          :: LGLOB          = 72
#if defined( GRIDREDUCED )
        INTEGER          :: LLPAR          = 47    ! Reduced vertical grid
        INTEGER,    PARAMETER :: LLTROP_FIX = 38    ! -- 47 levels
        INTEGER,    PARAMETER :: LLTROP    = 38
        INTEGER,    PARAMETER :: LLSTRAT   = 44
#else
        INTEGER          :: LLPAR          ! Full vertical grid
        INTEGER,    PARAMETER :: LLTROP_FIX = 40    ! -- 72 levels
        INTEGER,    PARAMETER :: LLTROP    = 40

```

```

        INTEGER,    PARAMETER :: LLSTRAT    = 59
#endif
        REAL(fpp),  PARAMETER :: PTOP      = 0.01e+0_fpp

#elif defined( MERRA2 ) && defined( GRID4x5 )

!-----
! MERRA2: 4 x 5
!-----
INTEGER          :: IGLOB    = 72
INTEGER          :: JGLOB    = 46
INTEGER          :: LGLOB    = 72
#if defined( GRIDREDUCED )
INTEGER          :: LLPAR     = 47      ! Reduced vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 38      ! -- 47 levels
INTEGER, PARAMETER :: LLTROP  = 38
INTEGER, PARAMETER :: LLSTRAT = 44
#else
INTEGER          :: LLPAR     = 72      ! Full vertical grid
INTEGER, PARAMETER :: LLTROP_FIX = 40      ! -- 72 levels
INTEGER, PARAMETER :: LLTROP  = 40
INTEGER, PARAMETER :: LLSTRAT = 59
#endif
REAL(fpp),  PARAMETER :: PTOP      = 0.01e+0_fpp

#endif
#endif

!-----
! For stratospheric chemistry
!-----
INTEGER          :: LLCHEM
INTEGER          :: LLCHEM_FIX

!=====
! OTHER PARAMETERS
!=====

! NSURFTYPE - Maximum number of surface types: 73 olson
! NTYPE     - Maximum number of veg types in a CTM grid box
! NPOLY     - Number of coefficients for polynomial fits
INTEGER, PARAMETER :: NSURFTYPE = 73
#if defined( EXTERNAL_GRID ) || defined( EXTERNAL_TYPE )
!-----
!          %%%%%%%%% GEOS-Chem HP (with ESMF & MPI) %%%%%%%%%
! For testing the ESMF interface to GEOS-Chem with a grid that
! is smaller than the usual 72x46, increase NTYPE (bmy, 12/4/12)
!-----

```

```

    INTEGER,    PARAMETER :: NTYPE    = 50
#else
    !-----
    !          %%%%%%%%% GEOS-Chem CLASSIC (with OpenMP) %%%%%%%%%
    ! Current practice in the std GEOS-Chem is to set NTYPE to 25,
    ! which is large enough if using the Olson 2001 land map at
    ! 0.25 x 0.25 resolution. (bmy, 12/4/12)
    !-----
    INTEGER,    PARAMETER :: NTYPE    = 25
#endif
    INTEGER,    PARAMETER :: NPOLY    = 20

-----

Prior to 3/19/18:
This is only used to dimension the ND14 array, but we should use LLPAR
instead. (bmy, 3/19/18)
    ! LLCONVM - Max number of layers for convection
    INTEGER          :: LLCONVM

-----

    ! MAXFAM -- Max number of families for prod and loss output
    INTEGER,    PARAMETER :: MAXFAM = 40

    ! MAXMEM is maximum number of families of prod and loss
    ! moved from input_mod and diag_pl_mod to here (hotp 7/31/09)
    ! MAXMEM also increased from 10 to 20 by FP
    ! MAXMEM increased from 20 to 22 for bromine (jpp, mpayer, 12/28/11)
    INTEGER,    PARAMETER :: MAXMEM = 22

    ! MAXPL increased from 100 to 500 and moved from diag_pl_mod
    ! to here by FP (hotp 7/31/09)
    INTEGER,    PARAMETER :: MAXPL = 500

    ! NDUST -- Number of FAST-J aerosol size bins (rvm, bmy, 11/15/01)
    INTEGER,    PARAMETER :: NDUST = 7

    ! NRHAER -- Number of aerosols undergoing hygroscopic growth
    INTEGER,    PARAMETER :: NRHAER = 5

    ! NSTRATAER -- Number of stratospheric aerosols (SDE 04/17/13)
    INTEGER,    PARAMETER :: NSTRATAER = 2

    ! NAER -- number of other aerosol categories (rvm, bmy, 2/27/02)
    ! Now set in Init_CMN_SIZE below (mps, 1/3/18)
    INTEGER          :: NAER

    ! NRH -- number of relative humidity bins (rvm, bmy, 2/27/02)
    INTEGER,    PARAMETER :: NRH = 5

```

```

#if defined( TOMAS )
    INTEGER,    PARAMETER :: TOMASSPEC = 8

# if defined( TOMAS40 )
    INTEGER,    PARAMETER :: NDSTBIN   = 40
    INTEGER,    PARAMETER :: TOMASBIN  = 40
# elif defined( TOMAS15 )
    INTEGER,    PARAMETER :: NDSTBIN   = 15
    INTEGER,    PARAMETER :: TOMASBIN  = 15
# elif defined( TOMAS12 )
    INTEGER,    PARAMETER :: NDSTBIN   = 12
    INTEGER,    PARAMETER :: TOMASBIN  = 12
# else
    ! NDSTBIN -- redimensioned for TOMAS (dwest, bmy, 2/1/10)
    INTEGER,    PARAMETER :: NDSTBIN   = 30
    INTEGER,    PARAMETER :: TOMASBIN  = 30 ! Number of TOMAS bins
# endif

#else
    ! NDSTBIN -- number of dust size bins for transport (tdf, bmy, 3/31/04)
    INTEGER,    PARAMETER :: NDSTBIN   = 4
#endif

    ! NJDAY is the array that is used to determine when it is time
    ! to end a GEOS-Chem run.  Moved this here from CMN_DIAG_mod.F,
    ! so that we can phase that module out. (bmy, 1/16/18)
    INTEGER          :: NJDAY(366)

```

REMARKS:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% NOTE: THIS WAS MADE INTO A MODULE IN ORDER TO REMOVE COMMON BLOCKS %%
%% WE WILL KEEP THIS FOR NOW.  EVENTUALLY WE MIGHT MIGRATE THESE DATA %%
%% INTO A DERIVED TYPE OBJECT. (bmy, 12/3/12)                               %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

REVISION HISTORY:

- (1) Now set LLTROP = 20 for GEOS-3 (bmy, 4/12/01)
- (2) Eliminated obsolete commented-out code (bmy, 4/20/01)
- (3) Now set MAXFAM = 12 for more P-L families (bmy, 6/28/01)
- (4) Comment out {IJL}GCMPAR -- these are obsolete (bmy, 9/24/01)
- (5) Also set LLPAR = 30 for GEOS-3, will regrid online (bmy, 9/24/01)
- (6) Removed obsolete code from 9/01 (bmy, 10/23/01)
- (7) Removed NAIR, LAIREMS, these are now defined
in "aircraft_nox_mod.f" (bmy, 2/14/02)
- (8) Eliminated commented-out code from 2/14/02. Also added NAER
and NRH parameters for aerosols. (rvm, bmy, 2/27/02)

- (9) Removed IM, JM, IMX, JMX to avoid namespace pollution. This is needed to get the new TPCORE to work. Also changed RCS ID tag comment character from "C" to "!" to allow freeform compilation. (bmy, 6/25/02)
- (10) Removed obsolete code from 6/02 (bmy, 8/26/02)
- (11) Added NUMDEP_SULF in a common block for sulfate dry deposition. Also set MAXDEP=31 and NNPARG=31 for coupled fullchem/sulfate simulations. (rjp, bdf, bmy, 11/15/02)
- (12) Removed IO, JO; these are now superseded by "grid_mod.f" (bmy, 2/11/03)
- (13) Added parameters for GEOS-4 (bmy, 6/18/03)
- (14) Now defines both 55 level and 30 level GEOS-4 grids. Also define LLTROP=19 for GEOS-4 grids. Also remove obsolete GEOS-2 grid declarations. (bmy, 10/31/03)
- (15) LLTROP should be 17 for GEOS-4...based on the ND55 diagnostic when computed for 2003 met fields (bmy, 2/18/04)
- (16) Increase NNPARG from 31 to 39 for carbon & dust tracers. Also declare NDSTBIN as # of dust bins. (rvn, tdf, bmy, 4/1/04)
- (17) Increase NNPARG to 41 for seasalt tracers (rjp, bec, bmy, 4/20/04)
- (18) Increase NNPARG to 50 for SOA tracers (rjp, bmy, 7/15/04)
- (19) Now use NESTED_CH and NESTED_NA cpp switches to define parameters for 1x1 nested grids. Also add parameters for the 1 x 1.25 global grid. (bmy, 12/1/04)
- (20) Now add parameters for GCAP and GEOS-5 grids. Remove references to obsolete LGEOSCO and FULLCHEM Cpp switches (bmy, 6/24/05)
- (21) Now add I1x1 and J1x1 parameters for data on the 1x1 GEOS grid. (bmy, 10/24/05)
- (22) Increase NNPARG to 52 (bmy, 12/6/05)
- (23) Increase NNPARG to 54 (dkh, bmy, 5/22/06)
- (24) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
- (25) Added variable tropopause case (bmy, phs, bdf, 8/21/06)
- (26) Set LLTROP to level of first box entirely above 20km for GEOS-3 and GEOS-4 (phs, 9/14/06)
- (27) Bug fix: set LLTROP_FIX = LLPARG for GCAP (bmy, 11/29/06)
- (28) Reset vertical coordinates for GEOS-5. Also renamed GRID30LEV to GRIDREDUCED (bmy, 4/3/07)
- (29) New parameters for GEOS-5 nested grid (yxw, dan, bmy, 11/6/08)
- (30) NEMPARG set to 12 to emit O3 and HNO3 (phs, 4/3/08)
- (31) Add tracers to NNPARG = 73. (tmf, 1/7/09)
- (32) NEMPARG set to 21 to emit new tracers for GLYX chemistry (tmf, ccc, 3/2/09)
- (33) NEMPARG set to 3 to emit MBO, MONX (tmf, ccc, 3/2/09)
- (34) Added EUROPE grid parameters (amv, 10/19/09)
- 18 Dec 2009 - Aaron van D - Added NESTED_EU grid parameters
- 18 Dec 2009 - R. Yantosca - Added ProTeX headers
- 25 Jan 2010 - R. Yantosca - Set NNPARG=320 for TOMAS simulations
- 25 Jan 2010 - R. Yantosca - Define TOMASBIN and TOMASPEC for TOMAS sims
- 08 Feb 2010 - F. Paulot - Increase NNPARG, NEMPARG and NEMPARG

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt  ! Input Options object
INTEGER,         OPTIONAL    :: value_I_LO  ! Lower lon index on this CPU
INTEGER,         OPTIONAL    :: value_J_LO  ! Lower lat index on this CPU
INTEGER,         OPTIONAL    :: value_I_HI  ! Upper lon index on this CPU
INTEGER,         OPTIONAL    :: value_J_HI  ! Upper lat index on this CPU
INTEGER,         OPTIONAL    :: value_IM    ! # of lons   on this CPU
INTEGER,         OPTIONAL    :: value_JM    ! # of lats   on this CPU
INTEGER,         OPTIONAL    :: value_LM    ! # of levels on this CPU
INTEGER,         OPTIONAL    :: value_IM_WORLD! # of lons in the global grid
INTEGER,         OPTIONAL    :: value_JM_WORLD! # of lats in the global grid
INTEGER,         OPTIONAL    :: value_LM_WORLD! # of levs in the global grid
INTEGER,         OPTIONAL    :: value_LLSTROP ! # of levs in troposphere
INTEGER,         OPTIONAL    :: value_LLSTRAT ! # of levs in stratosphere

```

OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC          ! Success or failure?

```

REVISION HISTORY:

```

15 Oct 2012 - M. Long      - Initial version
15 Oct 2012 - R. Yantosca - Added ProTeX Headers, use F90 format/indents
22 Oct 2012 - R. Yantosca - Renamed to GIGC_Init_Dimensions
03 Dec 2012 - R. Yantosca - Rewritten for clarity. Also pass optional
                           arguments to carry values from ESMF environment
03 Dec 2012 - R. Yantosca - Now allocate DLON, DLAT arrays here.
04 Aug 2015 - C. Keller    - Added value_LLSTROP and value_LLSTRAT.
03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input_Opt%LUCX
05 Jul 2018 - H. Lin      - Initialize DLAT to 0 (previously missing)

```

1.5.2 Cleanup_Cmn_Size

Subroutine CLEANUP_CMN_SIZE deallocates all module arrays.

INTERFACE:

```

SUBROUTINE Cleanup_CMN_SIZE( am_I_Root, RC )

```

USES:

```

USE ErrCode_Mod

```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

```
3 Dec 2012 - R. Yantosca - Initial version
```

1.6 Fortran: Module Interface state_met_mod.F90

Module STATE_MET_MOD contains the derived type used to define the Meteorology State object for GEOS-Chem.

This module also contains the routines that allocate and deallocate memory to the Meteorology State object. The Meteorology State object is not defined in this module. It must be declared as variable in the top-level driver routine, and then passed to lower-level routines as an argument.

INTERFACE:

```
MODULE State_Met_Mod
  USES:
  USE ErrorCode_Mod
  USE Precision_Mod
  USE Registry_Mod
```

```
  IMPLICIT NONE
  PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
  PUBLIC :: Init_State_Met
  PUBLIC :: Cleanup_State_Met
  PUBLIC :: Get_Metadata_State_Met
```

PRIVATE MEMBER FUNCTIONS:

```
  PRIVATE :: Register_MetField
```

PUBLIC DATA MEMBERS:

```
  !=====
  ! Derived type for Meteorology State
  !=====
  TYPE, PUBLIC :: MetState
```

```

!-----
! Surface fields
!-----
REAL(fp), POINTER :: ALBD      (:, : ) ! Visible surface albedo [1]
INTEGER,  POINTER :: ChemGridLev (:, : ) ! Chemistry grid level
REAL(fp), POINTER :: CLDFRC    (:, : ) ! Column cloud fraction [1]
INTEGER,  POINTER :: CLDTOPS   (:, : ) ! Max cloud top height [levels]
REAL(fp), POINTER :: EFLUX     (:, : ) ! Latent heat flux [W/m2]
!REAL(fp), POINTER :: EVAP     (:, : ) ! Surface evap [kg/m2/s]
REAL(fp), POINTER :: FRCLND    (:, : ) ! Olson land fraction [1]
REAL(fp), POINTER :: FRLAKE    (:, : ) ! Fraction of lake [1]
REAL(fp), POINTER :: FRLAND    (:, : ) ! Fraction of land [1]
REAL(fp), POINTER :: FRLANDIC  (:, : ) ! Fraction of land ice [1]
REAL(fp), POINTER :: FROCEAN   (:, : ) ! Fraction of ocean [1]
REAL(fp), POINTER :: FRSEAICE  (:, : ) ! Sfc sea ice fraction
REAL(fp), POINTER :: FRSNO     (:, : ) ! Sfc snow fraction
!REAL(fp), POINTER :: GRN      (:, : ) ! Greenness fraction
REAL(fp), POINTER :: GWETROOT  (:, : ) ! Root soil wetness [1]
REAL(fp), POINTER :: GWETTOP   (:, : ) ! Top soil moisture [1]
REAL(fp), POINTER :: HFLUX     (:, : ) ! Sensible heat flux [W/m2]
REAL(fp), POINTER :: LAI       (:, : ) ! Leaf area index [m2/m2]
! (online)
REAL(fp), POINTER :: LWI       (:, : ) ! Land/water indices [1]
REAL(fp), POINTER :: PARDR     (:, : ) ! Direct photosynthetically
! active radiation [W/m2]
REAL(fp), POINTER :: PARDF     (:, : ) ! Diffuse photosynthetically
! active radiation [W/m2]
REAL(fp), POINTER :: PBLH      (:, : ) ! PBL height [m]
INTEGER,  POINTER :: PBL_TOP_L (:, : ) ! PBL top layer [1]
REAL(fp), POINTER :: PHIS      (:, : ) ! Surface geopotential height
! [m2/s2]
REAL(fp), POINTER :: PRECANV   (:, : ) ! Anvil precip @ ground
! [kg/m2/s]
REAL(fp), POINTER :: PRECCON   (:, : ) ! Conv precip @ ground
! [kg/m2/s]
REAL(fp), POINTER :: PRECTOT   (:, : ) ! Total precip @ ground
! [kg/m2/s]
REAL(fp), POINTER :: PRECLSC   (:, : ) ! LS precip @ ground [kg/m2/s]
!REAL(fp), POINTER :: PRECSNO  (:, : ) ! Snow precip [kg/m2/s]
REAL(fp), POINTER :: PS1_WET   (:, : ) ! Wet surface pressure at
! start of timestep [hPa]
REAL(fp), POINTER :: PS2_WET   (:, : ) ! Wet surface pressure at
! end of timestep [hPa]
REAL(fp), POINTER :: PSC2_WET  (:, : ) ! Wet interpolated surface
! pressure [hPa]
REAL(fp), POINTER :: PS1_DRY   (:, : ) ! Dry surface pressure at
! start of timestep [hPa]
REAL(fp), POINTER :: PS2_DRY   (:, : ) ! Dry surface pressure at

```

```

! end of timestep [hPa]
REAL(fp), POINTER :: PSC2_DRY      (:,:) ! Dry interpolated surface
! pressure [hPa]
!REAL(fp), POINTER :: RADLWG      (:,:) ! Net longwave radiation @
! ground [W/m2]
REAL(fp), POINTER :: SEAICE00     (:,:) ! Sea ice coverage 00-10%
REAL(fp), POINTER :: SEAICE10     (:,:) ! Sea ice coverage 10-20%
REAL(fp), POINTER :: SEAICE20     (:,:) ! Sea ice coverage 20-30%
REAL(fp), POINTER :: SEAICE30     (:,:) ! Sea ice coverage 30-40%
REAL(fp), POINTER :: SEAICE40     (:,:) ! Sea ice coverage 40-50%
REAL(fp), POINTER :: SEAICE50     (:,:) ! Sea ice coverage 50-60%
REAL(fp), POINTER :: SEAICE60     (:,:) ! Sea ice coverage 60-70%
REAL(fp), POINTER :: SEAICE70     (:,:) ! Sea ice coverage 70-80%
REAL(fp), POINTER :: SEAICE80     (:,:) ! Sea ice coverage 80-90%
REAL(fp), POINTER :: SEAICE90     (:,:) ! Sea ice coverage 90-100%
REAL(fp), POINTER :: SLP           (:,:) ! Sea level pressure [hPa]
REAL(fp), POINTER :: SNODP        (:,:) ! Snow depth [m]
REAL(fp), POINTER :: SNOMAS       (:,:) ! Snow mass [kg/m2]
REAL(fp), POINTER :: SUNCOS       (:,:) ! COS(solar zenith angle) at
! current time
REAL(fp), POINTER :: SUNCOSmid    (:,:) ! COS(solar zenith angle) at
! midpoint of chem timestep
REAL(fp), POINTER :: SWGDN        (:,:) ! Incident radiation @ ground
! [W/m2]
REAL(fp), POINTER :: TO3          (:,:) ! Total overhead O3 column [DU]
REAL(fp), POINTER :: TROPP        (:,:) ! Tropopause pressure [hPa]
INTEGER, POINTER :: TropLev       (:,:) ! Tropopause level [1]
REAL(fp), POINTER :: TropHt       (:,:) ! Tropopause height [km]
REAL(fp), POINTER :: TS           (:,:) ! Surface temperature [K]
REAL(fp), POINTER :: TSKIN        (:,:) ! Surface skin temperature [K]
REAL(fp), POINTER :: U10M         (:,:) ! E/W wind speed @ 10m ht [m/s]
REAL(fp), POINTER :: USTAR        (:,:) ! Friction velocity [m/s]
REAL(fp), POINTER :: UVALBEDO     (:,:) ! UV surface albedo [1]
REAL(fp), POINTER :: V10M         (:,:) ! N/S wind speed @ 10m ht [m/s]
REAL(fp), POINTER :: ZO           (:,:) ! Surface roughness height [m]
REAL(fp), POINTER :: CNV_FRC      (:,:) ! Convective fraction [1]

!-----
! 3-D Fields
!-----
REAL(fp), POINTER :: AREA_M2      (:,:,) ! Grid box surface area [cm2]
REAL(fp), POINTER :: CLDF         (:,:,) ! 3-D cloud fraction [1]
REAL(fp), POINTER :: CMFMC        (:,:,) ! Cloud mass flux [kg/m2/s]
REAL(fp), POINTER :: DQRCU        (:,:,) ! Conv precip production rate
! [kg/kg/s] (assume per
! dry air)
REAL(fp), POINTER :: DQRLSAN      (:,:,) ! LS precip prod rate [kg/kg/s]
! (assume per dry air)

```

```

REAL(fp), POINTER :: DTRAIN      (:, :, :) ! Detrainment flux [kg/m2/s]
REAL(fp), POINTER :: OMEGA      (:, :, :) ! Updraft velocity [Pa/s]
REAL(fp), POINTER :: OPTD       (:, :, :) ! Visible optical depth [1]
REAL(fp), POINTER :: PEDGE      (:, :, :) ! Wet air press @ level
! edges [hPa]
REAL(fp), POINTER :: PFICU      (:, :, :) ! Dwn flux ice prec:conv
! [kg/m2/s]
REAL(fp), POINTER :: PFILSAN    (:, :, :) ! Dwn flux ice prec:LS+anv
! [kg/m2/s]
REAL(fp), POINTER :: PFLCU      (:, :, :) ! Dwn flux liq prec:conv
! [kg/m2/s]
REAL(fp), POINTER :: PFLLSAN    (:, :, :) ! Dwn flux ice prec:LS+anv
! [kg/m2/s]
!REAL(fp), POINTER :: PV        (:, :, :) ! Potential vorticity
! [kg*m2/kg/s]
REAL(fp), POINTER :: QI         (:, :, :) ! Ice mixing ratio
! [kg/kg dry air]
REAL(fp), POINTER :: QL         (:, :, :) ! Water mixing ratio
! [kg/kg dry air]
REAL(fp), POINTER :: REEVAPCN   (:, :, :) ! Evap of precip conv [kg/kg/s]
! (assume per dry air)
REAL(fp), POINTER :: REEVAPLS   (:, :, :) ! Evap of precip LS+anvil
! [kg/kg/s] (assume per
! dry air)
REAL(fp), POINTER :: RH         (:, :, :) ! Relative humidity [%]
REAL(fp), POINTER :: SPHU       (:, :, :) ! Specific humidity
! [g H2O/kg tot air]
REAL(fp), POINTER :: SPHU1      (:, :, :) ! Specific humidity at start
! of timestep [g/kg]
REAL(fp), POINTER :: SPHU2      (:, :, :) ! Specific humidity at end
! of timestep [g/kg]
REAL(fp), POINTER :: T          (:, :, :) ! Temperature [K]
REAL(fp), POINTER :: TAUCLI     (:, :, :) ! Opt depth of ice clouds [1]
REAL(fp), POINTER :: TAUCLW     (:, :, :) ! Opt depth of H2O clouds [1]
REAL(fp), POINTER :: TMPU1      (:, :, :) ! Temperature at start of
! timestep [K]
REAL(fp), POINTER :: TMPU2      (:, :, :) ! Temperature at end of
! timestep [K]
REAL(fp), POINTER :: U          (:, :, :) ! E/W component of wind [m s-1]
REAL(fp), POINTER :: UPDVVEL    (:, :, :) ! Updraft vertical velocity
! [hPa/s]
REAL(fp), POINTER :: V          (:, :, :) ! N/S component of wind [m s-1]

```

```

!-----
! Air quantities assigned in AIRQNT
!-----

```

```

! Note on pressures: PMID is calculated from PEDGE,
! and dry air pressures assume constant RH and T across grid box

```

```

REAL(fp), POINTER :: PEDGE_DRY      (:,:,) ! Dry air partial pressure
! @ level edges [hPa]
REAL(fp), POINTER :: PMID          (:,:,) ! Average wet air pressure [hPa]
! defined as arithmetic
! average of edge pressures
REAL(fp), POINTER :: PMID_DRY     (:,:,) ! Dry air partial pressure [hPa]
! defined as arithmetic avg
! of edge pressures
REAL(fp), POINTER :: THETA        (:,:,) ! Potential temperature [K]
REAL(fp), POINTER :: TV           (:,:,) ! Virtual temperature [K]
REAL(fp), POINTER :: MAIRDEN      (:,:,) ! Moist air density [kg/m3]
REAL(fp), POINTER :: AIRDEN       (:,:,) ! Dry air density [kg/m3]
REAL(fp), POINTER :: AIRNUMDEN    (:,:,) ! Dry air density [molec/cm3]
REAL(fp), POINTER :: AVGW         (:,:,) ! Water vapor volume mixing
! ratio [vol H2O/vol dry air]
REAL(fp), POINTER :: BXHEIGHT     (:,:,) ! Grid box height [m] (dry air)
REAL(fp), POINTER :: DELP         (:,:,) ! Delta-P (wet) across box [hPa]
REAL(fp), POINTER :: DELP_DRY     (:,:,) ! Delta-P (dry) across box [hPa]
REAL(fp), POINTER :: AD           (:,:,) ! Dry air mass [kg] in grid box
REAL(fp), POINTER :: AIRVOL       (:,:,) ! Grid box volume [m3] (dry air)
REAL(fp), POINTER :: DP_DRY_PREV  (:,:,) ! Previous State_Met%DELP_DRY

!-----
! Offline land type, leaf area index, and chlorophyll fields
!-----
INTEGER, POINTER :: IREG          (:, : ) ! # of landtypes in box (I,J)
INTEGER, POINTER :: ILAND         (:,:,) ! Land type at (I,J);
! 1..IREG(I,J)
INTEGER, POINTER :: IUSE          (:,:,) ! Fraction (per mil) of box
! (I,J) occupied by each land
! type
REAL(fp), POINTER :: MODISLAI     (:, : ) ! Daily LAI computed from
! monthly offline MODIS [m2/m2]
REAL(fp), POINTER :: MODISCHLR   (:, : ) ! Daily chlorophyll-a computed
! from offline monthly MODIS
REAL(fp), POINTER :: XLAI        (:,:,) ! MODIS LAI per land type,
! for this month
REAL(fp), POINTER :: XCHLR       (:,:,) ! MODIS CHLR per land type,
! for this month
REAL(fp), POINTER :: LandTypeFrac (:,:,) ! Olson frac per type (I,J,type)
REAL(fp), POINTER :: XLAI_NATIVE (:,:,) ! avg LAI per type (I,J,type)
REAL(fp), POINTER :: XCHLR_NATIVE (:,:,) ! avg CHLR per type (I,J,type)

!-----
! Fields for querying in which vertical regime a grid box is in
! or if a grid box is near local noon solar time
!-----
LOGICAL, POINTER :: InChemGrid    (:,:,) ! Are we in the chemistry grid?

```

```

LOGICAL, POINTER :: InPbl      (:, :, :) ! Are we in the PBL?
LOGICAL, POINTER :: InStratMeso (:, :, :) ! Are we in the stratosphere
!                                     ! or mesosphere?
LOGICAL, POINTER :: InStratosphere (:, :, :) ! Are we in the stratosphere?
LOGICAL, POINTER :: InTroposphere (:, :, :) ! Are we in the troposphere?
REAL(fp), POINTER :: LocalSolarTime (:, : ) ! Local solar time
LOGICAL, POINTER :: IsLocalNoon (:, : ) ! Is it local noon (between 11
!                                     ! and 13 local solar time?

```

```

!-----
! Registry of variables contained within State_Met
!-----
CHARACTER(LEN=3)          :: State      = 'MET'      ! Name of this state
TYPE(MetaRegItem), POINTER :: Registry => NULL()    ! Registry object

```

END TYPE MetState

REMARKS:

In MERRA2, PS and SLP are kept in Pa (not converted to hPa).

REVISION HISTORY:

```

19 Oct 2012 - R. Yantosca - Initial version, split off from gc_type_mod.F90
23 Oct 2012 - R. Yantosca - Added QI, QL met fields to the derived type
15 Nov 2012 - M. Payer    - Added all remaining met fields
12 Dec 2012 - R. Yantosca - Add IREG, ILAND, IUSE fields for dry deposition
13 Dec 2012 - R. Yantosca - Add XLAI, XLAI2 fields for dry deposition
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
15 Nov 2013 - R. Yantosca - Now denote that RH fields have units of [%]
10 Oct 2014 - C. Keller   - Added ITY (needed for GIGC). For now, this value
                           is just initialized to 1.0 and not modified any
                           more.
05 Nov 2014 - M. Yannetti - Changed REAL*8 to REAL(fp)
12 Feb 2015 - C. Keller   - Added UPDVVEL (for use in wet scavenging).
24 Feb 2015 - E. Lundgren - Add PEDGE_DRY, PMID_DRY, and MAIRDEN
03 Mar 2015 - E. Lundgren - Add TV (virtual temperature)
16 Apr 2015 - E. Lundgren - Add mean pressures PMEAN and PMEAN_DRY. Clarify
                           definition of PMID as arithmetic average P.
                           Add MOISTMW to use TCVV with moist mixing ratio.
25 May 2015 - C. Keller   - Removed SUNCOSmid5 (now calculated by HEMCO).
08 Jul 2015 - E. Lundgren - Add XCHLR and XCHLR2 for organic marine aerosols
11 Aug 2015 - R. Yantosca - Extend #ifdefs for MERRA2 met fields
22 Sep 2015 - E. Lundgren - Add SWGDN for incident radiation at ground
28 Oct 2015 - E. Lundgren - Add previous delta-P and specific humidity for
                           tracer mass conservation in mixing ratio update
04 Mar 2016 - C. Keller   - Add CNV_FRC for convective fraction. Currently
                           not a standard GEOS-FP output, only used in
                           online model (ESMF).

```


21 Dec 2015 - M. Sulprizio- Add AIRNUMDEN, which is the same as AIRDEN but has units molec/cm3 for the chemistry routines.

17 Mar 2016 - M. Sulprizio- Remove OPTDEP. Instead, we now solely use OPTD.

03 May 2016 - E. Lundgren - Add PSC2_DRY, PS1_DRY, and PS2_DRY

06 Jul 2016 - E. Lundgren - Rename PS1, PS2, and PSC1: add '_WET' suffix

06 Jul 2016 - E. Lundgren - Add DELP_DRY and DP_DRY_PREV

19 Jul 2016 - E. Lundgren - Remove PMEAN, PMEAN_DRY, MOISTMW, and ADMOIST

16 Aug 2016 - M. Sulprizio- Rename from gign_state_chm_mod.F90 to state_chm_mod.F90. The "gign" nomenclature is no longer used.

18 Oct 2016 - E. Lundgren - Remove XLAI2, CHLR2; add MODISLAI, MODISCHLR to replace modis_lai_mod-level GC_LAI and GC_CHLR

19 Oct 2016 - E. Lundgren - Use NSURFTYPE as the # of land types

03 Feb 2017 - M. Sulprizio- Add OMEGA for use in sulfate_mod.F (Q. Chen)

26 Jun 2017 - R. Yantosca - Added StateName and Registry to type MetState

27 Jun 2017 - R. Yantosca - Add fields of State_Met to the registry

24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA and remove obsolete met fields from State_Met

07 Sep 2017 - E. Lundgren - Add Register_MetField interface for init

13 Sep 2017 - M. Sulprizio- Remove DELP_PREV and SPHU_PREV; they're not used

14 Sep 2017 - M. Sulprizio- Comment out met fields that aren't actually used in GEOS-Chem (EVAP, GRN, PRECSNO, PV, RADLWG)

26 Sep 2017 - E. Lundgren - Remove Lookup_State_Met and Print_State_Met

07 Nov 2017 - R. Yantosca - Add tropht and troplev fields

08 Jan 2018 - R. Yantosca - Added logical query fields

31 Jan 2018 - E. Lundgren - Remove underscores from diagnostic names

1.6.1 Init_State_Met

Subroutine INIT_STATE_MET allocates all fields of the meteorology state object.

INTERFACE:

```
SUBROUTINE Init_State_Met(am_I_Root, State_Met, RC )
```

USES:

```
USE CMN_SIZE_MOD, ONLY : IIPAR, JJPARG, LLPARG, NSURFTYPE
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState),  INTENT(INOUT)  :: State_Met    ! Obj for meteorology state
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC           ! Return code
```

REMARKS:

For consistency, maybe this should be moved to a different module.

REVISION HISTORY:

19 Oct 2012 - R. Yantosca - Initial version, based on gc_environment_mod.F90
19 Oct 2012 - R. Yantosca - Now pass all dimensions as arguments
23 Oct 2012 - R. Yantosca - Now allocate QI, QL fields
15 Nov 2012 - M. Payer - Added all remaining met fields
16 Nov 2012 - R. Yantosca - Now zero all fields after allocating
27 Nov 2012 - R. Yantosca - Now allocate SUNCOS fields (IM,JM)
12 Dec 2012 - R. Yantosca - Now allocate the IREG, ILAND, IUSE fields
13 Dec 2012 - R. Yantosca - Now allocate the XLAI, XLAI2 fields
07 Mar 2013 - R. Yantosca - Now allocate PF*LSAN, PF*CU fields properly
for GEOS-5.7.x met (they are edged)
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
22 Aug 2014 - R. Yantosca - Allocate PBL_TOP_L field
05 Nov 2014 - R. Yantosca - Now use 0.0_fp instead of Od0
06 Nov 2014 - R. Yantosca - Now make all fields (IM,JM,LM) instead of
(LM,JM,IM), to facilitate use w/in GEOS-5 GCM
05 Oct 2016 - R. Yantosca - Swapped order of HKETA and HKBETA allocation
28 Nov 2016 - R. Yantosca - Nullify fields that may or may not be allocated
01 Jun 2017 - C. Keller - Initialize UPDVVEL to -999.0 to ensure that
GET_VUD (wetscav_mod.F) works properly.
26 Jun 2017 - R. Yantosca - Now register each variable after it's allocated
24 Aug 2017 - R. Yantosca - Now register level-edged variables appropriately
07 Sep 2017 - E. Lundgren - Abstract the metadata and method add to registry
16 Nov 2017 - E. Lundgren - Get grid params from CMN_Size_Mod not arguments

1.6.2 Cleanup_State_Met

Subroutine CLEANUP_STATE_MET deallocates all fields of the meteorology state object.

INTERFACE:

```
SUBROUTINE Cleanup_State_Met( am_I_Root, State_Met, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met    ! Obj for meteorology state
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)  :: RC          ! Return code
```

REVISION HISTORY:

```
19 Oct 2012 - R. Yantosca - Initial version, based on gc_environment_mod.F90
23 Oct 2012 - R. Yantosca - Now deallocate QI, QL fields
15 Nov 2012 - M. Payer    - Added all remaining met fields
19 Nov 2012 - R. Yantosca - Segregate DEALLOCATE statements w/ #ifdefs
                        for each met field data product type
27 Nov 2012 - R. Yantosca - Now deallocate the SUNCOS fields
12 Dec 2012 - R. Yantosca - Now deallocate the IREG, ILAND, IUUSE fields
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
22 Aug 2014 - R. Yantosca - Deallocate PBL_TOP_L field
```

1.6.3 Get_Metadata_State_Met

Subroutine GET_METADATA_STATE_MET retrieves basic information about each State_Met field.

INTERFACE:

```
SUBROUTINE Get_Metadata_State_Met( am_I_Root, metadataID, Found, &
                                   RC,          Desc,          Units, &
                                   Rank,       Type,          VLoc )
```

USES:

```
USE Charpak_Mod,          ONLY: To_UpperCase
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Is this the root CPU?
CHARACTER(LEN=*), INTENT(IN)  :: metadataID ! State_Met field ID
```

OUTPUT PARAMETERS:

```
LOGICAL,          INTENT(OUT) :: Found      ! Item found?
INTEGER,          INTENT(OUT) :: RC         ! Return code
CHARACTER(LEN=255), OPTIONAL  :: Desc      ! Long name string
CHARACTER(LEN=255), OPTIONAL  :: Units     ! Units string
INTEGER,          OPTIONAL    :: Rank      ! # of dimensions
INTEGER,          OPTIONAL    :: Type      ! Desc of data type
INTEGER,          OPTIONAL    :: VLoc      ! Vertical placement
```

REMARKS:

REVISION HISTORY:

```
28 Aug 2017 - E. Lundgren - Initial version
01 Nov 2017 - R. Yantosca - Now get To_UpperCase from charpak_mod.F90
```

1.6.4 Register_MetField_Rfp_2D

Registers a 2-D State_Met field (flexible precision).

INTERFACE:

```

SUBROUTINE Register_MetField_Rfp_2D( am_I_Root, metadataID, Ptr2Data,      &
                                     State_Met, RC                       )

```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(fp),         POINTER         :: Ptr2Data(:, :) ! pointer to met
TYPE(MetState),   INTENT(IN)      :: State_Met     ! Obj for met state

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC              ! Success/failure

```

REMARKS:

REVISION HISTORY:

07 Sep 2017 - E. Lundgren - Initial version

1.6.5 Register_MetField_Rfp_3D

Registers a 3-D State_Met field (flexible precision).

INTERFACE:

```

SUBROUTINE Register_MetField_Rfp_3D( am_I_Root, metadataID, Ptr2Data,      &
                                       State_Met, RC                       )

```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(fp),         POINTER         :: Ptr2Data(:, :, :) ! pointer to met
TYPE(MetState),   INTENT(IN)      :: State_Met     ! Obj for met state

```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)  :: RC              ! Success/failure
```

REMARKS:**REVISION HISTORY:**

07 Sep 2017 - E. Lundgren - Initial version

1.6.6 Register_MetField_Int_2D

Registers a 2-D State_Met field (integer precision).

INTERFACE:

```
SUBROUTINE Register_MetField_Int_2D( am_I_Root, metadataID, Ptr2Data,      &
                                     State_Met, RC                          )
```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)    :: metadataID     ! Name
INTEGER,          POINTER       :: Ptr2Data(:, :) ! pointer to met
TYPE(MetState),  INTENT(INOUT) :: State_Met     ! Obj for met state
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC              ! Success/failure
```

REMARKS:**REVISION HISTORY:**

07 Sep 2017 - E. Lundgren - Initial version

1.6.7 Register_MetField_Int_3D

Registers a 3-D State_Met field (integer precision).

INTERFACE:

```
SUBROUTINE Register_MetField_Int_3D( am_I_Root, metadataID, Ptr2Data,      &
                                     State_Met, RC                          )
```

USES:

```
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,           INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
INTEGER,           POINTER        :: Ptr2Data(:, :, :) ! pointer to met
TYPE(MetState),   INTENT(IN)      :: State_Met      ! Obj for met state
```

OUTPUT PARAMETERS:

```
INTEGER,           INTENT(OUT)    :: RC              ! Success/failure
```

REMARKS:**REVISION HISTORY:**

```
07 Sep 2017 - E. Lundgren - Initial version
```

1.7 Fortran: Module Interface registry_mod.F90

Contains derived types and methods to create a registry of each variable contained within a given module. This will allow the user to obtain a pointer to each module variable by searching for its name.

INTERFACE:

```
MODULE Registry_Mod
```

USES:

```
USE Precision_Mod
USE Registry_Params_Mod
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: Registry_AddField
PUBLIC  :: Registry_Lookup
PUBLIC  :: Registry_Print
PUBLIC  :: Registry_Destroy
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: MetaRegItem_AddNew
PRIVATE :: MetaRegItem_Create
PRIVATE :: MetaRegItem_Insert
PRIVATE :: MetaRegItem_Destroy
```

```

! PUBLIC TYPES::
!=====
! Derived type for a REGISTRY ITEM (a single registry entry).
! This represents a single module variable, plus some metadata.
!=====
TYPE, PUBLIC :: RegItem

!-----
! Identifying info
!-----
CHARACTER(LEN=67)    :: FullName      ! e.g. "STATE_VARIABLE"
CHARACTER(LEN=4 )   :: State         ! Name of state
CHARACTER(LEN=63)   :: Variable      ! Name of variable

!-----
! Metadata
!-----
CHARACTER(LEN=255)  :: Description    ! Longer description
REAL(fp)            :: MemoryInKb     ! Memory use in Kb
INTEGER             :: KindVal        ! Numerical KIND value
INTEGER             :: Rank           ! Dimensions of data
CHARACTER(LEN=255)  :: Units          ! Units of data
CHARACTER(LEN=3)    :: DimNames       ! e.g. "xyz", "yz", "y", "t"
LOGICAL             :: OnLevelEdges   ! Is data on level edges (T/F)?

!-----
! Pointers to floating point data (flexible precision)
!-----
REAL(fp), POINTER   :: Ptr0d          ! For 0D flex-prec data
REAL(fp), POINTER   :: Ptr1d (: )     ! For 1D flex-prec data
REAL(fp), POINTER   :: Ptr2d (:, : )  ! For 2D flex-prec data
REAL(fp), POINTER   :: Ptr3d (:, :, : ) ! For 3D flex-prec data

!-----
! Pointers to floating point data (8-byte precision)
!-----
REAL(f8), POINTER   :: Ptr0d_8        ! For 0D 8-byte data
REAL(f8), POINTER   :: Ptr1d_8(: )    ! For 1D 8-byte data
REAL(f8), POINTER   :: Ptr2d_8(:, : ) ! For 2D 8-byte data
REAL(f8), POINTER   :: Ptr3d_8(:, :, : ) ! For 3D 8-byte data

!-----
! Pointers to floating point data (4-byte precision)
!-----
REAL(f4), POINTER   :: Ptr0d_4        ! For 0D 4-byte data
REAL(f4), POINTER   :: Ptr1d_4(: )    ! For 1D 4-byte data
REAL(f4), POINTER   :: Ptr2d_4(:, : ) ! For 2D 4-byte data
REAL(f4), POINTER   :: Ptr3d_4(:, :, : ) ! For 3D 4-byte data

```

```

!-----
! Pointers to integer data
!-----
INTEGER, POINTER    :: Ptr0d_I      ! For 0D int data
INTEGER, POINTER    :: Ptr1d_I(: )  ! For 1D int data
INTEGER, POINTER    :: Ptr2d_I(:, : ) ! For 2D int data
INTEGER, POINTER    :: Ptr3d_I(:, :, : ) ! For 3D int data

END TYPE RegItem

!=====
! Derived type for a METAREGISTRY ITEM (a linked-list of REGISTRY ITEMS)
!=====
TYPE, PUBLIC :: MetaRegItem
    TYPE(MetaRegItem), POINTER :: Next => NULL() ! Pointer to next node
    TYPE(RegItem ), POINTER :: Item => NULL() ! Registry item within
END TYPE MetaRegItem

```

DEFINED PARAMETERS:**REMARKS:**

In Fortran 2003, the maximum variable name length is 63 characters, so we have declared the various character name fields of RegItem accordingly.

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version
27 Jun 2017 - R. Yantosca - Added integer data fields, and description
29 Jun 2017 - R. Yantosca - Added Ptr1DI to type RegItem
30 Jun 2017 - R. Yantosca - Add more pointers to 4-byte and integer data
23 Aug 2017 - R. Yantosca - Added OnLevelEdges field
31 Oct 2017 - R. Yantosca - Move Str2Hash, To_UpperCase to charpak_mod.F90

1.7.1 Registry_AddField

Creates a REGISTRY ITEM, which contains information (i.e. metadata plus a pointer to the data) about a variable within a module. The REGISTRY ITEM will then be added to the METAREGISTRY ITEM, which is the master list of all variables in the module. This will allow the user to obtain a pointer to any variable by searching for its name.

INTERFACE:

```

SUBROUTINE Registry_AddField( am_I_Root, Registry, State,      &
                             Variable, RC, Description,      &
                             Units, DimNames, OnLevelEdges,  &

```



```

Data0d,    Data1d,    Data2d,           &
Data3d,    Data0d_8,  Data1d_8,         &
Data2d_8,  Data3d_8,  Data0d_4,         &
Data1d_4,  Data2d_4,  Data3d_4,         &
Data0d_I,  Data1d_I,  Data2d_I,         &
Data3d_I                                     )

```

USES:

```

USE CharPak_Mod, ONLY : Str2Hash31, To_Uppercase
USE ErrCode_Mod

```

INPUT PARAMETERS:

```

! General identifying information
LOGICAL,          INTENT(IN)          :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)          :: State          ! State name
CHARACTER(LEN=*), INTENT(IN)          :: Variable       ! variable
CHARACTER(LEN=*), OPTIONAL            :: Description    ! Long description
CHARACTER(LEN=*), OPTIONAL            :: Units         ! Units of data
CHARACTER(LEN=*), OPTIONAL            :: DimNames      ! "xyz", "xy", "t"
LOGICAL,          OPTIONAL            :: OnLevelEdges  ! Set =T if data
                                                    ! is on level edges

! Floating-point data targets (flexible precision)
REAL(fp),         OPTIONAL, TARGET :: Data0d          ! 0D flex-prec data
REAL(fp),         OPTIONAL, TARGET :: Data1d (:      ) ! 1D flex_prec data
REAL(fp),         OPTIONAL, TARGET :: Data2d (:, :   ) ! 2D flex-prec data
REAL(fp),         OPTIONAL, TARGET :: Data3d (:, :, ) ! 3D flex-prec data

! Floating-point data targets (8-byte precision)
REAL(f8),         OPTIONAL, TARGET :: Data0d_8        ! 0D flex-prec data
REAL(f8),         OPTIONAL, TARGET :: Data1d_8(:      ) ! 1D flex_prec data
REAL(f8),         OPTIONAL, TARGET :: Data2d_8(:, :   ) ! 2D flex-prec data
REAL(f8),         OPTIONAL, TARGET :: Data3d_8(:, :, ) ! 3D flex-prec data

! Floating-point data targets (4-byte precision)
REAL(f4),         OPTIONAL, TARGET :: Data0d_4        ! 0D 4-byte data
REAL(f4),         OPTIONAL, TARGET :: Data1d_4(:      ) ! 1D 4-byte data
REAL(f4),         OPTIONAL, TARGET :: Data2d_4(:, :   ) ! 2D 4-byte data
REAL(f4),         OPTIONAL, TARGET :: Data3d_4(:, :, ) ! 3D 4-byte data

! Integer data targets
INTEGER,          OPTIONAL, TARGET :: Data0d_I        ! 1D int data
INTEGER,          OPTIONAL, TARGET :: Data1d_I(:      ) ! 1D int data
INTEGER,          OPTIONAL, TARGET :: Data2d_I(:, :   ) ! 2D int data
INTEGER,          OPTIONAL, TARGET :: Data3d_I(:, :, ) ! 3D int data

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetaRegItem), POINTER          :: Registry      ! Registry object

```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)      :: RC          ! Success/failure
```

REMARKS:

Internally, the REGISTRY ITEM will be referred to by its fullname field, which is "STATE_VARIABLE". Fullname will be defined automatically from the STATE and VARIABLE inputs as STATE_VARIABLE, unless variable is in State_Diag, in which case STATE_ is not appended as a prefix.

REVISION HISTORY:

```
23 Jun 2017 - R. Yantosca - Initial version
26 Jun 2017 - R. Yantosca - Changed "StateName" to "State", "Name" to
                          "Variable", and added "MemoryInKb"
27 Jun 2017 - R. Yantosca - Now assigns description and KIND value
29 Jun 2017 - R. Yantosca - Added DataIdI for 1D integer data
14 Jul 2017 - R. Yantosca - Now use KINDVAL parameters defined at the top
                          of the module to denote the different data types
23 Aug 2017 - R. Yantosca - Added OnLevelEdges argument so that we can
                          register data placed on level edges
24 Aug 2017 - R. Yantosca - Added the DimNames argument
25 Aug 2017 - R. Yantosca - Added DataOd_8 and DataId_8 so that we can
                          register netCDF index variables. Most other
                          data should be either REAL(fp) or REAL(f4)
25 Sep 2017 - E. Lundgren - Only use state name prefix if not from state_diag
06 Oct 2017 - R. Yantosca - Add pointers for 2D and 3D, 8-byte data
```

1.7.2 Registry_Lookup

Get a pointer to any variable in a module (aka "state") by searching for its name. Also returns associated metadata.

INTERFACE:

```
SUBROUTINE Registry_Lookup( am_I_Root,   Registry, State,      &
                          Variable,    RC,      Description,  &
                          Dimensions,   KindVal, MemoryInKb,  &
                          OnLevelEdges, Rank,    Units,         &
                          DimNames,     Ptr0d,   Ptr1d,         &
                          Ptr2d,        Ptr3d,   Ptr0d_8,       &
                          Ptr1d_8,      Ptr2d_8,  Ptr3d_8,       &
                          Ptr0d_4,      Ptr1d_4,  Ptr2d_4,       &
                          Ptr3d_4,      Ptr0d_I,  Ptr1d_I,       &
                          Ptr2d_I,      Ptr3d_I  )
```

USES:

```
USE Charpak_Mod, ONLY : Str2Hash31, To_UpperCase
USE ErrCode_Mod
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN) :: am_I_Root          ! Root CPU?
TYPE(MetaRegItem), POINTER :: Registry            ! Registry obj
CHARACTER(LEN=*), INTENT(IN) :: State              ! State name
CHARACTER(LEN=*), INTENT(IN) :: Variable          ! Variable name

```

OUTPUT PARAMETERS:

```

! Required outputs
INTEGER,          INTENT(OUT) :: RC                ! Success or failure

! Optional outputs
CHARACTER(LEN=255), OPTIONAL :: Description       ! Description of data
INTEGER,          OPTIONAL :: KindVal             ! Numerical KIND value
REAL(fp),         OPTIONAL :: MemoryInKb         ! Memory usage
INTEGER,          OPTIONAL :: Rank                ! Size of data
INTEGER,          OPTIONAL :: Dimensions(3)       ! Dimensions of data
CHARACTER(LEN=255), OPTIONAL :: Units            ! Units of data
CHARACTER(LEN=3),  OPTIONAL :: DimNames          ! "xyz", "xz", "t" etc.
LOGICAL,          OPTIONAL :: OnLevelEdges       ! Is the data defined
! on level edges (T/F)

! Floating-point data pointers (flex-precision)
REAL(fp),  POINTER, OPTIONAL :: Ptr0d            ! 0D flex-prec data
REAL(fp),  POINTER, OPTIONAL :: Ptr1d ( :      ) ! 1D flex-prec data
REAL(fp),  POINTER, OPTIONAL :: Ptr2d (:, :    ) ! 2D flex-prec data
REAL(fp),  POINTER, OPTIONAL :: Ptr3d (:, :, :  ) ! 3D flex-prec data

! Floating-point data pointers (4-byte precision)
REAL(f8),  POINTER, OPTIONAL :: Ptr0d_8         ! 0D 8-byte data
REAL(f8),  POINTER, OPTIONAL :: Ptr1d_8( :      ) ! 1D 8-byte data
REAL(f8),  POINTER, OPTIONAL :: Ptr2d_8( :, :    ) ! 2D flex-prec data
REAL(f8),  POINTER, OPTIONAL :: Ptr3d_8( :, :, :  ) ! 3D flex-prec data

! Floating-point data pointers (4-byte precision)
REAL(f4),  POINTER, OPTIONAL :: Ptr0d_4         ! 0D 4-byte data
REAL(f4),  POINTER, OPTIONAL :: Ptr1d_4( :      ) ! 1D 4-byte data
REAL(f4),  POINTER, OPTIONAL :: Ptr2d_4( :, :    ) ! 2D 4-byte data
REAL(f4),  POINTER, OPTIONAL :: Ptr3d_4( :, :, :  ) ! 3D 4-byte data

! Integer data pointers
INTEGER,    POINTER, OPTIONAL :: Ptr0d_I        ! 0D integer data
INTEGER,    POINTER, OPTIONAL :: Ptr1d_I( :      ) ! 1D integer data
INTEGER,    POINTER, OPTIONAL :: Ptr2d_I( :, :    ) ! 2D integer data
INTEGER,    POINTER, OPTIONAL :: Ptr3d_I( :, :, :  ) ! 3D integer data

```

REMARKS:

Internally, the REGISTRY ITEM will be referred to by its fullname field, which is "STATE_VARIABLE". Fullname will be defined automatically from the STATE and VARIABLE inputs as STATE_VARIABLE, unless variable is in State_Diag, in which case STATE_ is not appended as a prefix.

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version
 26 Jun 2017 - R. Yantosca - Changed "StateName" to "State", "Name" to "Variable", and added "MemoryInKb"
 27 Jun 2017 - R. Yantosca - Also added "Description" and "KindVal" outputs
 30 Jun 2017 - R. Yantosca - Added more pointers for 4-byte and integer data
 14 Jul 2017 - R. Yantosca - Now return the appropriate pointer variable based on the KINDVAL parameter, and set other passed pointer arguments to NULL.
 14 Jul 2017 - R. Yantosca - Now throw an error if the variable cannot be found in the registry
 23 Aug 2017 - R. Yantosca - Added optional OnLevelEdges argument
 24 Aug 2017 - R. Yantosca - Added optional DimNames argument
 25 Aug 2017 - R. Yantosca - Added optional Data0d_8 and Data1d_8 arguments
 25 Sep 2017 - E. Lundgren - Only use state name prefix if not from state_diag
 06 Oct 2017 - R. Yantosca - Add Ptr2d_8, Ptr3d_8 optional arguments
 01 Nov 2017 - R. Yantosca - Now use Str2Hash31 from charpak_mod.F90, which computes a hash from an input string of 31 chars
 01 Nov 2017 - R. Yantosca - Make the registry lookup case-insensitive

1.7.3 Registry_Print

Prints each REGISTRY ITEM belonging to a METAREGISTRY ITEM. In other words, this prints information about each field contained within a module.

INTERFACE:

```
SUBROUTINE Registry_Print( am_I_Root, Registry, RC, ShortFormat )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root      ! Are we on the root CPU?
LOGICAL,          OPTIONAL    :: ShortFormat    ! Print less information
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER    :: Registry      ! Registry of state fields
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC             ! Success or failure?
```

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version
 26 Jun 2017 - R. Yantosca - Also print memory usage in Kb
 27 Jun 2017 - R. Yantosca - Now print numeric KIND value and description
 27 Jun 2017 - R. Yantosca - Also print Ptr1dI, Ptr2dI, Ptr3dI
 29 Jun 2017 - R. Yantosca - Add SHORTFORMAT option to print less output
 23 Aug 2017 - R. Yantosca - Now print OnLevelEdges in full format
 24 Aug 2017 - R. Yantosca - Now use the DimNames field when printing
 25 Aug 2017 - R. Yantosca - Now print the vertical cell position: C or E
 25 Aug 2017 - R. Yantosca - Now print info about Data0d_8 and Data1d_8
 29 Aug 2017 - R. Yantosca - Minor fix, skip C/E printing for non-3D fields

1.7.4 Registry_Destroy

Destroys a METAREGISTRY ITEM (i.e. a linked list of REGISTRY ITEMS), each of which contains information (i.e. metadata, plus a pointer to the data source) for a field contained within a module.

INTERFACE:

```
SUBROUTINE Registry_Destroy( am_I_Root, Registry, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER    :: Registry    ! Registry of state fields
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC          ! Success or failure?
```

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version

1.7.5 MetaRegItem_AddNew

Wrapper for methods MetaRegItem_Create and MetaRegItem_Insert. Will create a METAREGISTRY ITEM (containing a REGISTRY ITEM) and (1) set it as the head node of a new linked list, or (2) append it to an existing linked list.

INTERFACE:

```
SUBROUTINE MetaRegItem_AddNew( am_I_Root, Node, Item, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
TYPE(RegItem),   POINTER     :: Item       ! REGISTRY ITEM object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER    :: Node      ! METAREGISTRY ITEM object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC        ! Success or failure
```

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version, based on code by Arjen Markus

1.7.6 MetaRegItem_Create

This method creates a new METAREGISTRY ITEM (to contain the supplied REGISTRY ITEM) and sets it as the head node of a linked list.

INTERFACE:

```
SUBROUTINE MetaRegItem_Create( am_I_Root, Node, Item, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
TYPE(RegItem),   POINTER     :: Item       ! REGISTRY ITEM object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER    :: Node      ! METAREGISTRY ITEM object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC        ! Success or failure
```

REMARKS:

This method is not intended to be called directly, but is rather wrapped by the MetaRegItem_AddNew method.

REVISION HISTORY:

23 Jun 2017 - R. Yantosca - Initial version, based on code by Arjen Markus

1.7.7 MetaRegItem_Insert

Creates a new METAREGISTRY ITEM (to contain the supplied REGISTRY ITEM), and pops it into an existing linked list, immediately following the head node.

INTERFACE:

```
SUBROUTINE MetaRegItem_Insert( am_I_Root, Node, Item, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root ! Are we on the root CPU?
TYPE(RegItem),    POINTER     :: Item      ! REGISTRY ITEM object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER     :: Node      ! METAREGISTRY ITEM object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC        ! Success or failure
```

REMARKS:

This method is not intended to be called directly, but is rather wrapped by the MetaRegItem_AddNew method.

REVISION HISTORY:

```
23 Jun 2017 - R. Yantosca - Initial version, based on code by Arjen Markus
06 Oct 2017 - R. Yantosca - Now insert new node at the head of the list
```

1.7.8 MetaRegItem_Destroy

This method will destroy the REGISTRY ITEM belonging to each METAREGISTRY ITEM (aka node) of a linked list. It will then destroy each METAREGISTRY ITEM itself.

INTERFACE:

```
SUBROUTINE MetaRegItem_Destroy( am_I_Root, List, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetaRegItem), POINTER      :: List      ! List of METAREGISTRY ITEMS
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC          ! Success or failure?
```

REMARKS:**REVISION HISTORY:**

```
23 Jun 2017 - R. Yantosca - Initial version, based on code by Arjen Markus
29 Jun 2017 - R. Yantosca - Now nullify pointers to integer fields
```

1.8 Fortran: Module Interface state_diag_mod.F90

Module STATE_DIAG_MOD contains the derived type used to define the Diagnostics State object for GEOS-Chem.

This module also contains the routines that allocate and deallocate memory to the Diagnostics State object. The Diagnostics State object is not defined in this module. It must be declared as variable in the top-level driver routine, and then passed to lower-level routines as an argument.

INTERFACE:

```
MODULE State_Diag_Mod
  USES:

  USE CMN_Size_Mod,    ONLY : IIPAR, JJPAR, LLPAR, NDUST
  USE DiagList_Mod
  USE ErrCode_Mod
  USE Precision_Mod
  USE Registry_Mod
  USE Species_Mod,    ONLY : Species
  USE State_Chm_Mod,  ONLY : ChmState

  IMPLICIT NONE
  PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Init_State_Diag
PUBLIC :: Cleanup_State_Diag
PUBLIC :: Get_Metadata_State_Diag
PUBLIC :: Get_NameInfo
PUBLIC :: Get_TagInfo
!PRIVATE MEMBER FUNCTIONS
PRIVATE :: Register_DiagField
```


PUBLIC DATA MEMBERS:

```

=====
! Derived type for Diagnostics State
=====
TYPE, PUBLIC :: DgnState

!-----
! Standard Simulation Diagnostic Arrays
!-----

! Concentrations
REAL(f8), POINTER :: SpeciesConc      (:,:,,:) ! Spc Conc for diag output

! Dry deposition
REAL(f4), POINTER :: DryDepChm        (:,:, ) ! Drydep flux in chemistry
REAL(f4), POINTER :: DryDepMix        (:,:, ) ! Drydep flux in mixing
REAL(f4), POINTER :: DryDep           (:,:, ) ! Total drydep flux
REAL(f4), POINTER :: DryDepVel        (:,:, ) ! Dry deposition velocity
!REAL(f4), POINTER :: DryDepRst_RA     (:,:, ) ! Aerodynamic resistance
!REAL(f4), POINTER :: DryDepRst_RB     (:,:, ) ! Aerodynamic resistance
!REAL(f4), POINTER :: DryDepRst_RC     (:,:, ) ! Total drydep resistance
!REAL(f4), POINTER :: DryDepRst_RI     (:, , ) ! Stomatal resistance
! Waiting for inputs on new resistance diagnostics commented out above

! Chemistry
REAL(f4), POINTER :: JVal              (:,:,,:) ! J-values, instantaneous
REAL(f4), POINTER :: JNoon            (:,:,,:) ! Noon J-values
REAL(f4), POINTER :: RxnRates         (:,:,,:) ! Reaction rates from KPP
REAL(f4), POINTER :: UVFluxDiffuse    (:,:, ) ! Diffuse UV flux per bin
REAL(f4), POINTER :: UVFluxDirect     (:,:, ) ! Direct UV flux per bin
REAL(f4), POINTER :: UVFluxNet        (:,:, ) ! Net UV flux per bin
REAL(f4), POINTER :: OHconcAfterChem  (:,:, ) ! OH, HO2, O1D, and O3P
REAL(f4), POINTER :: HO2concAfterChem (:,:, ) ! concentrations
REAL(f4), POINTER :: O1DconcAfterChem (:,:, ) ! upon exiting the
REAL(f4), POINTER :: O3PconcAfterChem (:,:, ) ! FlexChem solver
REAL(f4), POINTER :: Loss             (:,:,,:) ! Chemical loss of species
REAL(f4), POINTER :: Prod             (:,:,,:) ! Chemical prod of species

! Aerosol characteristics
REAL(f4), POINTER :: AerHygGrowth     (:,:,,:) ! Hygroscopic growth of spc
REAL(f4), POINTER :: AerAqVol         (:,:, ) ! Aerosol aqueous volume
REAL(f4), POINTER :: AerSurfAreaHyg   (:,:,,:) ! Surface area of
! hygroscopic grth species
REAL(f4), POINTER :: AerSurfAreaDust  (:,:, ) ! Mineral dust surface area
REAL(f4), POINTER :: AerSurfAreaSLA   (:,:, ) ! Strat liquid surf area
REAL(f4), POINTER :: AerSurfAreaPSC   (:,:, ) ! Polar strat cld surf area
REAL(f4), POINTER :: AerNumDenSLA     (:,:, ) ! Strat liquid # density

```

```

REAL(f4), POINTER :: AerNumDenPSC      (:,:, ) ! Polar strat cloud # den

! Aerosol optical depths
REAL(f4), POINTER :: AODDust           (:,:, ) ! Dust optical depth
REAL(f4), POINTER :: AODDustWL1       (:,:,,:) ! All bins 1st WL dust OD
REAL(f4), POINTER :: AODDustWL2       (:,:,,:) ! All bins 2nd WL dust OD
REAL(f4), POINTER :: AODDustWL3       (:,:,,:) ! All bins 3rd WL dust OD
REAL(f4), POINTER :: AODHygWL1        (:,:,,:) ! AOD for hygroscopic grth
REAL(f4), POINTER :: AODHygWL2        (:,:,,:) ! species @ input.geos rad
REAL(f4), POINTER :: AODHygWL3        (:,:,,:) ! wavelengths 1, 2, and 3
REAL(f4), POINTER :: AODSOAfromAqIsopWL1(:,:,!) ! AOD for SOA from aqueous
REAL(f4), POINTER :: AODSOAfromAqIsopWL2(:,:,!) ! isoprene, wavelengths
REAL(f4), POINTER :: AODSOAfromAqIsopWL3(:,:,!) ! 1, 2, and 3
REAL(f4), POINTER :: AODSLAWL1        (:,:, ) ! Strat liquid aerosol
REAL(f4), POINTER :: AODSLAWL2        (:,:, ) ! optical depths for
REAL(f4), POINTER :: AODSLAWL3        (:,:, ) ! wavelengths 1, 2, and 3
REAL(f4), POINTER :: AODPSCWL1        (:,:, ) ! Polar strat cloud
REAL(f4), POINTER :: AODPSCWL2        (:,:, ) ! optical depths for
REAL(f4), POINTER :: AODPSCWL3        (:,:, ) ! wavelengths 1, 2, and 3

! Aerosol mass and PM2.5
REAL(f4), POINTER :: AerMassASOA      (:,:, ) ! Aromatic SOA [ug/m3]
REAL(f4), POINTER :: AerMassBC        (:,:, ) ! Black carbon [ug/m3]
REAL(f4), POINTER :: AerMassINDIOL    (:,:, ) ! INDIOL [ug/m3]
REAL(f4), POINTER :: AerMassISN10A    (:,:, ) ! ISN10A [ug/m3]
REAL(f4), POINTER :: AerMassISOA      (:,:, ) ! ISOA [ug/m3]
REAL(f4), POINTER :: AerMassLVOCOA    (:,:, ) ! LVOCOA [ug/m3]
REAL(f4), POINTER :: AerMassNH4       (:,:, ) ! Nitrate [ug/m3]
REAL(f4), POINTER :: AerMassNIT       (:,:, ) ! NIT [ug/m3]
REAL(f4), POINTER :: AerMassOPOA      (:,:, ) ! OPOA [ug/m3]
REAL(f4), POINTER :: AerMassPOA       (:,:, ) ! POA [ug/m3]
REAL(f4), POINTER :: AerMassSAL       (:,:, ) ! Total seasalt [ug/m3]
REAL(f4), POINTER :: AerMassSO4       (:,:, ) ! Sulfate [ug/m3]
REAL(f4), POINTER :: AerMassSOAGX     (:,:, ) ! SOAGX [ug/m3]
REAL(f4), POINTER :: AerMassSOAIE     (:,:, ) ! SOAIE [ug/m3]
REAL(f4), POINTER :: AerMassSOAME     (:,:, ) ! SOAME [ug/m3]
REAL(f4), POINTER :: AerMassSOAMG     (:,:, ) ! SOAMG [ug/m3]
REAL(f4), POINTER :: AerMassTSOA      (:,:, ) ! Terpene SOA [ug/m3]
REAL(f4), POINTER :: BetaNO           (:,:, ) ! Beta NO [ug C/m3]
REAL(f4), POINTER :: PM25             (:,:, ) ! PM (r< 2.5 um) [ug/m3]
REAL(f4), POINTER :: TotalOA          (:,:, ) ! Sum of all OA [ug/m3]
REAL(f4), POINTER :: TotalOC          (:,:, ) ! Sum of all OC [ug/m3]
REAL(f4), POINTER :: TotalBiogenicOA  (:,:, ) ! Sum of biog OC [ug/m3]

! Advection
REAL(f4), POINTER :: AdvFluxZonal     (:,:,,:) ! EW Advective Flux
REAL(f4), POINTER :: AdvFluxMerid    (:,:,,:) ! NW Advective Flux
REAL(f4), POINTER :: AdvFluxVert     (:,:,,:) ! Vertical Advective Flux

```

```

! Mixing
REAL(f4), POINTER :: PBLMixFrac      (:,:, ) ! Frac of BL occupied by lev
REAL(f4), POINTER :: PBLFlux        (:,:,,:) ! BL mixing mass flux

! Convection
REAL(f4), POINTER :: CloudConvFlux  (:,:,,:) ! Cloud conv. mass flux
REAL(f4), POINTER :: WetLossConvFrac (:,:,,:) ! Fraction of soluble
                                                ! species lost in
                                                ! convective updraft
REAL(f4), POINTER :: WetLossConv    (:,:,,:) ! Loss in convect. updraft

! Wet deposition
REAL(f4), POINTER :: WetLossLS      (:,:,,:) ! Loss in LS rainout/washout
REAL(f4), POINTER :: PrecipFracLS   (:,:, ) ! Frac of box in LS precip
REAL(f4), POINTER :: RainFracLS     (:,:,,:) ! Frac lost to LS rainout
REAL(f4), POINTER :: WashFracLS     (:,:,,:) ! Frac lost to LS washout

! Carbon aerosols
REAL(f4), POINTER :: ProdBCPIfromBCPO(:,:, ) ! Prod BCPI from BCPO
REAL(f4), POINTER :: ProdOCPIfromOCPO(:,:, ) ! Prod OCPI from OCPO

! Sulfur aerosols prod & loss
REAL(f4), POINTER :: ProdSO2fromDMSandOH      (:,:,)
REAL(f4), POINTER :: ProdSO2fromDMSandNO3    (:,:,)
REAL(f4), POINTER :: ProdSO2fromDMS          (:,:,)
REAL(f4), POINTER :: ProdMSAfromDMS          (:,:,)
REAL(f4), POINTER :: ProdNITfromHNO3uptakeOnDust(:,:,)
REAL(f4), POINTER :: ProdSO4fromGasPhase     (:,:,)
REAL(f4), POINTER :: ProdSO4fromH2O2inCloud  (:,:,)
REAL(f4), POINTER :: ProdSO4fromO3inCloud    (:,:,)
REAL(f4), POINTER :: ProdSO4fromO2inCloudMetal (:,:,)
REAL(f4), POINTER :: ProdSO4fromO3inSeaSalt  (:,:,)
REAL(f4), POINTER :: ProdSO4fromOxidationOnDust (:,:,)
REAL(f4), POINTER :: ProdSO4fromUptakeOfH2SO4g (:,:,)
REAL(f4), POINTER :: ProdSO4fromHOBrInCloud  (:,:,)
REAL(f4), POINTER :: ProdSO4fromSR03        (:,:,)
REAL(f4), POINTER :: ProdSO4fromSRHOBr      (:,:,)
REAL(f4), POINTER :: ProdSO4fromO3s         (:,:,)
REAL(f4), POINTER :: LossHNO3onSeaSalt      (:,:,)

!-----
! Specialty Simulation Diagnostic Arrays
!-----

! Radon / Lead / Beryllium specialty simulation
REAL(f4), POINTER :: PbFromRnDecay  (:,:, ) ! Pb emitted from Rn decay
REAL(f4), POINTER :: RadDecay       (:,:,,:) ! Radioactive decay

```

```

! TOMAS aerosol microphysics specialty simulation

! CO2 specialty simulation

! CH4 specialty simulation

! Persistent Organic Pollutants specialty simulation

! Hg specialty simulation

! Radiation simulation (RRTMG)
REAL(f4), POINTER :: RadAllSkyLWSurf (:,:, ) ! All-sky LW rad @ surface
REAL(f4), POINTER :: RadAllSkyLWTOA (:,:, ) ! All-sky LW rad @ atm top
REAL(f4), POINTER :: RadAllSkySWSurf (:,:, ) ! All-sky SW rad @ surface
REAL(f4), POINTER :: RadAllSkySWTOA (:,:, ) ! All-sky SW rad @ atm top
REAL(f4), POINTER :: RadClrSkyLWSurf (:,:, ) ! Clr-sky SW rad @ surface
REAL(f4), POINTER :: RadClrSkyLWTOA (:,:, ) ! Clr-sky LW rad @ atm top
REAL(f4), POINTER :: RadClrSkySWSurf (:,:, ) ! Clr-sky SW rad @ surface
REAL(f4), POINTER :: RadClrSkySWTOA (:,:, ) ! Clr-sky SW rad @ atm top

!-----
! Registry of variables contained within State_Diag
!-----
CHARACTER(LEN=4)          :: State      = 'DIAG'   ! Name of this state
TYPE(MetaRegItem), POINTER :: Registry => NULL() ! Registry object

```

```
END TYPE DgnState
```

REMARKS:

TBD

REVISION HISTORY:

```

05 Jul 2017 - R. Yantosca - Initial version
22 Sep 2017 - E. Lundgren - Fill in content to allocate State_Diag; add
                           subroutines to get metadata and interface to
                           register fields
26 Sep 2017 - E. Lundgren - Remove Lookup_State_Diag and Print_State_Diag
05 Oct 2017 - R. Yantosca - Add separate drydep fields for chem & mixing
06 Oct 2017 - R. Yantosca - Declare SpeciesConc as an 8-byte real field
02 Nov 2017 - R. Yantosca - Update wetdep and convection diagnostic names

```

EOC

BOC

```

!MODULE INTERFACES:
INTERFACE Register_DiagField
MODULE PROCEDURE Register_DiagField_R4_2D

```

```

! real(fp) and real(8) 2D not implemented
MODULE PROCEDURE Register_DiagField_R4_3D
MODULE PROCEDURE Register_DiagField_Rfp_3D
! real(8) 3D not implemented
MODULE PROCEDURE Register_DiagField_R4_4D
! real(fp) 4D not implemented
MODULE PROCEDURE Register_DiagField_R8_4D
END INTERFACE Register_DiagField

```

PRIVATE TYPES:

```

! Shadow variables from Input_Opt
LOGICAL :: Is_UCX

```

```

CONTAINS
EOC

```

```

-----
                                GEOS-Chem Global Chemical Transport Model                                !
-----

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

\mbox{}\hrulefill\

```

```

\subsubsection [Init\_State\_Diag] {Init\_State\_Diag}

```

```

Subroutine INIT\_STATE\_DIAG allocates all fields of
the diagnostics state object.

```

```

\\

```

```

\\{\bf INTERFACE:}

```

```

\begin{verbatim}  SUBROUTINE Init_State_Diag( am_I_Root, Input_Opt, State_Chm, &
                                Diag_List, State_Diag, RC )

```

USES:

```

USE Input_Opt_Mod, ONLY : OptInput

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)      :: Input_Opt    ! Input Options object
TYPE(ChmState), INTENT(IN)      :: State_Chm     ! Chemistry state object
TYPE(DgnList),   INTENT(IN)      :: Diag_List     ! Diagnostics list object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(DgnState), INTENT(INOUT)    :: State_Diag   ! Diagnostic State object

```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Return code
```

REMARKS:

For consistency, maybe this should be moved to a different module.

REVISION HISTORY:

```
05 Jul 2017 - R. Yantosca - Initial version
22 Sep 2017 - E. Lundgren - Fill in content
06 Oct 2017 - R. Yantosca - State_Diag%SpeciesConc is now an 8-byte real
11 Oct 2017 - R. Yantosca - Bug fix: nAdvect is now defined properly
```

1.8.1 Cleanup_State_Diag

Subroutine CLEANUP_STATE_DIAG deallocates all fields of the meteorology state object.

INTERFACE:

```
SUBROUTINE Cleanup_State_Diag( am_I_Root, State_Diag, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root   ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnState), INTENT(INOUT)  :: State_Diag  ! Diagnostics State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Return code
```

REVISION HISTORY:

```
05 Jul 2017 - R. Yantosca - Initial version
05 Oct 2017 - R. Yantosca - Now put error trapping on deallocations
```

1.8.2 Get_Metadata_State_Diag

Subroutine GET_METADATA_STATE_DIAG retrieves basic information about each State_Diag field.

INTERFACE:

```
SUBROUTINE Get_Metadata_State_Diag( am_I_Root, metadataID, Found,  &
                                     RC,          Desc,          Units,  &
                                     TagId,      Rank,          Type,    &
                                     VLoc                                     )
```

USES:

```
USE Charpak_Mod,          ONLY: StrSplit, To_UpperCase
USE Registry_Params_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Is this the root CPU?
CHARACTER(LEN=*), INTENT(IN)  :: metadataID ! State_Diag field ID
```

OUTPUT PARAMETERS:

```
LOGICAL,          INTENT(OUT)      :: Found  ! Item found?
INTEGER,          INTENT(OUT)      :: RC     ! Return code
CHARACTER(LEN=255), INTENT(OUT), OPTIONAL :: Desc  ! Long name string
CHARACTER(LEN=255), INTENT(OUT), OPTIONAL :: Units ! Units string
CHARACTER(LEN=255), INTENT(OUT), OPTIONAL :: TagId ! Tag wildcard (wc)
INTEGER,          INTENT(OUT), OPTIONAL :: Rank  ! # of dimensions
INTEGER,          INTENT(OUT), OPTIONAL :: Type  ! Desc of data type
INTEGER,          INTENT(OUT), OPTIONAL :: VLoc  ! Vert placement
```

REMARKS:

If a diagnostic cannot use a wildcard, then set Tag=''.

REVISION HISTORY:

```
20 Sep 2017 - E. Lundgren - Initial version
06 Oct 2017 - R. Yantosca - State_Diag%SpeciesConc is now an 8-byte real
01 Nov 2017 - R. Yantosca - Now get To_UpperCase from charpak_mod.F90
02 Nov 2017 - R. Yantosca - Update metadata to be consistent w/ arrays
```

1.8.3 Get_TagInfo

Subroutine GET_TAGINFO retrieves basic information about tags given a wildcard string.

INTERFACE:

```
SUBROUTINE Get_TagInfo( am_I_Root, tagID, State_Chm, Found,      &
                        RC,          N,      tagName,  nTags      )
```

USES:**INPUT PARAMETERS:**

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Is this the root CPU?
CHARACTER(LEN=*), INTENT(IN)  :: tagID     ! ID of tag (e.g. wildcard)
TYPE(ChmState),  INTENT(IN)  :: State_Chm  ! Chemistry State object
INTEGER,          OPTIONAL    :: N         ! index (1 to # tags)
```

OUTPUT PARAMETERS:

```

LOGICAL,          INTENT(OUT) :: Found      ! Item found?
INTEGER,          INTENT(OUT) :: RC         ! Return code
CHARACTER(LEN=255), OPTIONAL :: tagName    ! tag name for index N
INTEGER,          OPTIONAL   :: nTags     ! # tags

```

REMARKS:**REVISION HISTORY:**

16 Nov 2017 - E. Lundgren - Initial version

1.8.4 Get_NameInfo

Subroutine GET_NAMEINFO retrieves a diagnostic name given a string in HISTORY.rc. This enables outputting a diagnostic name different from the input, useful for names that are set at run-time given information in one or more input files.

INTERFACE:

```

SUBROUTINE Get_NameInfo( am_I_Root, InName, OutName, RC )

```

USES:**INPUT PARAMETERS:**

```

LOGICAL,          INTENT(IN)  :: am_I_Root  ! Is this the root CPU?
CHARACTER(LEN=*), INTENT(IN)  :: InName     ! Name in HISTORY.rc

```

OUTPUT PARAMETERS:

```

CHARACTER(LEN=*), INTENT(OUT) :: OutName    ! Diagnostic output name
INTEGER,          INTENT(OUT) :: RC         ! Return code

```

REMARKS:**REVISION HISTORY:**

24 Jan 2018 - E. Lundgren - Initial version

1.8.5 Register_DiagField_R4_2D

Registers a 2-dimensional, 4-byte real field of State_Diag, so that we can include it in the netCDF diagnostic output archive.

INTERFACE:


```

SUBROUTINE Register_DiagField_R4_2D( am_I_Root, metadataID, Ptr2Data,      &
                                   State_Chm, State_Diag, RC              )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(f4),         POINTER         :: Ptr2Data(:, :) ! pointer to data
TYPE(ChmState),   INTENT(IN)      :: State_Chm     ! Obj for chem state
TYPE(DgnState),   INTENT(IN)      :: State_Diag    ! Obj for diag state

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC              ! Success/failure

```

REMARKS:**REVISION HISTORY:**

20 Sep 2017 - E. Lundgren - Initial version

1.8.6 Register_DiagField_R4_3D

Registers a 3-dimensional, 4-byte real field of State_Diag, so that we can include it in the netCDF diagnostic output archive.

INTERFACE:

```

SUBROUTINE Register_DiagField_R4_3D( am_I_Root, metadataID, Ptr2Data,      &
                                   State_Chm, State_Diag, RC              )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(f4),         POINTER         :: Ptr2Data(:, :, :) ! pointer to data
TYPE(ChmState),   INTENT(IN)      :: State_Chm     ! Obj for chem state
TYPE(DgnState),   INTENT(INOUT)   :: State_Diag    ! Obj for diag state

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC              ! Success/failure

```

REMARKS:**REVISION HISTORY:**

20 Sep 2017 - E. Lundgren - Initial version

1.8.7 Register_DiagField_R4_4D

Registers a 4-dimensional, 4-byte real field of State_Diag, so that we can include it in the netCDF diagnostic output archive.

INTERFACE:

```

SUBROUTINE Register_DiagField_R4_4D( am_I_Root, metadataID, Ptr2Data,      &
                                     State_Chm, State_Diag, RC           )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root          ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID         ! Name
REAL(f4),         POINTER         :: Ptr2Data(:, :, :, :) ! pointer to data
TYPE(ChmState),  INTENT(IN)      :: State_Chm          ! Obj for chem state
TYPE(DgnState),  INTENT(IN)      :: State_Diag         ! Obj for diag state

```

INPUT/OUTPUT PARAMETERS:

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC                 ! Success/failure

```

REMARKS:

REVISION HISTORY:

```

20 Sep 2017 - E. Lundgren - Initial version

```

1.8.8 Register_DiagField_Rfp_3D

Registers a 3-dimensional, 4-byte real field of State_Diag, so that we can include it in the netCDF diagnostic output archive.

INTERFACE:

```

SUBROUTINE Register_DiagField_Rfp_3D( am_I_Root, metadataID, Ptr2Data,  &
                                       State_Chm, State_Diag, RC          )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(fp),         POINTER         :: Ptr2Data(:, :, :) ! pointer to data
TYPE(ChmState),   INTENT(IN)      :: State_Chm      ! Obj for chem state
TYPE(DgnState),   INTENT(INOUT)   :: State_Diag     ! Obj for diag state

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC              ! Success/failure

```

REMARKS:**REVISION HISTORY:**

03 Nov 2017 - R. Yantosca - Initial version

1.8.9 Register_DiagField_R8_4D

Registers a 4-dimensional, 8-byte real field of State_Diag, so that we can include it in the netCDF diagnostic output archive.

INTERFACE:

```

SUBROUTINE Register_DiagField_R8_4D( am_I_Root, metadataID, Ptr2Data,      &
                                     State_Chm, State_Diag, RC              )

```

USES:

```

USE Registry_Params_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Root CPU?
CHARACTER(LEN=*), INTENT(IN)      :: metadataID     ! Name
REAL(f8),         POINTER         :: Ptr2Data(:, :, :, :) ! pointer to data
TYPE(ChmState),   INTENT(IN)      :: State_Chm      ! Obj for chem state
TYPE(DgnState),   INTENT(IN)      :: State_Diag     ! Obj for diag state

```

INPUT/OUTPUT PARAMETERS:**OUTPUT PARAMETERS:**

```

INTEGER,          INTENT(OUT)     :: RC              ! Success/failure

```

REMARKS:**REVISION HISTORY:**

20 Sep 2017 - E. Lundgren - Initial version

1.9 Fortran: Module Interface species_mod.F90

Module SPECIES_MOD contains types and routines to define the GEOS-Chem species object.

INTERFACE:

```
MODULE Species_Mod
  USES:
  USE Precision_Mod
```

```
  IMPLICIT NONE
  PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
  PUBLIC :: SpcData_Init
  PUBLIC :: SpcData_Cleanup
  PUBLIC :: Spc_Create
  PUBLIC :: Spc_GetIndx
  PUBLIC :: Spc_GetNumSpecies
  PUBLIC :: Spc_Print
```

PUBLIC TYPES:

```
!=====
! Counters for the species indices
!=====
INTEGER, PRIVATE :: AdvectCount = 0 ! Counter of advected species
INTEGER, PRIVATE :: AeroCount   = 0 ! Counter of aerosol species
INTEGER, PRIVATE :: DryDepCount = 0 ! Counter of dry-deposited species
INTEGER, PRIVATE :: GasSpcCount = 0 ! Counter of gas-phase species
INTEGER, PRIVATE :: HygGrthCount = 0 ! Counter of hygroscopic growth spc
INTEGER, PRIVATE :: KppVarCount = 0 ! Counter of variable KPP species
INTEGER, PRIVATE :: KppFixCount = 0 ! Counter of fixed KPP species
INTEGER, PRIVATE :: KppSpcCount = 0 ! Counter of species in KPP matrix
INTEGER, PRIVATE :: PhotolCount = 0 ! Counter of photolysis species
INTEGER, PRIVATE :: WetDepCount = 0 ! Counter of wet-deposited species
INTEGER, PRIVATE :: Hg0Count    = 0 ! Number of Hg0 tracers
INTEGER, PRIVATE :: Hg2Count    = 0 ! Number of Hg2 tracers
INTEGER, PRIVATE :: HgPCount    = 0 ! Number of HgP tracers

!=====
! Type for the Species Database object (vector of type Species)
!=====
TYPE, PUBLIC :: SpcPtr
  TYPE(Species), POINTER :: Info          ! Single entry of Species Database
END TYPE SpcPtr

!=====
```

```

! Type for individual species information
! (i.e. this is a single entry in the Species Database)
!=====
TYPE, PUBLIC :: Species

! Indices
INTEGER          :: ModelID          ! Model species ID
INTEGER          :: AdvectID         ! Advection index
INTEGER          :: AeroID           ! Aerosol species index
INTEGER          :: DryDepID         ! Dry deposition index
INTEGER          :: GasSpcID         ! Gas-phase species index
INTEGER          :: HygGrthID        ! Hygroscopic growth species index
INTEGER          :: KppVarId         ! KPP variable species index
INTEGER          :: KppFixId         ! KPP fixed species index
INTEGER          :: KppSpcID         ! KPP species index
INTEGER          :: PhotolID         ! Photolysis index
INTEGER          :: WetDepID         ! Wet deposition index

! Names
CHARACTER(LEN=31) :: Name            ! Short name
CHARACTER(LEN=80) :: FullName        ! Long name
CHARACTER(LEN=80) :: Formula         ! Chemical formula
INTEGER           :: NameHash        ! Integer hash for short name

! Logical switches
LOGICAL           :: Is_Advected     ! Is it advected?
LOGICAL           :: Is_Aero         ! Is it an aerosol species?
LOGICAL           :: Is_DryDep       ! Is it dry-deposited?
LOGICAL           :: Is_Gas          ! Is it a gas? If not, aerosol.
LOGICAL           :: Is_HygroGrowth ! Does it have hygroscopic growth?
LOGICAL           :: Is_ActiveChem   ! Is it an active chemical species?
LOGICAL           :: Is_FixedChem    ! Is it a fixed chemical species?
LOGICAL           :: Is_Kpp          ! Is it in the KPP mechanism?
LOGICAL           :: Is_Photolysis   ! Is it an photolysis species?
LOGICAL           :: Is_WetDep       ! Is it wet-deposited?
LOGICAL           :: Is_InRestart    ! Is it in the restart file?

! Molecular weights
REAL(fp)          :: MW_g            ! Species molecular weight [g/mol]
REAL(fp)          :: EmMW_g          ! Emitted molecular weight [g/mol]
REAL(fp)          :: MolecRatio      ! Mol carbon / mol species [1  ]

! Default background concentration
REAL(fp)          :: BackgroundVV    ! Background conc [v/v]

! Density and radius
REAL(fp)          :: Density          ! Density [kg/m3]
REAL(fp)          :: Radius           ! Radius [m]

```

```

! Henry's law parameters
REAL(f8)          :: Henry_K0          ! Liq./gas Henry const [M/atm ]
REAL(f8)          :: Henry_CR          ! d(ln K0) / d(1/T)   [K      ]
REAL(f8)          :: Henry_PKA        ! pKa for Henry const. correction

! Drydep parameters
LOGICAL           :: DD_AeroDryDep     ! Use AERO_SFCSII for drydep?
LOGICAL           :: DD_DustDryDep     ! Use DUST_SFCSII for drydep?
REAL(fp)          :: DD_DvzAerSnow    ! Vd for aerosols on snow [cm/s]
REAL(fp)          :: DD_DvzMinVal(2)  ! Min Vd for aerosols [cm/s]
REAL(fp)          :: DD_F0             ! F0 (reactivity) factor [1]
REAL(fp)          :: DD_KOA           ! KOA factor for POPG
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
!%% NOTE: We will eventually replace this with the common Henry's law
!%% parameters. But in order to replicate the prior behavior,
!%% we will need to supply the dry deposition code with the same
!%% HSTAR values that are currently set in INIT_DRYDEP. Therefore,
!%% add this field as a temporary placeholder for the Hstar quantity
!%% from drydep_mod.F. We will remove this later on. (bmy, 8/24/15)
!%%
REAL(fp)          :: DD_Hstar_Old      ! HSTAR value in drydep_mod [M/atm]
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

! Wetdep parameters, gas-phase species
LOGICAL           :: WD_LiqAndGas     ! Consider liquid and gas phases?
REAL(fp)          :: WD_ConvFacI2G    ! Conv. factor for ice/gas ratio
REAL(fp)          :: WD_RetFactor     ! Retention factor [1]

! Wetdep parameters, aerosol-phase species
LOGICAL           :: WD_Is_H2SO4     ! Flag to denote H2SO4 wetdep
LOGICAL           :: WD_Is_HNO3     ! Flag to denote HNO3 wetdep
LOGICAL           :: WD_Is_SO2      ! Flag to denote SO2 wetdep
LOGICAL           :: WD_CoarseAer    ! T=coarse aerosol; F=fine aerosol
REAL(fp)          :: WD_AerScavEff   ! Aerosol scavenging efficiency
REAL(fp)          :: WD_KcScaleFac(3) ! Temperature-dependent scale
! factors to multiply Kc rate
! (conv of condensate -> precip)
! in F_AEROSOL (wetscav_mod.F)
REAL(fp)          :: WD_RainoutEff(3) ! Temperature-dependent scale
! factors for rainout efficiency

! Microphysics parameters
LOGICAL           :: MP_SizeResAer   ! T=size-resolved aerosol (TOMAS)
LOGICAL           :: MP_SizeResNum   ! T=size-resolved aerosol number

! Tagged mercury parameters
LOGICAL           :: Is_Hg0          ! T=total or tagged Hg0 species

```

```

LOGICAL          :: Is_Hg2          ! T=total or tagged Hg2 species
LOGICAL          :: Is_HgP         ! T=total or tagged HgP species
INTEGER          :: Hg_Cat         ! Tagged Hg category number

END TYPE Species
!DEFINED PARAMETERS
!=====
! Missing value parameters
!=====
INTEGER, PARAMETER :: MISSING_INT = -999          ! Integer
REAL(fp), PARAMETER :: MISSING     = -999e+0_fp   ! Flexible precision
REAL(f8), PARAMETER :: MISSING_R8  = -999e+0_f8   ! 8-byte precision
REAL(fp), PARAMETER :: ZERO        = 0.0e+0_fp    ! Flexible precision
REAL(f8), PARAMETER :: ZERO_R8     = 0.0e+0_f8    ! 8-byte precision

REAL(fp), PARAMETER :: MISSING_MW  = -1.0_fp      ! Missing MW values

!=====
! Missing species concentration value if not in restart file and special
! background value not defined
!=====
REAL(fp), PARAMETER, PUBLIC :: MISSING_VV = 1.0e-20_fp ! Missing spc conc

```

REMARKS:

- (1) The emission molecular weight is the molecular weight of the emitted compound. This value is only different to MW_g if the emitted compound does not correspond to the transported species, e.g. if emissions are in kg C4H10 but the corresponding species is transported as mass Carbon.
- (2) MolecRatio is the ratio between # of species molecules per emitted molecule, e.g. 4 if emissions are kg C4H10 but model species are kg C.

REVISION HISTORY:

```

28 Feb 2014 - C. Keller - Initial version
22 Jul 2015 - R. Yantosca - Updated and cleaned up a bit
18 Aug 2015 - R. Yantosca - Added indices for drydep, wetdep, transport
18 Aug 2015 - R. Yantosca - Added missing value parameters
31 Aug 2015 - R. Yantosca - Add AdvectId
24 Sep 2015 - R. Yantosca - Make WD_RainoutEff a 3-element vector:
                        (1) T < 237K; (2) 237K < T < 258 K (3) T > 258K
24 Sep 2015 - R. Yantosca - Rename WD_ConvFactor to WD_ConvFacI2G
25 Sep 2015 - R. Yantosca - Rename WD_SizeResAer to MP_SizeResAer
25 Sep 2015 - R. Yantosca - Add MP_SizeResBin for microphysics size bins
30 Sep 2015 - R. Yantosca - Renamed DD_A_Density to Density
30 Sep 2015 - R. Yantosca - Renamed DD_A_Radius to Radius
30 Sep 2015 - R. Yantosca - Added WD_Is_HNO3 and WD_Is_SO2 fields to flag
                        special cases of HNO3 and SO2 wet deposition

```

01 Oct 2015 - R. Yantosca - Add field DD_DvzMinVal
 16 Oct 2015 - E. Lundgren - Add WD_Is_H2SO4 field to flag special case of
 H2SO4 wet deposition for microphysics
 22 Apr 2016 - R. Yantosca - Added Is_Hg0, Is_Hg2, Is_HgP species
 04 May 2016 - R. Yantosca - Added fast name lookup via hashing
 09 May 2016 - R. Yantosca - Add Is_Kpp, KppVarId, KppFixId to type Species
 21 Jun 2016 - M. Sulprizio- Add Is_PhotoLysis, Is_ActiveChem, and
 Is_FixedChem to type Species
 25 Jul 2016 - E. Lundgren - Add Is_InRestart to track which species are
 read in versus set to default background values
 02 Aug 2016 - M. Sulprizio- Remove function Get_KPPIndx, it is not used.
 KppSpCid is set in species_database_mod.F90 where
 KppVarId and KppFixId are set.
 04 Aug 2016 - R. Yantosca - Add parameter MISSING_MW = -1.0
 10 Aug 2016 - E. Lundgren - Add BackgroundVV field for default background
 and missing background concentration param [v/v]
 31 Oct 2017 - R. Yantosca - Move Str2Hash, To_UpperCase to species_mod.F90
 16 Nov 2017 - E. Lundgren - Add Is_HygroGrowth for cloud diagnostics
 27 Nov 2017 - E. Lundgren - Complete implementation for gas, aerosol, and
 photolysis species categories (id and count)

1.9.1 SpcData_Init

Routine SpcData_Init initializes species database object. This is an array where each element is of type Species. This object holds the metadata for each species (name, molecular weight, Henry's law constants, drydep info, wetdep info, etc.

INTERFACE:

```
SUBROUTINE SpcData_Init( am_I_Root, nSpecies, SpecDB, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! root CPU?
INTEGER,          INTENT(IN)    :: nSpecies     ! # of species
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(SpcPtr),     POINTER       :: SpecDB(:)    ! Species database
INTEGER,          INTENT(INOUT) :: RC          ! Return code
```

REVISION HISTORY:

20 Aug 2013 - C. Keller - Adapted from gignc_state_chm_mod.F90
 22 Jul 2015 - R. Yantosca - Cosmetic changes

1.9.2 Spc_GetIndx

Function Spc_GetIndx returns the index of a given species in the species data base object. You can search by the short name or the full name of the species.

INTERFACE:

```
FUNCTION Spc_GetIndx( Name, SpecDB ) RESULT( Indx )
```

USES:

```
USE CharPak_Mod, ONLY : Str2Hash14, To_UpperCase
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*) , INTENT(IN) :: Name          ! Species name
TYPE(SpcPtr) ,      POINTER    :: SpecDB(:)    ! Species Database object
```

RETURN VALUE:

```
INTEGER                :: Indx                ! Index of this species
```

REMARKS:

The input name field has will get copied to an internal string that is 14 characters long, for input into the Str2Hash function. 14 characters is about the longest species name for GEOS-Chem. We can modify this if need be.

REVISION HISTORY:

```
09 Oct 2012 - M. Long      - Initial version, based on gc_esmf_utils_mod.F90
22 Jul 2015 - R. Yantosca - Cosmetic changes
04 May 2016 - R. Yantosca - Now use hash comparison, it's faster
04 May 2016 - R. Yantosca - Renamed to Spc_GetIndx
05 May 2016 - R. Yantosca - The NAME argument is now of variable length
15 Jun 2016 - M. Sulprizio- Make species name uppercase before computing hash
01 Nov 2017 - R. Yantosca - Now use Str2Hash14 from charpak_mod.F90, which
                           computes a hash from an input string of 14 chars
```

```
EOP!EOP
```

```
-----
BOC
```

LOCAL VARIABLES:

```
INTEGER                :: N, Hash
CHARACTER(LEN=14)     :: Name14
```

```
!=====
! Spc_GetIndex begins here!
!=====
```

```
! Initialize the output value
```

```

Indx   = -1

! Make species name uppercase for hash algorithm
Name14 = To_UpperCase( Name )

! Compute the hash corresponding to the given species name
Hash   = Str2Hash14( Name14 )

! Loop over all entries in the Species Database object
DO N = 1, SIZE( SpecDB )

    ! Compare the hash we just created against the list of
    ! species name hashes stored in the species database
    IF( Hash == SpecDB(N)%Info%NameHash ) THEN
        Indx = SpecDB(N)%Info%ModelID
        EXIT
    ENDIF
ENDDO

END FUNCTION Spc_GetIndx
EOC
-----
                        GEOS-Chem Global Chemical Transport Model                !
-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\mbox{}\hrulefill\

\subsection [SpcData\_Cleanup] {SpcData\_Cleanup}

Routine SpcData\_Cleanup cleans up the passed species
collection object
\\
\\{\bf INTERFACE:}
\begin{verbatim}  SUBROUTINE SpcData_Cleanup( SpecDB )
INPUT/OUTPUT PARAMETERS:
    TYPE(SpcPtr), POINTER :: SpecDB(:) ! Species database object

REVISION HISTORY:
    20 Aug 2013 - C. Keller - Adapted from gigc_state_chm_mod.F90
    22 Jul 2015 - R. Yantosca - Cosmetic changes
    08 Oct 2015 - R. Yantosca - Bug fix, make sure the size of SpecDb
                                is zero before deallocating each element

```

```

REAL(fp),          OPTIONAL      :: DD_DvzMinVal(2)  ! Min Vd for aerosols
                                                         ! (cf GOCART) [cm/s]
REAL(fp),          OPTIONAL      :: DD_FO           ! Drydep reactivity [1]
REAL(fp),          OPTIONAL      :: DD_KOA          ! Drydep KOA parameter
!%%%%%%%%%%
!%% NOTE: We will eventually replace this with the common Henry's law
!%% parameters. But in order to replicate the prior behavior,
!%% we will need to supply the dry deposition code with the same
!%% HSTAR values that are currently set in INIT_DRYDEP. Therefore,
!%% add this field as a temporary placeholder for the Hstar quantity
!%% from drydep_mod.F. We will remove this later on. (bmy, 8/24/15)
!%%
REAL(fp),          OPTIONAL      :: DD_Hstar_Old     ! HSTAR, drydep [M/atm]
!%%%%%%%%%%
LOGICAL,           OPTIONAL      :: MP_SizeResAer    ! Size resolved aerosol?
LOGICAL,           OPTIONAL      :: MP_SizeResNum    ! Size resolved aer #?
REAL(fp),          OPTIONAL      :: WD_RetFactor     ! Wetdep retention factor
LOGICAL,           OPTIONAL      :: WD_LiqAndGas     ! Liquid and gas phases?
REAL(fp),          OPTIONAL      :: WD_ConvFacI2G    ! Factor for ice/gas ratio
REAL(fp),          OPTIONAL      :: WD_AerScavEff    ! Aerosol scavenging eff.
REAL(fp),          OPTIONAL      :: WD_KcScaleFac(3) ! Factor to multiply Kc
                                                         ! rate in F_AEROSOL

REAL(fp),          OPTIONAL      :: WD_RainoutEff(3) ! Rainout efficiency
LOGICAL,           OPTIONAL      :: WD_CoarseAer     ! Coarse aerosol?
LOGICAL,           OPTIONAL      :: Is_Advected      ! Is it advected?
LOGICAL,           OPTIONAL      :: Is_Drydep        ! Is it dry deposited?
LOGICAL,           OPTIONAL      :: Is_Gas           ! Gas (T) or aerosol (F)?
LOGICAL,           OPTIONAL      :: Is_HygroGrowth   ! Is hygroscopic growth?
LOGICAL,           OPTIONAL      :: Is_Photolysis    ! Is it photolysis spc?
LOGICAL,           OPTIONAL      :: Is_Wetdep        ! Is it wet deposited?
LOGICAL,           OPTIONAL      :: Is_InRestart     ! Is it in restart file?
LOGICAL,           OPTIONAL      :: Is_Hg0           ! Denotes Hg0 species
LOGICAL,           OPTIONAL      :: Is_Hg2           ! Denotes Hg2 species
LOGICAL,           OPTIONAL      :: Is_HgP           ! Denotes HgP species
INTEGER,           OPTIONAL      :: KppSpcId         ! KPP species ID
INTEGER,           OPTIONAL      :: KppVarId         ! KPP variable species ID
INTEGER,           OPTIONAL      :: KppFixId         ! KPP fixed species ID

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(Species),    POINTER        :: ThisSpc         ! Object w/ species info

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)    :: RC              ! Return code

```

REMARKS:

(1) If Fullname is not specified, it will use the value assigned to Name.

- (2) If EmMw_g is not specified, it will use the value assigned to MW_g.
- (3) If MolecRatio is not specified, it will be set to 1.
- (4) WD_Is_HNO3 is automatically set according to Name.
- (5) WD_Is_SO2 is automatically set according to Name.
- (6) All other fields, if not specified, will be set to MISSING
- (7) If Is_Gas = T, aerosol-specific fields will be set to MISSING.
(except for HNO3 and SO2, which wet scavenge as aerosols).
- (8) If Is_Gas = F, gas-phase specific-fields will be set to MISSING.
- (9) If Is_Advected = T, this will automatically update AdvectId.
- (10) If Is_Drydep = T, this will automatically update DryDepId.
- (11) If Is_Wetdep = T, this will automatically update WetDepId.

REVISION HISTORY:

- 20 Aug 2013 - C. Keller - Adapted from gignc_state_chm_mod.F90
- 22 Jul 2015 - R. Yantosca - Added RetFactor and drydep parameters
- 31 Aug 2015 - R. Yantosca - Now also compute AdvectId
- 04 Sep 2015 - R. Yantosca - Add arguments WD_RainoutEff, WD_CoarseAer,
and WD_SizeResAer
- 24 Sep 2015 - R. Yantosca - Added WD_KcScaleFac argument
- 22 Apr 2016 - R. Yantosca - Added Is_Hg0, Is_Hg2, Is_HgP
- 04 May 2016 - R. Yantosca - Now construct hash value from short name
- 15 Jun 2016 - M. Sulprizio- Make species name uppercase before computing hash
- 21 Jun 2016 - M. Sulprizio- Add optional argument Is_Photolysis. Also set
Is_ActiveChem and Is_Fixed Chem according to
KppVarId and KPPFixId.
- 06 Jul 2016 - R. Yantosca - Add more error checks to avoid uninit'd fields
- 18 Jul 2016 - M. Sulprizio- Remove special handling of ISOPN and MMN for
DryDepCount. Family tracers have been eliminated.
- 25 Jul 2016 - E. Lundgren - Add optional argument Is_InRestart
- 02 Aug 2016 - M. Sulprizio- Add optional argument KppSpcId
- 04 Aug 2016 - R. Yantosca - Now set missing molecular weights to -1,
which seems to avoid numerical roundoff
- 10 Aug 2016 - E. Lundgren - Add default background concentration argument
- 01 Nov 2017 - R. Yantosca - Now use Str2Hash14 from charpak_mod.F90, which
computes a hash from an input string of 14 chars
- 27 Nov 2017 - E. Lundgren - Add additional species categories

1.9.4 Spc_Print

Routine Spc_Create prints the fields of the species object.

INTERFACE:

```
SUBROUTINE Spc_Print( am_I_Root, ThisSpc, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)      :: am_I_Root      ! Are we on the root CPU?
TYPE(Species),    POINTER         :: ThisSpc        ! Object w/ species info
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT) :: RC          ! Return code
```

REMARKS:

Optional fields are not printed out if they are not defined (i.e. if they have a "missing data value" of -999).

REVISION HISTORY:

27 Jul 2015 - R. Yantosca - Initial version

1.9.5 Spc_GetNumSpecies

Routine Spc_GetNumSpecies returns the number of advected, dry-deposited, and wet-deposited species to an external routine.

INTERFACE:

```
SUBROUTINE Spc_GetNumSpecies( nAdvect,  nAero,   nDryDep, nGasSpc,  &
                              nHygGrth, nKppVar, nKppFix, nKppSpc,  &
                              nPhotol,  nWetDep,                                     &
                              nHg0Cats, nHg2Cats, nHgPCats          )
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: nAdvect    ! # of advected species
INTEGER, INTENT(OUT) :: nAero      ! # of aerosol species
INTEGER, INTENT(OUT) :: nDryDep    ! # of dry-deposited species
INTEGER, INTENT(OUT) :: nGasSpc    ! # of gas-phase species
INTEGER, INTENT(OUT) :: nHygGrth   ! # of species with hygroscopic growth
INTEGER, INTENT(OUT) :: nKppVar    ! # of variable KPP species
INTEGER, INTENT(OUT) :: nKppFix    ! # of fixed KPP species
INTEGER, INTENT(OUT) :: nKppSpc    ! # of species in KPP mechanism
INTEGER, INTENT(OUT) :: nPhotol    ! # of photolysis species
INTEGER, INTENT(OUT) :: nWetDep    ! # of wet-deposited species
INTEGER, INTENT(OUT) :: nHg0Cats   ! # of Hg0 categories
INTEGER, INTENT(OUT) :: nHg2Cats   ! # of Hg2 categories
INTEGER, INTENT(OUT) :: nHgPCats   ! # of HgP categories
```

REVISION HISTORY:

02 Sep 2015 - R. Yantosca - Initial version
 25 Apr 2016 - R. Yantosca - Also return the # of Hg0, Hg2, HgP categories
 18 May 2016 - R. Yantosca - Also return the # of KPP chemical species
 27 Nov 2017 - E. Lundgren - Add additional species categories for aerosol,
 gas, hygroscopic growth, active KPP, fixed KPP,
 and photolysis species

1.10 Fortran: Module Interface CMN_O3_mod.F

Common blocks for anthro emissions (via SMVGEAR!)

INTERFACE:

MODULE CMN_O3_MOD

USES:USE CMN_SIZE_MOD, ONLY : IIPAR, JJPARG, LLPARG
USE PRECISION_MODIMPLICIT NONE
PUBLIC**PUBLIC DATA MEMBERS:**

```

!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
!%% NOTE: THESE ARE USED FOR DIAGNOSTICS WHICH ARE OBSOLETE %%
!%% WE WILL EVENTUALLY REMOVE THESE IN THE NEAR FUTURE. %%
!%% (bmy, 5/22/15) %%
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

! SAVEOH = array to save OH fields
! SAVEHO2 = array to save HO2 fields (rvm, bmy, 2/27/02)
! SAVEEOH = array to save EOH fields (evf, 5/21/13)
! SAVEGLYX= array to save GLYX fields (eam, 8/25/14 )
! SAVEOA = array for total organic aerosol (trc 6 of ND42, eam,7/10/14)
!REAL(fp), ALLOCATABLE :: SAVEOH(:,:,:)
!REAL(fp), ALLOCATABLE :: SAVEHO2(:,:,:)
!REAL(fp), ALLOCATABLE :: SAVEO1D(:,:,:)
!REAL(fp), ALLOCATABLE :: SAVEO3P(:,:,:)
REAL(fp), ALLOCATABLE :: SAVEEOH(:,:,:)
REAL(fp), ALLOCATABLE :: SAVEGLYX(:,:,:)
REAL(fp), ALLOCATABLE :: SAVEOA(:,:,:)

```

REMARKS:

NOTE: Now NEMPARA = max no. of anthropogenic emissions
NEMPARB = max no. of biogenic emissions

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% NOTE: THIS MODULE IS NOW DEPRECATED. IT WILL BE REMOVED WHEN %%
%% THE GEOS-CHEM EMISSIONS MODULE IS UPDATED (SOMETIME SOON) %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

REVISION HISTORY:

23 Aug 2011 - M. Long - Converted to Module from Header file
29 Mar 2013 - M. Payer - Removed FRAC03, FRACNO, FRACNO2, SAVENO, SAVENO2,
and SAVENO3. They are no longer needed because

03, NO, NO2, and NO3 are now tracers.

13 Aug 2013 - M. Sulprizio- Increase last dimension of EMISS_BVOC to include sesquiterpenes and add EMISTNAP for SOA + semivolatile POA simulation (H. Pye)

21 Feb 2014 - M. Sulprizio- Added SAVE01D and SAVE03P for UCX (S.D. Eastham)

03 Dec 2014 - M. Yannetti - Added PRECISION_MOD

22 May 2015 - R. Yantosca - Remove arrays made obsolete by HEMCO

03 Jan 2017 - M. Sulprizio- Added SAVEEOH (E. Fischer)

1.10.1 Init_Cmn_O3

Subroutine INIT_CMN_O3 allocates all module arrays.

INTERFACE:

```
SUBROUTINE Init_CMN_O3( am_I_Root, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

19 Nov 2012 - R. Yantosca - Added ProTeX headers

01 Feb 2013 - R. Yantosca - Now allocate EMISS_BVOC to 19 emission species

29 Mar 2013 - M. Payer - Removed FRACO3, FRACNO, FRACNO2, SAVENO, SAVENO2, and SAVENO3. They are no longer needed because 03, NO, NO2, and NO3 are now tracers.

13 Aug 2013 - M. Sulprizio- Increase last dimension of EMISS_BVOC to include sesquiterpenes and add EMISTNAP for SOA + semivolatile POA simulation (H. Pye)

22 May 2015 - R. Yantosca - Remove arrays made obsolete by HEMCO

1.10.2 Cleanup_Cmn_O3

Subroutine CLEANUP_CMN_O3 allocates all module arrays.

INTERFACE:

```
SUBROUTINE Cleanup_CMN_O3( am_I_Root, RC )
```


USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

```
19 Nov 2012 - R. Yantosca - Added ProTeX headers
29 Mar 2013 - M. Payer    - Removed FRAC03, FRACNO, FRACNO2, SAVENO, SAVENO2,
                           and SAVENO3. They are no longer needed because
                           03, NO, NO2, and NO3 are now tracers.
22 May 2015 - R. Yantosca - Remove arrays made obsolete by HEMCO
```

1.11 Fortran: Module Interface inquireMod

Module inquireMod contains functions to find free and unopened logical file units (LUNs) for Fortran I/O. **INTERFACE:**

```
MODULE inquireMod
```

USES:

```
#if defined( ESMF_ )
! We only need to refer to these modules if we are connecting
! to the GEOS-5 GCM via the ESMF/MAPL framework (bmy, 8/3/12)
USE ESMF
USE MAPL_Mod
#endif
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: findFreeLUN
PUBLIC  :: I_Am_UnOPENed
```

REVISION HISTORY:

```
14 Jun 2012 - E. Nielsen - Initial version
03 Aug 2012 - R. Yantosca - Block off ESMF-specific code with #ifdefs
03 Aug 2012 - R. Yantosca - Cosmetic changes
```

1.11.1 findFreeLUN

Inquire for an existing, but unopened, logical unit number

INTERFACE:

```
FUNCTION findFreeLUN( b ) RESULT( lun )
```

USES:

```
IMPLICIT NONE
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN), OPTIONAL :: b ! Not really used here
```

RETURN VALUE:

```
INTEGER :: lun
```

REVISION HISTORY:

```
14 Jun 2012 - E. Nielsen - Initial version
03 Aug 2012 - R. Yantosca - Block off ESMF-specific code with #ifdefs
03 Aug 2012 - R. Yantosca - Cosmetic changes
06 Aug 2012 - R. Yantosca - Now make LUN range 11..199
```

1.11.2 I_Am_UnOPENed

Inquire as to the availability of a given logical unit

INTERFACE:

```
FUNCTION I_Am_UnOPENed( n ) RESULT( TorF )
```

USES:

```
IMPLICIT NONE
```

INPUT PARAMETERS:

```
INTEGER :: n ! Logical unit # to test
```

RETURN VALUE:

```
LOGICAL :: TorF ! .TRUE. means the file is unopened
```

REVISION HISTORY:

```
14 Jun 2012 - E. Nielsen - Initial version
03 Aug 2012 - R. Yantosca - Block off ESMF-specific code with #ifdefs
03 Aug 2012 - R. Yantosca - Cosmetic changes
```

1.12 Fortran: Module Interface CMN_DIAG_mod.F

Module CMN_DIAG_mod contains size parameters and global variables for the GEOS-Chem diagnostic arrays. This is mostly historical baggage.

INTERFACE:

```
MODULE CMN_DIAG_MOD
```

USES:

```
USE CMN_SIZE_MOD
USE CMN_FJX_MOD, ONLY : W_
USE PRECISION_MOD, ONLY : fpp => fp      ! For GEOS-Chem Precision (fp)

IMPLICIT NONE
PUBLIC
```

DEFINED PARAMETERS:

```
!#####
!##### NOTE: We need to keep these variables! #####
!#####
INTEGER, PARAMETER :: PD40=4
INTEGER             :: ND40
INTEGER             :: ND48, ND49, ND50, ND51, ND63 ! timeseries

!#####
!##### NOTE: These variables can be blocked out! #####
!#####
#if defined( BPCH_DIAG ) || defined( BPCH_TIMESER )
INTEGER, PARAMETER :: PD05=20
INTEGER, PARAMETER :: PD06=NDSTBIN
! +3 for SEAC4RS SOA tracers (jaf, 6/25/13)
INTEGER, PARAMETER :: PD07=15
INTEGER, PARAMETER :: PD08=4
INTEGER, PARAMETER :: PD09=6
INTEGER, PARAMETER :: PD10=20
INTEGER, PARAMETER :: PD11=5
INTEGER, PARAMETER :: PD16=2
INTEGER, PARAMETER :: PD17=8
INTEGER, PARAMETER :: PD18=8
INTEGER, PARAMETER :: PD19=3
INTEGER, PARAMETER :: PD21=8+(NRHAER+NDUST)*3+(NRHAER*2)+
&                          (NSTRATAER*3)
INTEGER, PARAMETER :: PD22=76
INTEGER, PARAMETER :: PD43=4
INTEGER, PARAMETER :: PD52=6
INTEGER, PARAMETER :: PD55=3
! Potential temperature diagnostic (hotp 7/31/09)
```

```

    INTEGER, PARAMETER :: PD57=1
    INTEGER, PARAMETER :: PD58=15
#if defined( TOMAS )
    ! Special settings for TOMAS aerosol microphysics (win, bmy, 1/22/10)
    INTEGER, PARAMETER :: PD59=TOMASBIN*TOMASSPEC
    INTEGER, PARAMETER :: PD60=TOMASBIN*TOMASSPEC
    INTEGER, PARAMETER :: PD61=2
#else
    ! Normal settings for non-TOMAS simulations
    INTEGER, PARAMETER :: PD59=0
    INTEGER, PARAMETER :: PD60=1
    INTEGER, PARAMETER :: PD61=0
#endif
    INTEGER          :: PD64
    INTEGER          :: PD65
    INTEGER, PARAMETER :: PD66=7
    INTEGER, PARAMETER :: PD67=23 ! (Lin, 31/03/09)
    INTEGER, PARAMETER :: PD68=8
    INTEGER, PARAMETER :: PD69=1

    !number of rad flux and optics output types
    !8 flux and 3*3=9 optics
    INTEGER, PARAMETER :: PD72R=17
    !total number of possible rad outputs (types*specs)
    !there are 11 possible flux output 'species' but
    !only 8 possible optics output 'species'
    !for simplicity we take the largest and put up with
    !some redundancy (should be 88+72=160)
    INTEGER, PARAMETER :: PD72=187 ! Radiation (Ridley 10/2012)

    ! (eam, 2015): ISORROPIA pH, H+, & aerosol water
    INTEGER, PARAMETER :: PD73=3
#endif

    !=====
    ! Variables for printing out selected tracers in diagnostic output
    !=====
#if defined( RRTMG )
    INTEGER, PARAMETER :: MAX_DIAG    = 187
#else
    ! SDE 2013-11-17: Increased to 80 for UCX
    INTEGER, PARAMETER :: MAX_DIAG    = 80
#endif

    INTEGER, PARAMETER :: MAX_TRACER = 600      ! Large placeholder value

PUBLIC DATA MEMBERS:

#if defined( BPCH_DIAG ) || defined( BPCH_TIMESER )

```

```
=====
! Diagnostic counters & time variables
=====
INTEGER :: KDA48

=====
! Variables for the number of levels in multi-level diagnostics
! Removed LD03 -- this is now in diag03_mod.f (bmy, 1/21/05)
! Added LD09 (bmy, 6/27/05)
! Added LD54 (phs, 9/22/06)
! Added LD10 (phs, 9/18/07)
! Added LD31 (bmy, 5/8/07)
! Added LD52 (jaegle, 02/26/09)
! Added LD59, LD60, LD61 (bmy, 1/22/10)
! Added LD57 (hotp 7/31/09)
=====
INTEGER :: LD12, LD13, LD14, LD15, LD16, LD17, LD18, LD21, LD22
INTEGER :: LD24, LD25, LD26, LD37, LD38, LD39, LD43, LD45, LD47
INTEGER :: LD54, LD64, LD65, LD66, LD68, LD01, LD02, LD05, LD07
INTEGER :: LD09, LD10, LD31, LD52, LD19, LD57, LD58, LD59, LD60
INTEGER :: LD61, LD62, LD73

=====
! NDxx diagnostic flags
=====
INTEGER :: ND01, ND02, ND05, ND06, ND07, ND08, ND09, ND10, ND11
INTEGER :: ND12, ND13, ND14, ND15, ND16, ND17, ND18, ND19, ND20
INTEGER :: ND21, ND22, ND23, ND24, ND25, ND26, ND27, ND28, ND29
INTEGER :: ND30, ND31, ND32, ND33, ND34, ND35, ND36, ND37, ND38
INTEGER :: ND39, ND43, ND44, ND45, ND46, ND47
INTEGER :: ND52, ND54, ND55, ND57, ND58, ND59
INTEGER :: ND60, ND61, ND62, ND64, ND65, ND66, ND67, ND68
INTEGER :: ND69, ND70, ND71, ND72, ND73, ND74, ND75

#endif

=====
! Variables for printing out selected tracers in diagnostic output
=====
INTEGER :: TINDEX(MAX_DIAG,MAX_TRACER)
INTEGER :: TCOUNT(MAX_DIAG)
INTEGER :: TMAX(MAX_DIAG)

#if defined( BPCH_DIAG ) || defined( BPCH_TIMESER )
=====
! OH, J-Value, and 2-PM diagnostic arrays (bmy, 9/25/98)
! Move this here for now (bmy, 7/20/04)
! Removed HR1_NO and HR2_NO (mpayer, 11/8/13)
=====
```

```

      REAL(fpp)   :: HR1_JV, HR2_JV
      REAL(fpp)   :: HR1_OH, HR2_OH, HR1_OTH, HR2_OTH
#endif

```

REMARKS:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% NOTE: THIS MODULE IS DEPRECATED.  MANY OF THE QUANTITIES STORED %%
%% in CMN_DIAG_MOD ARE NOW INCLUDED IN THE INPUT OPTIONS OBJECT, %%
%% BASED ON THE DERIVED TYPE IN Headers/input_opt_mod.F90. %%
%% (bmy, 11/19/12) %%
%% %%
%% ALSO NOTE: IN GEOS-CHEM v11-01, WE WILL BE RETIRING THE BPCH %%
%% DIAGNOSTICS.  MANY OF THESE PARAMETERS CAN THEN BE REMOVED. %%
%% (bmy, 6/24/16) %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

REVISION HISTORY:

- (1) Changed RCS ID tag comment character from "C" to "!" to allow freeform compilation. Also added & continuation characters in column 73 to allow header files to be included in F90 freeform files. Also converted PARAMETER statements to F90 syntax. (bmy, 6/25/02)
- (2) Add LD05 for sulfate prod/loss (rjp, bdf, bmy, 9/20/02)
- (3) Removed obsolete variables NTAU0, IDAY0, JDATE0, JYEAR0, KDACC, KDADYN, KDACONV, KDASRCE, KDACHEM, KDA3FLDS, KDA6FLDS, KDI6FLDS, KDKZZFLDS (bmy, 2/11/03)
- (4) Fix for LINUX - remove & from column 73 (bmy, 6/27/03)
- (5) Added LD03 for Kr85 Prod/loss diagnostic (bmy, 8/20/03)
- (6) Removed obsolete arrays (bmy, 1/21/05)
- (7) Rename MAXDIAG to MAX_DIAG and MAXTRACER to MAX_TRACER in order to avoid name conflicts with "gamap_mod.f" (bmy, 5/3/05)
- (8) Remove reference to TRCOFFSET (bmy, 5/16/06)
- (9) Added multi level LD54 to common CDIAG1 (phs, 9/22/06)
- (10) Added multi level LD10 to common CDIAG1. Set PD10 to 20. Set PD66 to 6. (phs, 9/18/07)
- (11) Added LD52 to common CDIAG1 (jaegle, 02/26/09)
- (12) Add GLYX, MGLY, SOAG, SOAM, and a few other tracers to AD17, AD18 for archiving rainout and washout fraction. (tmf, 1/7/09)
- (13) Add GLYX, MGLY J value archive. (tmf, 1/7/09)
- (14) Add GLYX, MGLY, SOAG, SOAM, and a few other tracers to AD37, AD38, AD39 for archiving rainout and washout flux. (tmf, 1/7/09)
- (15) Add GLYX, MGLY, GLYC, 6 PANs, SOAG, SOAM dry dep, PD44 = 41. (tmf, 1/7/09)
- (16) Add biogenic C2H4 emission, PD46 = 6. (tmf, 1/20/09)
- (17) Add one met field to ND67 (EFLUX). (ccc, 5/14/09)
- (18) Add declarations for PD58 and PD60, LD19, LD58. (kjlw,8/18/09)
- (19) Redimension PD59, PD60, PD61 for TOMAS microphysics. Added LD59, LD60,

LD61 to common block. Reset MAX_TRACER to NNPAR+1 for TOMAS.
(win, bmy, 1/22/10)
(20) Add LD57 and PD57 (potential temperature) (hotp, 3/15/10)
03 Aug 2010 - R. Yantosca - Added ProTeX headers
03 Aug 2010 - P. Kasibhatla & R. Yantosca - Now set MAX_TRACER to NNPAR+6
to match ND09 diagnostic
09 Mar 2011 - R. Yantosca - Updated MAX_TRACER for APM (G. Luo)
03 Aug 2011 - M. Long - Converted from Header file to Module
08 Nov 2013 - M. Sulprizio- Remove HR1_NO and HR2_NO. They are no longer
needed for ND43 because NO, NO2, and NO3 are now
tracers.
02 Dec 2014 - M. Yannetti - Added PRECISION_MOD
15 Dec 2014 - M. Sulprizio- Moved radiation diagnostic from ND72 to ND72 to
avoid conflicts with hourly max ppbv diagnostic.
23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer
compatible with the FlexChem implementation
21 Jul 2016 - R. Yantosca - Remove instances of NNPAR & obsolete parameters

1.12.1 Init_Cmn_Diag

Subroutine INIT_CMN_DIAG initializes quantities based on the grid-independent size parameters.

INTERFACE:

```
SUBROUTINE Init_CMN_DIAG( am_I_Root, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

REVISION HISTORY:

```
19 Nov 2012 - R. Yantosca - Added ProTeX headers
```

1.13 Fortran: Module Interface *errcode_mod.F90*

Module ERRCODE_MOD contains the error codes (i.e. that report success or failure) returned by GEOS-Chem routines.

INTERFACE:

```
MODULE ErrCode_Mod
```

USES:

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: GC_Error
PUBLIC :: GC_Warning
PUBLIC :: GC_CheckVar
```

DEFINED PARAMETERS:

```
INTEGER, PUBLIC, PARAMETER :: GC_SUCCESS = 0 ! Routine returns success
INTEGER, PUBLIC, PARAMETER :: GC_FAILURE = -1 ! Routine returns failure
```

REMARKS:

The error codes are returned by routines at various levels of GEOS-Chem.

REVISION HISTORY:

```
19 Oct 2012 - R. Yantosca - Initial version
16 Aug 2016 - M. Sulprizio- Rename from gigc_errcode_mod.F90 to
                    errcode_mod.F90. The "gigc" nomenclature is
                    no longer used.
23 Jun 2017 - R. Yantosca - Moved subroutine GC_Error here
27 Jun 2017 - R. Yantosca - Added routine GC_CheckVar
20 Dec 2017 - R. Yantosca - Added routine GC_Warning
```

1.13.1 GC_Error

Subroutine GC_Error prints an error message and sets RC to GC_FAILURE. Note that this routine does not stop a run, but it will cause a stop at a higher level if you add a catch for RC /= GC_SUCCESS.

INTERFACE:

```
SUBROUTINE GC_Error( ErrMsg, RC, ThisLoc, Instr )
```

USES:

```
USE Charpak_Mod, ONLY : WordWrapPrint
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN )           :: ErrMsg ! Message to display
CHARACTER(LEN=*), INTENT(IN ), OPTIONAL :: ThisLoc ! Location of error
CHARACTER(LEN=*), INTENT(IN ), OPTIONAL :: Instr  ! Other instructions
```

INPUT/OUTPUT PARAMETERS:


```
INTEGER,          INTENT(INOUT)          :: RC          ! Error code
```

REVISION HISTORY:

```
13 Aug 2015 - E. Lundgren - Initial version, based on C. Keller's HCO_ERROR
16 Aug 2016 - M. Sulprizio- Rename from GIGC_ERROR to GC_ERROR
23 Jun 2017 - R. Yantosca - Now moved from error_mod.F to errcode_mod.F90
28 Aug 2017 - R. Yantosca - Now flush the error msg to stdout/log file
28 Nov 2017 - R. Yantosca - Add separator lines to make msgs more visible
22 Jan 2018 - R. Yantosca - Add optional instructions text
```

1.13.2 GC_Warning

Subroutine GC_Warning prints an warning (i.e. non-fatal error message) and sets RC to GC_SUCCESS.

INTERFACE:

```
SUBROUTINE GC_Warning( WarnMsg, RC, ThisLoc, Instr )
```

USES:

```
USE Charpak_Mod, ONLY : WordWrapPrint
!
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN  )          :: WarnMsg ! Message to display
CHARACTER(LEN=*), INTENT(IN  ), OPTIONAL :: ThisLoc ! Location of warning
CHARACTER(LEN=*), INTENT(IN  ), OPTIONAL :: Instr  ! Other instructions
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT)          :: RC
```

REVISION HISTORY:

```
13 Aug 2015 - E. Lundgren - Initial version, based on C. Keller's HCO_ERROR
16 Aug 2016 - M. Sulprizio- Rename from GIGC_ERROR to GC_ERROR
23 Jun 2017 - R. Yantosca - Now moved from error_mod.F to errcode_mod.F90
28 Aug 2017 - R. Yantosca - Now flush the error msg to stdout/log file
08 Nov 2017 - R. Yantosca - Add separator lines to make msgs more visible
22 Jan 2018 - R. Yantosca - Add optional instructions text
```

1.13.3 GC_CheckVar

Wrapper routine for GC_Error. Prints an error message if there is an allocation or registration error. This is intended to be called from the state initialization method (e.g. Init_State_Met).

INTERFACE:

```
SUBROUTINE GC_CheckVar( Variable, Operation, RC )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)  :: Variable  ! Name of variable to check
INTEGER,          INTENT(IN)  :: Operation ! 0=Allocate
                                                ! 1=Register
                                                ! 2=Deallocate
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT) :: RC          ! Success or failure
```

REMARKS:

You also need to add an
 IF (RC /= GC_SUCCESS) RETURN
 from the calling routine for proper error handling.

REVISION HISTORY:

27 Jun 2017 - R. Yantosca - Initial version

1.14 Fortran: Module Interface precision_mod.F

Module PRECISION_MOD is used to change the precision of many variables throughout GEOS-Chem at compile-time.

INTERFACE:

```
MODULE PRECISION_MOD
```

USES:

```
IMPLICIT NONE
PRIVATE
```

DEFINED PARAMETERS:

```
!=====
! Set parameters for floating precision
! FP will be set to either 4-byte or 8-byte precision at compile
! time. Most variables can now declared with REAL(fp).
!=====
#if defined( USE_REAL8 )

! Use 8-byte floating point precision when asked.
INTEGER, PARAMETER, PUBLIC :: fp = KIND( REAL( 0.0, 8 ) )

#else
```

```

! Use 4-byte floating point by default.
INTEGER, PARAMETER, PUBLIC :: fp = KIND( REAL( 0.0, 4 ) )

#endif

!=====
! Set parameters for fixed precision
! Not all variables can be converted into the flexible precision.
! Some may have to be still declared as either 4-byte or 8-byte
! floating point. Use these parameters for such variables.
!=====

! KIND parameter for 4-byte precision
INTEGER, PARAMETER, PUBLIC :: f4 = KIND( REAL( 0.0, 4 ) )

! KIND parameter for 8-byte precision
INTEGER, PARAMETER, PUBLIC :: f8 = KIND( REAL( 0.0, 8 ) )

```

REMARKS:

This module is designed to help avoid hard-coding precision.

REVISION HISTORY:

```

(1 ) Created. (myannetti, 11/04/14)
23 Nov 2016 - R. Yantosca - Now rewrite KIND definitions to prevent 4-byte
and 8-byte variables from being elevated
when using -r8 (or equivalent flags)

```

1.15 Fortran: Module Interface diaglist_mod.F90

Module diaglist_mod.F90 contains the derived types and subroutines used for reading and storing user-configured diagnostic information from the history configuration file, specifically names and information derived from the names. The diagnostics list is used to allocate diagnostics stored in container State.Diag and to declare exports in GCHP. It does not store collection information. A module-level collection list containing names all collections that are declared in HISTORY.rc with names not commented out is also in this module. This is used to prevent adding diagnostics to the diagnostic list that are in collections not turned on in HISTORY.rc, thereby preventing their analogous State_Diag array initialization in GEOS-Chem. **INTERFACE:**

```
MODULE DiagList_Mod
```

USES:

```
USE ErrCode_Mod
USE Precision_Mod
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```

PUBLIC  :: Init_DiagList
PUBLIC  :: Print_DiagList
PUBLIC  :: Check_DiagList
PUBLIC  :: Cleanup_DiagList
PUBLIC  :: Search_CollList
!PRIVATE MEMBER FUNCTIONS
PRIVATE :: Init_DiagItem
PRIVATE :: InsertBeginning_DiagList
PRIVATE :: Search_DiagList

! Private collection list in Init_DiagList
! Make public for more widespread use?
PRIVATE :: Init_CollItem
PRIVATE :: Set_CollItem
PRIVATE :: InsertBeginning_CollList
PRIVATE :: Print_CollList
PRIVATE :: Cleanup_CollList
!PUBLIC DATA TYPES:
!=====
! Derived type for Diagnostics List
!=====
TYPE, PUBLIC :: DgnList
    TYPE(DgnItem), POINTER  :: head
END TYPE DgnList

!=====
! Derived type for Diagnostics Item (unique name in HISTORY.rc)
!=====
TYPE, PUBLIC :: DgnItem
    CHARACTER(LEN=63)      :: name
    CHARACTER(LEN=63)      :: state
    CHARACTER(LEN=63)      :: metadataID
    CHARACTER(LEN=63)      :: registryID
    LOGICAL                 :: isWildcard
    CHARACTER(LEN=7)        :: wildcard
    LOGICAL                 :: isTagged
    CHARACTER(LEN=63)      :: tag
    ! could also add a list of collections this diagnostic is part of
    TYPE(DgnItem), POINTER :: next
END TYPE DgnItem

!=====
! Configurable Settings Used for Diagnostic Names at Run-time
!=====
CHARACTER(LEN=5), PUBLIC  :: RadWL(3)    ! Wavelengths configured in rad menu
LOGICAL,                PUBLIC  :: IsFullChem ! Is this a fullchem simulation?

```

```

=====
! Derived type for Collections List
=====
TYPE, PUBLIC :: CollList
    TYPE(ColItem), POINTER :: head
END TYPE CollList

!=====
! Derived type for Collections Item (uncommented in HISTORY.rc)
!=====
TYPE, PUBLIC :: ColItem
    CHARACTER(LEN=63)      :: cname
    TYPE(ColItem), POINTER :: next
END TYPE ColItem

```

PUBLIC DATA MEMBERS:

```

TYPE(CollList), PUBLIC :: CollList      ! Collection list object

```

REVISION HISTORY:

```

22 Sep 2017 - E. Lundgren - Initial version
01 Nov 2017 - R. Yantosca - Moved ReadOneLine, CleanText to charpak_mod.F90
25 Jan 2018 - E. Lundgren - Add collection items and linked list
01 Feb 2018 - E. Lundgren - Rename module: diagnostics_mod -> diaglist_mod
                        and make collection name list publicly available

```

1.15.1 Init_DiagList

Reads the HISTORY.rc and input.geos input files to get determine which GEOS-Chem diagnostics have been requested. Then it uses this information to initialize the master list of diagnostics, aka, the DiagList object.

NOTE: This routine has to be called before any of the GEOS-Chem objects Input_Opt, State_Chm, State_Met, and State_Diag are created. When using GCHP, we must create the ESMF/MAPL export objects for the diagnostics in the Set_Services routine. Set_Services is called before GEOS-Chem is initialized.

INTERFACE:

```

SUBROUTINE Init_DiagList ( am_I_Root, historyConfigFile, DiagList, RC )

```

USES:

```

USE Charpak_Mod
USE InquireMod,      ONLY: findFreeLun

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root
CHARACTER(LEN=*), INTENT(IN)    :: historyConfigFile
!INPUT AND OUTPUT PARAMETERS:
TYPE(DgnList),   INTENT(INOUT)  :: DiagList

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC

```

REVISION HISTORY:

```

22 Sep 2017 - E. Lundgren - initial version
28 Nov 2017 - C. J. Lee   - Allow state MET variables to have underscores
29 Dec 2017 - C. Keller   - Look for GEOSCHEMCHEM instead of GIGCchem when
                           on NCCS Discover.
22 Jan 2018 - E. Lundgren - Allow use of AOD wavelengths in diagnostic names
26 Jan 2018 - E. Lundgren - Read collection names and skip collection diags
                           where HISTORY.rc declared col name commented out

```

1.15.2 Init_DiagItem

Initializes a DiagItem object, which contains information about a single GEOS-Chem diagnostic. Several DiagItem objects will be linked together in the master diagnostics list (DiagList).

INTERFACE:

```

SUBROUTINE Init_DiagItem ( am_I_Root, NewDiagItem, name,      state,      &
                           metadataID, registryID, isWildcard, wildcard, &
                           isTagged,  tag,              RC )

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root
CHARACTER(LEN=*), OPTIONAL      :: name
CHARACTER(LEN=*), OPTIONAL      :: state
CHARACTER(LEN=*), OPTIONAL      :: metadataID
CHARACTER(LEN=*), OPTIONAL      :: registryID
LOGICAL,          OPTIONAL      :: isWildcard
CHARACTER(LEN=*), OPTIONAL      :: wildcard
LOGICAL,          OPTIONAL      :: isTagged
CHARACTER(LEN=*), OPTIONAL      :: tag

```

OUTPUT PARAMETERS:

```

TYPE(DgnItem), POINTER :: NewDiagItem
INTEGER,          OPTIONAL :: RC

```

REVISION HISTORY:

```

21 Sep 2017 - E. Lundgren - Initial version

```

1.15.3 Init_ColItem

Initializes a ColItem object, which contains information about a single GEOS-Chem collection. Several ColItem objects will be linked together in the master collections list (ColList).

INTERFACE:

```
SUBROUTINE Init_ColItem ( am_I_Root, NewCollItem, cname, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root
TYPE(ColItem),   POINTER     :: NewCollItem
CHARACTER(LEN=*)  :: cname
```

OUTPUT PARAMETERS:

```
INTEGER,          OPTIONAL    :: RC
```

REVISION HISTORY:

25 Jan 2018 - E. Lundgren - Initial version

1.15.4 Set_ColItem

Sets a ColItem object, which contains information about a single GEOS-Chem collection. Several ColItem objects will be linked together in the master collections list (ColList).

INTERFACE:

```
SUBROUTINE Set_ColItem ( am_I_Root, Collname, CollList, Found, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root
CHARACTER(LEN=*)  :: CollName
TYPE(CollList)   :: CollList
```

OUTPUT PARAMETERS:

```
LOGICAL,          OPTIONAL    :: Found
INTEGER,          OPTIONAL    :: RC
```

REVISION HISTORY:

25 Jan 2018 - E. Lundgren - Initial version

1.15.5 InsertBeginning_DiagList

Inserts a new node at the beginning of the DiagList linked list object.

INTERFACE:

```
SUBROUTINE InsertBeginning_DiagList ( am_I_Root, DiagItem, DiagList, RC )
```

USES:

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root  
TYPE(DgnItem),   POINTER        :: DiagItem
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnList),   INTENT(INOUT)  :: DiagList
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC
```

REVISION HISTORY:

22 Sep 2017 - E. Lundgren - initial version

1.15.6 InsertBeginning_CollList

Inserts a new node at the beginning of the CollList linked list object.

INTERFACE:

```
SUBROUTINE InsertBeginning_CollList ( am_I_Root, CollItem, CollList, RC )
```

USES:

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root  
TYPE(CollItem),   POINTER        :: CollItem
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(CollList),   INTENT(INOUT)  :: CollList
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC
```

REVISION HISTORY:

25 Jan 2018 - E. Lundgren - initial version

1.15.7 Search_DiagList

Searches for a given diagnostic name within the DiagList diagnostic list object.

INTERFACE:

```
SUBROUTINE Search_DiagList ( am_I_Root, DiagList, name, found, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  
TYPE(DgnList),   INTENT(IN)  :: DiagList  
CHARACTER(LEN=*), INTENT(IN)  :: name
```

OUTPUT PARAMETERS:

```
LOGICAL,          INTENT(OUT) :: found  
INTEGER,         INTENT(OUT) :: RC
```

REVISION HISTORY:

22 Sep 2017 - E. Lundgren - Initial version

1.15.8 Search_CollList

Searches for a given collection name within the CollList collection list object.

INTERFACE:

```
SUBROUTINE Search_CollList ( am_I_Root, CollList, name, found, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  
TYPE(CollList),  INTENT(IN)  :: CollList  
CHARACTER(LEN=*), INTENT(IN)  :: name
```

OUTPUT PARAMETERS:

```
LOGICAL,          INTENT(OUT) :: found  
INTEGER,         INTENT(OUT) :: RC
```

REVISION HISTORY:

25 Jan 2018 - E. Lundgren - Initial version

1.15.9 Check_DiagList

Returns TRUE if a diagnostic name (or just a substring of a diagnostic name) is found in the DiagList diagnostic list object.

INTERFACE:

```
SUBROUTINE Check_DiagList( am_I_Root, DiagList, substr, found, RC )
```

USES:

```
USE Charpak_Mod, ONLY : To_UpperCase
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(DgnList),   INTENT(IN)  :: DiagList     ! Diagnostic list object
CHARACTER(LEN=*), INTENT(IN)  :: substr      ! Substring
```

OUTPUT PARAMETERS:

```
LOGICAL,          INTENT(OUT) :: found        ! Was a match found (T/F)?
INTEGER,          INTENT(OUT) :: RC          ! Success or failure?
```

REVISION HISTORY:

```
22 Sep 2017 - E. Lundgren - Initial version
01 Nov 2017 - R. Yantosca - Now use To_UpperCase from charpak_mod.F90
```

1.15.10 Print_DiagList

Subroutine Print_DiagList prints information for all DiagItem members in a DiagList linked list.

INTERFACE:

```
SUBROUTINE Print_DiagList( am_I_Root, DiagList, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! root CPU?
TYPE(DgnList),   INTENT(IN)  :: DiagList
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT) :: RC          ! Success?
```

REVISION HISTORY:

```
22 Sep 2017 - E. Lundgren - Initial version
```

1.15.11 Print_ColList

Subroutine Print_ColList prints information for all ColItem members in a ColList linked list.

INTERFACE:

```
SUBROUTINE Print_ColList( am_I_Root, ColList, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! root CPU?
TYPE(ColList),    INTENT(IN)    :: ColList
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER,          INTENT(INOUT) :: RC           ! Success?
```

REVISION HISTORY:

25 Jan 2018 - E. Lundgren - Initial version

1.15.12 Cleanup_DiagList

Subroutine Cleanup_DiagList deallocates a DiagList object and all of its member objects including the linked list of DiagItem objects.

INTERFACE:

```
SUBROUTINE Cleanup_DiagList ( am_I_Root, DiagList, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnList),    INTENT(INOUT) :: DiagList
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success?
```

REVISION HISTORY:

21 Sep 2017 - E. Lundgren - Initial version

1.15.13 Cleanup_CollList

Subroutine Cleanup_CollList deallocates a CollList object and all of its member objects including the linked list of CollItem objects.

INTERFACE:

```
SUBROUTINE Cleanup_CollList ( am_I_Root, CollList, RC )
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(CollList),  INTENT(INOUT) :: CollList
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success?
```

REVISION HISTORY:

```
21 Sep 2017 - E. Lundgren - Initial version
```

1.16 Fortran: Module Interface species_database_mod.F90

Module SPECIES_DATABASE_MOD contains routines to set up a database object containing physical properties for each GEOS-Chem species. This allows us to consolidate all species properties into a single data structure, for convenience.

INTERFACE:

```
MODULE Species_Database_Mod
```

USES:

```
USE Precision_Mod
```

```
IMPLICIT NONE
```

```
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: Init_Species_Database
PUBLIC  :: Cleanup_Species_Database
```

```
#if defined ( EXTERNAL_GRID ) || defined( EXTERNAL_FORCING )
```

```
!-----
```

```
!          %%%%%%%%% GEOS-Chem HP (with ESMF & MPI) %%%%%%%%%
```

```
! Cleanup routines for restoring the internal state of this
```

```
! module are exposed, so the DB can be reset from an external
```

```
! interface to perform multiple initializations of
! chemistry states. (hplin, 6/4/18)
```

```
!-----
```

```
PUBLIC   :: Cleanup_Work_Arrays
#endif
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: TranUc
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Uncomment this if you want to use the new henry's law constants
%% compiled by Katie Travis and posted on this wiki page:
%% http://wiki.geos-chem.org/Physical_properties_of_GEOS-Chem_species
#define NEW_HENRY_CONSTANTS 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

REMARKS:

REVISION HISTORY:

```
28 Aug 2015 - R. Yantosca - Initial version
02 Aug 2016 - M. Sulprizio- Add KppSpCId to store all KPP species incices.
```

1.16.1 Init_Species_Database

Initializes the GEOS-Chem species database object. You can add information about new species to this routine.

INTERFACE:

```
SUBROUTINE Init_Species_Database( am_I_Root, Input_Opt, SpcData, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE Species_Mod
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)  :: am_I_Root      ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt     ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(SpcPtr), POINTER     :: SpcData(:)    ! Vector with species info
```

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: RC ! Success or failure?

REMARKS:

For detailed information about the species database (and the physical properties that are specified there), please see the following GEOS-Chem wiki pages:

(1) wiki.geos-chem.org/GEOS_Chem_species_database

(2) wiki.geos-chem.org/Physical_properties_of_GEOS-Chem_species

References for the new Henry's law constants:

(1) Sander et al [2015]: <http://henrys-law.org>

The Hcp (aka K0) parameter listed on the wiki page:

http://wiki.geos-chem.org/Physical_properties_of_GEOS-Chem_species have units of [mol m⁻³ Pa⁻¹]. But the GEOS-Chem species object expects this parameter to have units of [M atm⁻¹]. Therefore, when we pass the Hcp parameter to routine Spc_Create via the HENRY_K0 argument, we will multiply by the proper conversion factor (9.86923e-3) to convert [mol m⁻³ Pa⁻¹] to [M atm⁻¹].

REVISION HISTORY:

- 22 Jul 2015 - R. Yantosca - Initial version
- 01 Sep 2015 - R. Yantosca - Add Henry K0, CR constants for DMS, ACET
- 02 Sep 2015 - R. Yantosca - Corrected typos for some SOA species
- 24 Sep 2015 - R. Yantosca - WD_RainoutEff is now a 3-element vector
- 24 Sep 2015 - R. Yantosca - Add WD_KcScaleFAC, a 3-element vector
- 30 Sep 2015 - R. Yantosca - DD_A_Density is renamed to Density
- 30 Sep 2015 - R. Yantosca - DD_A_Radius is renamed to Radius
- 01 Oct 2015 - R. Yantosca - Added DD_DvzMinVal field to put a minimum deposition velocity for sulfate aerosols
- 14 Oct 2015 - E. Lundgren - Treat H2SO4 as an aerosol for TOMAS
- 18 Nov 2015 - M. Sulprizio- Add passive tracers PASV to RnPbBe simulation
- 16 Dec 2015 - R. Yantosca - Use MW_g = 31.4 g/mol for SO4s and NITs
- 15 Mar 2016 - R. Yantosca - Added tagged CO tracer names
- 22 Apr 2016 - R. Yantosca - Now define Is_Hg0, Is_Hg2, Is_HgP fields
- 19 May 2016 - R. Yantosca - Remove DryDepId_PAN and DryDepId_HNO3; we shall now explicitly compute a drydep velocity for all GEOS-Chem species.
- 21 Jun 2016 - M. Sulprizio- Set Is_PhotoLysis to T for all species included in FAST-JX photolysis. Also added new species that are in FAST-JX photolysis but not already defined here.
- 18 Jul 2016 - M. Sulprizio- Remove family tracers ISOPN, MMN, CFCX, HCFCX and replace with their constituents.
- 02 Aug 2016 - M. Sulprizio- Add KppSpcId as argument passed to Spc_Create
- 11 Aug 2016 - E. Lundgren - Define special background conc for some species
- 22 Nov 2016 - M. Sulprizio- Move aerosol densities for BC, OC, and SO4 here from aerosol_mod.F
- 23 Feb 2017 - M. Sulprizio- Change MolecRatio for ALK4 from 4 to 4.3

(B. Henderson)
 13 Jun 2017 - M. Sulprizio- Add species for mechanistic isoprene SOA
 (E. Marais)
 27 Nov 2017 - E. Lundgren - Add SALA, SALC, OCPO/OCPI, BCPO/BCPI, and SO4
 as hygroscopic growth species for cloud diags

1.16.2 Cleanup_Species_Database

Finalizes the vector with species information.

INTERFACE:

```
SUBROUTINE Cleanup_Species_Database( am_I_Root, SpcData, RC )
```

USES:

```
USE ErrCode_Mod
USE Species_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(SpcPtr),    POINTER     :: SpcData(:)    ! Species database object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC           ! Success or failure?
```

REMARKS:

REVISION HISTORY:

```
22 Jul 2015 - R. Yantosca - Initial version
```

1.16.3 TranUc

Tranlate a character variable to all upper case letters. Non-alphabetic characters are not affected. The original "text" is destroyed.

INTERFACE:

```
SUBROUTINE TranUc( text )
```

INPUT/OUTPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(INOUT) :: text
```

AUTHOR:

Robert D. Stewart, May 19, 1992 (part of CHARPAK)

REMARKS:

Keep a private shadow copy of this routine here so as not to incur a dependency with GeosUtil/charpak_mod.F. This lets us keep species_datbase_mod.F90 in the Headers/ folder together with state_chm_mod.F90 and species_mod.F90.

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Initial version

1.16.4 Unique_Species_Names

Stores the list of unique species names (i.e. removing duplicates from the list of advected species and the the list of KPP species) for later use. Also computes the corresponding indices for the KPP variable and fixed species arrays (VAR and FIX, respectively).

INTERFACE:

```
SUBROUTINE Unique_Species_Names( am_I_Root, Input_Opt, nSpecies, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE GcKpp_Monitor,    ONLY : Spc_Names
USE GcKpp_Parameters, ONLY : NFIX, NSPEC, NVAR
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: nSpecies     ! Number of unique species
INTEGER,          INTENT(OUT) :: RC           ! Success or failure
```

REMARKS:

This may not be the fastest search algorithm (because it relies on string comparisons). But it is only executed at startup so we can live with it. We could make it faster by hashing but that seems like overkill.

REVISION HISTORY:

09 May 2016 - R. Yantosca - Initial version
 02 Aug 2016 - M. Sulprizio- Add KppSpcId; Also only set KppVarId if loop
 indexis <= NVAR.

1.16.5 Cleanup_Work_Arrays

Cleans working (temporary) arrays used by this module, restoring them to an unused state. It is called at the end of Init_Species_Database or by an external module when needed to reinitialize the species DB.

INTERFACE:

```
SUBROUTINE Cleanup_Work_Arrays()
```

REMARKS:

This routine allows Species_Database_Mod to be initialized more than once in the same CPU, if called externally before re-initializing a State_Chm derived type object.

REVISION HISTORY:

```
06 May 2016 - R. Yantosca - Initial version
05 Jul 2018 - H.P. Lin    - Add missing KppSpId
```

1.17 Fortran: Module Interface physconstants.F

PhysConstants contains GEOS-Chem specific PHYSICAL CONSTANTS and DERIVED QUANTITIES.

INTERFACE:

```
MODULE PHYSCONSTANTS
```

USES:

```
USE PRECISION_MOD    ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

```
PUBLIC
```

DEFINED PARAMETERS:

```
! AIRMW : Average molecular weight of dry air [g/mol]
```

```
REAL(fp), PARAMETER :: AIRMW    = 28.97e+0_fp
```

```
! H2OMW : Molecular weight of water [g/mol]
```

```
REAL(fp), PARAMETER :: H2OMW    = 18.016e+0_fp
```

```
! AVO    : Avogadro's number [particles/mol]
```

```
! Now use more precise value 6.022140857e+23 instead of 6.022e+23?
```

```
! Source: NIST, 2014 (ewl, 1/7/16) (NEED TO CHANGE HEMCO)
```

```
REAL(fp), PARAMETER :: AVO      = 6.022140857e+23_fp
```

```
! g0     : Acceleration due to gravity at earth's surface [m/s^2]
```

```
! Now use more precise value of 9.80665 instead of 9.8 (ewl, 1/7/16)
! Source: NIST, 2014 (NEED TO CHANGE HEMCO)
REAL(fp), PARAMETER :: g0      = 9.80665e+0_fp

! g0_100 : 100 / g0
REAL(fp), PARAMETER :: g0_100 = 100.e+0_fp / g0

! PI      : Double-Precision value of PI
REAL(fp), PARAMETER :: PI      = 3.14159265358979323e+0_fp

! PI_180  : Number of radians per degree
REAL(fp), PARAMETER :: PI_180 = PI / 180e+0_fp

! Re      : Radius of Earth [m]
REAL(fp), PARAMETER :: Re      = 6.375e+6_fp

! Rd      : Gas Constant in Dry Air [J/K/kg]
REAL(fp), PARAMETER :: Rd      = 287.0e+0_fp

! Rv      : Gas Constant for water vapor [J/K/kg]
REAL(fp), PARAMETER :: Rv      = 461.0e+0_fp

! Rdg0    = Rd      / g0
REAL(fp), PARAMETER :: Rdg0    = Rd / g0

! SCALE_HEIGHT : Scale height of atmosphere [m]
REAL(fp), PARAMETER :: SCALE_HEIGHT = 7600.e+0_fp

! VON_KARMAN : Von Karman's constant [.]
REAL(fp), PARAMETER :: VON_KARMAN = 0.4e+0_fp

! RSTARG   : Molar gas constant [J/K/mol]
! Now use more precise value 8.3144598 instead of 8.31450? (ewl, 1/7/16)
! Source: NIST, 2014 (NEED TO CHANGE HEMCO)
REAL(fp), PARAMETER :: RSTARG   = 8.3144598e+0_fp

! XNUMOLAIR : Molecules dry air per kg dry air
REAL(fp), PARAMETER :: XNUMOLAIR = AVO / ( AIRMW * 1.e-3_fp)

! BOLTZ    : Boltzmann's constant [J/K] (Source: NIST, 2014)
REAL(fp), PARAMETER :: BOLTZ    = 1.38064852e-23_fp

! ATM      : Standard atmosphere [Pa] (Source: NIST, 2014)
REAL(fp), PARAMETER :: ATM      = 1.01325e+5_fp

! Condensation vapor pressure
! ** NEED SOURCE **
! We think 6.1078 hPa is the saturation vapor pressure at 273.16 K, the
```

```

!   triple point of water, but this needs to be confirmed (mps, 4/21/16)
!   Use BOLTZ [J/K] rather than BOLTG [ergs/K] from comode_loop_mod
!   (ewl, 1/4/16)
REAL(fp), PARAMETER :: CONSVAP = 6.1078e+03_fp /
&                               ( BOLTZ * 1e+7_fp )

! Gas constant in: [L.atm/K.mole]
REAL(fp), PARAMETER :: RGASLTM = 8.2057e-2_fp

! Molecular weight of carbon (kg/mol)
REAL(fp), PARAMETER :: MWCARB = 12.01e-3_fp

```

!REFERENCES:

(1) NIST, 2014. Website: <http://physics.nist.gov/cuu/Constants/index.html>

REVISION HISTORY:

```

25 Jun 2002 - R. Yantosca - Initial version
23 Aug 2011 - M. Long      - Converted to Module from Header file
23 Jul 2014 - R. Yantosca - Add Von Karman's constant here
23 Jul 2014 - R. Yantosca - Add Avogadro's number here
23 Jul 2014 - R. Yantosca - List constants in alphabetical order
02 Dec 2014 - M. Yannetti - Added PRECISION_MOD
06 Jan 2014 - E. Lundgren - Added constants and clarified comments
24 Mar 2015 - E. Lundgren - Add xnumolair, previously in tracer_mod
08 Jan 2016 - E. Lundgren - Update values to NIST 2014 and use AIRMW in
                           definition of XNUMOLAIR. Add BOLTZ.
08 Jan 2016 - E. Lundgren - Rename this file to physconstants.F. Previously
                           it was called CMN_GCTM_mod.F
21 Apr 2016 - M. Sulprizio- Move CONSVAP here from comode_loop_mod.F

```

1.18 Fortran: Module Interface input_opt_mod.F90

Module INPUT_OPT_MOD contains the derived type for GEOS-Chem options and logical switches.

INTERFACE:

```
MODULE Input_Opt_Mod
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```

PUBLIC :: Set_Input_Opt
PUBLIC :: Set_Input_Opt_Advect
PUBLIC :: Set_Input_Opt_Passive
PUBLIC :: Cleanup_Input_Opt

```

PUBLIC DATA MEMBERS:

```

=====
! Derived type for Input Options
=====
TYPE, PUBLIC :: OptInput

!-----
! General Runtime & Distributed Comp Info
!-----
INTEGER          :: NPES      ! Number of MPI procs
INTEGER          :: myCpu     ! Local MPI process handle
INTEGER          :: MPICOMM   ! MPI Communicator Handle
LOGICAL          :: HPC       ! Is this an HPC (ESMF or otherwise) sim?
LOGICAL          :: RootCPU   ! Is this the root cpu?

!-----
! SIZE PARAMETER fields
!-----
INTEGER          :: MAX_DIAG
INTEGER          :: MAX_FAM

!-----
! SIMULATION MENU fields
!-----
INTEGER          :: NYMdb
INTEGER          :: NHMSb
INTEGER          :: NYMDe
INTEGER          :: NHMSe
CHARACTER(LEN=255) :: RUN_DIR
CHARACTER(LEN=255) :: IN_RST_FILE
CHARACTER(LEN=255) :: DATA_DIR
CHARACTER(LEN=255) :: CHEM_INPUTS_DIR
CHARACTER(LEN=255) :: RES_DIR
CHARACTER(LEN=255) :: GEOS_FP_DIR
CHARACTER(LEN=255) :: MERRA2_DIR
LOGICAL          :: LCAPTROP
REAL(fp)         :: OZONOPAUSE
INTEGER          :: NESTED_IO
INTEGER          :: NESTED_JO
CHARACTER(LEN=255) :: HcoConfigFile

!-----
! PASSIVE SPECIES MENU fields

```

```

!-----
INTEGER                :: NPASSIVE
INTEGER                :: NPASSIVE_DECAY
CHARACTER(LEN=63), POINTER :: PASSIVE_NAME    (:)
INTEGER,                POINTER :: PASSIVE_ID    (:)
REAL(fp),               POINTER :: PASSIVE_MW    (:)
REAL(fp),               POINTER :: PASSIVE_TAU   (:)
REAL(fp),               POINTER :: PASSIVE_INITCONC(:)
INTEGER,                POINTER :: PASSIVE_DECAYID (:)

!-----
! ADVECTED SPECIES MENU fields
!-----
INTEGER                :: N_ADVECT
CHARACTER(LEN=255), POINTER :: AdvectSpc_Name(:)
INTEGER                :: SIM_TYPE
CHARACTER(LEN=255)      :: SIM_NAME
LOGICAL                :: LSPLIT
LOGICAL                :: ITS_A_RnPPbBe_SIM
LOGICAL                :: ITS_A_CH3I_SIM
LOGICAL                :: ITS_A_FULLCHEM_SIM
LOGICAL                :: ITS_A_HCN_SIM
LOGICAL                :: ITS_A_TAGO3_SIM
LOGICAL                :: ITS_A_TAGCO_SIM
LOGICAL                :: ITS_A_C2H6_SIM
LOGICAL                :: ITS_A_CH4_SIM
LOGICAL                :: ITS_AN_AEROSOL_SIM
LOGICAL                :: ITS_A_MERCURY_SIM
LOGICAL                :: ITS_A_CO2_SIM
LOGICAL                :: ITS_A_H2HD_SIM
LOGICAL                :: ITS_A_POPS_SIM

!-----
! AEROSOL MENU fields
!-----
LOGICAL                :: LSULF
LOGICAL                :: LMETALCATSO2
LOGICAL                :: LCARB
LOGICAL                :: LBRC
LOGICAL                :: LSOA
LOGICAL                :: LMPOA
LOGICAL                :: LSVPOA
LOGICAL                :: LOMOC
LOGICAL                :: LDUST
LOGICAL                :: LDEAD
LOGICAL                :: LSSALT
LOGICAL                :: LDSTUP
REAL(fp),              POINTER :: SALA_REDGE_um(:)

```

```
REAL(fp),          POINTER :: SALC_REDGE_um(:)
LOGICAL            :: LGRAVSTRAT
LOGICAL            :: LSOLIDPSC
LOGICAL            :: LHOMNUCNAT
REAL(fp)           :: T_NAT_SUPERCOOL
REAL(fp)           :: P_ICE_SUPERSAT
LOGICAL            :: LPSCCHEM
LOGICAL            :: LSTRATOD
!for BC absorption enhancement, (xnw, 8/24/15)
LOGICAL            :: LBCAE
REAL(fp)           :: BCAE_1
REAL(fp)           :: BCAE_2
```

```
!-----
```

```
! EMISSIONS MENU fields
```

```
!-----
```

```
LOGICAL            :: LEMIS
INTEGER            :: TS_EMIS
LOGICAL            :: LBIOFUEL
LOGICAL            :: LOTDLOC
LOGICAL            :: LSOILNOX
LOGICAL            :: LWARWICK_VSLS
LOGICAL            :: LSSABr2
LOGICAL            :: LFIX_PBL_BRO
LOGICAL            :: LCH4EMIS
LOGICAL            :: LCH4SBC
LOGICAL            :: LOCSEMIS
LOGICAL            :: LCFCEMIS
LOGICAL            :: LCLEMIS
LOGICAL            :: LBREMIS
LOGICAL            :: LN2OEMIS
LOGICAL            :: LBASICEMIS
LOGICAL            :: LSETH2O
INTEGER            :: CFCYEAR
LOGICAL            :: LFUTURECFC
```

```
!-----
```

```
! CO MENU fields
```

```
!-----
```

```
LOGICAL            :: LPCO_CH4
LOGICAL            :: LPCO_NMVOC
```

```
!-----
```

```
! CO2 MENU fields
```

```
!-----
```

```
LOGICAL            :: LFOSSIL
LOGICAL            :: LCHEMCO2
LOGICAL            :: LBIODIURNAL
```

```
LOGICAL          :: LBIONETCLIM
LOGICAL          :: LOCEAN
LOGICAL          :: LSHIP
LOGICAL          :: LPLANE
LOGICAL          :: LFFBKGRD
LOGICAL          :: LBIOSPHTAG
LOGICAL          :: LFOSSILTAG
LOGICAL          :: LSHIPTAG
LOGICAL          :: LPLANETAG

!-----
! CHEMISTRY MENU fields
!-----
LOGICAL          :: LCHEM
LOGICAL          :: LSCHEM
LOGICAL          :: LLINOZ
LOGICAL          :: LSYNOZ
INTEGER          :: TS_CHEM
REAL(fp)         :: GAMMA_HO2
LOGICAL          :: LUCX
LOGICAL          :: LCH4CHEM
LOGICAL          :: LACTIVEH2O
LOGICAL          :: LINITSPEC
INTEGER, POINTER :: NTLOOPNCS(:)
LOGICAL          :: USE_ONLINE_03
LOGICAL          :: USE_03_FROM_MET
LOGICAL          :: USE_TOMS_03
INTEGER          :: LLSTRAT

!-----
! RADIATION MENU fields
!-----
LOGICAL          :: LRAD
LOGICAL          :: LLWRAD
LOGICAL          :: LSWRAD
LOGICAL, POINTER :: LSKYRAD(:)
INTEGER          :: TS_RAD
INTEGER          :: NWVSELECT
REAL(8), ALLOCATABLE :: WVSELECT(:)
CHARACTER(LEN=5), ALLOCATABLE :: STRWVSELECT(:)

!-----
! TRANSPORT MENU fields
!-----
LOGICAL          :: LTRAN
LOGICAL          :: LFILL
INTEGER          :: TPCORE_IORD
INTEGER          :: TPCORE_JORD
```

```

INTEGER                :: TPCORE_KORD
INTEGER                :: TS_DYN

!-----
! CONVECTION MENU fields
!-----
LOGICAL                :: LCONV
LOGICAL                :: LTURB
LOGICAL                :: LNLPBL
INTEGER                :: TS_CONV

!-----
! DEPOSITION MENU fields
!-----
LOGICAL                :: LDRYD
LOGICAL                :: LWETD
REAL(fp)              :: WETD_CONV_SCAL
LOGICAL                :: PBL_DRYDEP

!-----
! GAMAP MENU fields
!-----
CHARACTER(LEN=255)    :: GAMAP_DIAGINFO
CHARACTER(LEN=255)    :: GAMAP_TRACERINFO

!-----
! OUTPUT MENU fields
!-----
INTEGER,              POINTER :: NJDAY(:)

!-----
! DIAGNOSTIC MENU fields
!-----
CHARACTER(LEN=255)    :: HistoryInputFile
INTEGER                :: ND01,          LD01
INTEGER                :: ND02,          LD02
INTEGER                :: ND03,          LD03
INTEGER                :: ND04,          LD04
INTEGER                :: ND05,          LD05
INTEGER                :: ND06,          LD06
INTEGER                :: ND07,          LD07
INTEGER                :: ND08,          LD08
INTEGER                :: ND09,          LD09
INTEGER                :: ND10,          LD10
INTEGER                :: ND11,          LD11
INTEGER                :: ND12,          LD12
INTEGER                :: ND13,          LD13
INTEGER                :: ND14,          LD14

```


INTEGER	:: ND15,	LD15
INTEGER	:: ND16,	LD16
INTEGER	:: ND17,	LD17
INTEGER	:: ND18,	LD18
INTEGER	:: ND19,	LD19
INTEGER	:: ND20,	LD20
INTEGER	:: ND21,	LD21
INTEGER	:: ND22,	LD22
INTEGER	:: ND23,	LD23
INTEGER	:: ND24,	LD24
INTEGER	:: ND25,	LD25
INTEGER	:: ND26,	LD26
INTEGER	:: ND27,	LD27
INTEGER	:: ND28,	LD28
INTEGER	:: ND29,	LD29
INTEGER	:: ND30,	LD30
INTEGER	:: ND31,	LD31
INTEGER	:: ND32,	LD32
INTEGER	:: ND33,	LD33
INTEGER	:: ND34,	LD34
INTEGER	:: ND35,	LD35
INTEGER	:: ND36,	LD36
INTEGER	:: ND37,	LD37
INTEGER	:: ND38,	LD38
INTEGER	:: ND39,	LD39
INTEGER	:: ND40,	LD40
INTEGER	:: ND41,	LD41
INTEGER	:: ND42,	LD42
INTEGER	:: ND43,	LD43
INTEGER	:: ND44,	LD44
INTEGER	:: ND45,	LD45
INTEGER	:: ND46,	LD46
INTEGER	:: ND47,	LD47
INTEGER	:: ND48,	LD48
INTEGER	:: ND49,	LD49
INTEGER	:: ND50,	LD50
INTEGER	:: ND51,	LD51
INTEGER	:: ND52,	LD52
INTEGER	:: ND53,	LD53
INTEGER	:: ND54,	LD54
INTEGER	:: ND55,	LD55
INTEGER	:: ND56,	LD56
INTEGER	:: ND57,	LD57
INTEGER	:: ND58,	LD58
INTEGER	:: ND59,	LD59
INTEGER	:: ND60,	LD60
INTEGER	:: ND61,	LD61
INTEGER	:: ND62,	LD62

```

INTEGER          :: ND63,          LD63
INTEGER          :: ND64,          LD64
INTEGER          :: ND66,          LD66
INTEGER          :: ND67,          LD67
INTEGER          :: ND68,          LD68
INTEGER          :: ND69,          LD69
INTEGER          :: ND70,          LD70
INTEGER          :: ND71,          LD71
INTEGER          :: ND72,          LD72
INTEGER          :: ND73,          LD73

INTEGER          :: TS_DIAG
LOGICAL          :: LPRT
INTEGER,         POINTER :: TINDEX(:, :)
INTEGER,         POINTER :: TCOUNT(:)
INTEGER,         POINTER :: TMAX(:)
LOGICAL          :: DO_DIAG_WRITE

! Collection ids
INTEGER          :: DIAG_COLLECTION
INTEGER          :: GC_RST_COLLECTION ! Used only for NetCDF

!-----
! PLANEFLIGHT MENU fields
!-----
LOGICAL          :: DO_PF
CHARACTER(LEN=255) :: PF_IFILE
CHARACTER(LEN=255) :: PF_OFILE

!-----
! ND48 MENU fields
!-----
LOGICAL          :: DO_ND48
CHARACTER(LEN=255) :: ND48_FILE
INTEGER          :: ND48_FREQ
INTEGER          :: ND48_N_STA
INTEGER,         POINTER :: ND48_IARR(:)
INTEGER,         POINTER :: ND48_JARR(:)
INTEGER,         POINTER :: ND48_LARR(:)
INTEGER,         POINTER :: ND48_NARR(:)

!-----
! ND49 MENU fields
!-----
LOGICAL          :: DO_ND49
INTEGER          :: N_ND49
CHARACTER(LEN=255) :: ND49_FILE
INTEGER,         POINTER :: ND49_TRACERS(:)

```

```

INTEGER                :: ND49_FREQ
INTEGER                :: ND49_IMIN
INTEGER                :: ND49_IMAX
INTEGER                :: ND49_JMIN
INTEGER                :: ND49_JMAX
INTEGER                :: ND49_LMIN
INTEGER                :: ND49_LMAX

!-----
! ND50 MENU fields
!-----
LOGICAL                :: DO_ND50
INTEGER                :: N_ND50
CHARACTER(LEN=255)    :: ND50_FILE
LOGICAL                :: LND50_HDF
INTEGER,               POINTER :: ND50_TRACERS(:)
INTEGER                :: ND50_IMIN
INTEGER                :: ND50_IMAX
INTEGER                :: ND50_JMIN
INTEGER                :: ND50_JMAX
INTEGER                :: ND50_LMIN
INTEGER                :: ND50_LMAX

!-----
! ND51 MENU fields
!-----
LOGICAL                :: DO_ND51
INTEGER                :: N_ND51
CHARACTER(LEN=255)    :: ND51_FILE
LOGICAL                :: LND51_HDF
INTEGER,               POINTER :: ND51_TRACERS(:)
REAL(fp)              :: ND51_HR_WRITE
REAL(fp)              :: ND51_HR1
REAL(fp)              :: ND51_HR2
INTEGER                :: ND51_IMIN
INTEGER                :: ND51_IMAX
INTEGER                :: ND51_JMIN
INTEGER                :: ND51_JMAX
INTEGER                :: ND51_LMIN
INTEGER                :: ND51_LMAX

!-----
! ND51b MENU fields
!-----
LOGICAL                :: DO_ND51b
INTEGER                :: N_ND51b
CHARACTER(LEN=255)    :: ND51b_FILE
LOGICAL                :: LND51b_HDF

```

```

INTEGER,          POINTER :: ND51b_TRACERS(:)
REAL(fp)          :: ND51b_HR_WRITE
REAL(fp)          :: ND51b_HR1
REAL(fp)          :: ND51b_HR2
INTEGER           :: ND51b_IMIN
INTEGER           :: ND51b_IMAX
INTEGER           :: ND51b_JMIN
INTEGER           :: ND51b_JMAX
INTEGER           :: ND51b_LMIN
INTEGER           :: ND51b_LMAX

!-----
! ND63 MENU fields
!-----
LOGICAL           :: DO_ND63
INTEGER           :: N_ND63
CHARACTER(LEN=255) :: ND63_FILE
INTEGER,          POINTER :: ND63_TRACERS(:)
INTEGER           :: ND63_FREQ
INTEGER           :: ND63_IMIN
INTEGER           :: ND63_IMAX
INTEGER           :: ND63_JMIN
INTEGER           :: ND63_JMAX

!-----
! PROD LOSS MENU fields
!-----
LOGICAL           :: DO_SAVE_PL
INTEGER           :: ND65, LD65
LOGICAL           :: DO_SAVE_O3
LOGICAL           :: DO_SAVE_PCO
INTEGER           :: NFAM
CHARACTER(LEN=255), POINTER :: FAM_NAME(:)
CHARACTER(LEN=255), POINTER :: FAM_TYPE(:)

!-----
! NESTED GRID MENU fields
!-----
LOGICAL           :: ITS_A_NESTED_GRID
LOGICAL           :: LWINDO
LOGICAL           :: LWINDO2x25
LOGICAL           :: LWINDO_NA
CHARACTER(LEN=255) :: TPBC_DIR_NA
LOGICAL           :: LWINDO_EU
CHARACTER(LEN=255) :: TPBC_DIR_EU
LOGICAL           :: LWINDO_CH
CHARACTER(LEN=255) :: TPBC_DIR_CH
LOGICAL           :: LWINDO_AS

```

```
CHARACTER(LEN=255)      :: TPBC_DIR_AS
LOGICAL                 :: LWINDO_CU
CHARACTER(LEN=255)      :: TPBC_DIR
INTEGER                 :: NESTED_TS
INTEGER                 :: NESTED_I1
INTEGER                 :: NESTED_J1
INTEGER                 :: NESTED_I2
INTEGER                 :: NESTED_J2
INTEGER                 :: NESTED_IOW
INTEGER                 :: NESTED_JOW
INTEGER                 :: NESTED_IOE
INTEGER                 :: NESTED_JOE

!-----
! BENCHMARK MENU fields
!-----
LOGICAL                 :: LSTDRUN
CHARACTER(LEN=255)      :: STDRUN_INIT_FILE
CHARACTER(LEN=255)      :: STDRUN_FINAL_FILE

!-----
! MERCURY MENU fields
!-----
INTEGER                 :: ANTHRO_Hg_YEAR
CHARACTER(LEN=255)      :: HG_SCENARIO
LOGICAL                 :: USE_CHECKS
LOGICAL                 :: LDYNOCEAN
LOGICAL                 :: LPREINDHG
LOGICAL                 :: LGTMM
CHARACTER(LEN=255)      :: GTMM_RST_FILE
LOGICAL                 :: LARCTICRIV
LOGICAL                 :: LKRedUV

!-----
! CH4 MENU fields
!-----
LOGICAL                 :: GOSAT_CH4_OBS
LOGICAL                 :: TCCON_CH4_OBS

!-----
! POPS MENU fields
!-----
CHARACTER(LEN=3)        :: POP_TYPE
LOGICAL                 :: CHEM_PROCESS
REAL(fp)                :: POP_XMW
REAL(fp)                :: POP_KOA
REAL(fp)                :: POP_KBC
REAL(fp)                :: POP_K_POPG_OH
```

```

REAL(fp)                :: POP_K_POPP_03A
REAL(fp)                :: POP_K_POPP_03B
REAL(fp)                :: POP_HSTAR
REAL(fp)                :: POP_DEL_H
REAL(fp)                :: POP_DEL_Hw

!-----
! Fields for interface to GEOS-5 GCM
!-----
LOGICAL                  :: haveImpRst
LOGICAL                  :: AlwaysSetH2O

!-----
! Fields for LINOZ strat chem
!-----
INTEGER                  :: LINOZ_NLEVELS
INTEGER                  :: LINOZ_NLAT
INTEGER                  :: LINOZ_NMONTHS
INTEGER                  :: LINOZ_NFIELDS
REAL(fp),                POINTER :: LINOZ_TPARAM(:, :, :, :, :)

END TYPE OptInput

```

REMARKS:

This will eventually replace the switches in logical_mod.F.

REVISION HISTORY:

```

01 Nov 2012 - R. Yantosca - Initial version, based on logical_mod.F
                                newer Olson 2001 land map & drydep inputs
07 Nov 2012 - R. Yantosca - Added Input_Opt%ITS_A*_SIM fields
08 Nov 2012 - R. Yantosca - Added APM MENU fields
09 Nov 2012 - R. Yantosca - Added LD* variables for diagnostic levels
28 Nov 2012 - R. Yantosca - Add USE_OLSON_2001 logical flag
22 May 2013 - M. Payer - Add GAMMA_HO2 variable for chemistry menu
26 Feb 2013 - M. Long - Add extra fields from input.geos
26 Feb 2013 - M. Long - Bug fix: timesteps are now INTEGER, not LOGICAL
28 Feb 2013 - R. Yantosca - Add haveImpRst field for GEOS-5 GCM interface
08 Mar 2013 - R. Yantosca - Add myCpu field to pass CPU # to GEOS-Chem
15 Mar 2013 - R. Yantosca - Add fields for LINOZ strat chemistry
27 Mar 2013 - R. Yantosca - Add extra fields for tagged CO2
27 Mar 2013 - R. Yantosca - Add extra fields for tagged EDGAR
29 Mar 2013 - R. Yantosca - Add DO_DIAG_WRITE field (to shut diags in MPI)
22 Jul 2013 - M. Sulprizio- Add extra fields for RCP emissions
31 Jul 2013 - M. Sulprizio- Add extra field for AEIC aircraft emissions and
                                remove LAIRNOX field
13 Aug 2013 - M. Sulprizio- Add extra fields for semivolatile POA (H. Pye)
22 Aug 2013 - R. Yantosca - Add fields for soil NOx & species restart files

```

- 17 Sep 2013 - M. Sulprizio- Add LDSTUP flag for acid uptake on dust aerosols
- 26 Sep 2013 - R. Yantosca - Renamed GEOS_57_DIR to GEOS_FP_DIR
- 03 Oct 2013 - M. Sulprizio- Removed obsolete LMFACT for flux correction
- 03 Oct 2013 - M. Sulprizio- Removed obsolete LAVHRRRLAI and LMODISLAI
- 13 Dec 2013 - M. Sulprizio- Add USE_O3_FROM_MET logical flag
- 16 Apr 2014 - M. Sulprizio- Add field for PSC restart file
- 23 Jun 2014 - R. Yantosca - Add POP_EMISDIR field for POPs simulation
- 25 Jun 2014 - R. Yantosca - Now add Input_Opt%SIM_TYPE field
- 29 Sep 2014 - R. Yantosca - Now add Input_Opt%N_DUST_BINS field
- 03 Dec 2014 - M. Yannetti - Added PRECISION_MOD
- 03 Dec 2014 - M. Sulprizio- Add fields for Radiation Menu
- 16 Dec 2014 - R. Yantosca - Removed JLOP, JLOP_PREV; these are in State_Chm
- 01 Apr 2015 - R. Yantosca - Add extra nested-grid fields
- 09 Apr 2015 - M. Sulprizio- Removed fields for NAPEMISS, POAEMISSSCALE, and PST_RST_FILE. These options are now handled by HEMCO.
- 11 Aug 2015 - R. Yantosca - Add MERRA2_DIR field to OptInput
- 26 Jan 2016 - E. Lundgren - Add fields for netcdf diagnostics
- 04 Feb 2016 - C. Keller - Add LINITSPEC. Used in ESMF to initialize species concentrations from globchem.dat.
- 04 Feb 2016 - M. Sulprizio- Add Hg_CAT and Hg_CAT_FULL arrays for tagged Hg simulations
- 27 Apr 2016 - R. Yantosca - Remove Hg_Cat, Hg_Cat_Full fields
- 17 May 2016 - R. Yantosca - Remove TRACER_N_CONST, TRACER_CONST, ID_EMITTED, TRACER_COEFF
- 31 May 2016 - E. Lundgren - Remove TRACER_MW_KG, TRACER_MW_G, and XNUMOL
- 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer compatible with the FlexChem implementation
- 13 Jul 2016 - R. Yantosca - Remove some unused drydep fields
- 16 Aug 2016 - M. Sulprizio- Rename from gigc_input_opt_mod.F90 to input_opt_mod.F90. The "gigc" nomenclature is no longer used.
- 29 Aug 2016 - M. Sulprizio- Rename N_TRACERS to N_ADVECT and TRACER_NAME to AdvectSpc_Name to reflect that we now refer to tracers as advected species
- 20 Sep 2016 - R. Yantosca - LND51_HDF and LND51b_HDF are now declared as LOGICAL, not INTEGER. This chokes Gfortran.
- 03 Oct 2016 - R. Yantosca - LWINDO_CU has to be LOGICAL, not INTEGER
- 16 Jun 2017 - M. Sulprizio- Remove switches for CH4 emissions since these are now handled by HEMCO
- 12 Jul 2017 - R. Yantosca - Add Input_Opt%HistoryInputFile field
- 13 Jul 2017 - E. Lundgren - Add passive species variables
- 24 Aug 2017 - M. Sulprizio- Remove obsolete options: GCAP_DIR, GEOS_4_DIR, GEOS_5_DIR, MERRA_DIR, TEMP_DIR, LUNZIP, LWAIT
- 13 Sep 2017 - M. Sulprizio- Remove USE_OLSON_2001. Olson 2001 is now the default.
- 14 Sep 2017 - M. Sulprizio- Add USE_ONLINE_O3 and USE_TOMS_O3 to options for overhead O3 in chemistry menu

02 Nov 2017 - R. Yantosca - Bug fix: LBIOFUEL should be LOGICAL
 07 Nov 2017 - R. Yantosca - Remove LVARTROP; it's not needed
 29 Dec 2017 - C. Keller - Added LLSTRAT. Used in gc_environment_mod.F90
 29 Dec 2017 - C. Keller - Added AlwaysSetH2O.
 04 Apr 2018 - E. Lundgren - Remove MAX_PASV; use # from input.geos instead

1.18.1 Set_Input_Opt

Subroutine SET_INPUT_OPT initializes all GEOS-Chem options carried in Input Options derived type object.

INTERFACE:

```
SUBROUTINE Set_Input_Opt( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(Input_Opt), INTENT(INOUT) :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)     :: RC          ! Success or failure?
```

REMARKS:

Set the following fields of Input_Opt outside of this routine:

```
(1 ) Input_Opt%MAX_DIAG      : Max # of diagnostics
(3 ) Input_Opt%MAX_MEMB      : Max # of members per family tracer
(4 ) Input_Opt%MAX_FAM       : Max # of P/L diagnostic families
(5 ) Input_Opt%MAX_DEP       : Max # of dry depositing species
(6 ) Input_Opt%MAX_PASV      : Max # of passive species
(7 ) Input_Opt%LINOZ_NLEVELS : Number of levels    in LINOZ climatology
(8 ) Input_Opt%LINOZ_NLAT    : Number of latitudes in LINOZ climatology
(9 ) Input_Opt%LINOZ_NMONTHS : Number of months   in LINOZ climatology
(10) Input_Opt%LINOZ_NFIELDS : Number of species  in LINOZ climatology
```

We also need to implement better error checking.

REVISION HISTORY:

```
01 Nov 2012 - R. Yantosca - Initial version
07 Nov 2012 - R. Yantosca - Now add size parameter fields to Input_Opt
                           that can be set prior to calling this routine
09 Nov 2012 - R. Yantosca - Now zero LD* fields for diagnostic levels
```


28 Nov 2012 - R. Yantosca - Now set USE_OLSON_2001 logical flag
29 Nov 2012 - M. Payer - Add Input_Opt%ITS_A_POPS_SIM
26 Feb 2013 - M. Long - Add extra fields from input.geos
28 Feb 2013 - R. Yantosca - Add haveImpRst field for GEOS-5 GCM interface
08 Mar 2013 - R. Yantosca - Now initialize the myCpu field
15 Mar 2013 - R. Yantosca - Now initialize the LINOZ_TPARM field
27 Mar 2013 - R. Yantosca - Add extra fields for tagged CO2
27 Mar 2013 - R. Yantosca - Add extra fields for EDGAR
29 Mar 2013 - R. Yantosca - Add DO_DIAG_WRITE field (to shut diags in MPI)
22 Apr 2013 - R. Yantosca - Now dimension ND48_*ARR to 1000 so that we are
consistent with the settings in diag48_mod.F
22 Jul 2013 - M. Sulprizio- Add extra fields for RCP emissions
07 Aug 2013 - M. Sulprizio- Add extra fields for SOA + SVPOA simulation
22 Aug 2013 - R. Yantosca - Add fields for soil NOx & species restart files
12 Sep 2013 - M. Sulprizio- Double size of IDDEP to account for dust
alkalinity (tdf 04/10/08)
17 Sep 2013 - M. Sulprizio- Add LDSTUP flag for acid uptake on dust aerosols
26 Sep 2013 - R. Yantosca - Renamed GEOS_57_DIR to GEOS_FP_DIR
25 Jun 2014 - R. Yantosca - Now initialize Input_Opt%SIM_TYPE field
03 Dec 2014 - M. Yannetti - Added PRECISION_MOD
05 Mar 2015 - R. Yantosca - Added RES_DIR, CHEM_INPUTS_DIR fields
06 Mar 2015 - R. Yantosca - Now initialize directory names with './'
01 Apr 2015 - R. Yantosca - Now initialize extra nested-grid fields
04 Mar 2016 - C. Keller - Added WETD_CONV_SCAL, LSYNOZ, LCAPTROP, and
OZONOPAUSE. These are only used within ESMF.
17 May 2016 - R. Yantosca - Remove TRACER_N_CONST, TRACER_CONST, ID_EMITTED,
TRACER_COEFF
31 May 2016 - E. Lundgren - Remove TRACER_MW_G, TRACER_MW_KG, and XNUMOL
13 Jul 2016 - R. Yantosca - Remove some obsolete drydep fields
13 Jul 2016 - R. Yantosca - Remove ID_TRACER, NUMDEP
16 Mar 2017 - R. Yantosca - Remove obsolete family and drydep variables
17 Mar 2017 - R. Yantosca - Remove IDDEP, DUSTREFF, DUSTDEN
12 Jul 2017 - R. Yantosca - Initialize Input_Opt%HistoryInputFile field
02 Nov 2017 - R. Yantosca - LWINDO_CU should be .FALSE., not 0
07 Nov 2017 - R. Yantosca - Remove LVARTROP; it's not needed
08 Mar 2018 - R. Yantosca - Bug fix, remove reference to TINDEXT here

1.18.2 Set_Input_Opt_Advect

Subroutine SET_INPUT_OPT_ADVECT initializes all GEOS-Chem options carried in Input Options derived type object that depend on the number of advected species (Input_Opt

INTERFACE:

```
SUBROUTINE Set_Input_Opt_Advect( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REMARKS:

NOTE: These arrays are all for bpch diagnostics, and will eventually be removed from GEOS-Chem.

REVISION HISTORY:

```
26 Jan 2018 - M. Sulprizio- Initial version
04 Apr 2018 - E. Lundgren - Renamed from Set_Input_Opt_Extra to
                    Set_Input_Opt_Advect
```

1.18.3 Set_Input_Opt_Passive

Subroutine SET_INPUT_OPT_PASSIVE initializes all GEOS-Chem options carried in Input Options derived type object that depend on the number of passive species (Input_Opt

INTERFACE:

```
SUBROUTINE Set_Input_Opt_Passive( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REMARKS:

NOTE: These arrays are all for bpch diagnostics, and will eventually be removed from GEOS-Chem.

REVISION HISTORY:

```
04 Apr 2018 - E. Lundgren - Initial version
```

1.18.4 Cleanup_Input_Opt

Subroutine CLEANUP_INPUT_OPT deallocates all allocatable fields of the Input Options object.

INTERFACE:

```
SUBROUTINE Cleanup_Input_Opt( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)  :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC           ! Success or failure
```

REVISION HISTORY:

```

02 Nov 2012 - R. Yantosca - Initial version
07 Nov 2012 - R. Yantosca - Now deallocate fields from prod/loss menu
26 Feb 2013 - M. Long      - Now deallocate extra fields from input.geos
15 Mar 2013 - R. Yantosca - Now deallocate the LINOZ_TPARM field
17 May 2016 - R. Yantosca - Remove TRACER_N_CONST, TRACER_CONST, ID_EMITTED,
                           TRACER_COEFF
31 May 2016 - E. Lundgren - Remove TRACER_MW_G, TRACER_MW_KG, and XNUMOL
13 Jul 2016 - R. Yantosca - Remove ID_TRACER
16 Mar 2017 - R. Yantosca - Remove obsolete family & drydep fields
17 Mar 2017 - R. Yantosca - Remove IDDEP, DUSTREFF, DUSTDEN

```

2 GEOS-Chem driver and other operations

These modules contain routines to perform core operations such as initializing a GEOS-Chem simulation, timestepping, reading data from restart files, and cleanup.

2.1 Fortran: Module Interface main.F

Program GEOS_CHEM is the main level driver program for the GEOS-Chem model of atmospheric chemistry and composition.

INTERFACE:

```
PROGRAM GEOS_Chem
```

USES:

```

!-----
! Parameters to define floating-point variables
!-----
USE PRECISION_MOD,  ONLY : fpp => fp  ! Flexible precision
USE PRECISION_MOD,  ONLY : f4         ! 4-byte floating point
USE PRECISION_MOD,  ONLY : f8         ! 8-byte floating point

!-----
! Basic GEOS-Chem modules
!-----
USE CMN_SIZE_MOD      ! Size parameters
USE DiagList_Mod     ! Derived type for diagnostics list
USE Diagnostics_Mod  ! Set select netcdf diagnostics
USE ErrCode_Mod      ! Error codes for success or failure
USE ERROR_MOD        ! For error checking
USE FILE_MOD         ! For file I/O
USE GEOS_TIMERS_MOD  ! For GEOS-Chem timers (optional)
USE GC_Environment_Mod ! For allocating derived type objects
USE GC_GRID_MOD      ! For defining the lons/lats/areas of the grid
USE Input_Opt_Mod    ! Derived type for Input Options
USE INPUT_MOD        ! For reading settings from "input.geos"
USE MAPPING_MOD      ! For regridding MODIS LAI
USE OLSON_LANDMAP_MOD ! Computes IREG, ILAND, IUUSE from Olson map
USE PhysConstants    ! Physical constants
USE PRESSURE_MOD     ! For computing pressure at grid boxes
USE Grid_Registry_Mod ! Registers horizontal/vertical grid metadata
USE RESTART_MOD      ! For restart file I/O
USE State_Chm_Mod    ! Derived type for Chemistry State object
USE State_Diag_Mod   ! Derived type for Diagnostics State object
USE State_Met_Mod    ! Derived type for Meteorology State object
USE TIME_MOD         ! For computing date & time
USE UnitConv_Mod     ! For species conc unit conversions

!-----
! GEOS-Chem chemistry modules
!-----
USE CARBON_MOD       ! For SOA simulation
USE CHEMISTRY_MOD    ! Driver routines for chemistry
USE MERCURY_MOD      ! For offline Hg simulation (driver)
USE OCEAN_MERCURY_MOD ! For offline Hg simulation (ocean model)
USE STRAT_CHEM_MOD   ! For linearized stratospheric chemistry
USE TOMS_MOD         ! For overhead O3 columns (for FAST-J)
USE UCX_MOD          ! For unified trop-strat chemistry (SDE)
USE UVALBEDO_MOD     ! For reading UV albedoes (for FAST-J)
USE SET_GLOBAL_CH4_MOD ! For setting global CH4 concentrations

```

```

!-----
! GEOS-Chem deposition modules
!-----
USE DEPO_MERCURY_MOD      ! Deposition for offline Hg simulation
USE DRYDEP_MOD           ! For dry deposition
USE WETSCAV_MOD          ! For wet deposition (rainout & washout)

!-----
! GEOS-Chem diagnostics modules
!-----
#if defined( BPCH_DIAG )
  USE BENCHMARK_MOD      ! For the 1-month benchmark simulations
  USE DIAG_MOD           ! G-C diagnostic arrays & counters
  USE DIAG41_MOD         ! For ND41 (afternoon PBL      ) diag
  USE DIAG42_MOD         ! For ND42 (SOA products      ) diag
#endif
#if defined( BPCH_DIAG ) || defined( BPCH_TIMESER )
  USE CMN_DIAG_MOD       ! Logical switches for G-C diagnostics
  USE DIAG48_MOD         ! For ND48 (station timeseries ) diag
  USE DIAG49_MOD         ! For ND49 (inst. timeseries  ) diag
  USE DIAG50_MOD         ! For ND50 (24h avg timeseries ) diag
  USE DIAG51_MOD         ! For ND51 (satellite timeseries) diag
  USE DIAG51b_MOD        ! For ND51b (satellite timeseries) diag
  USE DIAG63_MOD         ! For ND63 (PARANOX timeseries ) diag
#endif
  USE DIAG_OH_MOD        ! For ND43 (OH,HO2,etc. prod  ) diag
  USE HISTORY_MOD        ! Updated netCDF diagnostics
  USE PLANEFLIGHT_MOD    ! For ND40 (plane flight track ) diag
  USE GOSAT_CH4_MOD      ! For GOSAT observation operator
  USE TCCON_CH4_MOD      ! For TCCON observation operator
  USE HCOI_GC_MAIN_MOD   ! Writes out HEMCO diagnostics (C. Keller)

!-----
! GEOS-Chem dynamics modules
!-----
USE CONVECTION_MOD      ! For deep cloud convection
USE LINOZ_MOD           ! For LINOX linear strat chemistry
USE PBL_MIX_MOD         ! To compute PBL height
USE TPCORE_BC_MOD       ! For nested-grid boundary conditions
USE TRANSPORT_MOD       ! Driver routines for advection
USE VDIFF_MOD           ! For non-local PBL mixing (J. Lin)

!-----
! GEOS-Chem emissions modules
!-----
USE EMISSIONS_MOD       ! For interfacing with HEMCO emissions
USE MIXING_MOD          ! performs tracer mixing
USE MODIS_LAI_MOD       ! For MODIS leaf area indices (replacement)

```

```

!-----
! GEOS-Chem met field I/O modules
!-----
USE DAO_MOD           ! Met field definitions
USE GEOSFP_READ_MOD   ! For reading GEOS-FP data
USE MERRA2_READ_MOD   ! For reading MERRA2 data
#if defined( EXCHANGE )
  USE EXCHANGE_MOD     ! For two-way coupled simulations
#endif

#if defined( RRTMG )
!-----
! Radiation modules
!-----
USE RRTMG_RAD_TRANSFER_MOD, ONLY: DO_RRTMG_RAD_TRANSFER,
&                               LW_UFLUX, LW_DFLUX,
&                               SW_UFLUX, SW_DFLUX,
&                               LW_UFLUXC, LW_DFLUXC,
&                               SW_UFLUXC, SW_DFLUXC,
&                               INIT_SURFACE_RAD,
&                               READ_SURFACE_RAD,
&                               INIT_STRAT_CLIM,
&                               READ_STRAT_CLIM,
&                               INIT_MCICA_CLOUDS,
&                               SET_SPECMASK
USE CMN_FJX_MOD,          ONLY: NSPECRADMENU,
&                          LSPECRADMENU
USE rrtmg_lw_init,  ONLY : rrtmg_lw_ini
USE rrtmg_sw_init,  ONLY : rrtmg_sw_ini

#endif

```

IMPLICIT NONE

REMARKS:

```

GGGGGG EEEEEEE 0000 SSSSSS      CCCCCC H   H EEEEEEE M   M
G       E       0   0 S          C       H   H E       M M M M
G  GGG EEEEEEE 0   0 SSSSSS      C       HHHHHHH EEEEEEE M   M M
G   G E       0   0 S          C       H   H E       M   M
GGGGGG EEEEEEE 0000 SSSSSS      CCCCCC H   H EEEEEEE M   M

```

(formerly known as the Harvard-GEOS model)
for 4 x 5, 2 x 2.5 global grids and hi-res nested grids

Contact: GEOS-Chem Support Team (geos-chem-support@as.harvard.edu)

See the GEOS-Chem Web Site:

<http://acmg.seas.harvard.edu/geos/>

and the GEOS-Chem User's Guide:

<http://acmg.seas.harvard.edu/geos/doc/man/>

and the GEOS-Chem wiki:

<http://wiki.seas.harvard.edu/geos-chem/>

for the most up-to-date GEOS-Chem documentation on the following topics:

- installation, compilation, and execution
- coding practice and style
- input files and met field data files
- horizontal and vertical resolution
- modification history

REVISION HISTORY:

- 13 Aug 2010 - R. Yantosca - Added ProTeX headers
- 13 Aug 2010 - R. Yantosca - Add modifications for MERRA (treat like GEOS-5)
- 19 Aug 2010 - R. Yantosca - Now call MERRA met field reader routines
- 02 Feb 2011 - S. Kim - Call Compute_OD after wet deposition
- 05 Oct 2011 - R. Yantosca - Now get SUNCOS30 array from routine COSSZA
- 07 Oct 2011 - R. Yantosca - Rename SUNCOS30 to SUNCOS_MID, which is the cos(SZA) at the midpt of the chemistry timestep
- 02 Feb 2012 - R. Yantosca - Added modifications for GEOS-5.7.x met fields
- 06 Feb 2012 - R. Yantosca - Reorganize USE statements for clarity
- 06 Feb 2012 - R. Yantosca - Renamed NN to NNN to avoid name confusion
- 07 Feb 2012 - R. Yantosca - Split off met field I/O into internal routines READ_INITIAL_MET_FIELDS and READ_MET_FIELDS
- 07 Feb 2012 - M. Payer - Replace call to COSSZA with GET_COSINE_SZA
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 06 Mar 2012 - R. Yantosca - Now call READ_TOMS every month (this was formerly done within routine "fast_j.F")
- 06 Mar 2012 - R. Yantosca - Add subroutine GET_OVERHEAD_O3_FOR_FASTJ which calls COMPUTE_OVERHEAD_O3 (in toms_mod.F) to pre-compute the overhead O3 column for FAST-J photolysis. This removes code from "set_prof.F" to facilitate the GI model.
- 19 Mar 2012 - R. Yantosca - Now call routines from olson_landmap_mod.F90 to read the Olson land map data
- 04 Apr 2012 - R. Yantosca - Now call updated LAI routines from new module modis_lai_mod.F90. Retire routine RDLAI.

05 Apr 2012 - R. Yantosca - Removed reference to LXTRA, it's obsolete
11 Apr 2012 - R. Yantosca - Replace lai_mod.F with modis_lai_mod.F90
11 Apr 2012 - R. Yantosca - Now call INIT_MODIS_LAI (in modis_lai_mod.F90)
here so that we don't have to call it from
megan_mod.F and mercury_mod.F separately.
17 Apr 2012 - R. Yantosca - Need to set the mapping variable to NULL()
10 Jun 2012 - L. Murray - Remove references to UPBDFLX_MOD.F
31 Jul 2012 - R. Yantosca - Now pass am_I_Root variable to lower-level
routines in order to allow PRINT and WRITE
statements to execute on the root CPU. This
is needed for compatibility w/ the GEOS-5 GCM.
13 Aug 2012 - R. Yantosca - Now call FILL_CHEM_STATE_IDs to populate
the CHEM_STATE object ID and name fields
18 Oct 2012 - R. Yantosca - Rename LOCAL_MET object to State_Met
18 Oct 2012 - R. Yantosca - Rename CHEM_STATE object to State_Chm
18 Oct 2012 - R. Yantosca - Now pass am_I_Root, RC arguments to routines
ALLOCATE_ALL, INIT_ALL when using -DDEVEL
19 Oct 2012 - R. Yantosca - Now reference gigc_state_chm_mod.F90
19 Oct 2012 - R. Yantosca - Now reference gigc_state_met_mod.F90
25 Oct 2012 - R. Yantosca - Define logical doDebugPrt for ND70 output
25 Oct 2012 - R. Yantosca - Add descriptive comments for DEVEL #ifdefs
25 Oct 2012 - R. Yantosca - Now reference gigc_errcode_mod.F90
01 Nov 2012 - R. Yantosca - Now read soil NOx restart file
01 Nov 2012 - R. Yantosca - Now reference gigc_input_opt_mod.F90
08 Nov 2012 - R. Yantosca - Now pass Input_Opt as an arg to DO_CHEMISTRY
01 Nov 2012 - R. Yantosca - Now read soil NOx restart file
14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
to various subroutines
15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
15 Nov 2012 - R. Yantosca - Bring Input_Opt out of the DEVEL tags
26 Feb 2013 - R. Yantosca - Add placeholder tag for Input_Opt%MAX_DEP
05 Mar 2013 - R. Yantosca - Now pass am_I_Root, Input_Opt, RC to routine
DO_PBL_MIX_2 (for non-local PBL mixing)
15 Mar 2013 - R. Yantosca - Now set Input_Opt%LINOZ_N* fields here
26 Mar 2013 - S.D. Eastham - Added initialization of rare tracers
29 Mar 2013 - R. Yantosca - Bring code out of DEVEL blocks
30 May 2013 - R. Yantosca - Now pass Input_Opt object to STDRUN routine
03 Jun 2013 - R. Yantosca - Use routines from updated mercury_mod.F
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
23 Oct 2013 - R. Yantosca - Now pass am_I_root, Input_Opt, RC to INIT_DAO
13 Dec 2013 - M. Sulprizio - Now set USE_O3_FROM_MET logical flag during
initialization stage
11 Apr 2014 - R. Yantosca - Now remove call to INIT_GLOBAL_CH4; this is
now done from routine GIGC_Init_Extra
19 May 2014 - C. Keller - Added INIT_CHEMISTRY
19 May 2014 - C. Keller - Added HEMCO
23 Jun 2014 - R. Yantosca - Removed references to unix_cmds_mod.F

23 Jun 2014 - R. Yantosca - Removed references to directory_mod.F
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
15 Jul 2014 - R. Yantosca - Now reference grid_mod.F90, regrid_a2a_mod.F90
15 Jul 2014 - R. Yantosca - Now call Init_Map_A2A to store shadow variables
within regrid_a2a_mod.F90. This helps to
break dependencies for the HEMCO implementation.
25 Jul 2014 - R. Yantosca - Remove reference to commsoil_mod.F90
22 Aug 2014 - R. Yantosca - Now save areas [m2] in State_Met%AREA_M2
15 Dec 2014 - M. Yannetti - Added PRECISION_MOD
06 Jan 2015 - R. Yantosca - Added two-way coupled simulation options
17 Feb 2015 - E. Lundgren - Remove STT and TCVV pointers
25 Feb 2015 - E. Lundgren - Remove MAKE-RH call since now in AIRQNT
16 Mar 2015 - E. Lundgren - Change tracer main units from kg to kg/kg
24 Mar 2015 - E. Lundgren - Now pass Input_Opt to Check_STT
31 Mar 2015 - E. Lundgren - Move post-transport AIRQNT call to transport_mod
16 Apr 2015 - R. Yantosca - Remove call to INIT_DAO; it's obsolete
12 Aug 2015 - R. Yantosca - Add support for MERRA2 meteorology
03 Sep 2015 - R. Yantosca - Now call SETUP_WETSCAV instead of INIT_WETSCAV
25 Jan 2016 - R. Yantosca - Call LINE_BUFFER to force PGI compiler to flush
STDOUT (unit=6) output to disk during a run
03 Feb 2016 - E. Lundgren - Use routine MAKE_RESTART_FILES for all GC rsts
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
02 May 2016 - R. Yantosca - Now pass Input_Opt to cleanup_gigc_state_chm
18 May 2016 - M. Sulprizio- Remove call to INIT_COMODE; it's obsolete
06 Jun 2016 - M. Sulprizio- Remove call to FILL_CHEM_STATE_IDS; this routine
was made obsolete by the species database
22 Jun 2016 - R. Yantosca - Add error checks to prevent calling UCX routines
when we are running specialty simulations
12 Jul 2016 - E. Lundgren - Remove binary punch restart file option
29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
16 Jun 2017 - M. Sulprizio- Add call for GOSAT CH4 observation operator
17 Aug 2017 - M. Sulprizio- Add call for TCCON CH4 observation operator
23 Aug 2017 - R. Yantosca - Now add at timer for the History component
24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA
30 Aug 2017 - R. Yantosca- Move the HISTORY UPDATE block before call to
SET_CURRENT_TIME, for clarity
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
now the default.
14 Sep 2017 - M. Sulprizio- Remove setting of USE_O3_FROM_MET. This is now
set in the chemistry menu of input.geos.
22 Jan 2018 - R. Yantosca - Attach error traps after every routine w/ RC

2.1.1 display_grid_and_model

Internal Subroutine DISPLAY_GRID_AND_MODEL displays the appropriate messages for the given model grid and machine type. It also prints the starting time and date (local time) of the GEOS-Chem simulation.

INTERFACE:

```
SUBROUTINE Display_Grid_and_Model()
```

REVISION HISTORY:

```
02 Dec 2003 - R. Yantosca - Initial version
13 Aug 2010 - R. Yantosca - Added ProTeX headers
13 Aug 2010 - R. Yantosca - Added extra output
02 Feb 2012 - R. Yantosca - Added output for GEOS-5.7.x met fields
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
19 Mar 2012 - R. Yantosca - Now echo info for 0.25 x 0.3125 runs
19 Mar 2012 - R. Yantosca - Now echo info if ISORROPIA is turned off
22 Oct 2012 - R. Yantosca - Now echo info if -DDEVEL is used
01 Mar 2017 - R. Yantosca - Now echo info if netCDF can do compression
16 Mar 2017 - M. Sulprizio- Print model version to log file
```

2.1.2 ctm_flush

Internal subroutine CTM_FLUSH flushes certain diagnostic file buffers to disk.

CTM_FLUSH should normally be called after each diagnostic output, so that in case the run dies, the output files from the last diagnostic timestep will not be lost.

FLUSH is an intrinsic FORTRAN subroutine and takes as input the unit number of the file to be flushed to disk.

INTERFACE:

```
SUBROUTINE CTM_Flush()
```

REVISION HISTORY:

```
31 Aug 2000 - R. Yantosca - Initial version
13 Aug 2010 - R. Yantosca - Added ProTeX headers
06 Aug 2012 - R. Yantosca - IU_BPCH is only global file LUN still needed
```

2.1.3 display_end_time

Internal subroutine DISPLAY_END_TIME prints the ending time of the GEOS-Chem simulation.

INTERFACE:

```
SUBROUTINE Display_End_Time()
```

REVISION HISTORY:

03 May 2005 - R. Yantosca - Initial version
13 Aug 2010 - R. Yantosca - Added ProTeX headers

2.1.4 read_initial_met_fields

Internal subroutine READ_INITIAL_MET_FIELDS calls the various routines to read met fields at the beginning of a GEOS-Chem simulation. This code was moved out of the main routine for clarity, due to the many `#if defined()` blocks that are required.

INTERFACE:

```
SUBROUTINE Read_Initial_Met_Fields()
```

REMARKS:

All variables used in this routine are declared above in the main program, and as such, are visible here.

Also calls the following routines:

- (1) AVGPOLE (average pressure @ poles) when I3 or I6 fields are read
- (2) LIGHTNING (lightning NOx emissions) when A3 or A6 fields are read

REVISION HISTORY:

07 Feb 2012 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
23 Oct 2013 - R. Yantosca - Now pass Input_Opt to GET_A6_FIELDS
23 Oct 2013 - R. Yantosca - Now pass Input_Opt to GET_MERRA_A3_FIELDS
24 Jun 2014 - R. Yantosca - Now pass Input_Opt to other routines
24 Jun 2014 - R. Yantosca - Cosmetic changes, line up arguments
12 Aug 2015 - R. Yantosca - Call routines for reading MERRA2 fields

2.1.5 read_met_fields

Internal subroutine READ_MET_FIELDS calls the various routines to read met fields in the main GEOS-Chem timestepping loop. This code was moved out of the main routine for clarity, due to the many `#if defined()` blocks that are required.

INTERFACE:

```
SUBROUTINE READ_MET_FIELDS()
```

REMARKS:

All variables used in this routine are declared above in the main program, and as such, are visible here.

Also calls the following routines:

- (1) AVGPOLE (average pressure @ poles) when I3 or I6 fields are read
- (2) LIGHTNING (lightning NOx emissions) when A3 or A6 fields are read

REVISION HISTORY:

07 Feb 2012 - R. Yantosca - Initial version
 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
 04 Jan 2013 - M. Payer - Call UPDATE_T_DAY for MERRA and GEOS-5.7.2 (tmf)
 23 Oct 2013 - R. Yantosca - Now pass Input_Opt to GET_A6_FIELDS
 24 Jun 2014 - R. Yantosca - Now pass Input_Opt to other routines
 24 Jun 2014 - R. Yantosca - Cosmetic changes, line up arguments
 12 Aug 2015 - R. Yantosca - Now call routines to read MERRA2 met data

2.1.6 get_overhead_o3_for_fastj

Internal subroutine GET_OVERHEAD_O3_FOR_FASTJ

INTERFACE:

```

SUBROUTINE Get_Overhead_O3_For_FastJ( am_I_Root )
INPUT ARGUMENTS:

LOGICAL, INTENT(IN ) :: am_I_Root ! Root CPU?

```

REMARKS:

This routine makes use of variables declared in above in the main program (which are visible in all sub-programs below the CONTAINS statement).

The original code was done in FAST-J routine "set_prof.F", but has been split off to facilitate development of the grid-independent model.

REVISION HISTORY:

07 Mar 2012 - R. Yantosca - Initial version
 14 Nov 2013 - R. Yantosca - For GEOS-FP, read O3 from met field files
 13 Dec 2013 - M. Sulprizio- Moved USE_O3_FROM_MET to the Input_Opt object and set in initialization stage of GEOS_CHEM
 22 Oct 2014 - C. Keller - Added am_I_Root

2.2 Fortran: Module Interface input_mod.F

Module INPUT_MOD contains routines that read the GEOS-Chem input file at the start of the run and pass the information to several other GEOS-Chem F90 modules.

INTERFACE:

```
MODULE INPUT_MOD
```

USES:

```
USE CharPak_Mod,    ONLY : MAXDIM => MAXSTRLEN
USE inquireMod,     ONLY : findFreeLUN
USE PRECISION_MOD   ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: Read_Input_File
PUBLIC  :: Do_Error_Checks
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE  :: SPLIT_ONE_LINE
PRIVATE  :: READ_SIMULATION_MENU
PRIVATE  :: READ_ADVECTED_SPECIES_MENU
PRIVATE  :: READ_AEROSOL_MENU
PRIVATE  :: READ_EMISSIONS_MENU
PRIVATE  :: READ_CHEMISTRY_MENU
PRIVATE  :: READ_RADIATION_MENU
PRIVATE  :: READ_TRANSPORT_MENU
PRIVATE  :: READ_CONVECTION_MENU
PRIVATE  :: READ_DEPOSITION_MENU
PRIVATE  :: READ_OUTPUT_MENU
PRIVATE  :: READ_DIAGNOSTIC_MENU
PRIVATE  :: SET_TINDEX
PRIVATE  :: READ_ND49_MENU
PRIVATE  :: READ_ND50_MENU
PRIVATE  :: READ_ND51_MENU
PRIVATE  :: READ_ND51b_MENU
PRIVATE  :: READ_ND63_MENU
PRIVATE  :: READ_PROD_LOSS_MENU
PRIVATE  :: READ_NESTED_GRID_MENU
PRIVATE  :: READ_BENCHMARK_MENU
PRIVATE  :: READ_CH4_MENU
PRIVATE  :: READ_PASSIVE_SPECIES_MENU
PRIVATE  :: VALIDATE_DIRECTORIES
PRIVATE  :: CHECK_DIRECTORY
PRIVATE  :: CHECK_TIME_STEPS
```

PRIVATE :: IS_LAST_DAY_GOOD

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
- (1) Now references LSOA in READ_AEROSOL_MENU (bmy, 9/28/04)
 - (2) Fixed error checks and assign LSPLIT for tagged Hg. Also now refernces LAVHRLAI from "logical_mod.f" (eck, bmy, 12/20/04)
 - (3) Updated for crystalline/aqueous aerosol tracers. Also moved routine IS_LAST_DAY_GOOD here from "main.f". Also now references "ocean_mercury_mod.f". Also now open the bpch file for output in READ_DIAGNOSTIC_MENU instead of in "main.f". (cas, sas, bmy, 2/3/05)
 - (4) Now references "diag03_mod.f" and "diag41_mod.f". Fixed minor bugs. Now references FILE_EXISTS from "file_mod.f". Updated comments. (bmy, 3/28/05)
 - (5) Now modified for GEOS-5 and GCAP met fields. Also now set LSPLIT correctly for HCN/CH3CN simulation. (swu, xyp, bmy, 6/30/05)
 - (6) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (7) Now read LMEGAN switch for MEGAN biogenics. Now read variable DATA_DIR_1x1 for 1x1 emissions files, etc. Now reference XNUMOL and XNUMOLAIR from "tracer_mod.f" (tmf, bmy, 10/25/05)
 - (8) Now read LEMEP switch for EMEP emissions (bdf, bmy, 11/1/05)
 - (9) Now added MERCURY MENU section. Also fixed bug in READ_ND48_MENU. (eck, cdh, bmy, 3/6/06)
 - (10) Now read LGFED2BB switch for GFED2 biomass emissions (bmy, 4/5/06)
 - (11) Bug fix for GCAP in IS_LAST_DAY_GOOD. Also now read LCTH, LMFLUX, LPRECON in READ_EMISSIONS_MENU. (bmy, 5/10/06)
 - (12) Updated for ND42 SOA concentration diagnostic (dkh, bmy, 5/22/06)
 - (13) Modified for future emissions (swu, bmy, 6/1/06)
 - (14) Modified for BRAVO emissions (rjp, kfb, bmy, 6/26/06)
 - (15) Remove support for GEOS-1 and GEOS-STRAT met fields. Also modified for David Streets' emissions. (bmy, 8/17/06)
 - (16) Modified for variable tropopause. Also set dimension of ND28 diag for GFED2 or default biomass burning. Now read if Time Spent in Troposphere is wanted (phs, bmy, 10/17/06)
 - (17) Now modified for OTD-LIS local redistribution. Remove references to GEOS-1 and GEOS-STRAT run dirs. (bmy, 11/5/07)
 - (18) New error traps for OTD-LIS scaling, dependent on met field type. Bug fix, create string variables for ERROR_STOP. Bug fix: use ND52 in call to SET_TINDEX in READ_DIAGNOSTIC_MENU. (ltm, bmy, 2/11/08)
 - (19) Bug fix: use (0,0) in call to INIT_TRANSFER (phs, 6/17/08)
 - (20) Minor fix in READ_TRANSPORT_MENU (cdh, bmy, 7/7/08)
 - (21) Fixed typo READ_EMISSIONS_MENU for GEOS-3 (bmy, 10/30/08)
 - (22) Set upper limit on dynamic timestep for 0.5 x 0.666 nested grids (yxw, bmy, dan, 11/6/08)
 - (23) Now read LCAC switch for CAC emissions (amv, 1/09/2008)
 - (24) Move the call to NDXX_SETUP (phs, 11/18/08)
 - (25) Minor bug fix in READ_DIAGNOSTIC_MENU (tmf, 2/10/09)

- (26) Add LMEGANMONO switch in emission menu (ccc, 3/2/09)
- (27) Add LDICARB switch in aerosol menu (ccc, tmf, 3/10/09)
- (28) Now read LCOOKE in aerosol menu (phs, 5/18/09)
- (29) Add CH4_MENU in input.geos (kjl, 8/18/09)
- (30) Corrected typos in CHECK_TIME_STEPS (bmy, 8/21/09)
- (31) Now read LLINOZ in READ_SIMULATION_MENU (dbm, bmy, 10/16/09)
- (32) Remove reference to obsolete embedded chemistry stuff (bmy, 2/25/10)
- (33) Remove depreciated lightning options (ltm, bmy, 1/24/11)
- 25 Aug 2010 - R. Yantosca - Added modifications for MERRA
- 27 Aug 2010 - R. Yantosca - Added ProTeX headers
- 16 Feb 2011 - R. Yantosca - Add modifications for APM from G. Luo
- 29 Jul 2011 - R. Yantosca - Bug fix in READ_EMISSIONS_MENU for nested NA
- 07 Sep 2011 - P. Kasibhatla - Modified to include monthly GFED3
- 17 Jan 2012 - P. Kasibhatla - Modified to include daily and 3-hourly GFED3
- 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x met data
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 30 Jul 2012 - R. Yantosca - READ_INPUT_FILE now accepts am_I_Root from both the ESMF interface and main.F
- 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
- 03 Aug 2012 - R. Yantosca - Now make IU_GEOS a global module variable so that we can define it with findFreeLun
- 02 Nov 2012 - R. Yantosca - Now pass the Input Options object to routines; this will eventually replace logical_mod, etc.
- 26 Feb 2013 - M. Long - Now make INITIALIZE_GEOS_GRID a public routine
- 04 Mar 2013 - R. Yantosca - Add routine GIGC_Init_Extra to move some init calls out of the run stage when using ESMF
- 23 Apr 2013 - R. Yantosca - For TOMAS, rename READ_MICROPHYSICS_MENU to INIT_TOMAS_MICROPHYSICS
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 17 Sep 2013 - R. Yantosca - Increase MAXDIM from 255 to 500 for more tracers
- 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP everywhere
- 26 Sep 2013 - R. Yantosca - Now read GEOS_FP_DIR from Input_Opt everywhere
- 23 Jun 2014 - R. Yantosca - Remove references to logical_mod.F
- 23 Jun 2014 - R. Yantosca - Removed INIT_INPUT routine
- 14 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 24 Nov 2014 - C. Keller - Updates on timestep handling in ESMF environment
- 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer compatible with the FlexChem implementation
- 28 Jul 2016 - M. Sulprizio- Rename subroutine READ_TRACER_MENU to READ_ADVECTED_SPECIES_MENU
- 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
- 24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA; Also remove LUNZIP, LWAIT, and TEMP_DIR options since they were only used for BPCH met
- 01 Nov 2017 - R. Yantosca - Now define MAXDIM from charpak_mod.F90

01 Nov 2017 - R. Yantosca - Now get ReadOneLine from charpak_mod.F90
 08 Nov 2017 - R. Yantosca - Retire usage of ERROR_STOP; we need to send errors all the way up to the top level
 26 Jan 2018 - M. Sulprizio- Moved Initialize_Geos_Grid and GC_Init_Extra routines to gc_environment_mod.F90
 06 Feb 2018 - E. Lundgren - Change input timestep/freq units from min to sec

2.2.1 read_input_file

Subroutine READ_INPUT_FILE is the driver program for reading the GEOS-Chem input file "input.geos" from disk.

In an ESMF environment, all time steps (chemistry, convection, emissions, dynamics) must be specified externally before calling this routine. This is done in routine GIGC_Init_Simulation (gigc_initialization_mod.F90). The time steps specified in input.geos are ignored.

INTERFACE:

```
SUBROUTINE READ_INPUT_FILE( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_SIZE_MOD
USE CHARPAK_MOD,      ONLY : ReadOneLine, STREPL
USE ErrCode_Mod
USE FILE_MOD,         ONLY : IOERROR
USE Input_Opt_Mod,   ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt  ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,      INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
 (1) Now call DO_GAMAP from "gamap_mod.f" to create "diaginfo.dat" and "tracerinfo.dat" files after all diagnostic menus have been read in
 (2) Now call NDXX_setup from this routine (phs, 11/18/08)
 (3) Now call READ_ND51b_MENU (amv, bmy, 12/18/09)
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root from main.F, so that we can get rid of duplicate code in DEVEL blocks

03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
 03 Aug 2012 - R. Yantosca - Now make IU_GEOS a global module variable
 so that we can define it with findFreeLun
 15 Oct 2012 - R. Yantosca - Add EXTERNAL_GRID, XTERNAL_FORCING to #ifdef
 16 Oct 2012 - R. Yantosca - Don't call CHECK_TIME_STEPS if we are calling
 READ_INPUT_FILE from the ESMF interface
 09 Nov 2012 - R. Yantosca - Now pass Input_Opt to lower-level routines
 06 Dec 2012 - R. Yantosca - Now call CHECK_TIME_STEPS when we are connecting
 to the GEOS-5 GCM via the ESMF environment,
 19 Mar 2013 - R. Yantosca - When using ESMF interface to GEOS-5, append
 ".rc" to input.geos (instead of ___rc)
 04 Apr 2013 - R. Yantosca - Now pass objects to DO_GAMAP routine
 23 Jun 2014 - R. Yantosca - Now do not call INIT_INPUT, this is replaced
 by the INIT_GIGC_INPUT_OPT routine
 15 Apr 2015 - R. Yantosca - Also define Input_Opt%ITS_A_NESTED_GRID here
 so that we can pass it to INITIALIZE_GEOS_GRID
 04 Aug 2015 - M. Long - Removed ".rc" file specifier. Not necessary.
 16 Dec 2015 - R. Yantosca - Now pass State_Chm to READ_TRACER_MENU
 04 Sep 2015 - C. Keller - Added passive tracer menu
 16 Jun 2016 - E. Lundgren - Move TOMAS init to GIGC_INIT_EXTRA
 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.2 split_one_line

Subroutine SPLIT_ONE_LINE reads a line from the input file (via routine READ_ONE_LINE), and separates it into substrings.

SPLIT_ONE_LINE also checks to see if the number of substrings found is equal to the number of substrings that we expected to find. However, if you don't know a-priori how many substrings to expect a-priori, you can skip the error check.

INTERFACE:

```

SUBROUTINE SPLIT_ONE_LINE( SUBSTRS, N_SUBSTRS,
&                          N_EXP,   LOCATION,  RC )

```

USES:

```

USE Charpak_Mod, ONLY: ReadOneLine, StrSplit
USE ErrCode_Mod
USE File_Mod,     ONLY: IoError

```

INPUT PARAMETERS:

```

INTEGER,                INTENT(IN)  :: N_EXP      ! Expected # of substrs
CHARACTER(LEN=*),       INTENT(IN)  :: LOCATION   ! Name of calling routine

```

OUTPUT PARAMETERS:

```

! Array of substrings (separated by " ")
CHARACTER(LEN=255), INTENT(OUT) :: SUBSTRS(MAXDIM) ! Substrings
INTEGER,           INTENT(OUT) :: N_SUBSTRS      ! # of substrings
INTEGER,           INTENT(OUT) :: RC             ! Success/failure?

```

REVISION HISTORY:

```

20 Jul 2004 - R. Yantosca - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
17 Sep 2013 - R. Yantosca - Extend LINE to 500 chars to allow more tracers
8 Nov 2017 - R. Yantosca - Now return error conditon to calling routine

```

2.2.3 read_simulation_menu

Subroutine READ_SIMULATION_MENU reads the SIMULATION MENU section of the GEOS-Chem input file.

INTERFACE:

```

SUBROUTINE READ_SIMULATION_MENU( am_I_Root, Input_Opt, RC )

```

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE TIME_MOD

```

INPUT PARAMETERS:

```

LOGICAL,           INTENT(IN)   :: am_I_Root   ! Is this the root CPU?

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt   ! Input options

```

OUTPUT PARAMETERS:

```

INTEGER,           INTENT(OUT)  :: RC         ! Success or failure

```

REVISION HISTORY:

```

20 Jul 2004 - R. Yantosca - Initial version
(1) Bug fix: Read LSVGLB w/ the * format and not w/ '(a)'. (bmy, 2/23/05)
(2) Now read GEOS_5_DIR and GCAP_DIR from input.geos (swu, bmy, 5/25/05)
(3) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(4) Now references DATA_DIR_1x1 for 1x1 emissions files (bmy, 10/24/05)
(5) Now read switch for using variable tropopause or not (phs, 9/14/06)
(6) Remove references to GEOS-1 and GEOS-STRAT run dirs. Now calls
    INIT_TRANSFER (bmy, 11/5/07)
(7) Fix typo in "print to screen" section (phs, 6/1/08)

```

(8) Call INIT_TRANSFER w/ (0,0) instead of (I0,J0) (phs, 6/17/08)
 (10) Now read LLINOZ switch from input.geos file (dbm, bmy, 10/16/09)
 13 Aug 2010 - R. Yantosca - Now read MERRA_DIR
 19 Aug 2010 - R. Yantosca - Set LUNZIP=F for MERRA met fields.
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 01 Feb 2012 - R. Yantosca - Now read GEOS_57_DIR for GEOS-5.7.x met
 08 Feb 2012 - R. Yantosca - Set LUNZIP=F for GEOS-5.7.x met fields
 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
 01 Mar 2012 - R. Yantosca - Now call routine INITIALIZE_GEOS_GRID to
 initialize horizontal grid parameters
 10 Jun 2012 - L. Murray - Move Linoz to chemistry menu
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 06 Dec 2012 - R. Yantosca - Now get NYMdb, NHMSb, NYMDe, NHMSe from the ESMF
 environment when connecting to the GEOS-5 GCM
 11 Dec 2012 - R. Yantosca - ACCEPT_DATE_TIME_FROM_ESMF has now been renamed
 to ACCEPT_EXTERNAL_DATE_TIME
 19 May 2014 - C. Keller - Now read HEMCO configuration file.
 23 Jun 2014 - R. Yantosca - Remove references to logical_mod.F
 12 Aug 2015 - R. Yantosca - Add support for MERRA2
 12 Aug 2015 - R. Yantosca - Add support for 05x0625 grids
 13 Aug 2015 - R. Yantosca - Bug fix: prefix RES_DIR in front of MERRA2_DIR
 27 Jul 2016 - M. Sulprizio- Remove LSVGLB and OUT_RST_FILE options. Restart
 files are now always saved out and the output
 restart file name is hardcoded in restart_mod.F.
 09 Aug 2016 - E. Lundgren - Remove call to routine set_restart; use input
 rst filename directly in restart_mod.F.
 07 Nov 2017 - R. Yantosca - Don't assign to Input_Opt%VARTROP, and will
 eventually remove the line from input.geos
 08 Nov 2017 - R. Yantosca - Return error condition to calling program
 26 Jan 2018 - M. Sulprizio- Remove obsolete options LVARTROP and reading of
 DATA_DIR_1x1

2.2.4 read_timestep_menu

Subroutine READ_TIMESTEP_MENU reads the TIMESTEP MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_TIMESTEP_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

09 Aug 2017 - M. Sulprizio- Initial version

08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.5 read_advected_species_menu

Subroutine READ_ADVECTED_SPECIES_MENU reads the ADVECTED SPECIES MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ADVECTED_SPECIES_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_SIZE_MOD
```

```
USE ErrCode_Mod
```

```
USE Input_Opt_Mod, ONLY : OptInput, Set_Input_Opt_Advect
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version

(1) Now set LSPLIT correctly for Tagged Hg simulation (eck, bmy, 12/13/04)

(2) Now initialize ocean mercury module (sas, bmy, 1/20/05)

(3) Now set LSPLIT correctly for Tagged HCN/CH3CN sim (xyp, bmy, 6/30/05)

(4) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)

(5) Now reference XNUMLAIR from "tracer_mod.f" (bmy, 10/25/05)

(6) Now move call to INIT_OCEAN_MERCURY to READ_MERCURY_MENU (bmy, 2/24/06)

(7) Now do not call SET_BIOTRCE anymore; it's obsolete (bmy, 4/5/06)
 (8) Add SET_BIOTRCE to initialize IDBxxxxs. (fp, 2/26/10)
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 13 Aug 2012 - R. Yantosca - Now pass CHEM_STATE as an argument (DEVEL only)
 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 07 Nov 2012 - R. Yantosca - Now define Input_Opt%ITS_A*_SIM fields
 28 Oct 2013 - R. Yantosca - Set Input_Opt%ITS_A_SPECIALTY_SIM = .FALSE.
 when running GEOS-Chem in an ESMF environment
 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
 21 Aug 2014 - R. Yantosca - Bug fix: add ITS_A_RnPbBe_SIM to the test
 that defines the ITS_A_SPECIALTY_SIM flag
 24 Mar 2015 - E. Lundgren - Move init_tracer to within APM ifdef block
 23 Jun 2016 - R. Yantosca - Remove call to TRACERID, it's obsolete
 28 Jul 2016 - M. Sulprizio- Rename TRACER MENU to ADVECTED SPECIES MENU;
 Remove ID, MW, and member columns from menu
 08 Nov 2017 - R. Yantosca - Return error condition to calling program
 26 Jan 2018 - M. Sulprizio- Remove line for setting N_ADVECT and instead
 count the number of advected species listed in
 this menu and set N_ADVECT accordingly

2.2.6 read_aerosol_menu

Subroutine READ_AEROSOL_MENU reads the AEROSOL MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_AEROSOL_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)  :: Input_Opt   ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REMARKS:

Move error checks that depend on species indices to the subroutine DO_ERROR_CHECKS. This is now called from GC_INIT_EXTRA, after the initialization of the species database.

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
 (1) Now reference LSOA (bmy, 9/28/04)
 (2) Now stop run if LSOA=T and SOA tracers are undefined (bmy, 11/19/04)
 (3) Now reference LCRYST from "logical_mod.f". Also now check to make prevent aerosol tracers from being undefined if the corresponding logical switch is set. (cas, bmy, 1/14/05)
 (4) Now also require LSSALT=T when LSULF=T, since we now compute the production of SO4 and NIT w/in the seasalt aerosol (bec, bmy, 4/13/05)
 (5) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 (6) Now update error check for SOG4, SOA4 (dkh, bmy, 6/1/06)
 (7) Add LDICARB switch to cancel SOG condensation onto OC aerosols. (ccc, tmf, 3/10/09)
- 27 Aug 2010 - R. Yantosca - Added ProTeX headers
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F
- 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
- 11 Apr 2013 - S.D. Eastham- Added gravitational settling flag
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust aerosols (T.D. Fairlie)
- 16 Apr 2014 - M. Sulprizio- Now read path for PSC restart file
- 23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
- 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
- 09 Apr 2015 - M. Sulprizio- Removed options for DEAD dust emissions, NAP emissions, POA emissions scale, and the PSC restart file. These options are now handled by HEMCO.
- 08 Jul 2015 - E. Lundgren - Added LMPOA option for marine organic aerosols
- 16 Jun 2016 - K. Travis - Now define species ID's with the Ind_ function
- 22 Jun 2016 - R. Yantosca - Move error checks to DO_ERROR_CHECKS routine
- 16 Jan 2018 - M. Sulprizio- Add option for using spatially and seasonally varying OM/OC from Philip et al. (2014)
- 26 Jan 2018 - M. Sulprizio- Remove obsolete options for LCRYST and LDICARB

BOC

LOCAL VARIABLES:

```

! Scalars
INTEGER                :: N, T

! Strings
CHARACTER(LEN=255)     :: ErrMsg, ThisLoc

! Arrays
CHARACTER(LEN=255)     :: SUBSTRS(MAXDIM)

```

```
!=====
! READ_AEROSOL_MENU begins here!
!=====

! Initialize
RC      = GC_SUCCESS
ErrMsg = 'Error reading the "input.geos" file!'
ThisLoc =
&      ' -> at Read_Aerosol_Menu (in module GeosCore/input_mod.F)'

! Error check
IF ( CT1 /= 2 ) THEN
    ErrMsg = 'SIMULATION MENU & ADVECTED SPECIES MENU ' //
&          'must be read in first!'
    CALL GC_Error( ErrMsg, RC, ThisLoc )
    RETURN
ENDIF

! Use online sulfate aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSULF', RC )
IF ( RC /= GC_SUCCESS ) THEN
    CALL GC_Error( ErrMsg, RC, ThisLoc )
    RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSULF

! Use metal catalyzed oxidation of SO2?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LMETALCATSO2', RC )
IF ( RC /= GC_SUCCESS ) THEN
    CALL GC_Error( ErrMsg, RC, ThisLoc )
    RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LMETALCATSO2

! Use online carbon aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LCARB', RC )
IF ( RC /= GC_SUCCESS ) THEN
    CALL GC_Error( ErrMsg, RC, ThisLoc )
    RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LCARB

! Use brown carbon aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LBRC', RC )
IF ( RC /= GC_SUCCESS ) THEN
    CALL GC_Error( ErrMsg, RC, ThisLoc )
    RETURN
```

```
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LBRC

! Use secondary organic aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSOA', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSOA

! SOAupdate: Add Semi-volatile POA switch (hotp 8/9/09)
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSVPOA', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSVPOA

! Use spatially and seasonally varying OM/OC?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LOMOC', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LOMOC

! Use online dust aerosols ?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LDUST', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LDUST

! Use SO2 and HNO3 uptake on dust aerosols
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LDSTUP', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LDSTUP

! Use online sea-salt aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSSALT', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
```



```
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSSALT

! Accum mode seasalt radii bin edges [um]
CALL SPLIT_ONE_LINE( SUBSTRS, N, 2, 'SALA_REDGE_um', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
DO T = 1, N
  READ( SUBSTRS(T), * ) Input_Opt%SALA_REDGE_um(T)
ENDDO

! Coarse mode seasalt radii bin edges [um]
CALL SPLIT_ONE_LINE( SUBSTRS, N, 2, 'SALC_REDGE_um', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
DO T = 1, N
  READ( SUBSTRS(T), * ) Input_Opt%SALC_REDGE_um(T)
ENDDO

! Use marine organic aerosols?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LMPOA', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LMPOA

! Apply gravitational settling in stratosphere?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LGRAVSTRAT', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LGRAVSTRAT

! Use solid polar stratospheric clouds (PSCs)?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSOLIDPSC', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSOLIDPSC

! Allow homogeneous nucleation of NAT?
```

```
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LHOMNUCNAT', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LHOMNUCNAT

! NAT supercooling requirement (K)
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'T_NAT_SUPERCOOL', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%T_NAT_SUPERCOOL

! Ice supersaturation ratio requirement
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'P_ICE_SUPERSAT', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%P_ICE_SUPERSAT

! Perform PSC-related heterogeneous chemistry?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LPSCCHEM', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LPSCCHEM

! Include stratospheric aerosols optical depths?
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LSTRATOD', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LSTRATOD

! Include BC absorption enhancement due to coating? (xnw, 8/24/15)
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'LBCAE', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%LBCAE

! Define BC absorption enhancement (xnw, 8/24/15)
```

```

CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'BCAE_1', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%BCAE_1

! Define BC absorption enhancement (xnw, 8/24/15)
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'BCAE_2', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF
READ( SUBSTRS(1:N), * ) Input_Opt%BCAE_2

! Separator line
CALL SPLIT_ONE_LINE( SUBSTRS, N, 1, 'separator 2', RC )
IF ( RC /= GC_SUCCESS ) THEN
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF

!=====
! Error checks
!=====

! Make sure that SALA, SALC bins are contiguous
IF ( Input_Opt%SALA_REEDGE_um(2) /=
&   Input_Opt%SALC_REEDGE_um(1) ) THEN
  ErrMsg = 'SALA and SALC bin edges are not contiguous!'
  CALL GC_Error( ErrMsg, RC, ThisLoc )
  RETURN
ENDIF

! Turn off switches for simulations that don't use aerosols
IF ( ( .not. Input_Opt%ITS_A_FULLCHEM_SIM ) .and.
&   ( .not. Input_Opt%ITS_AN_AEROSOL_SIM ) ) THEN
  Input_Opt%LSULF      = .FALSE.
  Input_Opt%LMETALCATSO2 = .FALSE.
  Input_Opt%LCARB      = .FALSE.
  Input_Opt%LBRC       = .FALSE.
  Input_Opt%LSOA       = .FALSE.
  Input_Opt%LDUST      = .FALSE.
  Input_Opt%LSSALT     = .FALSE.
  Input_Opt%LMPOA      = .FALSE.
  Input_Opt%LSVPOA     = .FALSE.
  Input_Opt%LOMOC      = .FALSE.
  Input_Opt%LBCAE      = .FALSE.

```

```

ENDIF

! Return success
RC = GC_SUCCESS

!=====
! Print to screen
!=====
IF ( am_I_Root ) THEN
  WRITE( 6, '(/,a)' ) 'AEROSOL MENU'
  WRITE( 6, '( a)' ) '-----'
  WRITE( 6, 100 ) 'Online SULFATE AEROSOLS? : ',
& Input_Opt%LSULF
  WRITE( 6, 100 ) 'Metal catalyzed SO2 ox.? : ',
& Input_Opt%LMETALCATSO2
  WRITE( 6, 100 ) 'Online CARBON AEROSOLS? : ',
& Input_Opt%LCARB
  WRITE( 6, 100 ) 'Brown Carbon Aerosol? : ',
& Input_Opt%LBRC
  WRITE( 6, 100 ) 'Online COMPLEX SOA? : ',
& Input_Opt%LSOA
  WRITE( 6, 100 ) 'Semivolatile POA? : ',
& Input_Opt%LSVPOA
  WRITE( 6, 100 ) 'Spatial & seasonal OM/OC? : ',
& Input_Opt%LOMOC
  WRITE( 6, 100 ) 'Online DUST AEROSOLS? : ',
& Input_Opt%LDUST
  WRITE( 6, 100 ) 'Acid uptake on dust? : ',
& Input_Opt%LDSTUP
  WRITE( 6, 100 ) 'Online SEA SALT AEROSOLS? : ',
& Input_Opt%LSSALT
  WRITE( 6, 110 ) 'Accum SEA SALT radii [um] : ',
& Input_Opt%SALA_REDEGE_um(1),
& Input_Opt%SALA_REDEGE_um(2)
  WRITE( 6, 110 ) 'Coarse SEA SALT radii [um] : ',
& Input_Opt%SALC_REDEGE_um(1),
& Input_Opt%SALC_REDEGE_um(2)
  WRITE( 6, 100 ) 'MARINE ORGANIC AEROSOLS? : ',
& Input_Opt%LMPOA
  WRITE( 6, 100 ) 'Settle strat. aerosols? : ',
& Input_Opt%LGRAVSTRAT
  WRITE( 6, 100 ) 'Online SOLID PSC aerosols? : ',
& Input_Opt%LSOLIDPSC
  WRITE( 6, 100 ) 'Allow hom. NAT nucleation? : ',
& Input_Opt%LHOMNUCNAT
  WRITE( 6, 120 ) 'NAT supercooling requirement: ',
& Input_Opt%T_NAT_SUPERCOOL
  WRITE( 6, 120 ) 'Ice supersaturation req. : ',

```

```

&          ((Input_Opt%P_ICE_SUPER SAT-1)*1.e+2_fp)
WRITE( 6, 100 ) 'Perform PSC het. chemistry? : ',
&          Input_Opt%LPSCCHEM
WRITE( 6, 100 ) 'Use strat. aerosol OD?      : ',
&          Input_Opt%LSTRATOD
WRITE( 6, 100 ) 'BC Absorption Enhancement? : ',
&          Input_Opt%LBCAE
!for BC absorption enhancement, (xnw, 8/24/15)
WRITE( 6, 105 ) 'Hydrophilic BC AE factor   : ',
&          Input_Opt%BCAE_1
WRITE( 6, 105 ) 'Hydrophobic BC AE factor   : ',
&          Input_Opt%BCAE_2
ENDIF

```

```

100 FORMAT( A, L5      )
105 FORMAT( A, f6.2 )
110 FORMAT( A, f6.2, ' - ', f6.2 )
120 FORMAT( A, f6.2, 'K' )

```

```
END SUBROUTINE READ_AEROSOL_MENU
```

```
EOC
```

```
-----
GEOS-Chem Global Chemical Transport Model      !
-----
```

```
%////////////////////////////////////
```

```
\mbox{}\hrulefill\
```

```
\subsection [read\_emissions\_menu] {read\_emissions\_menu}
```

```
Subroutine READ\_EMISSIONS\_MENU reads the EMISSIONS MENU
section of the GEOS-Chem input file.
```

```
\\
```

```
\\{\bf INTERFACE:}
```

```
\begin{verbatim}      SUBROUTINE READ\_EMISSIONS\_MENU( am_I_Root, Input_Opt, RC )
```

```
USES:
```

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE TIME_MOD,      ONLY : GET_YEAR

```

```
INPUT PARAMETERS:
```

```
LOGICAL,          INTENT(IN)      :: am_I_Root    ! Is this the root CPU?
```

```
INPUT/OUTPUT PARAMETERS:
```

```
TYPE(OptInput),  INTENT(INOUT)    :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: RC ! Success or failure

REMARKS:

The Ind_() function now defines all species ID's. It returns -1 if a species cannot be found. Therefore now test for Ind_() > 0 for a valid species.

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
- (1) Now read LNEI99 -- switch for EPA/NEI99 emissions (bmy, 11/5/04)
 - (2) Now read LAVHRRRLAI-switch for using AVHRR-derived LAI (bmy, 12/20/04)
 - (3) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (4) Now read LMEGAN -- switch for MEGAN biogenics (tmf, bmy, 10/20/05)
 - (5) Now read LEMEP -- switch for EMEP emissions (bdf, bmy, 11/1/05)
 - (6) Now read LGFED2BB -- switch for GFED2 biomass emissions (bmy, 4/5/06)
 - (7) Now read LOTDLIS, LCTH, LMFLUX, LPRECON for lightning options (bmy, 5/10/06)
 - (8) Now read LBRAVO for BRAVO Mexican emissions (rjp, kfb, bmy, 6/26/06)
 - (9) Now read LEDGAR for EDGAR emissions (avd, bmy, 7/11/06)
 - (10) Now read LSTREETS for David Streets' emissions (bmy, 8/17/06)
 - (11) Kludge: Reset LMFLUX or LPRECON to LCTH, as the MFLUX and PRECON lightning schemes have not yet been implemented. Rename LOTDLIS to LOTDREG. Also read LOTDLOC for the OTD-LIS local redistribution of lightning flashes (cf B. Sauvage). Make sure LOTDREG and LOTDLOC are not both turned on at the same time. (bmy, 1/31/07)
 - (12) Add LOTDSCALE to the list of LNOx options (ltm, bmy, 9/24/07)
 - (13) Add new error traps for OTD-LIS options, dependent on met field type (ltm, bmy, 11/29/07)
 - (14) Bug fix, create string variables for ERROR_STOP (bmy, 1/24/08)
 - (15) Now read LCAC for CAC emissions (amv, 1/09/2008)
 - (16) Now read LEDGARSHIP, LARCSHIP and LEMEPSHIP (phs, 12/5/08)
 - (17) Fixed typo in message for GEOS-3 (bmy, 10/30/08)
 - (18) Now read LVISTAS (amv, 12/2/08)
 - (19) Now read L8DAYBB, L3HRBB and LSYNOPBB for GFED2 8-days and 3hr emissions, and LICARTT for corrected EPA (phs, yc, 12/17/08)
 - (20) Add a specific switch for MEGAN emissions for monoterpenes and MBO (ccc, 2/2/09)
 - (21) Now read LICOADSSHIP (cklee, 6/30/09)
 - (22) Bug fix: for now, if LEMEPSHIP is turned on but LEMEP is turned off, just turn off LEMEPSHIP and print a warning msg. (mak, bmy, 10/18/09)
 - (23) Now accounts for NEI2005 (amv, phs, 10/9/09)
 - (24) Included optional flag for using MODIS LAI data (mpb,2009).
 - (25) Included optional flag for using PCEEA model (mpb, 2009)
 - (26) Now force settings for EU, NA, CC nested grids (amv, bmy, 12/18/09)
 - (27) Now force MEGAN to use MODIS LAI (ccarouge, bmy, 2/24/10)

- (28) Add separate switch for NOx fertilizer. (fp, 2/29/10)
- (29) Add scaling for isoprene and NOx emissions. (fp, 2/29/10)
- (30) Remove depreciated lightning options. (ltm, 1/25,11)
- 27 Aug 2010 - R. Yantosca - Added ProTeX headers
- 27 Aug 2010 - R. Yantosca - Added warning msg for MERRA
- 29 Jul 2011 - L. Zhang - Fix bug that turns off CAC/BRAVO emissions inadvertently during nested NA simulations
- 11 Aug 2011 - E. Leibensperger - Added flag for historical emissions and base year
- 07 Sep 2011 - P. Kasibhatla - Add modifications for GFED3
- 14 Feb 2012 - R. Yantosca - Reorganize error checks for logical switches
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 05 Apr 2012 - R. Yantosca - Now call INIT_MODIS_LAI
- 05 Apr 2012 - R. Yantosca - Reorganized USE statements for clarity
- 10 Apr 2012 - R. Yantosca - Bug fix: do not turn off LAVHRLAI or LMODISLAI when emissions are turned off. LAI is used in other areas of the code.
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F
- 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
- 22 Jul 2013 - M. Sulprizio- Now copy LRCP, LRCPSHIP, LRCPAIR to Input_Opt
- 31 Jul 2013 - M. Sulprizio- Now copy LAEIC to Input_Opt; Add check to make sure LAEIC and LRCPAIR are not both on
- 22 Aug 2013 - R. Yantosca - Now read path for soil NOx restart file
- 26 Sep 2013 - R. Yantosca - Remove SEAC4RS C-preprocessor switch
- 03 Oct 2013 - M. Sulprizio- Removed obsolete options for LAVHRLAI and LMODISLAI. MODIS LAI data are now read from netCDF files.
- 03 Jun 2014 - R. Yantosca - Radically simplify this routine since most emissions options are now specified in HEMCO
- 23 Jun 2014 - R. Yantosca - Remove reference to logical_mod.F
- 25 Jun 2014 - R. Yantosca - Move call to INIT_MODIS_LAI to GIGC_INIT_EXTRA
- 04 Sep 2014 - R. Yantosca - Bug fix: Now use Input_Opt%OTDLOC since we have now removed logical_mod.F
- 24 Jun 2015 - R. Yantosca- Now always turn off emissions for mass cons test
- 16 Jun 2016 - K. Travis - Now define species ID's with the Ind_ function
- 22 Jun 2016 - R. Yantosca - Move some error checks to DO_ERROR_CHECKS
- 20 Sep 2016 - R. Yantosca - Use "I4" format to read in CFCYEAR
- 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.7 read_co_sim_menu

Subroutine READ_CO_SIM_MENU reads the CO SIM MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_CO_SIM_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_03_MOD
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,          ONLY : ERROR_STOP
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT)  :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

```
24 Mar 2017 - J. Fisher - Initial version
```

2.2.8 read_co2_sim_menu

Subroutine READ_CO2_SIM_MENU reads the CO2 SIM MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_CO2_SIM_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT)  :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```


REVISION HISTORY:

02 Mar 2009 - R. Nassar - Initial version
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 07 Sep 2011 - P. Kasibhatla - Modified for GFED3
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 03 Jun 2014 - R. Yantosca - Now specify biomass, biofuel options in HEMCO
 23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
 13 Apr 2015 - R. Nassar - Simplified CO2 menu since options are now in HEMCO
 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.9 read_chemistry_menu

Subroutine READ_CHEMISTRY_MENU reads the CHEMISTRY MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_CHEMISTRY_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
 (1) added optional test on KPTRACER (phs, 6/17/09)
 (2) Remove reference to obsolete embedded chemistry stuff in "CMN"
 (bmy, 2/25/10)
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 10 Jun 2012 - L. Murray - Move all strat chemistry switches here
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F

01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 06 Dec 2012 - R. Yantosca - Now get TS_CHEM from the ESMF environment
 when we are connecting to the GEOS-5 GCM
 11 Dec 2012 - R. Yantosca - ACCEPT_DATE_TIME_FROM_ESMF has now been renamed
 to ACCEPT_EXTERNAL_DATE_TIME
 22 May 2013 - M. Payer - Now read in GAMMA_HO2. Recommended value is 0.2
 based on Jacon et al (2000) and Mao et al (2013).
 22 Aug 2013 - R. Yantosca - Now read in path for species restart file
 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
 27 Jul 2016 - M. Sulprizio- Remove LSVCSPEC and SPEC_RST_FILE. Restart files
 are now always saved out and the output restart
 file name is hardcoded in restart_mod.F.
 14 Sep 2017 - M. Sulprizio- Add all options for overhead O3 used in FAST-JX
 (USE_ONLINE_O3, USE_O3_FROM_MET, & USE_TOMS_O3)
 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.10 read_radiation_menu

Subroutine READ_RADIATION_MENU reads the RADIATION MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_RADIATION_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_FJX_MOD
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)      :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,      INTENT(OUT)     :: RC           ! Success or failure
```

REVISION HISTORY:

18 Jun 2013 - D. Ridley - Initial version
 03 Dec 2014 - M. Sulprizio- Now save fields to the Input_Opt object
 10 Dec 2014 - M. Sulprizio- Add error checks for RRTMG switches
 10 Mar 2017 - C. Keller - Wrapped print out in am_I_Root statement.
 08 Nov 2017 - R. Yantosca - Return error condition to calling program
 23 Mar 2018 - E. Lundgren - Store wavelength data in Input_Opt instead of
 variables formerly declared in CMN_FJX_MOD.F

2.2.11 read_transport_menu

Subroutine READ_TRANSPORT_MENU reads the TRANSPORT MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_TRANSPORT_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
 - (1) Now define MAX_DYN for 1 x 1.25 grid (bmy, 12/1/04)
 - (2) Update text in error message (bmy, 2/23/05)
 - (3) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (4) Don't stop run if TS_DYN > MAX_DYN but transport is turned off
(cdh, bmy, 7/7/08)
 - (5) Set MAX_DYN for the 0.5 x 0.666 nested grid (yxw, dan, bmy, 11/6/08)
 - 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 - 10 Jun 2012 - L. Murray - Move strat to chemistry menu
 - 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
running with the traditional driver main.F
 - 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 - 06 Dec 2012 - R. Yantosca - Now get TS_DYN from the ESMF environment, if
we are connecting to the GEOS-5 GCM
 - 11 Dec 2012 - R. Yantosca - ACCEPT_DATE_TIME_FROM_ESMF has now been renamed
to ACCEPT_EXTERNAL_DATE_TIME
 - 03 Oct 2013 - M. Sulprizio- Removed obsolete option for flux correction. This
was used for GEOS-3, which has been retired.
 - 23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
 - 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
 - 20 Sep 2016 - R. Yantosca - Use "I8" format for write statement
 - 18 Apr 2017 - C. Holmes - Fix incorrect format statement in error msg
 - 08 Nov 2017 - R. Yantosca - Return error condition to calling program
-

2.2.12 read_convection_menu

Subroutine READ_CONVECTION_MENU reads the CONVECTION MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_CONVECTION_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)      :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt   ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,      INTENT(OUT)     :: RC           ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
( 1 ) Add option for new non-local PBL scheme. And a check on GEOS-5,
      LNL PBL turned to false if GEOS-5 is not used (lin, ccc 5/13/09)
27 Aug 2010 - R. Yantosca - Now allow non-local PBL for MERRA met data
27 Aug 2010 - R. Yantosca - Added ProTeX headers
02 Feb 2012 - R. Yantosca - Added modifications for MERRA met data
13 Apr 2012 - R. Yantosca - Fixed typo ( defined( GEOS_FP ) should have
                          been !defined( GEOS_FP ) )
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                          running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
01 Mar 2013 - R. Yantosca - Now set TS_CONV to the same value as TS_DYN
                          when connecting to the GEOS-5 GCM.
12 Aug 2015 - R. Yantosca - Now allow non-local PBL mixing for MERRA2 met
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.13 read_deposition_menu

Subroutine READ_DEPOSITION_MENU reads the DEPOSITION MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_DEPOSITION_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
(1 ) Now print an informational message for tagged Hg (bmy, 12/15/04)
(2 ) We need to call WETDEPID for both wetdep and cloud convection
      since this sets up the list of soluble tracers (bmy, 3/1/05)
(3 ) Remove references to obsolete CO_OH simulation (bmy, 6/24/05)
(4 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
16 Feb 2011 - R. Yantosca - Add modifications for APM from G. Luo
31 Jul 2012 - R. Yantosca - Now pass am_I_Root to INIT_DRYDEP
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                          running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
26 Feb 2013 - R. Yantosca - Now call INIT_DUST jere to facilitate
                          connecting to the GEOS-5 GCM
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
25 Jun 2014 - R. Yantosca - Now set USE_OLSON_2001 = T for GEOS-FP met;
                          this was originally done in INIT_INPUT
05 Mar 2015 - C. Keller   - Added PBL_DRYDEP to define if dry
                          deposition rates are calculated over
                          full PBL or 1st model level only.
                          Set to .TRUE. (full PBL) if full PBL
                          mixing is used, to false otherwise.
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
                          now the default.
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.14 read_gamap_menu

Subroutine READ_GAMAP_MENU reads the GAMAP MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_GAMAP_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

```
25 Apr 2005 - R. Yantosca - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
```

2.2.15 read_output_menu

Subroutine READ_OUTPUT_MENU reads the OUTPUT MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_OUTPUT_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC          ! Success or failure
```

REMARKS:

NJDAY is now located in CMN_SIZE_mod.F, so that we can eventually retire the obsolete CMN_DIAG_mod.F. (bmy, 1/16/18)

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
03 Aug 2012 - R. Yantosca - IU_GEOS is now a global module variable
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
28 Feb 2013 - R. Yantosca - Don't call IS_LAST_DAY_GOOD when using ESMF
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.16 read_diagnostic_menu

Subroutine READ_DIAGNOSTIC_MENU reads the DIAGNOSTIC MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_DIAGNOSTIC_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_SIZE_MOD          ! Size parameters
USE CMN_DIAG_MOD          ! Needed for timeseries diags (binary only)
#if defined( BPCH_DIAG )
  USE BPCH2_MOD,          ONLY : OPEN_BPCH2_FOR_WRITE
  USE DIAG03_MOD,         ONLY : ND03,          PD03,          PD03_PL
  USE DIAG04_MOD,         ONLY : ND04,          PD04
  USE DIAG41_MOD,         ONLY : ND41,          PD41
  USE DIAG42_MOD,         ONLY : ND42,          PD42
  USE DIAG53_MOD,         ONLY : ND53,          PD53
  USE DIAG56_MOD,         ONLY : ND56,          PD56
#endif
USE DRYDEP_MOD,          ONLY : NUMDEP
USE ErrCode_Mod
USE FILE_MOD,           ONLY : IU_BPCH
USE Input_Opt_Mod,      ONLY : OptInput
USE TIME_MOD,           ONLY : GET_NYMDb, GET_NHMSb, EXPAND_DATE
```

INPUT PARAMETERS:

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?

INPUT/OUTPUT PARAMETERS:

TYPE(Optional), INTENT(INOUT) :: Input_Opt ! Input options

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: RC ! Success or failure

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
- (1) Now reference IU_BPCH from "file_mod.f" and OPEN_BPCH2_FOR_WRITE from "bpch2_mod.f". Now opens the bpch file for output here instead of w/in "main.f" (bmy, 2/3/05)
 - (2) Now references "diag03_mod.f" and "diag41_mod.f". Now turn off ND38 when both LWETD=F and LCONV=F. Now calls EXPAND_DATE to replace YYYYMMDD and HHMMSS tokens in the bpch file name with the actual starting date & time of the run. (bmy, 3/25/05)
 - (3) Now get diag info for ND09 for HCN/CH3CN sim (bmy, 6/27/05)
 - (4) Now references "diag04_mod.f" (bmy, 7/26/05)
 - (5) Now make sure all USE statements are USE, ONLY. Also remove reference to DIAG_MOD, it's not needed. (bmy, 10/3/05)
 - (6) Now remove reference to NBIOTRCE; Replace w/ NBIOMAX. (bmy, 4/5/06)
 - (7) Now reference ND56, PD56, INIT_DIAG56 from "diag56_mod.f" (bmy, 5/10/06)
 - (8) Now reference ND42, PD42, INIT_DIAG42 from "diag42_mod.f" (dkh, bmy, 5/22/06)
 - (9) Now set max dimension for GFED2 or default biomass (bmy, 9/22/06)
 - (10) Bug fix: Should use ND52 in call to SET_TINDEX (cdh, bmy, 2/11/08)
 - (11) Remove call to NDXX_SETUP; this is now called in READ_INPUT_FILE. (phs, 11/18/08)
 - (12) Now set TINDEX with PD45=NNPAR+1 tracers instead of N_TRACERS. (tmf, 2/10/09)
 - (13) NBIOMAX now in CMN_SIZE (fp, 6/2009)
- 27 Aug 2010 - R. Yantosca - Added ProTeX headers
- 26 May 2011 - R. Yantosca - For ND17, ND18, ND37, ND38, ND39, we need to set N_TMP = N_TRACERS, or else wetdep tracers with indices higher than #32 won't print out.
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F
- 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
- 06 Mar 2013 - H. Amos - merge C. Friedman's POP code
- 08 Nov 2013 - M. Sulprizio- Remove HR1_NO, and HR2_NO from ND43 diagnostic.
- 03 Feb 2014 - R. Yantosca - Remove references to TINDEX, TCOUNT, and TMAX from diag_mod. They are in CMN_SIZE_mod.F. Lee Murray reports this causes the compilation to choke on Macintosh platforms.


```

23 Jun 2014 - R. Yantosca - Now pass Input_Opt, RC to INIT_DIAG_OH
24 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
15 Dec 2014 - M. Sulprizio- Moved radiation diagnostic from ND71 to ND72 to
                    avoid conflicts with hourly max ppbv diagnostic.
15 Jan 2015 - R. Yantosca - Now define Input_Opt%DIAG_COLLECTION
22 May 2015 - R. Yantosca - Remove variables made obsolete by HEMCO
23 Jun 2016 - R. Yantosca - Now call INIT_DIAG_OH from GIGC_Init_Extra
20 Jul 2016 - R. Yantosca - Remove references to NNPAR
08 Nov 2017 - R. Yantosca - Return error condition to calling program

```

2.2.17 set_tindex

Subroutine SET_TINDEX sets the TINDEX and TMAX arrays, which determine how many tracers to print to the punch file.

INTERFACE:

```

SUBROUTINE SET_TINDEX( am_I_Root,
&                    N_DIAG, L_DIAG, SUBSTRS, N, NMAX )

```

USES:

```

#if defined( BPCH_DIAG )
#if defined( TOMAS )
USE CHARPAK_MOD, ONLY : TXTEXT    ! (win, 7/14/09)
#endif
USE CMN_DIAG_MOD           ! TMAX, TINDEX
USE CMN_SIZE_MOD           ! Size parameters
#endif

```

INPUT PARAMETERS:

```

INTEGER,      INTENT(IN) :: N_DIAG    ! GEOS-Chem diagnostic #
INTEGER,      INTENT(IN) :: N         ! # of valid substrs passed
INTEGER,      INTENT(IN) :: NMAX     ! Max # of tracers allowed
INTEGER,      INTENT(IN) :: L_DIAG   ! # of levels to save
CHARACTER(LEN=255), INTENT(IN) :: SUBSTRS(N) ! Substrs passed from
                                                ! READ_DIAGNOSTIC_MENU
LOGICAL,      INTENT(IN) :: am_I_Root ! Is this the root CPU?

```

REMARKS:

NOTE: This routine converts to a stub when BPCH_DIAG=n (bmy, 1/16/18)

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
 (1) Bug fix: now do not drop the last tracer number if "all" is not
 explicitly specified (tmf, bmy, 11/15/04)
 27 Aug 2010 - R. Yantosca - Added ProTeX headers

2.2.18 read_planeflight_menu

Subroutine READ_PLANEFLIGHT_MENU reads the PLANEFLIGHT MENU section of the GEOS-Chem input file. This turns on the ND40 flight track diagnostic.

INTERFACE:

```
SUBROUTINE READ_PLANEFLIGHT_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_DIAG_MOD           ! ND40
USE CMN_SIZE_MOD           ! MAXFAM
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE PLANEFLIGHT_MOD, ONLY : SET_PLANEFLIGHT
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)  :: Input_Opt   ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.19 read_nd48_menu

Subroutine READ_ND48_MENU reads the ND48 MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ND48_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
(1 ) Bug fix: ND48 stations should now be read correctly. (bmy, 3/6/06)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.20 read_nd49_menu

Subroutine READ_ND49_MENU reads the ND49 MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ND49_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)  :: RC          ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.21 read_nd50_menu

Subroutine READ_ND50_MENU reads the ND50 MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ND50_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)   :: am_I_Root   ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt   ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)  :: RC          ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
(1 ) Now include option to save ND51 diagnostic to HDF5 file format
      (amv, bmy, 12/21/09)
(2 ) Increase tracer number to 121. (ccc, 4/20/10)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.22 read_nd51_menu

Subroutine READ_ND51_MENU reads the ND51 MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ND51_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
(1 ) Now include option to save ND51 diagnostic to HDF5 file format
      (amv, bmy, 12/21/09)
(2 ) Increase # of tracers to 121 (ccc, 4/20/10)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                          running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.23 read_nd51b_menu

Subroutine READ_ND51b_MENU reads the ND51 MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_ND51b_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

```
21 Dec 2009 - Aaron van D - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
```

2.2.24 read_nd63_menu

Subroutine READ_ND63_MENU reads the ND63 MENU section of the GEOS-Chem input file. (gvinken, 02/25/11)

INTERFACE:

```
SUBROUTINE READ_ND63_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

25 Feb 2011 - G. Vinken - Initial version
 07 Feb 2012 - M. Payer - Added ProTeX headers
 24 Feb 2012 - M. Payer - Renamed routine from READ_ND59_MENU to
 READ_ND63 MENU. ND59 is used by TOMAS.
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
 01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
 20 Sep 2016 - R. Yantosca - Fixed typo (N_ND63 instead of ND63) when
 saving to Input_Opt%ND63_TRACERS

2.2.25 read_prod_loss_menu

Subroutine READ_PROD_LOSS_MENU reads the PROD AND LOSS MENU section of the GEOS-Chem input file

INTERFACE:

```
SUBROUTINE READ_PROD_LOSS_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
#if defined( BPCH_DIAG )
  USE CMN_DIAG_MOD
#endif
  USE ErrCode_Mod
  USE gckpp_Parameters, ONLY : NFAM
  USE gckpp_Monitor,   ONLY : FAM_NAMES
  USE Input_Opt_Mod,   ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)  :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC           ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
 (1) Bug fixes. Only error check # of prod/loss families for TagOx and
 TagCO runs if DO_SAVE_PL=T. Also turn off this diagnostic for
 the offline aerosol run. (bmy, 10/29/04)
 (2) Add error trap is P/L families are asked when using KPP. (ccc, 3/10/10)
 27 Aug 2010 - R. Yantosca - Added ProTeX headers
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when

running with the traditional driver main.F

08 Nov 2012 - R. Yantosca - Now save fields to the Input_Opt object

23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F

25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F

14 Jan 2016 - C. Keller - Now refer to Input_Opt%LKPP instead of LKPP

27 Mar 2017 - R. Yantosca - For tagCO, ND65 slots < nAdvect are L(CO)

08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.26 read_nested_grid_menu

Subroutine READ_NESTED_GRID_MENU reads the NESTED GRID MENU section of the GEOS-CHEM input file.

INTERFACE:

```
SUBROUTINE READ_NESTED_GRID_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure
```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version

(1) Now give user the option of saving out nested grid boundary conditions at 2 x 2.5 resolution for the EU, CH, or NA grids (amv, bmy, 12/18/09)

27 Aug 2010 - R. Yantosca - Added ProTeX headers

30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F

01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments

23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F

08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.27 read_benchmark_menu

Subroutine READ_BENCHMARK_MENU reads the BENCHMARK MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_BENCHMARK_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.28 read_mercury_menu

Subroutine READ_MERCURY_MENU reads the BENCHMARK MENU section of the GEOS-Chem input file.

INTERFACE:

```
SUBROUTINE READ_MERCURY_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

```
24 Feb 2006 - R. Yantosca - Initial version
( 1) Update for Chris Holmes's mercury version. (ccc, 5/6/10)
( 2) Add options to use GTMM for mercury soil emissions (ccc, 9/16/09)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
25 Apr 2016 - R. Yantosca - Now init mercury modules from GIGC_INIT_EXTRA
27 Jul 2016 - M. Sulprizio- Remove IN_HG_RST_FILE and OUT_HG_RST_FILE.
                           Hg restart fields are now stored in the GEOS-Chem
                           netCDF restart file.
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.29 read_ch4_menu

Subroutine READ_CH4_MENU reads the CH4 MENU section of the GEOS-Chem input file; this defines emissions options for CH4 tagged simulations.

INTERFACE:

```
SUBROUTINE READ_CH4_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REVISION HISTORY:

```
03 Aug 2009 - K. Wecht, C. Pickett-Heaps - Initial version
27 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
19 Feb 2014 - R. Yantosca - Add warning for CH4 budget (which is now
                           controlled by an #ifdef in global_ch4_mod.F)
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
20 Sep 2016 - R. Yantosca - Rewrote IF statement to avoid Gfortran error
16 Jun 2017 - M. Sulprizio- Remove emission options since they are now
                           handled by HEMCO
16 Jun 2017 - M. Sulprizio- Add option for GOSAT CH4 observation operator
17 Aug 2017 - M. Sulprizio- Add option for TCCON CH4 observation operator
18 Oct 2017 - M. Sulprizio- Remove CH4 budget option
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.30 read_pops_menu

Subroutine READ_POPS_MENU reads the POPS MENU section of the GEOS-Chem input file; this defines emissions options for POPs simulations.

INTERFACE:

```
SUBROUTINE READ_POPS_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)          :: am_I_Root   ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REVISION HISTORY:

01 Oct 2012 - C. Friedman - Initial version
 26 Nov 2012 - M. Payer - Added ProTeX headers
 29 Nov 2012 - M. Payer - Now pass Input_Opt, RC as arguments
 26 Mar 2013 - R. Yantosca - Now pass Input_Opt to INIT_POPS
 14 Apr 2014 - R. Yantosca - Now echo all POPs parameters to log file
 23 Jun 2014 - R. Yantosca - Now use Input_Opt%POP_EMITDIR
 25 Aug 2014 - M. Sulprizio- Move call to INIT_POPS to GIGC_Init_Extra
 27 Aug 2014 - M. Sulprizio- Remove POP_EMITDIR, emissions are now handled
 by HEMCO
 08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.31 read_passive_species_menu

Subroutine READ_PASSIVE_SPECIES_MENU reads the passive species menu section of the GEOS-Chem input file; this defines passive species to be used for this simulation.

INTERFACE:

```
SUBROUTINE READ_PASSIVE_SPECIES_MENU( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput, Set_Input_Opt_Passive
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)           :: am_I_Root   ! Is this the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt  ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC        ! Success or failure?
```

REVISION HISTORY:

04 Sep 2015 - C. Keller - Initial version
 14 Jul 2017 - E. Lundgren - Store passive species values in Input_Opt and
 remove dependency on passive_species_mod.F90
 02 Aug 2017 - R. Yantosca - Bug fix: Now use NP as the loop index over
 passive species so as to not conflict with
 the returned # of substrings
 08 Nov 2017 - R. Yantosca - Return error condition to calling program
 04 Apr 2018 - E. Lundgren - Call routine to init Input_Opt passive spc arrays

2.2.32 validate_directories

Subroutine VALIDATE_DIRECTORIES makes sure that each of the directories that we have read from the GEOS-Chem input file are valid. Also, trailing separator characters will be added.

INTERFACE:

```
SUBROUTINE VALIDATE_DIRECTORIES( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE GC_GRID_MOD,    ONLY : ITS_A_NESTED_GRID
USE TIME_MOD,       ONLY : EXPAND_DATE
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REVISION HISTORY:

```
20 Jul 2004 - R. Yantosca - Initial version
(1 ) Now make sure all USE statements are USE, ONLY.  Now also validate
      GCAP and GEOS-5 directories. (bmy, 10/3/05)
(2 ) Now references DATA_DIR_1x1 from directory_mod.f (bmy, 10/24/05)
(3 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
(4 ) Now check TPBC_DIR_NA, TPBC_DIR_CH, TPBC_DIR_EU (amv, bmy, 12/18/09)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
27 Aug 2010 - R. Yantosca - Now check MERRA directory
08 Feb 2012 - R. Yantosca - Now check GEOS-5.7.x directory
09 Feb 2012 - R. Yantosca - Rewrote USE statements for clarity
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
20 Jun 2014 - R. Yantosca - Now use fields from Input_Opt
05 Mar 2015 - R. Yantosca - Now also check CHEM_INPUTS_DIR
12 Aug 2015 - R. Yantosca - Now validate Input_Opt%MERRA2_DIR
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.33 check_directory

Subroutine CHECK_DIRECTORY makes sure that the given directory is valid. Also a trailing slash character will be added if necessary.

INTERFACE:

```
SUBROUTINE CHECK_DIRECTORY( Input_Opt, DIR, RC )
```

USES:

```
USE ErrCode_Mod
USE FILE_MOD,      ONLY : File_Exists
USE Input_Opt_Mod, ONLY : OptInput
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT) :: Input_Opt  ! Input Options object
CHARACTER(LEN=*), INTENT(INOUT) :: DIR       ! Dir to be checked
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure
```

REVISION HISTORY:

```
20 Mar 2003 - R. Yantosca - Initial version
(1 ) Now references FILE_EXISTS from "file_mod.f" (bmy, 3/23/05)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
20 Jun 2014 - R. Yantosca - Now added Input_Opt object via arg list
08 Nov 2017 - R. Yantosca - Return error condition to calling program
```

2.2.34 check_time_steps

Subroutine CHECK_TIME_STEPS computes the smallest dynamic time step for the model, based on which operation are turned on. This is called from routine READ_INPUT_FILE, after all of the timesteps and logical flags have been read from "input.geos".

INTERFACE:

```
SUBROUTINE CHECK_TIME_STEPS( am_I_Root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE TIME_MOD,      ONLY : SET_TIMESTEPS
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT) :: RC           ! Success or failure?

```

REVISION HISTORY:

```

20 Jul 2004 - R. Yantosca - Initial version
(1 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(2 ) Add TS_DIAG, the largest time steps used for diagnostics.
      And test that all time steps are multiple of the smallest one.
      (ccc, 5/13/09)
(3 ) Corrected typos -99999 instead of -999999 (phs, bmy, 8/21/09)
(4 ) Now compute TS_SUN_2 which is 1/2 of the chemistry timestep (or
      smallest timestep if LCHEM=LEMIS=LDRYD=F). This is used to compute
      SUNCOS at the midpoint of the timestep instead of the beginning.
      (bmy, 4/27/10)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
07 Oct 2011 - R. Yantosca - Add extra error checks for centralizing
                          chemistry timestep algorithm
07 Oct 2011 - R. Yantosca - Remove TS_SUN_2 from call to SET_TIMESTEPS
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                          running with the traditional driver main.F
23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F
08 Nov 2017 - R. Yantosca - Return error condition to calling program

```

2.2.35 is_last_day_good

Subroutine IS_LAST_DAY_GOOD tests to see if there is output scheduled on the last day of the run.

INTERFACE:

```

SUBROUTINE IS_LAST_DAY_GOOD( Input_Opt, RC )

```

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE JULDAY_MOD,     ONLY : JULDAY
USE TIME_MOD,       ONLY : GET_NYMDe, ITS_A_LEAPYEAR, YMD_EXTRACT

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt  ! Input options
INTEGER,          INTENT(OUT)  :: RC         ! Success or failure?

```

REVISION HISTORY:

20 Jul 2004 - R. Yantosca - Initial version
(1) Moved to "input_mod.f" from "main.f" (bmy, 1/11/05)
(2) Now call ITS_A_LEAPYEAR with FORCE=.TRUE. to always return whether
the year Y would be a leap year, regardless of met field type.
(swu, bmy, 4/24/06)
27 Aug 2010 - R. Yantosca - Added ProTeX headers
08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.2.36 do_error_checks

Makes sure that certain species are defined in order to proceed with a certain option. Halts the simulation with an error message if incorrect inputs would have caused a simulation to crash.

INTERFACE:

```
SUBROUTINE Do_Error_Checks( am_I_root, Input_Opt, RC )
```

USES:

```
USE ErrCode_Mod  
USE Input_Opt_Mod, ONLY : OptInput  
USE State_Chm_Mod, ONLY : Ind_
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)      :: am_I_Root
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput),  INTENT(INOUT)   :: Input_Opt
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)    :: RC
```

REMARKS:

These error checks were originally called when the various menus were read in from disk. However, in order to use the Ind_() function to look up given species indices, we need to call this after the Species Database (which is in State_Chm) is initialized. Therefore, we have now moved these error checks to this routine, which is now called from GC_Init_Extra.

The Ind_() function now defines all species ID's. It returns -1 if a species cannot be found. The prior behavior was to return 0 if a species wasn't found. Therefore, in order to preserve the logic of the error checks, we must force any -1's returned by Ind_() to 0's in this subroutine.

REVISION HISTORY:

22 Jun 2016 - R. Yantosca - Initial version
08 Nov 2017 - R. Yantosca - Now return error condition to calling routine

2.3 Fortran: Module Interface gc_environment_mod.F90

Module GC_ENVIRONMENT_MOD establishes the runtime environment for the GEOS-Chem. It is designed to receive model parameter and geophysical environment information and allocate memory based upon it.

It provides routines to do the following:

- Allocate geo-spatial arrays
- Initialize met. field derived type.
- Initialize Chemistry, Meteorology, Emissions, and Physics States

INTERFACE:

```
MODULE GC_Environment_Mod
  !USES
  USE Precision_Mod
```

```
  IMPLICIT NONE
  PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: GC_Allocate_All
PUBLIC  :: GC_Init_StateObj
PUBLIC  :: GC_Init_Grid
PUBLIC  :: GC_Init_Extra
PUBLIC  :: GC_Init_Regridding
```

PRIVATE MEMBER FUNCTIONS:

```
#if defined( TOMAS )
  PRIVATE :: INIT_TOMAS_MICROPHYSICS
#endif
```

REMARKS:

For consistency, we should probably move the met state initialization to the same module where the met state derived type is contained.

REVISION HISTORY:

```
26 Jan 2012 - M. Long      - Created module file
13 Aug 2012 - R. Yantosca - Added ProTeX headers
19 Oct 2012 - R. Yantosca - Removed routine INIT_LOCAL_MET, this is now
                           handled in Headers/gigc_state_met_mod.F90
22 Oct 2012 - R. Yantosca - Renamed to gigc_environment_mod.F90
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
28 Aug 2015 - R. Yantosca - Remove Get_nSchm_nSchmBry; stratospheric
                           chemistry fields are now read by HEMCO
16 Aug 2016 - M. Sulprizio- Rename from gigc_environment_mod.F90 to
                           gc_environment_mod.F90. The "gigc" nomenclature
                           is no longer used.
```

2.3.1 gc_allocate_all

Subroutine GC_ALLOCATE_ALL allocates all LAT/LON ALLOCATABLE arrays for global use by the GEOS-Chem either as a standalone program or module.

INTERFACE:

```

SUBROUTINE GC_Allocate_All( am_I_Root,      Input_Opt,      &
                           RC,             value_I_LO,      &
                           value_J_LO,    value_I_HI,      &
                           value_J_HI,    value_IM,        &
                           value_JM,      value_LM,        &
                           value_IM_WORLD, value_JM_WORLD, &
                           value_LM_WORLD )

```

USES:

```

#if defined( BPCH_DIAG )
  USE CMN_DIAG_Mod,      ONLY : Init_CMN_DIAG
#endif
  USE CMN_FJX_MOD,      ONLY : Init_CMN_FJX
  USE CMN_O3_Mod,       ONLY : Init_CMN_O3
  USE CMN_SIZE_Mod,     ONLY : Init_CMN_SIZE
  USE ErrCode_Mod
  USE Input_Opt_Mod
  USE VDIFF_PRE_Mod,    ONLY : Init_Vdiff_Pre

```

IMPLICIT NONE

INPUT PARAMETERS:

LOGICAL,	INTENT(IN)	:: am_I_Root	! Are we on the root CPU?
TYPE(OptInput),	INTENT(IN)	:: Input_Opt	! Input Options object
INTEGER,	OPTIONAL	:: value_I_LO	! Min local lon index
INTEGER,	OPTIONAL	:: value_J_LO	! Min local lat index
INTEGER,	OPTIONAL	:: value_I_HI	! Max local lon index
INTEGER,	OPTIONAL	:: value_J_HI	! Max local lat index
INTEGER,	OPTIONAL	:: value_IM	! Local # of lons
INTEGER,	OPTIONAL	:: value_JM	! Local # of lats
INTEGER,	OPTIONAL	:: value_LM	! Local # of levels
INTEGER,	OPTIONAL	:: value_IM_WORLD	! Global # of lons
INTEGER,	OPTIONAL	:: value_JM_WORLD	! Global # of lats
INTEGER,	OPTIONAL	:: value_LM_WORLD	! Global # of levels

OUTPUT PARAMETERS:

INTEGER,	INTENT(OUT)	:: RC	! Success or failure?
----------	-------------	-------	-----------------------

REMARKS:

For error checking, return up to the main routine w/ an error code.
This can be improved upon later.

REVISION HISTORY:

```

26 Jan 2012 - M. Long      - Initial version
13 Aug 2012 - R. Yantosca - Added ProTeX headers
17 Oct 2012 - R. Yantosca - Add am_I_Root, RC as arguments
22 Oct 2012 - R. Yantosca - Renamed to GIGC_Allocate_All
30 Oct 2012 - R. Yantosca - Now pass am_I_Root, RC to SET_COMMSOIL_MOD
01 Nov 2012 - R. Yantosca - Now zero the fields of the Input Options object
16 Nov 2012 - R. Yantosca - Remove this routine from the #ifdef DEVEL block
27 Nov 2012 - R. Yantosca - Now pass Input_Opt to INIT_COMODE_LOOP
03 Dec 2012 - R. Yantosca - Now pass am_I_Root, RC to INIT_CMN_SIZE
03 Dec 2012 - R. Yantosca - Add optional arguments to accept dimension
                           size information from the ESMF interface
13 Dec 2012 - R. Yantosca - Remove reference to obsolete CMN_DEP_mod.F
13 Dec 2012 - R. Yantosca - Remove reference to obsolete CMN_mod.F
23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN_NOX_mod.F
25 Jul 2014 - R. Yantosca - Remove reference to obsolete commsoil_mod.F90
25 Jul 2014 - R. Yantosca - Now call INIT_GET_NDEP (GeosCore/get_ndep_mod.F)
04 Aug 2015 - C. Keller    - Now pass LLTROP and LLSTRAT to INIT_CMN_SIZE.
17 Jun 2016 - R. Yantosca - Move call to INIT_GET_NDEP to GIGC_INIT_EXTRA
                           which is called after species database init
30 Jun 2016 - M. Sulprizio - Remove call to INIT_COMODE_LOOP; it's obsolete
20 Dec 2017 - R. Yantosca - Return when encountering errors
29 Dec 2017 - C. Keller    - Now accept value of LLSTRAT from Input_Opt
14 Mar 2018 - E. Lundgren  - Input_Opt parameter is IN only, not INOUT

```

2.3.2 gc_init_stateobj

Subroutine GC_INIT_STATEOBJ initializes the top-level data structures that are either passed to/from GC or between GC components (emis-*i*transport-*i*chem-*i*diagnostics-*i*etc)

INTERFACE:

```

SUBROUTINE GC_Init_StateObj( am_I_Root, Diag_List, Input_Opt, State_Chm, &
                             State_Diag, State_Met, RC
                             )

```

USES:

```

USE DiagList_Mod
USE Diagnostics_Mod
USE ErrCode_Mod
USE Input_Opt_Mod
USE State_Chm_Mod
USE State_Met_Mod
USE State_Diag_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
TYPE(DgnList),    INTENT(IN)      :: Diag_List    ! Diagnostics list object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input Options object
TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag   ! Diagnostics State object
TYPE(MetState), INTENT(INOUT) :: State_Met    ! Meteorology State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)    :: RC          ! Success or failure

```

REMARKS:

Need to add better error checking, currently we just return upon error.

REVISION HISTORY:

```

26 Jan 2012 - M. Long      - Initial version
13 Aug 2012 - R. Yantosca - Added ProTeX headers
16 Oct 2012 - R. Yantosca - Renamed LOCAL_MET argument to State_Met
16 Oct 2012 - R. Yantosca - Renamed GC_STATE argument to State_Chm
16 Oct 2012 - R. Yantosca - Call Init_Chemistry_State (in gc_type2_mod.F90,
                           which was renamed from INIT_CHEMSTATE)
19 Oct 2012 - R. Yantosca - Now reference gignc_state_met_mod.F90
19 Oct 2012 - R. Yantosca - Now reference gignc_state_chm_mod.F90
19 Oct 2012 - R. Yantosca - Now reference gignc_errcode_mod.F90
19 Oct 2012 - R. Yantosca - Now reference IGAS in Headers/comode_loop_mod.F
22 Oct 2012 - R. Yantosca - Renamed to GIGC_Init_All
26 Oct 2012 - R. Yantosca - Now call Get_nSchm, nSchmBry to find out the
                           number of strat chem species and Bry species
01 Nov 2012 - R. Yantosca - Now use LSCHM from logical_mod.F
09 Nov 2012 - R. Yantosca - Now pass Input Options object for GIGC
26 Feb 2013 - R. Yantosca - Now pass Input_Opt to Init_GIGC_State_Chm
28 Aug 2015 - R. Yantosca - Remove strat-chem options from call to
                           Init_GIGC_State_Chm; this is done by HEMCO
25 Jan 2016 - R. Yantosca - Bug fix: Declare Input_Opt as INTENT(INOUT),
                           to match the declaration in INIT_GIGC_STATE_CHM
28 Jan 2016 - M. Sulprizio- Remove NBIOMAX from call to Init_GIGC_State_Chm
30 Jun 2016 - M. Sulprizio- Remove nSpecies from call to Init_GIGC_State_Chm
26 Jun 2017 - R. Yantosca - Now call GC_ERROR to give better error feedback
05 Jul 2017 - R. Yantosca - Now initialize the Diagnostics State object
26 Sep 2017 - E. Lundgren - Pass diagnostics list object as argument
16 Nov 2017 - E. Lundgren - Do not pass IIPAR, JJPAR, LLPAR, NDUST, AER
                           since available from CMN_Size_Mod in routines
01 Feb 2018 - E. Lundgren - Initialize new diagnostics_mod module

```

2.3.3 gc_init_grid

Subroutine GC_INIT_GRID calls routines from gc_grid_mod.F90 to initialize the horizontal grid parameters.

INTERFACE:

```
SUBROUTINE GC_Init_Grid( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE GC_GRID_MOD,      ONLY : COMPUTE_GRID
USE GC_GRID_MOD,      ONLY : INIT_GRID
USE GC_GRID_MOD,      ONLY : SET_XOFFSET, SET_YOFFSET
USE Input_Opt_Mod,    ONLY : OptInput
USE TRANSFER_MOD,     ONLY : INIT_TRANSFER
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC           ! Success or failure?
```

REMARKS:

The module grid_mod.F90 has been modified to save grid parameters in 3D format, which will facilitate interfacing GEOS-Chem to a GCM.

The module global_grid_mod.F90 contains several of the global grid arrays (*_g) originally in grid_mod.F. These arrays are used in regridding GFED3 biomass emissions, which are available on a 0.5x0.5 global grid. The global arrays may need to be used in the future for regridding other emissions for nested grids.

REVISION HISTORY:

```
01 Mar 2012 - R. Yantosca - Initial version
01 May 2012 - M. Payer    - Add call to COMPUTE_GLOBAL_GRID for nested grids
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
                           running with the traditional driver main.F
01 Nov 2012 - R. Yantosca - Now pass Input_Opt, RC as arguments
30 Nov 2012 - R. Yantosca - Accept external DLON, DLAT from ESMF interface
26 Feb 2013 - R. Yantosca - Now pass I_LO, J_LO to COMPUTE_GRID
28 Feb 2013 - R. Yantosca - Bug fix for GEOS-5 interface: Now call
                           Compute_Grid with 1..IIPAR, I..JJPARG
01 Jul 2013 - R. Yantosca - Don't use 1/2 sized polar boxes for GCAP
25 Jun 2014 - R. Yantosca - Now accept Input_Opt via the arg list
23 Jul 2014 - R. Yantosca - Remove reference to obsolete global_grid_mod
08 Nov 2017 - R. Yantosca - Return error condition to calling program
26 Jan 2018 - M. Sulprizio - Moved to gc_environment_mod.F90 from input_mod.F
                           and renamed from Initialize_Grid to GC_Init_Grid
```

2.3.4 gc_init_extra

Subroutine GC_INIT_EXTRA initializes other GEOS-Chem modules that have not been initialized in either GC_Allocate_All or GC_Init_all.

INTERFACE:

```
SUBROUTINE GC_Init_Extra( am_I_Root, Diag_List, Input_Opt, &
                        State_Chm, State_Diag, RC          )
```

USES:

```
USE Aerosol_Mod,      ONLY : Init_Aerosol
USE Carbon_Mod,      ONLY : Init_Carbon
USE CO2_Mod,         ONLY : Init_CO2
USE C2H6_Mod,        ONLY : Init_C2H6
USE Depo_Mercury_Mod, ONLY : Init_Depo_Mercury
USE Diag03_Mod,      ONLY : Init_Diag03
USE Diag04_Mod,      ONLY : Init_Diag04
USE Diag20_Mod,      ONLY : Init_Diag20
USE Diag41_Mod,      ONLY : Init_Diag41
USE Diag42_Mod,      ONLY : Init_Diag42
USE Diag48_Mod,      ONLY : Init_Diag48
USE Diag49_Mod,      ONLY : Init_Diag49
USE Diag50_Mod,      ONLY : Init_Diag50
USE Diag51_Mod,      ONLY : Init_Diag51
USE Diag51b_Mod,     ONLY : Init_Diag51b
USE Diag53_Mod,      ONLY : Init_Diag53
USE Diag56_Mod,      ONLY : Init_Diag56
USE Diag63_Mod,      ONLY : Init_Diag63
USE Diag_OH_Mod,     ONLY : Init_Diag_OH
USE DiagList_Mod,   ONLY : DgnList
USE Drydep_Mod,     ONLY : Init_Drydep
USE Dust_Mod,       ONLY : Init_Dust
USE ErrCode_Mod
USE Error_Mod,      ONLY : Debug_Msg
USE Gamap_Mod,      ONLY : Do_Gamap
USE Get_Ndep_Mod,   ONLY : Init_Get_Ndep
USE Global_CH4_Mod, ONLY : Init_Global_CH4
USE Input_Mod,      ONLY : Do_Error_Checks
USE Input_Opt_Mod,  ONLY : OptInput
USE Linoz_Mod,      ONLY : Init_Linoz
USE Land_Mercury_Mod, ONLY : Init_Land_Mercury
USE Mercury_Mod,    ONLY : Init_Mercury
USE Modis_Lai_Mod,  ONLY : Init_Modis_Lai
USE Ocean_Mercury_Mod, ONLY : Init_Ocean_Mercury
USE POPs_Mod,       ONLY : Init_POPs
USE Seasalt_Mod,    ONLY : Init_SeaSalt
USE State_Chm_Mod,  ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
```

```

USE Sulfate_Mod,      ONLY : Init_Sulfate
USE Tagged_CO_Mod,   ONLY : Init_Tagged_CO
USE Tagged_O3_Mod,   ONLY : Init_Tagged_O3
USE Toms_Mod,        ONLY : Init_Toms
USE TPCORE_BC_Mod,   ONLY : Init_TPCORE_BC
USE Vdiff_Pre_Mod,   ONLY : Set_Vdiff_Values
USE WetScav_Mod,     ONLY : Init_WetScav

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(DgnList ), INTENT(IN)  :: Diag_List    ! Diagnostics list object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt    ! Input Options object
TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry state object
TYPE(DgnState), INTENT(INOUT) :: State_Diag   ! Diagnostics State object

```

INPUT/OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)   :: RC            ! Success or failure?

```

REMARKS:

Several of the INIT routines now called within GC_Init_Extra had originally been called from the Run method. We now gather these INIT routines here so that they may be called from the Initialization method. This is necessary when connecting GEOS-Chem to the GEOS-5 GCM via ESMF.

REVISION HISTORY:

```

04 Mar 2013 - R. Yantosca - Initial revision
05 Mar 2013 - R. Yantosca - Now call INIT_AEROSOL (GeosCore/aerosol_mod.F)
15 Mar 2013 - R. Yantosca - Now call INIT_LINOZ (GeosCore/linoz_mod.F)
29 Mar 2013 - R. Yantosca - Now call INIT_TROPOPAUSE (so that we can pass
                           a LVARTROP from Input_Opt and not logical_mod.F)
10 Apr 2014 - R. Yantosca - Now call INIT_TAGGED_CO
10 Apr 2014 - R. Yantosca - Now call INIT_TAGGED_OX and INIT_GLOBAL_CH4
11 Apr 2014 - R. Yantosca - Now call INIT_C2H6 and INIT_HCN_CH3CN
14 Apr 2014 - R. Yantosca - Also call INIT_C2H6 if it's a fullchem sim
                           since we read C2H6 emissions from c2h6_mod.F
25 Jun 2014 - R. Yantosca - Now call INIT_MODIS_LAI
25 Jun 2014 - R. Yantosca - Now call SET_VDIFF_VALUES (vdiff_pre_mod.F90)
25 Aug 2014 - M. Sulprizio - Now call INIT_POPS
16 Mar 2015 - R. Yantosca - Now call INIT_TOMS here
28 Aug 2015 - R. Yantosca - Now initialize drydep & wetdep here, so that
                           we can take advantage of the species database
03 Sep 2015 - R. Yantosca - Now call INIT_WETSCAV instead of WETDEPID
21 Sep 2015 - R. Yantosca - Now pass State_Chm to INIT_POPS
22 Sep 2015 - R. Yantosca - Bug fix: only call INIT_WETSCAV if convection,

```

```

                wetdep, or chemistry ist turned on. This
                replicates the prior behavior,
23 Sep 2015 - R. Yantosca - Now pass State_Chm to INIT_SEASALT
23 Sep 2015 - R. Yantosca - Now pass State_Chm to INIT_SULFATE
25 Apr 2016 - R. Yantosca - Now call INIT_DIAG03 here
25 Apr 2016 - R. Yantosca - Now initialize all mercury modules from here
23 Jun 2016 - R. Yantosca - Now call INIT_DIAG_OH from here
16 Aug 2016 - M. Sulprizio- Rename from GIGC_Init_Extra to GC_Init_Extra.
                The "gigc" nomenclature is no longer used.
05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument and
                pass it to the various init routines
07 Nov 2017 - R. Yantosca - Now accept Diag_List as an argument
07 Nov 2017 - R. Yantosca - Return error condition to main level
26 Jan 2018 - M. Sulprizio- Moved to gc_environment_mod.F90 from input_mod.F

```

2.3.5 gc_init_regridding

Internal subroutine GC_INIT_REGRIDDING passes several variables to regrid_a2a_mod.F90, where they are locally shadowed.

INTERFACE:

```
SUBROUTINE GC_Init_Regridding( am_I_Root, Input_Opt, RC )
```

USES:

```

USE CMN_SIZE_Mod
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE GC_Grid_Mod
USE Regrid_A2A_Mod

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt   ! Input Options object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT) :: RC           ! Success or failure?

```

REMARKS:

This routine is a wrapper for Init_Map_A2A in regrid_a2a_mod.F90. Passing variables via Init_Map_A2A helps us to remove dependencies on other GEOS-Chem routines from regrid_a2a_mod.F90. This in turn facilitates the implementation of the HEMCO emissions package.

REVISION HISTORY:

15 Jul 2014 - R. Yantosca - Initial version
 05 Mar 2015 - R. Yantosca - Now read data w/r/t ExtData/HEMCO
 26 Jan 2018 - M. Sulprizio- Moved to gc_environment_mod.F90 from main.F and
 renamed from Initialize_Regridding to
 GC_Init_Regridding

2.3.6 init_tomas_microphysics

Subroutine INIT_TOMAS_MICROPHYS will initialize the TOMAS microphysics package.
 This replaces the former subroutine READ_MICROPHYSICS_MENU.

INTERFACE:

```
SUBROUTINE INIT_TOMAS_MICROPHYSICS( am_I_Root, Input_Opt, &
                                     State_Chm, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Chm_Mod,     ONLY : Ind_
USE TOMAS_MOD,         ONLY : INIT_TOMAS
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
TYPE(ChmState), INTENT(IN)  :: State_Chm    ! Chemistry state
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(OptInput), INTENT(INOUT) :: Input_Opt  ! Input options
```

OUTPUT PARAMETERS:

```
INTEGER,      INTENT(OUT)   :: RC           ! Success or failure
```

REMARKS:

We now invoke TOMAS by compiling GEOS-Chem and setting either the TOMAS=yes (30 bins, default) or TOMAS40=yes (40 bins, optional) switches. The old LTOMAS logical switch is now obsolete because all of the TOMAS code is segregated from the rest of GEOS-Chem with #ifdef blocks. Therefore, we no longer need to read the microphysics menu, but we still need to apply some error checks and then call INIT_TOMAS. (bmy, 4/23/13)

The Ind_() function now defines all species ID's. It returns -1 if a species cannot be found. The prior behavior was to return 0 if a species wasn't found. Therefore, in order to preserve the logic of the error checks, we must force any -1's returned by Ind_() to 0's in this subroutine.

REVISION HISTORY:

- 20 Jul 2004 - R. Yantosca - Initial version
- (1) Now read LNEI99 -- switch for EPA/NEI99 emissions (bmy, 11/5/04)
 - (2) Now read LAVHRRLAI-switch for using AVHRR-derived LAI (bmy, 12/20/04)
 - (3) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (4) Now read LMEGAN -- switch for MEGAN biogenics (tmf, bmy, 10/20/05)
 - (5) Now read LEMEP -- switch for EMEP emissions (bdf, bmy, 11/1/05)
 - (6) Now read LGFED2BB -- switch for GFED2 biomass emissions (bmy, 4/5/06)
 - (7) Now read LOTDLIS, LCTH, LMFLUX, LPRECON for lightning options
(bmy, 5/10/06)
 - (8) Now read LBRAVO for BRAVO Mexican emissions (rjp, kfb, bmy, 6/26/06)
 - (9) Now read LEDGAR for EDGAR emissions (avd, bmy, 7/11/06)
 - (10) Now read LSTREETS for David Streets' emissions (bmy, 8/17/06)
 - (11) Kludge: Reset LMFLUX or LPRECON to LCTH, as the MFLUX and PRECON lightning schemes have not yet been implemented. Rename LOTDLIS to LOTDREG. Also read LOTDLOC for the OTD-LIS local redistribution of lightning flashes (cf B. Sauvage). Make sure LOTDREG and LOTDLOC are not both turned on at the same time. (bmy, 1/31/07)
 - (12) Add LOTDSCALE to the list of LNOx options (ltm, bmy, 9/24/07)
 - (13) Add new error traps for OTD-LIS options, dependent on met field type (ltm, bmy, 11/29/07)
 - (14) Bug fix, create string variables for ERROR_STOP (bmy, 1/24/08)
 - (15) Now read LCAC for CAC emissions (amv, 1/09/2008)
 - (16) Now read LEDGARSHIP, LARCSHIP and LEMEPSHIP (phs, 12/5/08)
 - (17) Fixed typo in message for GEOS-3 (bmy, 10/30/08)
 - (18) Now read LVISTAS (amv, 12/2/08)
 - (19) Now read L8DAYBB, L3HRBB and LSYNOPBB for GFED2 8-days and 3hr emissions, and LICARTT for corrected EPA (phs, yc, 12/17/08)
 - (20) Add a specific switch for MEGAN emissions for monoterpenes and MBO (ccc, 2/2/09)
 - (21) Now read LICOADSSHIP (cklee, 6/30/09)
 - (22) Bug fix: for now, if LEMEPSHIP is turned on but LEMEP is turned off, just turn off LEMEPSHIP and print a warning msg. (mak, bmy, 10/18/09)
 - (23) Now accounts for NEI2005 (amv, phs, 10/9/09)
 - (24) Included optional flag for using MODIS LAI data (mpb,2009).
 - (25) Included optional flag for using PCEEA model (mpb, 2009)
 - (26) Now force settings for EU, NA, CC nested grids (amv, bmy, 12/18/09)
 - (27) Now force MEGAN to use MODIS LAI (ccarouge, bmy, 2/24/10)
 - (28) Add separate switch for NOx fertilizer. (fp, 2/29/10)
 - (29) Add scaling for isoprene and NOx emissions. (fp, 2/29/10)
- 27 Aug 2010 - R. Yantosca - Added ProTeX headers
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F
- 23 Apr 2013 - R. Yantosca - Renamed to INIT_TOMAS_MICROPHYS
- 30 Jan 2014 - R. Yantosca - INIT_TOMAS accepts am_I_Root, Input_Opt, RC
- 16 Jun 2016 - K. Travis - Now define species ID's with the Ind_ function
- 16 Jun 2016 - E. Lundgren - INIT_TOMAS now accepts State_Chm

22 Jun 2016 - R. Yantosca - Force -1's returned by Ind_() to zeroes,
in order to preserve the program logic
08 Nov 2017 - R. Yantosca - Return error condition to calling program

2.4 Fortran: Module Interface restart_mod.F

Module RESTART_MOD contains variables and routines which are used to read and write restart files for GEOS-Chem species in units of [v/v] dry mixing ratio, which represents mol species per mol dry air.

INTERFACE:

```
MODULE RESTART_MOD
```

USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE inquireMod,        ONLY : findFreeLUN
USE PhysConstants
USE Precision_Mod
USE UnitConv_Mod,      ONLY : Convert_Spc_Units
```

```
IMPLICIT NONE
```

```
PRIVATE
```

```
!PUBLIC MEMBER FUNCTIONS
```

```
PUBLIC  :: READ_GC_RESTART
PUBLIC  :: WRITE_GC_RESTART
PUBLIC  :: INIT_GC_RESTART
PUBLIC  :: COMPARE_TRC_SPC ! temporary for tracer removal debugging
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE  :: CHECK_RST_DIMENSIONS
```

DEFINED PARAMETERS:

```
! Prefix of GEOS-Chem output restart filename
CHARACTER(LEN=255), PARAMETER :: Output_GC_Rst_Prefix =
&                                'GEOSChem_restart'
```

REVISION HISTORY:

25 Jun 2002 - R. Yantosca - Initial version
(1) Moved routines "make_restart_file.f" and "read_restart_file.f" into
this module. Also now internal routines to "read_restart_file.f"
are now a part of this module. Now reference "file_mod.f" to get
file unit numbers and error checking routines. (bmy, 6/25/02)

- (2) Now reference AD from "dao_mod.f". Now reference "error_mod.f".
Also added minor bug fix for ALPHA platform. (bmy, 10/15/02)
- (3) Now references "grid_mod.f" and the new "time_mod.f" (bmy, 2/11/03)
- (4) Added error-check and cosmetic changes (bmy, 4/29/03)
- (5) Removed call to COPY_STT_FOR_OX, it's obsolete (bmy, 8/18/03)
- (6) Add fancy output (bmy, 4/26/04)
- (7) Added routine SET_RESTART. Now reference "logical_mod.f" and
"tracer_mod.f" (bmy, 7/20/04)
- (8) Removed obsolete routines TRUE_TRACER_INDEX and COPY_DATA_FOR_CO_OH
(bmy, 6/28/05)
- (9) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (10) Now pass TAU via the arg list in MAKE_RESTART_FILE (bmy, 12/15/05)
- (11) Add MAKE_CSPEC_FILE and READ_CSPEC_FILE routines to save and read
CSPEC_FULL restart files (dkh, 02/12/09)
- 11 Jul 2011 - R. Yantosca - Corrected mis-indexing problem w/ the
CSPEC restart file
- 21 Jul 2011 - M. Long - Now include F77_CMN_SIZE instead of CMN_SIZE
- 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 19 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 09 Feb 2016 - E. Lundgren - Add NetCDF restart file routines
- 11 Jul 2016 - E. Lundgren - Store only species in restart files
- 12 Jul 2016 - E. Lundgren - Remove bpch restart file format option and
associated routines
- 09 Aug 2016 - E. Lundgren - Remove obsolete routines convert_tracer_to_vv
and set_restart and associated module vars;
replace all tracer language to species; rename
OUTPUT_RESTART_FILE to OUTPUT_GC_RST_PREFIX
- 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
- 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

2.4.1 init_gc_restart

Subroutine INIT_GC_RESTART initializes the GEOS-Chem restart file collection and populates it with containers. This collection includes restart concentrations for each species as well as additional data if running the mercury simulation.

INTERFACE:

```

SUBROUTINE INIT_GC_RESTART( am_I_Root, Input_Opt,
&                           State_Chm, RC           )

```

USES:

```

USE GC_GRID_MOD,           ONLY : AREA_M2
USE HCO_Error_Mod
USE HCO_Diagn_Mod
USE Input_Opt_Mod,        ONLY : OptInput

```

```

USE Species_Mod,      ONLY : Species
USE State_Chm_Mod,   ONLY : ChmState
USE TIME_MOD,        ONLY : GET_TS_CHEM

```

INPUT PARAMETERS:

```

LOGICAL,              INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput),      INTENT(INOUT)  :: Input_Opt    ! Input Options object
TYPE(ChmState),      INTENT(INOUT)  :: State_Chm     ! Chemistry state object

```

OUTPUT PARAMETERS:

```

INTEGER,              INTENT(OUT)   :: RC            ! Success or failure?

```

REVISION HISTORY:

```

09 Feb 2016 - E. Lundgren - Initial version
20 Apr 2016 - E. Lundgren - Implement ocean and snow Hg variables
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies and
                        NAMEGAS with SpcInfo%Name from species database
11 Jul 2016 - E. Lundgren - Remove tracer container; now only use species
12 Jul 2016 - E. Lundgren - Rename routine from init_gc_restart_nc

```

2.4.2 read_gc_restart

Subroutine READ_GC_RESTART reads species concentrations [mol/mol] from the GEOS-Chem restart file and uses them to initialize species concentrations in [kg/kg dry]. If species data are missing from the restart file, pre-configured background values are used. If using the mercury simulation, additional restart data are read from file.

INTERFACE:

```

SUBROUTINE READ_GC_RESTART( am_I_Root,  YYYYMMDD,  HHMMSS,
&                           Input_Opt,  State_Met,  State_Chm,  RC )

```

USES:

```

USE m_netcdf_io_open           ! netCDF file open
USE m_netcdf_io_read           ! netCDF read
USE m_netcdf_io_readattr       ! netCDF attribute reads
USE m_netcdf_io_get_dimlen     ! netCDF get dimensions
USE m_netcdf_io_close          ! netCDF file close
USE PHYSCONSTANTS,            ONLY : BOLTZ
USE Input_Opt_Mod,            ONLY : OptInput
USE Species_Mod,              ONLY : Species
USE State_Chm_Mod,            ONLY : ChmState

```


2.4.3 write_gc_restart

Subroutine WRITE_GC_RESTART writes species concentrations [mol/mol] to the GEOS-Chem restart file in NetCDF format. If using the mercury simulation, additional data such as ocean mercury values are written to the restart file.

INTERFACE:

```

SUBROUTINE WRITE_GC_RESTART( am_I_Root, YYYYMMDD, HHMMSS,
&                            Input_Opt, State_Chm, State_Met,
&                            RC )

```

USES:

```

USE Input_Opt_Mod,      ONLY : OptInput
USE Species_Mod,       ONLY : Species
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : EXPAND_DATE

! HEMCO
USE HCO_Error_Mod
USE HCO_Diagn_Mod,     ONLY : Diagn_Update
USE HCOIO_Diagn_Mod,   ONLY : HCOIO_Diagn_WriteOut

! For Hg simulation
USE OCEAN_MERCURY_MOD, ONLY : Hg0aq, Hg2aq, HgPaq, Hgaq_tot
USE OCEAN_MERCURY_MOD, ONLY : CHECK_OCEAN_MERCURY
USE DEPO_MERCURY_MOD,  ONLY : SNOW_HG_OC,  SNOW_HG_STORED_OC
USE DEPO_MERCURY_MOD,  ONLY : SNOW_HG_LN,  SNOW_HG_STORED_LN

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
INTEGER,      INTENT(IN)    :: YYYYMMDD     ! YYYY/MM/DD GMT date
INTEGER,      INTENT(IN)    :: HHMMSS       ! hh:mm:ss GMT time
TYPE(OptInput), INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(MetState), INTENT(IN)  :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)   :: RC           ! Success or failure?

```

REVISION HISTORY:

```

09 Feb 2016 - E. Lundgren - Initial version
20 Apr 2016 - E. Lundgren - Implement ocean and snow Hg variables

```

29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies and
 NAMEGAS with SpcInfo%Name from species database
 22 Jun 2016 - R. Yantosca - Now refer to Hg0_Id_List, Hg2_Id_List, and
 HgP_Id_List fields of State_Chm
 11 Jul 2016 - E. Lundgren - Remove tracers and write only species
 12 Jul 2016 - E. Lundgren - Rename from write_gc_restart_nc
 10 Aug 2016 - E. Lundgren - Input species concentrations are now kg/kg dry

2.4.4 check_rst_dimensions

Subroutine CHECK_RST_DIMENSIONS checks to see if dimensions read from the netCDF file match the defined GEOS-Chem dimensions. If not, then it will stop the GEOS-Chem simulation with an error message.

INTERFACE:

```

SUBROUTINE Check_Rst_Dimensions( lon, lat, lev, time,
&                               time_expected, LOC )

```

INPUT PARAMETERS:

```

INTEGER,          OPTIONAL, INTENT(IN)  :: lon           ! Lon dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lat           ! Lat dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lev           ! Alt dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time          ! Time dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time_expected ! Expected # of
                                                    ! time slots
CHARACTER(LEN=*), INTENT(IN)           :: LOC           ! Caller location

```

REMARKS:

Call this routine with keyword arguments, e.g

```

CALL CHECK_RST_DIMENSION( lon=X, lat=Y, lev=Z, &
                        time=T, time_expected=8, caller=caller )

```

REVISION HISTORY:

REVISION HISTORY:

(1) Added to "restart_mod.f". Now no longer allow initialization with less than a globally-sized data block. (bmy, 6/25/02)
 (2) Now reference GEOS_CHEM_STOP from "error_mod.f", which frees all allocated memory before stopping the run. (bmy, 10/15/02)
 11 Jul 2011 - R. Yantosca - Added ProTeX headers
 21 Jul 2011 - M. Long - Now include F77_CMN_SIZE instead of CMN_SIZE
 09 Feb 2016 - E. Lundgren - Add args and all args optional except location
 01 Apr 2016 - E. Lundgren - Rename this subroutine from check_dimensions to check_rst_dimensions for global clarity

2.4.5 compare_trc_spc

Subroutine COMPARE_TRC_SPC prints out species values to log in kg/kg dry for advected species, and prints out non-advected species molec/cm3. This can only be used where tracers were kg/kg dry.

INTERFACE:

```

SUBROUTINE COMPARE_TRC_SPC( am_I_Root, Input_Opt,
&                           State_Met, State_Chm, LOC, RC )

```

USES:

```

USE Input_Opt_Mod,      ONLY : OptInput
USE Species_Mod,       ONLY : Species
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,                INTENT(IN) :: am_I_Root  ! Are we on the root CPU?
TYPE(OptInput),        INTENT(IN) :: Input_Opt  ! Input Options object
TYPE(MetState),        INTENT(IN) :: State_Met  ! Meteorology State object
CHARACTER(LEN=*),      INTENT(IN) :: LOC
include "netcdf.inc"

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,                INTENT(OUT) :: RC      ! Success or failure?

```

REVISION HISTORY:

08 Aug 2016 - E. Lundgren - Initial version

2.5 Fortran: Module Interface cleanup.F

Subroutine CLEANUP deallocates the memory assigned to dynamically allocatable arrays just before exiting a GEOS-Chem simulation.

INTERFACE:

```

SUBROUTINE CLEANUP( am_I_Root, ERROR, RC )

```

USES:

```

USE AEROSOL_MOD,        ONLY : CLEANUP_AEROSOL
USE C2H6_MOD,           ONLY : CLEANUP_C2H6
USE CARBON_MOD,         ONLY : CLEANUP_CARBON

```

```

        USE CO2_MOD,                ONLY : CLEANUP_CO2
        USE CMN_FJX_Mod,            ONLY : Cleanup_CMN_FJX
        USE CMN_O3_Mod,            ONLY : Cleanup_CMN_O3
        USE CMN_SIZE_Mod,          ONLY : Cleanup_CMN_SIZE
#if defined( BPCH_DIAG )
        USE DIAG_MOD,              ONLY : CLEANUP_DIAG
        USE DIAG03_MOD,            ONLY : CLEANUP_DIAG03
        USE DIAG04_MOD,            ONLY : CLEANUP_DIAG04
        USE DIAG20_MOD,            ONLY : CLEANUP_DIAG20
        USE DIAG41_MOD,            ONLY : CLEANUP_DIAG41
        USE DIAG50_MOD,            ONLY : CLEANUP_DIAG50
        USE DIAG51_MOD,            ONLY : CLEANUP_DIAG51
        USE DIAG53_MOD,            ONLY : CLEANUP_DIAG53
#endif
        USE DIAG_OH_MOD,           ONLY : CLEANUP_DIAG_OH
        USE DRYDEP_MOD,            ONLY : CLEANUP_DRYDEP
        USE DUST_MOD,              ONLY : CLEANUP_DUST
        USE ErrCode_Mod
        USE ERROR_MOD,             ONLY : DEBUG_MSG
        USE FLEXCHEM_MOD,          ONLY : CLEANUP_FLEXCHEM
        USE GEOSFP_READ_MOD,       ONLY : CLEANUP_GEOSFP_READ
        USE GET_NDEP_MOD,          ONLY : CLEANUP_GET_NDEP
        USE GLOBAL_CH4_MOD,        ONLY : CLEANUP_GLOBAL_CH4
        USE GC_GRID_MOD,           ONLY : CLEANUP_GRID
        USE GEOS_TIMERS_MOD
        USE Grid_Registry_Mod,     ONLY : Cleanup_Grid_Registry
        USE HDF_MOD,               ONLY : CLEANUP_HDF
        USE History_Mod,           ONLY : History_Cleanup
        USE Input_Opt_Mod,         ONLY : OptInput
        USE ISOROPIAII_MOD,        ONLY : CLEANUP_ISOROPIAII
        USE LINOZ_MOD,             ONLY : CLEANUP_LINOZ
        USE MERCURY_MOD,           ONLY : CLEANUP_MERCURY
        USE MODIS_LAI_MOD,         ONLY : CLEANUP_MODIS_LAI
        USE OCEAN_MERCURY_MOD,     ONLY : CLEANUP_OCEAN_MERCURY
        USE DEPO_MERCURY_MOD,      ONLY : CLEANUP_DEPO_MERCURY
        USE LAND_MERCURY_MOD,      ONLY : CLEANUP_LAND_MERCURY
        USE MERRA2_READ_MOD,       ONLY : CLEANUP_MERRA2_READ
        USE PBL_MIX_MOD,           ONLY : CLEANUP_PBL_MIX
        USE PJC_PFIX_MOD,          ONLY : CLEANUP_PJC_PFIX
        USE PLANEFLIGHT_MOD,       ONLY : CLEANUP_PLANEFLIGHT
        USE PRESSURE_MOD,          ONLY : CLEANUP_PRESSURE
        USE Regrid_A2A_Mod,        ONLY : Cleanup_Map_A2a
        USE SEASALT_MOD,           ONLY : CLEANUP_SEASALT
        USE Set_Global_CH4_Mod,    ONLY : Cleanup_Set_Global_CH4
        USE SULFATE_MOD,           ONLY : CLEANUP_SULFATE
        USE STRAT_CHEM_MOD,        ONLY : CLEANUP_STRAT_CHEM
        USE TAGGED_CO_MOD,         ONLY : CLEANUP_TAGGED_CO
#if defined( TOMAS )

```

```

        USE TOMAS_MOD,                ONLY : CLEANUP_TOMAS  !sfarina, 1/16/13
#endif
        USE TOMS_MOD,                 ONLY : CLEANUP_TOMS
        USE TPCORE_FVDAS_MOD,         ONLY : EXIT_TPCORE
        USE TPCORE_WINDOW_MOD,        ONLY : EXIT_TPCORE_WINDOW
#if ! defined( ESMF_ )
        USE TRANSPORT_MOD,           ONLY : CLEANUP_TRANSPORT
#endif
        USE UCX_MOD,                 ONLY : CLEANUP_UCX
        USE VDIFFF_PRE_Mod,          ONLY : Cleanup_VDIFFF_PRE
        USE WETSCAV_MOD,             ONLY : CLEANUP_WETSCAV

        ! HEMCO
#if !defined(ESMF_)
        USE EMISSIONS_MOD,           ONLY : EMISSIONS_FINAL
#endif

#if defined( RRTMG )
        USE RRTMG_RAD_TRANSFER_MOD,  ONLY : CLEANUP_SURFACE_RAD
        USE RRTMG_RAD_TRANSFER_MOD,  ONLY : CLEANUP_STRAT_CLIM
        USE RRTMG_RAD_TRANSFER_MOD,  ONLY : CLEANUP_MCICA_CLOUDS
#endif

```

IMPLICIT NONE

INPUT PARAMETERS:

```

        LOGICAL,          INTENT(IN)  :: am_I_Root      ! Are we on the root CPU?
        LOGICAL,          INTENT(IN)  :: ERROR         ! Cleanup after error?

```

OUTPUT PARAMETERS:

```

        INTEGER,          INTENT(OUT) :: RC             ! Success or failure

```

REVISION HISTORY:

- 29 Nov 1999 - R. Yantosca - Initial version
- (1) CLEANUP is written in Fixed-Format F90.
 - (2) Now calls CLEANUP_WETSCAV, which deallocates arrays from "wetscav_mod.f". (bmy, 3/9/00)
 - (3) Add call to CLEANUP_SULFATE, which deallocates arrays from "sulfate_mod.f". Also now deallocate ND32 arrays. (bmy, 6/6/00)
 - (4) Add call to CLEANUP_DAO, which deallocates arrays from "dao_mod.f". (bmy, 6/26/00)
 - (5) Add call to CLEANUP_TAGGED_CO and CLEANUP_COMODE, which deallocates arrays from and "comode_mod.f". (bmy, 7/19/00)
 - (6) Add call to CLEANUP_GLOBAL_OH and CLEANUP_COMODE, which deallocates arrays from "global_oh_mod.f". (bmy, 7/28/00)
 - (7) Add calls to CLEANUP_BIOMASS and CLEANUP_BIOFUEL, which deallocates

- arrays from "biomass_mod.f" and "biofuel_mod.f". Also deallocate the AD32_bf array for the biofuel NOx diagnostic. (bmy, 9/12/00)
- (8) Add call to CLEANUP_DIAG51, to deallocate module arrays from "diag51_mod.f" (bmy, 11/29/00)
 - (9) Removed obsolete code from 11/29/00 (bmy, 12/21/00)
 - (10) Add call to CLEANUP_CH4, to deallocate module arrays from "global_ch4_mod.f" (bmy, 1/16/01)
 - (11) Now deallocate the AD34 array. Also updated comments and made some cosmetic changes. (bmy, 3/15/01)
 - (12) Now deallocate the AD12 array (bdf, bmy, 6/15/01)
 - (13) Add call to CLEANUP_ACETONE, to deallocate module arrays from "acetone_mod.f" Also deallocate AD11 array. Also deallocate variables from dao_mod.f last, to try to avoid bus error on SGI (bmy, 8/3/01)
 - (14) Added call to CLEANUP_UVALBEDO from "uvalbedo_mod.f". Also removed obsolete code from 9/01. Also only include references to CLEANUP_* subroutines in other modules for clarity. (bmy, 1/15/02)
 - (15) Added call to CLEANUP_C2H6 from "c2h6_mod.f" (bmy, 1/25/02)
 - (16) Added call to CLEANUP_AIRCRAFT_NOX from "aircraft_nox_mod.f" (bmy, 2/14/02)
 - (17) Now deallocate CTNO2, CTHO2, LTNO2, LTHO2 arrays (rvm, bmy, 2/27/02)
 - (18) Now reference CLEANUP_PLANEFLIGHT from "planeflight_mod.f". Now also deallocate AD01 and AD02 arrays. (mje, bmy, 8/7/02)
 - (19) Now reference cleanup routines from "global_nox_mod.f", "global_hno3_mod.f", "global_no3_mod.f", "drydep_mod.f", and "rpmares_mod.f". (bmy, 12/16/02)
 - (20) Now reference cleanup routine from "transport_mod.f" (bmy, 2/10/03)
 - (21) Now reference cleanup routine from "pjc_pfix_mod.f" and "tpcore_fvdas_mod.f90". (bmy, 5/9/03)
 - (22) Now reference cleanup routine from "toms_mod.f" (bmy, 7/14/03)
 - (23) Now reference cleanup routine from "carbon_mod.f", "dust_mod.f", and "dust_dead_mod.f". (bmy, 7/14/03)
 - (23) Now references cleanup routine from "lightning__nox_mod.f" (bmy, 4/14/04)
 - (24) Now references cleanup routine from "seasalt_mod.f" (bmy, 4/26/04)
 - (25) Now references cleanup routines from new modules (bmy, 7/20/04)
 - (26) Now calls cleanup routine from "epa_nei_mod.f" (bmy, 11/5/04)
 - (27) Now call CLEANUP_MERCURY from "mercury_mod.f" (eck, bmy, 12/7/04)
 - (28) Now call CLEANUP_OCEAN_MERCURY from "ocean_mercury_mod.f". Also reordered the calling sequence. (sas, bmy, 1/21/05)
 - (29) Now call CLEANUP_PBL_MIX from "pbl_mix_mod.f". Now call CLEANUP_DIAG41 from "diag41_mod.f". (bmy, 2/17/05)
 - (30) Now calls CLEANUP_HCN_CH3CN from "hcn_ch3cn_mod.f" (bmy, 6/23/05)
 - (31) Now calls CLEANUP_DIAG04, CLEANUP_CO2, and CLEANUP_TROPOPAUSE (bmy, 8/15/05)
 - (32) Now calls CLEANUP_LAI from "lai_mod.f", CLEANUP_MEGAN from "megan_mod.f" and CLEANUP_REGRID_1x1 from "regrid_1x1_mod.f" (tmf, bdf, bmy, 10/24/05)

- (33) Now calls CLEANUP_EMEP from "emep_mod.f" (bdf, bmy, 11/1/05)
- (34) Now calls CLEANUP_GC_BIOMASS and CLEANUP_GFED2_BIOMASS (bmy, 4/5/06)
- (35) Now calls CLEANUP_DIAG56 from "diag56_mod.f" and
CLEANUP_LIGHTNING_NOX_NL from "lightning_nox_nl_mod.f"
(ltm, bmy, 5/5/06)
- (36) Now references CLEANUP_BRAVO from "bravo_mod.f" and CLEANUP_EDGAR
from "edgar_mod.f" (bmy, 7/6/06)
- (37) Now calls CLEANUP_H2_HD from "h2_hd_mod.f" and CLEANUP_GLOBAL_01D
from "global_old_mod.f". Remove call to CLEANUP_LIGHTNING_NOX_NL
from "lightning_nox_nl_mod.f" (hup, phs, bmy, 10/2/07)
- (38) Now calls GEOS5_EXIT_TPCORE_WINDOW to finalize the TPCORE for
GEOS-5 nested window simulations (yxw, dan, bmy, 11/6/08)
- (39) Now references CLEANUP_CAC_ANTHRO (amv, phs, 3/10/08)
- (40) Now references CLEANUP_ARCTAS_SHIP (phs, 3/10/08)
- (41) Now references CLEANUP_VISTAS_ANTHRO (phs, 3/10/08)
- (41) Now references CLEANUP_LINOZ (phs, 10/16/09)
- (42) Now references CLEANUP_HDF (amv, bmy, 12/21/09)
- (43) Now references CLEANUP_ISOROPIAII (ccc, bmy, 1/29/09)
- (44) Now references CLEANUP_DEPO_MERCURY and CLEANUP_LAND_MERCURY
(ccc, 5/6/10)
- (45) Added call to CLEANUP_GLOBAL_OC, which deallocates arrays from
"global_oc_mod.f" (clf, 2/28/2011)
- (46) Added call to CLEANUP_GLOBAL_BC, which deallocates arrays from
"global_bc_mod.f" (clf, 2/28/2011)
- 13 Aug 2010 - R. Yantosca - Added ProTeX headers
- 28 Feb 2011 - C. Friedman - Added call to CLEANUP_GLOBAL_OC, which
deallocates arrays from "global_oc_mod.f"
- 22 Aug 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
- 07 Sep 2011 - P. Kasibhatla - Add modifications for GFED3
- 19 Mar 2012 - M. Payer - Remove call to CLEANUP_ACETONE. It is no longer
needed after removal of JO1D and RESP routines.
- 05 Apr 2012 - R. Yantosca - Now call CLEANUP_MODIS_LAI
- 11 Apr 2012 - R. Yantosca - Remove reference to obsolete lai_mod.F
- 01 May 2012 - M. Payer - Added calls for CLEANUP_GLOBAL_GRID and
CLEANUP_GRID
- 19 Nov 2012 - R. Yantosca - Add cleanup calls for modules in Headers;
these were omitted during development
- 28 Nov 2012 - R. Yantosca - Remove reference to CLEANUP_DAO; we have now
removed all allocatable arrays from dao_mod.F
- 03 Dec 2012 - R. Yantosca - Now call Cleanup_CMN_SIZE
- 13 Dec 2012 - R. Yantosca - Remove reference to obsolete CMN_DEP_mod.F
- 06 Mar 2013 - H. Amos - Merge C. Friedman's PAH code
- 14 Mar 2013 - M. Payer - Restore reference to CLEANUP_DAO. Some arrays
have been restored to dao_mod.F for use in
the vertical regridding of OH for offline
simulations.
- 24 Mar 2013 - S.D. Eastham- Switched TROPOPAUSE_MOD to CHEMGRID_MOD
- 04 Apr 2013 - S.D. Eastham- Added call for CLEANUP_UCX

- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
- 26 Sep 2013 - R. Yantosca - Now reference EXIT_GEOSFP_TPCORE_WINDOW
- 29 Oct 2013 - R. Yantosca - Now call CLEANUP_DAO for all simulations (if arrays are not allocated it'll exit gracefully)
- 03 Jun 2014 - R. Yantosca - Don't finalize HEMCO if emissions are turned off. This prevents a seg fault error.
- 03 Jun 2014 - R. Yantosca - Now pass Input_Opt via the argument list
- 03 Jun 2014 - R. Yantosca - Remove references to emissions modules that HEMCO renders obsolete
- 24 Jun 2014 - R. Yantosca - Remove CLEANUP_BROMOCARB; HEMCO replaces this
- 25 Jun 2014 - R. Yantosca - Remove reference to tracer_mod.F
- 21 Jul 2014 - R. Yantosca - Remove reference to regrid_1x1_mod.F
- 23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN_mod.F
- 23 Jul 2014 - R. Yantosca - Remove reference to obsolete global_grid_mod.F
- 23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN_NOX_mod.F
- 25 Jul 2014 - R. Yantosca - Remove reference to commsoil_mod.F90
- 25 Jul 2014 - R. Yantosca - Now call CLEANUP_GET_NDEP
- 15 Aug 2014 - R. Yantosca - Remove reference to biofuel_mod.F
- 15 Aug 2014 - R. Yantosca - Remove reference to biomass_mod.F
- 10 Sep 2014 - M. Sulprizio- Remove references to global_oc_mod.F and global_bc_mod.F
- 23 Sep 2014 - M. Sulprizio- Remove references to global_hno3_mod.F and global_no3_mod.F
- 12 Jan 2015 - R. Yantosca - Remove CLEANUP_UVALBEDO routine
- 11 Mar 2015 - R. Yantosca - Remove call to CLEANUP_GLOBAL_OH
- 11 Mar 2015 - R. Yantosca - Remove call to CLEANUP_GLOBAL_NOX
- 16 Mar 2015 - R. Yantosca - Remove call to CLEANUP_HCN_CH3CN
- 25 Mar 2015 - C. Keller - Now cleanup UCX before emissions.
- 16 Apr 2015 - R. Yantosca - Remove call to CLEANUP_DAO; it's obsolete
- 24 Mar 2016 - C. Keller - Remove input arg Input_Opt; this is cleaned up before routine CLEANUP is called.
- 18 May 2016 - M. Sulprizio- Remove call to CLEANUP_COMODE; it's obsolete
- 30 Jun 2016 - M. Sulprizio- Remove call to CLEANUP_COMODE_LOOP; it's obsolete
- 29 Nov 2016 - R. Yantosca- grid_mod.F90 is now gc_grid_mod.F90
- 09 Aug 2017 - R. Yantosca - Pass am_I_Root, RC to Cleanup_Grid
- 16 Aug 2017 - R. Yantosca - Now call History_Cleanup from here
- 23 Aug 2017 - R. Yantosca - Now time how long HISTORY_CLEANUP takes
- 23 Aug 2017 - R. Yantosca - Now call Cleanup_Grid_Registry
- 06 Nov 2017 - R. Yantosca - Remove call to Cleanup_Chemgrid
- 09 Nov 2017 - R. Yantosca - Return error condition to calling program

3 Transport modules

These modules contain routines to advect GEOS-Chem species using assimilated meteorology fields.

3.1 Fortran: Module Interface transport_mod.F

Module TRANSPORT_MOD is used to call the proper version of the TPCORE advection scheme for different met field datasets and their nested or global grids.

INTERFACE:

```
MODULE TRANSPORT_MOD
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
USE PRESSURE_MOD
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: CLEANUP_TRANSPORT
PUBLIC  :: DO_TRANSPORT
PUBLIC  :: INIT_TRANSPORT
PUBLIC  :: INIT_WINDOW_TRANSPORT
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: DO_GLOBAL_ADV
PRIVATE :: DO_WINDOW_TRANSPORT
```

REVISION HISTORY:

- 10 Mar 2003 - Y. Wang, R. Yantosca - Initial version
- (1) Now can select transport scheme for GEOS-3 winds. Added code for PJC pressure fixer. (bdf, bmy, 5/8/03)
- (2) Now delete DSIG array, it's obsolete. Also added new PRIVATE function GET_AIR_MASS to compute air masses from the input/output pressures from the new GEOS-4/fvDAS TPCORE. (bmy, 6/24/03)
- (3) Now references DEBUG_MSG from "error_mod.f". (bmy, 8/7/03)
- (4) Bug fix in DO_GLOBAL_TRANSPORT (bmy, 10/21/03)
- (5) IORD, JORD, KORD are now module variables. Now references "logical_mod.f" and "tracer_mod.f" (bmy, 7/20/04)
- (6) Add mass-flux diagnostics to TPCORE_FVDAS (bdf, bmy, 9/28/04)
- (7) Now references "diag_mod.f" (bmy, 9/28/04)
- (8) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
- (9) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (10) Now flip arrays in call to TPCORE_FVDAS (bmy, 6/16/06)
- (11) Added modifications for SUN compiler (bmy, 7/12/06)
- (12) Bug fixes in DO_GLOBAL_TRANSPORT (bmy, 11/29/06)
- (13) Split off GCAP, GEOS-3, GEOS-4/GEOS-5 specific calling sequences into separate subroutines. Also removed some obsolete module variables. (bmy, 10/30/07)

(14) Modifications for GEOS-5 nested grid (yxw, dan, bmy, 11/6/08)
 (15) Bug fix in mass balance in GCAP_GLOBAL_ADV and GEOS4_GEOS5_GLOBAL_ADV.
 (ccc, 2/17/09)

26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBDED switch
 26 Feb 2010 - R. Yantosca - Added ProTex Headers
 08 Mar 2010 - C. Carouge - Modify call to tpcore_fvdas. We do not re-order
 mass fluxes diagnostics anymore.

28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
 21 Jun 2012 - R. Yantosca - Now use pointers to flip arrays in vertical
 21 Jun 2012 - R. Yantosca - Comment out GEOS-3 window subroutine
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 26 Sep 2013 - R. Yantosca - Renamed "GEOS57" to "GEOSFP" in routine names
 24 Nov 2014 - M. Yannetti - Added PRECISION_MOD
 12 Feb 2015 - E. Lundgren - Added functionality for writing diags to netcdf
 15 Apr 2015 - R. Yantosca - Add TARGET spec to A_M2 array
 24 Jun 2015 - E. Lundgren - Now use kg/kg total air as tracer units
 (previously v/v) for advection
 24 Jun 2015 - E. Lundgren - Remove post-advection residual mass correction
 and GET_AIR_MASS function since no longer needed
 when using kg/kg total air in advection
 13 Aug 2015 - E. Lundgren - Tracer units are now input as kg/kg dry air
 (previously v/v)
 19 Jan 2016 - E. Lundgren - Move DiagnUpdate_Transport to diagnostics_mod
 and consolidate netcdf and bpch diagnostic code
 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer
 compatible with the FlexChem implementation
 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
 24 Aug 2017 - M. Sulprizio - Remove support for GCAP, GEOS-4, GEOS-5 and MERRA
 and rename routine GEOS4_GEOS5_GLOBAL_ADV to
 DO_GLOBAL_ADV

3.1.1 do.transport

Subroutine DO_TRANSPORT is the driver routine for the proper TPCORE program for GEOS-3, GEOS-4/GEOS-5, or window simulations.

INTERFACE:

```

SUBROUTINE DO_TRANSPORT( am_I_Root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, RC           )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE GC_GRID_MOD,    ONLY : ITS_A_NESTED_GRID
USE Input_Opt_Mod, ONLY : OptInput

```



```

    USE State_Chm_Mod, ONLY : ChmState
    USE State_Diag_Mod, ONLY : DgnState
    USE State_Met_Mod, ONLY : MetState
    USE TPCORE_BC_MOD, ONLY : INIT_TPCORE_BC
    USE TIME_MOD, ONLY : GET_TS_DYN
#if defined( USE_TEND )
    USE TENDENCIES_MOD
#endif

```

INPUT PARAMETERS:

```

    LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
    TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

    TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
    TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
    TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object

```

OUTPUT PARAMETERS:

```

    INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

REVISION HISTORY:

```

10 Mar 2003 - R. Yantosca - Initial version
(1) Removed IORD, JORD, KORD from the arg list. Also now removed
    reference to CMN, it's not needed. (bmy, 7/20/04)
(2) Now call separate routines for different met fields. (bmy, 10/30/07)
(3) Now references subroutine INIT_TPCORE_BC from tpcore_bc_mod.f and
    DO_GEOS5_FVDAS_WINDOW_TRANSPORT from
    "tpcore_geos5_fvdas_window_mod.f90". (yxw, dan, bmy, 11/6/08)
26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBED switch
26 Feb 2010 - R. Yantosca - Added ProTeX headers
06 Oct 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5.
28 Feb 2012 - R. Yantosca - Treat GEOS-5.7 in the same way as MERRA
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
31 Mar 2015 - E. Lundgren - Make State_Met INTENT(INOUT) for AIRQNT calls
19 Jul 2016 - R. Yantosca - Now bracket DO_TEND calls with #ifdef USE_TEND
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
09 Nov 2017 - R. Yantosca - Return error condition to calling program

```

3.1.2 do_global_adv

Subroutine DO_GLOBAL_ADV is the driver routine for TPCORE with the GMAO GEOS-FP or MERRA-2 met fields.

INTERFACE:

```

SUBROUTINE DO_GLOBAL_ADV( am_I_Root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, RC )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE DAO_MOD,              ONLY : AIRQNT
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD         ! NDxx flags
USE DIAG_MOD,           ONLY : MASSFLEW, MASSFLNS, MASSFLUP
#endif
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod,       ONLY : OptInput
USE State_Chm_Mod,       ONLY : ChmState
USE State_Diag_Mod,      ONLY : DgnState

USE State_Met_Mod,       ONLY : MetState
USE State_Met_Mod,       ONLY : MetState
USE PhysConstants        ! Physical constants
USE PJC_PFIX_MOD,        ONLY : DO_PJC_PFIX
USE TIME_MOD,            ONLY : GET_TS_DYN
USE TPCORE_FVDAS_MOD,    ONLY : TPCORE_FVDAS
USE UnitConv_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)  :: State_Met    ! Meteorology State object
TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

As of July 2016, we assume that all of the advected species are listed first in the species database. This is the easiest way to pass a slab to the TPCORE routine. This may change in the future. (bmy, 7/13/16)

Note: the mass flux diagnostic arrays (MASSFLEW, MASSFLNS and MASSFLUP) are incremented upside-down (level 1 = top of the atmosphere). The levels order is reversed only when written out to diagnostic output.

REVISION HISTORY:

30 Oct 2007 - R. Yantosca - Initial version
 (1) Split off the GEOS-4 & GEOS-5 relevant parts from the previous routine DO_GLOBAL_TRANSPORT (bmy, 10/30/07)
 (2) Activate the call to SAVE_GLOBAL_TPCORE_BC (yxw, dan, bmy, 11/6/08)
 (3) Bug fix in mass balance: only account for cells of STT with non-zero concentrations when doing the computation (ccc, bmy, 2/17/09)
 26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBED switch
 26 Feb 2010 - R. Yantosca - Added ProTeX headers
 16 Feb 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
 21 Jun 2012 - R. Yantosca - Now use pointers to flip indices in vertical
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
 04 Jun 2013 - R. Yantosca - Replace pointer STT with State_Chm%Tracers
 26 Feb 2015 - E. Lundgren - Replace GET_PEDGE with State_Met%PEDGE.
 31 Mar 2015 - E. Lundgren - Move AIRQNT call from main to within transport
 09 Jun 2015 - E. Lundgren - Remove dependency on tpcore_bc_mod since not used
 15 Jun 2015 - E. Lundgren - Tracer units in advection are now kg/kg total air (previously v/v)
 15 Jun 2015 - E. Lundgren - Remove mass residual adjustment to tracer concentration following advection
 13 Aug 2015 - E. lundgren - Remove v/v <-> kg/kg conversion
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
 20 Jul 2016 - R. Yantosca - Add minor bug fixes for netCDF diagnostics
 20 Jul 2016 - R. Yantosca - Now replace NNPARG with State_Chm%nAdvect
 20 Jul 2016 - R. Yantosca - Now pass State_Chm to DiagnUpdate_Transport_Flux
 03 Aug 2016 - R. Yantosca - Removed temporary tracer-removal code
 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

3.1.3 do_window_transport

Subroutine DO_WINDOW_TRANSPORT is the driver program for the proper TPCORE program for the GEOS-FP/MERRA2 nested-grid simulations.

INTERFACE:

```

SUBROUTINE DO_WINDOW_TRANSPORT( am_I_Root, Input_Opt, State_Met,
&                               State_Chm, State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE DAO_MOD,              ONLY : AIRQNT
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD          ! NDxx flags
USE DIAG_MOD,            ONLY : MASSFLEW, MASSFLNS, MASSFLUP

```

```

#endif
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD,          ONLY : GET_XOFFSET,   GET_YOFFSET
USE Input_Opt_Mod,       ONLY : OptInput
USE State_Chm_Mod,       ONLY : ChmState
USE State_Met_Mod,       ONLY : MetState
USE State_Diag_Mod,      ONLY : DgnState
USE PhysConstants        ! Physical constants
USE PJC_PFIX_WINDOW_MOD, ONLY : DO_PJC_PFIX_WINDOW
USE TIME_MOD,           ONLY : GET_TS_DYN
USE TPCORE_BC_MOD,       ONLY : IO_W, JO_W, I1_W, J1_W
USE TPCORE_BC_MOD,       ONLY : I2_W, J2_W, IM_W, JM_W, IGZD
USE TPCORE_BC_MOD,       ONLY : DO_WINDOW_TPCORE_BC
USE TPCORE_WINDOW_MOD,   ONLY : TPCORE_WINDOW
USE UnitConv_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)  :: State_Met    ! Meteorology State object
TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)  :: RC           ! Success or failure?

```

REMARKS:

As of July 2016, we assume that all of the advected species are listed first in the species database. This is the easiest way to pass a slab to the TPCORE routine. This may change in the future. (bmy, 7/13/16)

Note: the mass flux diagnostic arrays (MASSFLEW, MASSFLNS and MASSFLUP) are incremented upside-down (level 1 = top of the atmosphere). The levels order is reversed only when written out to diagnostic output.

REVISION HISTORY:

```

10 Mar 2003 - R. Yantosca - Initial version
26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBED switch
26 Feb 2010 - R. Yantosca - Added ProTeX headers
16 Feb 2011 - R. Yantosca - Add modifications for APM from G. Luo
21 Jun 2012 - R. Yantosca - Now use pointers to flip arrays in the vertical
04 Jun 2013 - R. Yantosca - Replace pointer STT with State_Chm%Tracers
26 Feb 2015 - E. Lundgren - Replace GET_PEDGE with State_Met%PEDGE.
31 Mar 2015 - E. Lundgren - Move AIRQNT call from main to within transport

```

01 Apr 2015 - L. Zhang - Add updates to skip the window region
 15 Apr 2015 - R. Yantosca - Add more pointers to avoid array temporaries
 15 Jun 2015 - E. Lundgren - Tracer units in advection are now kg/kg total air
 (previously v/v)
 15 Jun 2015 - E. Lundgren - Remove mass residual adjustment to tracer
 concentration prior to advection
 13 Aug 2015 - E. lundgren - Remove v/v <-> kg/kg conversion
 04 Nov 2015 - M. Sulprizio- Rename from DO_GEOSFP_WINDOW_TRANSPORT to
 DO_WINDOW_TRANSPORT for use with all nested grids
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 20 Jul 2016 - R. Yantosca - Add minor bug fixes for netCDF diagnostics
 20 Jul 2016 - R. Yantosca - Now replace NNPARG with State_Chm%nAdvect
 20 Jul 2016 - R. Yantosca - Now pass State_Chm to DiagnUpdate_Transport_Flux
 03 Aug 2016 - R. Yantosca - Removed temporary tracer-removal code
 11 Jul 2017 - R. Yantosca - Prevent segfault if ND24, ND25, or ND26 are off
 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

3.1.4 init_transport

Subroutine INIT_TRANSPORT initializes all module variables and arrays.

INTERFACE:

```
SUBROUTINE INIT_TRANSPORT( am_I_Root, Input_Opt, State_Diag, RC )
```

USES:

```
USE CMN_SIZE_MOD           ! Size parameters
USE ErrCode_Mod
USE ERROR_MOD,            ONLY : ALLOC_ERR
USE GC_GRID_MOD,         ONLY : GET_AREA_M2, GET_YMID_R
USE Input_Opt_Mod,       ONLY : OptInput
USE State_Diag_Mod,      ONLY : DgnState
USE PhysConstants        ! Re
USE TIME_MOD,            ONLY : GET_TS_DYN
USE TPCORE_FVDAS_MOD,    ONLY : INIT_TPCORE
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(DgnState),  INTENT(IN)  :: State_Diag    ! Diagnostics State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

- 10 Mar 2003 - R. Yantosca - Initial version
- (1) Now references GET_TS_DYN from "time_mod.f", INIT_TPCORE_FVDAS from "tpcore_fvdas_mod.f90", and GET_YMID_R from "grid_mod.f". Now also include "CMN_SETUP". (bdf, bmy, 4/28/03)
- (2) Remove reference to DSIG, it's obsolete. (bmy, 6/24/03)
- (3) Now references LEMBED & LTPFV from "logical_mod.f". Now references N_TRACERS from "tracer_mod.f". (bmy, 7/20/04)
- (4) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
- (5) Removed reference to USE_GEOS_4_TRANSPORT, STT_I1, STT_I2, STT_J1, STT_J2, variables (bmy, 10/30/07)
- (6) Deleted reference to CMN, it's not needed anymore (bmy, 11/6/08)
- 26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBED switch
- 26 Feb 2010 - R. Yantosca - Added ProTeX headers
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_M2(I,J,L) from grid_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET_YMID_R_W(I,J,L) from grid_mod.F90
- 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC
- 05 Dec 2014 - M. Yannetti - Added REAL_N_DYN
- 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
- 09 Nov 2017 - R. Yantosca - Return error condition to calling program

3.1.5 init_window_transport

Subroutine INIT_WINDOW_TRANSPORT initializes all module variables and arrays for the GEOS-FP/MERRA2 nested grid simulation.

INTERFACE:

```

SUBROUTINE INIT_WINDOW_TRANSPORT( am_I_Root,  Input_Opt,
&                                State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE ErrCode_Mod
USE ERROR_MOD,            ONLY : ALLOC_ERR
USE GC_GRID_MOD,         ONLY : GET_AREA_M2
USE GC_GRID_MOD,         ONLY : GET_YMID_R_W
USE Input_Opt_Mod,      ONLY : OptInput
USE PhysConstants        ! Re
USE State_Diag_Mod,     ONLY : DgnState
USE TIME_MOD,           ONLY : GET_TS_DYN
USE TPCORE_FVDAS_MOD,   ONLY : INIT_TPCORE
USE TPCORE_BC_MOD,      ONLY : IO_W, JO_W, I1_W, J1_W
USE TPCORE_BC_MOD,      ONLY : I2_W, J2_W, IM_W, JM_W
USE TPCORE_BC_MOD,      ONLY : IGZD, INIT_TPCORE_BC
USE TPCORE_WINDOW_MOD,  ONLY : INIT_WINDOW

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(DgnState),  INTENT(IN)  :: State_Diag    ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT) :: RC           ! Success or failure?

```

REVISION HISTORY:

```

06 Jun 2008 - D. Chen & R. Yantosca - Initial version
26 Feb 2010 - R. Yantosca - Removed references to obsolete LEMBED switch
26 Feb 2010 - R. Yantosca - Added ProTeX headers
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID_R_W(I,J,L) from grid_mod.F90
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
26 Sep 2013 - R. Yantosca - Renamed to INIT_GEOSFP_WINDOW_TRANSPORT
05 Dec 2014 - M. Yannetti - Added REAL_N_DYN
04 Nov 2015 - M. Sulprizio- Renamed from INIT_GEOSFP_WINDOW_TRANSPORT to
                          INIT_WINDOW_TRANSPORT for use with all nested
                          grids
26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

3.1.6 cleanup_transport

Subroutine CLEANUP_TRANSPORT deallocates all module arrays.

INTERFACE:

```

SUBROUTINE CLEANUP_TRANSPORT

```

REVISION HISTORY:

```

10 Mar 2003 - R. Yantosca - Initial version
(1 ) Remove reference to DSIG, it's obsolete. (bmy, 6/24/03)
(2 ) Remove obsolete embedded chemistry arrays (bmy, 10/30/07)
26 Feb 2010 - R. Yantosca - Added ProTeX headers
12 Feb 2015 - E. Lundgren - Added new diagnostics arrays for netcdf output

```

3.2 Fortran: Module Interface pjc_pfix_mod.F

Module Pjc_Pfix_Mod contains routines which implements the Philip Cameron-Smith pressure fixer for the new fvDAS transport scheme. (bdf, bmy, 5/8/03, 10/27/03)

INTERFACE:

```
MODULE Pjc_Pfix_Mod
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: Do_Pjc_Pfix
PUBLIC  :: Cleanup_Pjc_Pfix
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: Calc_Pressure
PRIVATE :: Calc_Advection_Factors
PRIVATE :: Adjust_Press
PRIVATE :: Init_Press_Fix
PRIVATE :: Do_Press_Fix_LLNL
PRIVATE :: Average_Press_Poles
PRIVATE :: Convert_Winds
PRIVATE :: Calc_Horiz_Mass_Flux
PRIVATE :: Calc_Divergence
PRIVATE :: Set_Press_Terms
PRIVATE :: Do_Divergence_Pole_Sum
PRIVATE :: Xpavg
PRIVATE :: Init_Pjc_Pfix
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)
Brendan Field and Bob Yantosca (5/8/03)
Modified for new GMI TPCORE by Claire Carouge (ccarouge@seas.harvard.edu)

REVISION HISTORY:

- (1) Bug fix for Linux/PGI compiler in routines ADJUST_PRESS and
INIT_PRESS_FIX. (bmy, 6/23/03)
- (2) Now make P1, P2 true surface pressure in DO_PJC_PFIX (bmy, 10/27/03)
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 19 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

3.2.1 Do_Pjc_Pfix

Subroutine Do_Pjc_Pfix is the driver routine for the Philip Cameron-Smith pressure fixer for the fvDAS transport scheme. (bdf, bmy, 5/8/03, 3/5/07)

We assume that the winds are on the A-GRID, since this is the input that the fvDAS transport scheme takes. (bdf, bmy, 5/8/03)

INTERFACE:

```
SUBROUTINE Do_Pjc_Pfix( D_DYN, P1, P2, UWND, VWND, XMASS, YMASS )
```

USES:

```
USE CMN_SIZE_MOD      ! Size parameters
USE PhysConstants     ! Physical constants
```

INPUT PARAMETERS:

```
! Dynamic timestep [s]
REAL(fp), INTENT(IN)  :: D_DYN

! True PSurface at middle of dynamic timestep [hPa]
REAL(fp), INTENT(IN)  :: P1(:, :)

! True PSurface at end   of dynamic timestep [hPa]
REAL(fp), INTENT(IN)  :: P2(:, :)

! Zonal (E-W) wind [m/s]
REAL(fp), INTENT(IN)  :: UWND(IIPAR, JJPARG, LLPARG)

! Meridional (N-S) wind [m/s]
REAL(fp), INTENT(IN)  :: VWND(IIPAR, JJPARG, LLPARG)
```

OUTPUT PARAMETERS:

```
! E-W mass fluxes [mixing ratio]
REAL(fp), INTENT(OUT) :: XMASS(IIPARG, JJPARG, LLPARG)

! N-S mass fluxes [mixing ratio]
REAL(fp), INTENT(OUT) :: YMASS(IIPARG, JJPARG, LLPARG)
```

AUTHOR:

Brendan Field and Bob Yantosca (5/8/03)

REMARKS:

- (1) Now P1 and P2 are "true" surface pressures, and not PS-PTOP. If using this P-fixer w/ GEOS-3 winds, pass true surface pressure to this routine. (bmy, 10/27/03)
- (2) Now define P2_TMP array for passing to ADJUST_PRESS (yxw, bmy, 3/5/07)

REVISION HISTORY:

- 02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).
- 04 Nov 2013 - M. Sulprizio- Eliminate array temporaries by accepting assumed-
shape dummy arguments
-

3.2.2 Calc_Pressure

Subroutine Calc_Pressure recalculates the new surface pressure from the adjusted air masses XMASS and YMASS. This is useful for debugging purposes. (bdf, bmy, 5/8/03)

INTERFACE:

```
SUBROUTINE Calc_Pressure( XMASS, YMASS, RGW_FV, PS_NOW, PS_AFTER )
```

USES:

```
USE CMN_SIZE_MOD ! Size parameters
```

INPUT PARAMETERS:

```
! E-W mass flux from pressure fixer
REAL(fp), INTENT(IN)  :: XMASS(IIPAR,JJP,LLPAR)

! N-S mass flux from pressure fixer
REAL(fp), INTENT(IN)  :: YMASS(IIPAR,JJP,LLPAR)

! Surface pressure - PTOP at current time
REAL(fp), INTENT(IN)  :: PS_NOW(IIPAR,JJP)

! 1 / ( SINE(J+1) - SINE(J) ) -- latitude factor
REAL(fp), INTENT(IN)  :: RGW_FV(JJP)
```

OUTPUT PARAMETERS:

```
! Surface pressure - PTOP adjusted by P-fixer
REAL(fp), INTENT(OUT) :: PS_AFTER(IIPAR,JJP)
```

AUTHOR:

Brendan Field and Bob Yantosca (5/8/03)

REVISION HISTORY:

```
02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                Declare all REAL variables as REAL(fp).
23 Jul 2014 - R. Yantosca - Removed reference to obsolete CMN_mod.F
```

3.2.3 Calc_Advection_Factors

Subroutine Calc_Advection_Factors calculates the relative area of each grid box, and the geometrical factors used by this modified version of TPCORE. These geometrical DO assume that the space is regularly gridded, but do not assume any link between the surface area and the linear dimensions.

INTERFACE:

```

SUBROUTINE Calc_Advection_Factors
& (mcor, rel_area, geofac, geofac_pc)

```

USES:

```

USE CMN_SIZE_MOD    ! Size parameters
USE PhysConstants   ! Physical constants

```

INPUT PARAMETERS:

```

! Area of grid box (m^2)
REAL(fp), INTENT(IN)  :: mcor(i1_gl :i2_gl, ju1_gl:j2_gl)

```

OUTPUT PARAMETERS:

```

! relative surface area of grid box (fraction)
REAL(fp), INTENT(OUT) :: rel_area(i1_gl :i2_gl, ju1_gl:j2_gl)

! Geometrical factor for meridional advection; geofac uses
! correct spherical geometry, and replaces acosp as the
! meridional geometrical factor in tpcore
REAL(fp), INTENT(OUT) :: geofac(ju1_gl:j2_gl)

! Special geometrical factor (geofac) for Polar cap
REAL(fp), INTENT(OUT) :: geofac_pc

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REMARKS:

Now reference PI from "CMN_GCTM" for consistency. Also force double-precision with the "D" exponent. (bmy, 5/6/03)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.4 Adjust_Press

Subroutine Adjust_Press initializes and calls the pressure fixer code.

INTERFACE:

```

SUBROUTINE Adjust_Press
& (metdata_name_org, do_timinterp_winds, new_met_rec,
&  met_grid_type, advec_consrv_opt, pmet2_opt, press_fix_opt,
&  tdt, geofac_pc, geofac, cose, cosp, rel_area, dap, dbk,
&  pctm1, pctm2, pmet2, uu, vv, xmass, ymass)

```

INPUT PARAMETERS:

```
! First part of metdata_name, e.g., "NCAR"
CHARACTER(LEN=*) :: metdata_name_org

! Time interpolate wind fields?
LOGICAL :: do_timinterp_winds

! New met record?
LOGICAL :: new_met_rec

! Met grid type, A or C
INTEGER :: met_grid_type

! Advection_conserve option
INTEGER :: advec_consrv_opt

! pmet2 option
INTEGER :: pmet2_opt

! pressure fixer option
INTEGER :: press_fix_opt

! Model time step [s]
REAL(fp) :: tdt

! Special geometrical factor (geofac) for Polar cap
REAL(fp) :: geofac_pc

! Geometrical factor for meridional advection; geofac uses
! correct spherical geometry, and replaces acosp as the
! meridional geometrical factor in tpcore
REAL(fp) :: geofac (ju1_g1:j2_g1)

! Cosines of grid box edges and centers
REAL(fp) :: cose (ju1_g1:j2_g1)
REAL(fp) :: cosp (ju1_g1:j2_g1)

! Pressure difference across layer from (ai * pt) term [hPa]
REAL(fp) :: dap (k1:k2)

! Difference in bi across layer - the dSigma term
REAL(fp) :: dbk (k1:k2)

! Relative surface area of grid box (fraction)
REAL(fp) :: rel_area( i1_g1:i2_g1, ju1_g1:j2_g1)

! Metfield surface pressure at t1+tdt [hPa]
```

```

REAL(fp)  :: pmet2(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! CTM surface pressure at t1 [hPa]
REAL(fp)  :: pctm1(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! CTM surface pressure at t1+tdt [hPa]
REAL(fp)  :: pctm2(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! Wind velocity, x direction at t1+tdt/2 [m/s]
REAL(fp)  :: uu(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

! Wind velocity, y direction at t1+tdt/2 [m/s]
REAL(fp)  :: vv(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

```

INPUT/OUTPUT PARAMETERS:

```

! Horizontal mass flux in E-W direction [hPa]
REAL(fp)  :: xmass(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

! Horizontal mass flux in N-S direction [hPa]
REAL(fp)  :: ymass(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
 Declare all REAL variables as REAL(fp).

3.2.5 Init_Press_Fix

Subroutine Init_Press_Fix initializes the pressure fixer.

INTERFACE:

```

SUBROUTINE Init_Press_Fix
& (metdata_name_org, met_grid_type, tdt, geofac_pc, geofac,
&   cose, cosp, dap, dbk, dps, dps_ctm, rel_area, pctm1, pmet2,
&   uu, vv, xmass, ymass)

```

INPUT PARAMETERS:

```

! Model Time step [s]
REAL(fp)  :: tdt

! First part of metdata_name, e.g., "NCAR"
CHARACTER(LEN=*) :: metdata_name_org

```

```

! Met grid type, A or C
INTEGER          :: met_grid_type

! Special geometrical factor (geofac) for Polar cap
REAL(fp)        :: geofac_pc

! Cosine of grid box edges and centers
REAL(fp)        :: cose(ju1_gl:j2_gl)
REAL(fp)        :: cosp(ju1_gl:j2_gl)

! Geometrical factor for meridional advection; geofac uses
! correct spherical geometry, and replaces acosp as the
! meridional geometrical factor in tpcore
REAL(fp)        :: geofac(ju1_gl:j2_gl)

! Pressure difference across layer from (ai * pt) term [hPa]
REAL(fp)        :: dap(k1:k2)

! Difference in bi across layer - the dSigma term
REAL(fp)        :: dbk(k1:k2)

! relative surface area of grid box (fraction)
REAL(fp)        :: rel_area( i1_gl:i2_gl, ju1_gl:j2_gl)

! Metfield surface pressure at t1 [hPa]
REAL(fp)        :: pmet2(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! CTM surface pressure at t1 [hPa]
REAL(fp)        :: pctm1(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! CTM surface pressure at t1+tdt [hPa]
REAL(fp)        :: pctm2(ilo_gl:ihi_gl, julio_gl:jhi_gl)

! Wind velocity, x direction at t1+tdt/2 [m/s]
REAL(fp)        :: uu(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

! Wind velocity, y direction at t1+tdt/2 [m/s]
REAL(fp)        :: vv(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

```

OUTPUT PARAMETERS:

```

! Horizontal mass flux in E-W direction [hPa]
REAL(fp) :: xmass(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

! Horizontal mass flux in N-S direction [hPa]
REAL(fp) :: ymass(ilo_gl:ihi_gl, julio_gl:jhi_gl, k1_gl:k2_gl)

! Change of surface pressure from met field pressure [hPa]

```

```

REAL(fp)  :: dps(i1_gl:i2_gl, ju1_gl:j2_gl)

! CTM surface pressure tendency [hPa]
REAL(fp)  :: dps_ctm(i1_gl:i2_gl, ju1_gl:j2_gl)

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
 Declare all REAL variables as REAL(fp).

3.2.6 Do_Press_Fix_Llnl

Subroutine Do_Press_Fix_Llnl fixes the mass fluxes to match the met field pressure tendency.

INTERFACE:

```

SUBROUTINE Do_Press_Fix_Llnl
& (geofac_pc, geofac, dbk, dps, dps_ctm, rel_area,
&  xmass, ymass, xmass_fixed, ymass_fixed)

```

INPUT PARAMETERS:

```

! Special geometrical factor (geofac) for Polar cap
REAL(fp), INTENT(IN)  :: geofac_pc

! Geometrical factor for meridional advection; geofac uses
! correct spherical geometry, and replaces acospi as the
! meridional geometrical factor in tpcore
REAL(fp), INTENT(IN)  :: geofac(ju1_gl:j2_gl)

! Difference in bi across layer - the dSigma term
REAL(fp), INTENT(IN)  :: dbk(k1:k2)

! Change of surface pressure from met field pressure [hPa]
REAL(fp), INTENT(IN)  :: dps(i1:i2, ju1:j2)

! Relative surface area of grid box (fraction)
REAL(fp), INTENT(IN)  :: rel_area(i1:i2, ju1:j2)

! Horizontal mass fluxes in E-W and N-S directions [hPa]
REAL(fp), INTENT(IN)  :: xmass(ilo:ihi, julio:jhi, k1:k2)
REAL(fp), INTENT(IN)  :: ymass(ilo:ihi, julio:jhi, k1:k2)

```

OUTPUT PARAMETERS:

```
! Sum over vertical of dpi calculated from original mass fluxes [hPa]
REAL(fp), INTENT(OUT) :: dps_ctm(i1:i2, ju1:j2)
```

```
! Horizontal mass flux in E-W and N-S directions after fixing [hPa]
REAL(fp), INTENT(OUT) :: xmass_fixed(ilo:ihi, julio:jhi, k1:k2)
REAL(fp), INTENT(OUT) :: ymass_fixed(ilo:ihi, julio:jhi, k1:k2)
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.7 Average_Press_Poles

Subroutine Average_Press_Poles averages pressure at the Poles when the Polar cap is enlarged. It makes the last two latitudes equal.

INTERFACE:

```
SUBROUTINE Average_Press_Poles
& (rel_area, press)
```

INPUT PARAMETERS:

```
! Relative surface area of grid box (fraction)
REAL(fp), INTENT(IN) :: rel_area(i1:i2, ju1:j2)
```

OUTPUT PARAMETERS:

```
! Surface pressure [hPa]
REAL(fp), INTENT(INOUT) :: press (ilo:ihi, julio:jhi)
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.8 Convert_Winds

Subroutine Convert_Winds converts winds on A or C grid to Courant # on C grid.

INTERFACE:

```
SUBROUTINE Convert_Winds
& (igd, tdt, cosp, crx, cry, uu, vv)
```

USES:

```
USE CMN_SIZE_MOD ! Size parameters
USE PhysConstants ! Re, PI
```

INPUT PARAMETERS:

```
! A or C grid
INTEGER, INTENT(IN) :: igd

! Model time step [s]
REAL(fp), INTENT(IN) :: tdt

! Cosine of grid box centers
REAL(fp), INTENT(IN) :: cosp(ju1_g1:j2_g1)

! Wind velocity in E-W (UU) and N-S (VV) directions at t1+tdt/2 [m/s]
REAL(fp), INTENT(IN) :: uu (ilo:ihi, julo:jhi, k1:k2)
REAL(fp), INTENT(IN) :: vv (ilo:ihi, julo:jhi, k1:k2)
```

OUTPUT PARAMETERS:

```
! Courant number in E-W (CRX) and N-S (CRY) directions
REAL(fp), INTENT(OUT) :: crx (ilo:ihi, julo:jhi, k1:k2)
REAL(fp), INTENT(OUT) :: cry (ilo:ihi, julo:jhi, k1:k2)
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REMARKS:

Use GEOS-CHEM physical constants Re, PI to be consistent with other usage everywhere (bmy, 5/5/03)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.9 Calc_Horiz_Mass_Flux

Subroutine Calc_Horiz_Mass_Flux calculates the horizontal mass flux for non-GISS met data.

INTERFACE:

```

SUBROUTINE Calc_Horiz_Mass_Flux
& (cose, delpm, uu, vv, xmass, ymass, tdt, cosp)

```

USES:

```

USE CMN_SIZE_MOD ! Size parameters
USE PhysConstants ! Re, Pi

```

INPUT PARAMETERS:

```

! Timestep [s]
REAL(fp), INTENT(IN)  :: tdt

! Cosine of grid box edges
REAL(fp), INTENT(IN)  :: cose (ju1_g1:j2_g1)

! Cosine of grid box centers
REAL(fp), INTENT(IN)  :: cosp (ju1_g1:j2_g1)

! Pressure thickness, the pseudo-density in a
! hydrostatic system at t1+tdt/2 (approximate) [hPa]
REAL(fp), INTENT(IN)  :: delpm(ilo:ihi, julio:jhi, k1:k2)

! E-W (UU) and N-S (VV) winds [m/s]
REAL(fp), INTENT(IN)  :: uu (ilo:ihi, julio:jhi, k1:k2)
REAL(fp), INTENT(IN)  :: vv (ilo:ihi, julio:jhi, k1:k2)

```

OUTPUT PARAMETERS:

```

! Horizontal mass flux in E-W and N-S directions [hPa]
REAL(fp), INTENT(OUT) :: xmass(ilo:ihi, julio:jhi, k1:k2)
REAL(fp), INTENT(OUT) :: ymass(ilo:ihi, julio:jhi, k1:k2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REMARKS:

Use GEOS-CHEM physical constants Re, PI to be consistent with other usage everywhere (bmy, 5/5/03)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.10 Calc_Divergence

Subroutine Calc_Divergence calculates the divergence.

INTERFACE:

```
SUBROUTINE Calc_Divergence
& (do_reduction, geofac_pc, geofac, dpi, xmass, ymass)
```

INPUT PARAMETERS:

```
! Set to F if called on Master; set to T if called by Slaves
! (NOTE: this doesn't seem to be used!)
LOGICAL, INTENT(IN)    :: do_reduction

! Special geometrical factor (geofac) for Polar cap
REAL(fp), INTENT(IN)  :: geofac_pc

! geometrical factor for meridional advection; geofac uses
! correct spherical geometry, and replaces acosp as the
! meridional geometrical factor in tpcore
REAL(fp), INTENT(IN)  :: geofac(ju1_gl:j2_gl)

! horizontal mass fluxes in E-W and N-S directions [hPa]
REAL(fp), INTENT(IN)  :: xmass (ilo:ihi, julo:jhi, k1:k2)
REAL(fp), INTENT(IN)  :: ymass (ilo:ihi, julo:jhi, k1:k2)
```

INPUT/OUTPUT PARAMETERS:

```
! Divergence at a grid point; used to calculate vertical motion [hPa]
REAL(fp), INTENT(INOUT) :: dpi (i1:i2, ju1:j2, k1:k2)
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Declare all REAL variables as REAL(fp).

3.2.11 Set_Press_Terms

Subroutine Set_Press_Terms sets the pressure terms.

INTERFACE:

```

SUBROUTINE Set_Press_Terms
& (dap, dbk, pres1, pres2, delp1, delpm, pu)

```

INPUT PARAMETERS:

```

! Pressure difference across layer from (ai * pt) term [hPa]
REAL(fp), INTENT(IN)  :: dap  (k1:k2)

! Difference in bi across layer - the dSigma term
REAL(fp), INTENT(IN)  :: dbk  (k1:k2)

! Surface pressure at t1 [hPa]
REAL(fp), INTENT(IN)  :: pres1(ilo:ihi, julo:jhi)

! Surface pressure at t1+tdt [hPa]
REAL(fp), INTENT(IN)  :: pres2(ilo:ihi, julo:jhi)

```

OUTPUT PARAMETERS:

```

! Pressure thickness, the psudo-density in a
! hydrostatic system at t1 [hPa]
REAL(fp), INTENT(OUT) :: delp1(ilo:ihi, julo:jhi, k1:k2)

! Pressure thickness, the psudo-density in a
! hydrostatic system at t1+tdt/2 (approximate) [hPa]
REAL(fp), INTENT(OUT) :: delpm(ilo:ihi, julo:jhi, k1:k2)

! Pressure at edges in "u" [hPa]
REAL(fp), INTENT(OUT) :: pu  (ilo:ihi, julo:jhi, k1:k2)

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
 Declare all REAL variables as REAL(fp).

3.2.12 Do_Divergence_Pole_Sum

Do_Divergence_Pole_Sum sets the divergence at the Poles.

INTERFACE:

```

SUBROUTINE Do_Divergence_Pole_Sum
& (do_reduction, geofac_pc, dpi, ymass)

```

INPUT PARAMETERS:

```

! Set to T if called on Master; set to F if called by Slaves
! (NOTE: This does not seem to be used!)
LOGICAL :: do_reduction

```

```

! Special geometrical factor (geofac) for Polar cap
REAL(fp) :: geofac_pc

```

```

! horizontal mass flux in N-S direction [hPa]
REAL(fp) :: ymass(ilo:ihi, julo:jhi, k1:k2)

```

OUTPUT PARAMETERS:

```

! Divergence at a grid point; used to calculate vertical motion [hPa]
REAL(fp) :: dpi ( i1:i2, j1:j2, k1:k2)

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
 Declare all REAL variables as REAL(fp).

3.2.13 Xpavg

!description: Subroutine Xpavg replaces each element of a vector with the average of the entire array. (bmy, 5/7/03)

INTERFACE:

```

SUBROUTINE Xpavg( P, IM )

```

USES:

```

! References to F90 modules
USE ERROR_MOD, ONLY : ERROR_STOP

```

INPUT PARAMETERS:

```

! Dimension of P
INTEGER, INTENT(IN) :: IM

```

INPUT/OUTPUT PARAMETERS:

```

! 1-D vector to be averaged
REAL(fp), INTENT(INOUT) :: P(IM)

```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
Now make all REAL variables REAL(fp).

3.2.14 Init_Pjc_Pfix

Subroutine Init_Pjc_Pfix allocates and initializes module arrays and variables. GMI dimension variables will be used for compatibility with the Phil Cameron-Smith P-fixer. (bdf, bmy, 5/8/03)

INTERFACE:

SUBROUTINE Init_Pjc_Pfix

USES:

! References to F90 modules
USE GC_GRID_MOD, ONLY : GET_AREA_M2, GET_YMID_R
USE ERROR_MOD, ONLY : ALLOC_ERR, ERROR_STOP
USE PRESSURE_MOD, ONLY : GET_AP, GET_BP

USE CMN_SIZE_MOD ! Size parameters
USE PhysConstants ! Re, PI, etc...

AUTHOR:

Brendan Field and Bob Yantosca (5/8/03)

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID_R(I,J,L) from grid_mod.F90

3.2.15 Cleanup_Pjc_Pfix

Subroutine Cleanup_Pjc_Pfix deallocates all module arrays (bmy, 5/8/03)

INTERFACE:

SUBROUTINE Cleanup_Pjc_Pfix

REVISION HISTORY:

02 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.

3.3 Fortran: Module Interface tpcore_bc_mod.F

Module TPCORE_BC_MOD contains modules and variables which are needed to save and read TPCORE nested-grid boundary conditions to/from disk.

INTERFACE:

```
MODULE TPCORE_BC_MOD
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
IMPLICIT NONE
PRIVATE
```

PUBLIC DATA MEMBERS:

```
!-----
! IO_W : Lon offset of TPCORE REGION [# boxes]
! JO_W : Lat offset of TPCORE REGION [# boxes]
! IM_W : Lon extent of TPCORE REGION [# boxes]
! JM_W : Lat extent of TPCORE REGION [# boxes]
! I1_W : Lower left-hand (LL) lon index of NESTED WINDOW
! J1_W : Lower left-hand (LL) lat index of NESTED WINDOW
! I2_W : Upper right-hand (UR) lon index of NESTED WINDOW
! J2_W : Upper right-hand (UR) lat index of NESTED WINDOW
! IGZD : ???
! Please also see the diagram in the REMARKS section.
!-----
INTEGER, PUBLIC :: IO_W, JO_W
INTEGER, PUBLIC :: IO_E, JO_E
INTEGER, PUBLIC :: I1_W, J1_W
INTEGER, PUBLIC :: I2_W, J2_W
INTEGER, PUBLIC :: IM_W, JM_W
INTEGER, PUBLIC :: IGZD
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: INIT_TPCORE_BC
PUBLIC :: DO_WINDOW_TPCORE_BC
PUBLIC :: SET_CLEAN_BC
PUBLIC :: SAVE_GLOBAL_TPCORE_BC
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: OPEN_BC_FILE
PRIVATE :: CLEAN_WINDOW_TPCORE_BC
PRIVATE :: READ_WINDOW_TPCORE_BC
PRIVATE :: GET_4x5_BC
PRIVATE :: GET_4x5_BC_INIT
PRIVATE :: GET_2x25_BC
PRIVATE :: GET_2x25_BC_INIT
PRIVATE :: ITS_TIME_FOR_BC
PRIVATE :: CLEANUP_TPCORE_BC
```

REMARKS:

Reference Diagram:

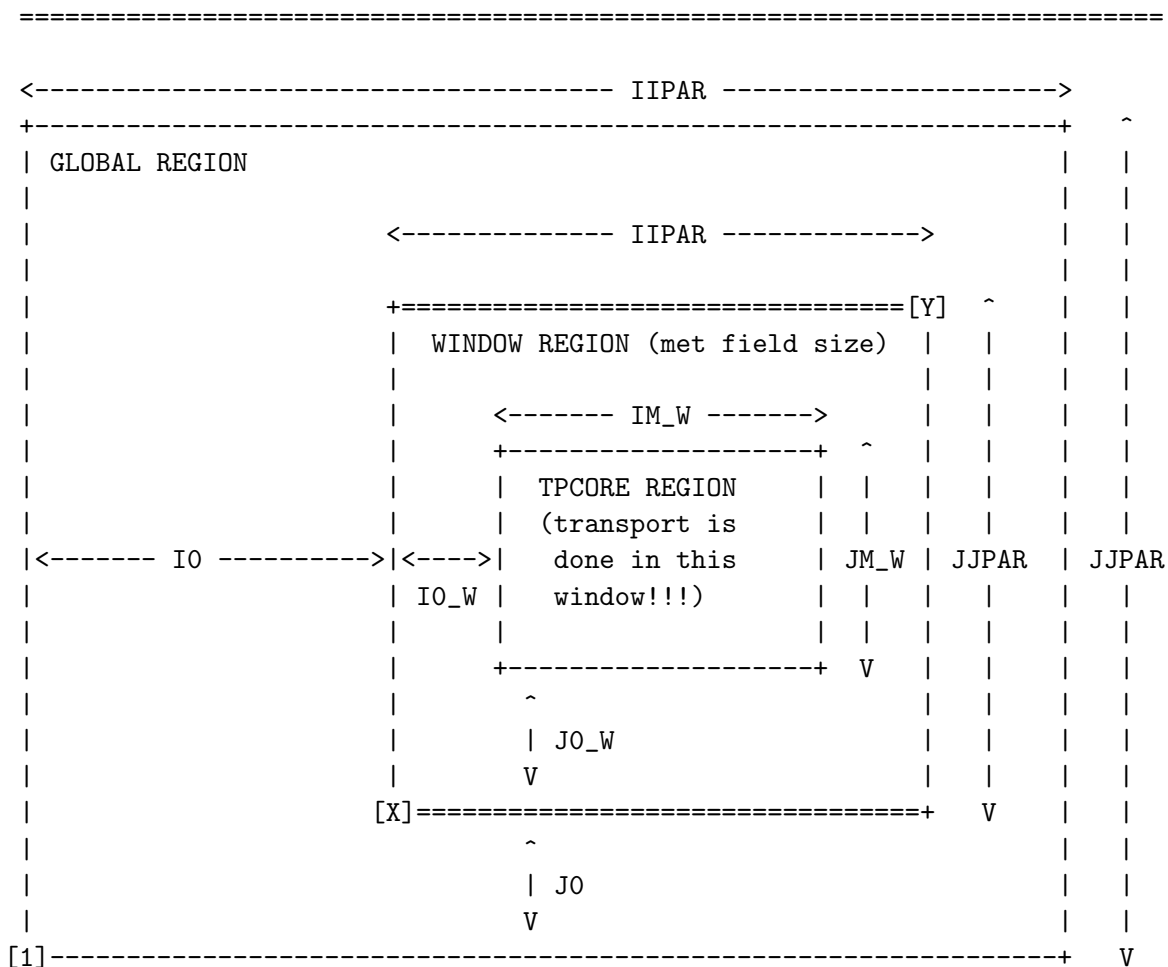


DIAGRAM NOTES:

- (a) The outermost box ("Global Region") is the global grid size. This region has IIPAR boxes in longitude and JJPARG boxes in latitude. The origin of the "Global Region" is at the south pole, at the lower left-hand corner (point [1]).
- (b) The next innermost box ("Window Region") is the nested-grid window. This region has IIPAR boxes in longitude and JJPARG boxes in latitude. This is the size of the trimmed met fields that will be used for a 1 x 1 "nested-grid" simulation.
- (c) The innermost region ("TPCORE Region") is the actual area in which TPCORE transport will be performed. Note that this region is smaller than the "Window Region". It is set up this way since a cushion of grid boxes is needed TPCORE Region for boundary conditions.
- (d) IO is the longitude offset (# of boxes) and JO is the latitude offset

- (# of boxes) which translate between the "Global Region" and the "Window Region".
- (e) IO_W is the longitude offset (# of boxes), and JO_W is the latitude offset (# of boxes) which translate between the "Window Region" and the "TPCORE Region".
 - (f) The lower left-hand corner of the "Window Region" (point [X]) has longitude and latitude indices (I1_W, J1_W). Similarly, the upper right-hand corner (point [Y]) has longitude and latitude indices (I2_W, J2_W).
 - (g) Note that if IO=0, JO=0, IO_W=0, JO_W=0, IIPAR=IIPAR, JJPARG=JJPARG specifies a global simulation. In this case the "Window Region" totally coincides with the "Global Region".
 - (h) In order for the nested-grid to work we must save out concentrations over the WINDOW REGION from a coarse model (e.g. 4x5) corresponding to the same WINDOW REGION at 1x1. These concentrations are copied along the edges of the 1x1 WINDOW REGION and are thus used as boundary conditions for TPCORE.

REVISION HISTORY:

- 04 Mar 2003 - R. Yantosca - Initial version
- (1) Bug fix for LINUX w/ TIMESTAMP_STRING (bmy, 9/29/03)
- (2) Now references "tracer_mod.f", "directory_mod.f", and "logical_mod.f" (bmy, 7/20/04)
- (3) Now get HALFPOLAR for GEOS or GCAP grids (bmy, 6/28/05)
- (4) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (5) Rename arguments in GET_4x5_BC to avoid name conflict (bmy, 10/24/05)
- (6) Now use EXPAND_DATE instead of obsolete DATE_STRING (bmy, 3/15/06)
- (7) Added 2x2.5 boundary condition output (created GET_2x25_BC).
Added multi-boundary condition output (NA, EU, CH and Custom region).
Internally defined boundary condition regions for NA, EU and CH.
(amv, bmy, 12/18/09)
- 16 Feb 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 15 May 2012 - R. Yantosca - Added ProTeX headers
- 06 Aug 2012 - R. Yantosca - Now make IU_BC, IU_BC_NA, IU_BC_EU, IU_BC_CH local variables for findFreeLUN
- 07 Sep 2012 - R. Yantosca - Minor fixes for numerical stability
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 21 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 01 Apr 2015 - R. Yantosca - Now add IO_E and JO_E
- 03 Aug 2015 - M. Sulprizio - Now make BC and BC_CU separate arrays to avoid confusion and error
- 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer compatible with the FlexChem implementation

29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
 24 Aug 2017 - M. Sulprizio- Remove support for GEOS-5 nested grids

3.3.1 set_clean_bc

Subroutine SET_CLEAN_BC initializes the CLEAN_BC logical flag. CLEAN_BC decides whether or not we will zero the nested-grid tpcore boundary conditions.

INTERFACE:

```
SUBROUTINE SET_CLEAN_BC( THIS_CLEAN_BC )
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: THIS_CLEAN_BC
```

REVISION HISTORY:

04 Mar 2003 - R. Yantosca - Initial versioni
 15 May 2012 - R. Yantosca - Added ProTeX headers

3.3.2 open_bc_file

Subroutine OPEN_BC_FILE opens the file which contains boundary conditions saved from the coarse-grid WINDOW REGION for either reading or writing.

INTERFACE:

```
SUBROUTINE OPEN_BC_FILE( am_I_Root, Input_Opt, RC,
&                          WINDOW,   FOR_READ,  FOR_WRITE )
```

USES:

```
#if defined( BPCH_TPBC )
  USE BPCH2_MOD,      ONLY : OPEN_BPCH2_FOR_WRITE
  USE BPCH2_MOD,      ONLY : OPEN_BPCH2_FOR_READ
#endif
  USE ErrCode_Mod
  USE Input_Opt_Mod, ONLY : OptInput
  USE inquireMod,    ONLY : findFreeLUN
  USE TIME_MOD,      ONLY : EXPAND_DATE,  GET_NYMD
  USE TIME_MOD,      ONLY : ITS_A_NEW_DAY
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)   :: am_I_Root   ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)   :: Input_Opt   ! Input Options object
LOGICAL, INTENT(IN), OPTIONAL :: FOR_READ    ! Open file for read?
LOGICAL, INTENT(IN), OPTIONAL :: FOR_WRITE   ! Open file for write?
INTEGER, INTENT(IN)           :: WINDOW      ! Specifies nested region
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REMARKS:

Pass these values via the WINDOW argument to do the following actions;

```
WINDOW = 1 : Save BC's to file (Custom window,      aka "CU")
WINDOW = 2 : Save BC's to file (North America window, aka "NA")
WINDOW = 3 : Save BC's to file (Europe window,      aka "EU")
WINDOW = 4 : Save BC's to file (China/SE Asia window, aka "CH")
WINDOW = 5 : Read BC's from file
```

REVISION HISTORY:

```
07 Mar 2003 - R. Yantosca - Initial version
(1 ) Now use ITS_A_NEW_DAY from "time_mod.f".  Now references TPBC_DIR
      from "directory_mod.f" (bmy, 7/20/04)
(2 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(3 ) DATE_STRING is now obsolete; use EXPAND_DATE instead (bmy, 3/15/06)
(4 ) Can now read files from different directories (amv, bmy, 12/18/09)
15 May 2012 - R. Yantosca - Added ProTeX headers
06 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
06 Aug 2012 - R. Yantosca - Cleaned up IF statement, added comments
06 Aug 2012 - R. Yantosca - Close existing files before opening new files
07 Aug 2012 - R. Yantosca - Now print LUN used to open file
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
26 Sep 2013 - R. Yantosca - Removed SEAC4RS C-preprocessor switch
```

3.3.3 save_global_tpcore_bc

Subroutine SAVE_GLOBAL_TPCORE_BC saves concentrations from the WINDOW REGION of a coarse-resolution model run to a bpch file. A new boundary conditions file is created for each day.

INTERFACE:

```
      SUBROUTINE SAVE_GLOBAL_TPCORE_BC( am_I_Root, Input_Opt,
&                                     State_Chm, RC           )
```

USES:

```
#if defined( BPCH_TPBC )
  USE BPCH2_MOD
#endif
  USE CMN_SIZE_MOD
  USE ErrCode_Mod
  USE Input_Opt_Mod,      ONLY : OptInput
  USE State_Chm_Mod,     ONLY : ChmState
```

```

USE TIME_MOD,          ONLY : GET_NYMD,   GET_NHMS
USE TIME_MOD,          ONLY : GET_TAU,    TIMESTAMP_STRING
USE UnitConv_Mod

```

INPUT PARAMETERS:

```

LOGICAL,              INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(Optional),      INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),      INTENT(INOUT)  :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,              INTENT(OUT)   :: RC           ! Success or failure?

```

REVISION HISTORY:

```

04 Mar 2003 - Y. Wang      - Initial version
(1 ) Now references N_TRACERS and STT from "tracer_mod.f".  Also now
      references TIMESTAMP_STRING from "time_mod.f".  (bmy, 7/20/04)
(2 ) Now call GET_HALFPOLAR from "bpch2_mod.f" to get the HALFPOLAR flag
      value for GEOS or GCAP grids (bmy, 6/28/05)
(3 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(4 ) Can now save files to different directories (amv, bmy, 12/18/09)
16 Feb 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
15 May 2012 - R. Yantosca - Added ProTeX headers
06 Aug 2012 - R. Yantosca - Added comments & cosmetic changes
06 Aug 2012 - R. Yantosca - Now use file unit variables from this module
                          instead of from GeosUtil/file_mod.F
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte
17 Feb 2014 - E. Lundgren - Move temporary unit change to vol/vol to within
                          this subroutine
13 Aug 2015 - E. Lundgren - Tracer units are now [kg/kg dry air]
30 Jun 2016 - R. Yantosca - Remove instances of STT.  Now get the advected
                          species ID from State_Chm%Map_Advect.
03 Aug 2016 - R. Yantosca - Remove temporary tracer removal code

```

3.3.4 do_window_tpcore_bc

Subroutine DO_WINDOW_TPCORE_BC is a driver routine for assigning TPCORE boundary conditions to the tracer array Spc.

INTERFACE:

```

SUBROUTINE DO_WINDOW_TPCORE_BC( am_I_Root, Input_Opt,
&                               State_Chm, RC           )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT) :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REVISION HISTORY:

```

07 Mar 2003 - R. Yantosca - Initial version
(1 ) Now references N_TRACERS and STT from "tracer_mod.f" (bmy, 7/20/04)
(2 ) Now can use 2 x 2.5 BC's (amv, bmy, 12/18/09)
16 Feb 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
08 Dec 2009 - R. Yantosca - Added ProTeX headers
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
13 Aug 2014 - E. Lundgren - Tracer units are now [kg/kg dry air]
03 Aug 2016 - R. Yantosca - Remove temporary tracer removal code

```

3.3.5 clean_window_tpcore_bc

Subroutine CLEAN_WINDOW_TPCORE_BC zeroes the boundary conditions array BC at each timestep. (bmy, 3/7/03, 12/18/09)

INTERFACE:

```

SUBROUTINE CLEAN_WINDOW_TPCORE_BC( am_I_Root, Input_Opt, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REVISION HISTORY:

07 Mar 2003 - M. Prather - Initial version
 (1) Now references N_TRACERS from "tracer_mod.f" (bmy, 7/20/04)
 (2) Now zeroes the arrays for the different regions (amv, bmy, 12/18/09)
 16 Feb 2011 - R. Yantosca - Add modifications for APM microphysics (G. Luo)
 15 May 2012 - R. Yantosca - Added ProTeX headers
 07 Sep 2012 - R. Yantosca - Simplify coding, remove parallel loops
 07 Sep 2012 - R. Yantosca - Now use 0e+0_fp instead of 0e0 to zero BC arrays
 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 26 Sep 2013 - R. Yantosca - Removed SEAC4RS C-preprocessor switch

3.3.6 read_window_tpcore_bc

Subroutine READ_WINDOW_TPCORE_BC reads tracer concentrations saved on the WINDOW REGION of a coarse-grid simulation (e.g. 4x5, 2x2.5). These concentrations will be used as boundary conditions for TPCORE transport.

INTERFACE:

```

SUBROUTINE READ_WINDOW_TPCORE_BC( am_I_Root, Input_Opt,
&                               State_Chm, RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE FILE_MOD,          ONLY : IOERROR
USE Input_Opt_Mod,    ONLY : OptInput
USE State_Chm_Mod,    ONLY : ChmState
USE TIME_MOD,         ONLY : GET_TAU, TIMESTAMP_STRING

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry state object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REVISION HISTORY:

07 Mar 2003 - R. Yantosca - Initial version
(1) LINUX has a problem putting a function call w/in a WRITE statement.
Now save output from TIMESTAMP_STRING to STAMP and print that.
(bmy, 9/29/03)
(2) Now references N_TRACERS from "tracer_mod.f" (bmy, 7/20/04)
(3) Rewritten to be more generic (amv, bmy, 12/18/09)
15 May 2012 - R. Yantosca - Added ProTeX headers
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC

3.3.7 get_4x5_bc_init

Function GET_4x5_BC_INIT initializes the MAP1x1 arrays.

INTERFACE:

```
SUBROUTINE GET_4x5_BC_INIT()
```

USES:

```
USE CMN_SIZE_MOD  
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID
```

REMARKS:

NOTE: This routine was originally written for the 1 x 1 nested grid, but this now works for other nested grid data. Keep variable names I_1x1, J_1x1, etc. unchanged for the present.

For now we will assume that we have saved tracer concentrations from a 4x5 window which overlays the corresponding 1x1 WINDOW REGION. These 4x5 tracer concentrations are used as boundary conditions for TPCORE.

REVISION HISTORY:

07 Mar 2003 - Y. Wang, R. Yantosca - Initial version
(1) Rename arguments to avoid conflict w/ I1x1, J1x1 parameters in CMN_SIZE. (bmy, 10/24/05)
01 Mar 2012 - R. Yantosca - Now use GET_XMID(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
15 May 2012 - R. Yantosca - Added ProTeX headers
10 Nov 2014 - C. Keller - Split off from GET_4x5_BC

3.3.8 get_4x5_bc

Function GET_4x5_BC returns a value from the 4x5 BC boundary conditions array at the location of a nested grid box.

INTERFACE:

```
FUNCTION GET_4x5_BC( I_1x1, J_1x1, L_1x1, N_1x1 ) RESULT( VALUE )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I_1x1 ! Nested-grid lon    index
INTEGER, INTENT(IN) :: J_1x1 ! Nested-grid lat    index
INTEGER, INTENT(IN) :: L_1x1 ! Nested-grid level  index
INTEGER, INTENT(IN) :: N_1x1 ! Nested-grid tracer index
```

RETURN VALUE:

```
REAL(fp)           :: VALUE ! 4 x 5 BC @ location of nested grid box
```

REMARKS:

NOTE: This routine was originally written for the 1 x 1 nested grid, but this now works for other nested grid data. Keep variable names I_1x1, J_1x1, etc. unchanged for the present.

For now we will assume that we have saved tracer concentrations from a 4x5 window which overlays the corresponding 1x1 WINDOW REGION. These 4x5 tracer concentrations are used as boundary conditions for TPCORE.

REVISION HISTORY:

```
07 Mar 2003 - Y. Wang, R. Yantosca - Initial version
(1 ) Rename arguments to avoid conflict w/ I1x1, J1x1 parameters in
      CMN_SIZE. (bmy, 10/24/05)
01 Mar 2012 - R. Yantosca - Now use GET_XMID(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
15 May 2012 - R. Yantosca - Added ProTeX headers
10 Nov 2014 - C. Keller   - Now initialize MAP1x1 array in separate routine
```

3.3.9 get_2x25_bc_init

Function GET_2x25_BC_INIT initializes the 2x25 mapping values.

INTERFACE:

```
SUBROUTINE GET_2x25_BC_INIT()
```

USES:

```
USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID
```

RETURN VALUE:

REMARKS:

NOTE: This routine was originally written for the 1 x 1 nested grid, but this now works for other nested grid data. Keep variable names I_1x1, J_1x1, etc. unchanged for the present.

For now we will assume that we have saved tracer concentrations from a 2 x 2.5 window which overlays the corresponding NESTED WINDOW REGION. These 2 x 2.5 tracer concentrations are used as boundary conditions for TPCORE.

REVISION HISTORY:

18 Dec 2009 - A. van Donkeelar - Initial version
 01 Mar 2012 - R. Yantosca - Now use GET_XMID(I,J,L) from grid_mod.F90
 01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
 15 May 2012 - R. Yantosca - Added ProTeX headers
 10 Nov 2014 - C. Keller - Split off from GET_2x25_BC. Now account for high-res midpoints on edges.

3.3.10 get_2x25_bc

Function GET_2x25_BC returns a value from the 2 x 2.5 BC boundary conditions array at the location of a nested grid box.

INTERFACE:

```
FUNCTION GET_2x25_BC( I_1x1, J_1x1, L_1x1, N_1x1 ) RESULT( VALUE )
```

USES:**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I_1x1 ! Nested-grid lon    index
INTEGER, INTENT(IN) :: J_1x1 ! Nested-grid lat    index
INTEGER, INTENT(IN) :: L_1x1 ! Nested-grid level  index
INTEGER, INTENT(IN) :: N_1x1 ! Nested-grid tracer index
```

RETURN VALUE:

```
REAL(fp)           :: VALUE ! 2 x 2.5 BC @ location of nested grid box
```

REMARKS:

NOTE: This routine was originally written for the 1 x 1 nested grid, but this now works for other nested grid data. Keep variable names I_1x1, J_1x1, etc. unchanged for the present.

For now we will assume that we have saved tracer concentrations from a 2 x 2.5 window which overlays the corresponding NESTED WINDOW REGION. These 2 x 2.5 tracer concentrations are used as boundary conditions for TPCORE.

REVISION HISTORY:

18 Dec 2009 - A. van Donkeelar - Initial version
01 Mar 2012 - R. Yantosca - Now use GET_XMID(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
15 May 2012 - R. Yantosca - Added ProTeX headers
10 Nov 2014 - C. Keller - Now initialize MAP1x1 array in separate routine
Also account for high-res midpoint on edges.

3.3.11 *its_time_for_bc*

Subroutine ITS_TIME_FOR_BC returns TRUE if it is time to read in the next set of boundary conditions for TPCORE, or FALSE otherwise.

INTERFACE:

```
FUNCTION ITS_TIME_FOR_BC() RESULT( FLAG )
```

USES:

```
USE TIME_MOD, ONLY : GET_ELAPSED_SEC
```

RETURN VALUE:

```
LOGICAL :: FLAG    ! =T if it's time to read BC's from disk
```

REVISION HISTORY:

05 Mar 2003 - R. Yantosca - Initial version
15 May 2012 - R. Yantosca - Added ProTeX headers
06 Feb 2018 - E. Lundgren - Replace GET_ELAPSED_MIN with GET_ELAPSED_SEC to
match new timestep units of seconds

3.3.12 *init_tpcore_bc*

Subroutine INIT_TPCORE_BC initializes module variables and arrays.

INTERFACE:

```
SUBROUTINE INIT_TPCORE_BC( am_I_Root, Input_Opt, State_Chm, RC )
```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,          ONLY : ALLOC_ERR
USE GC_GRID_MOD,        ONLY : GET_XOFFSET, GET_YOFFSET
USE GC_GRID_MOD,        ONLY : ITS_A_NESTED_GRID
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(ChmState),  INTENT(IN)  :: State_Chm     ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT) :: RC           ! Success or failure

```

REVISION HISTORY:

```

10 Feb 2003 - R. Yantosca - Initial version
(1 ) Now references N_TRACERS from "tracer_mod.f".  Now references LWINDO
      from "logical_mod.f".  Now references TPBC_DIR from "directory_mod.f".
      Now references ITS_A_NESTED_GRID from "grid_mod.f".  Also added
      arguments to take values from "input_mod.f". (bmy, 7/20/04)
15 May 2012 - R. Yantosca - Added ProTeX headers
07 Sep 2012 - R. Yantosca - Now use 0e+0_fp instead of 0e0 to zero BC arrays
17 Oct 2012 - M. Payer    - Bug fix: Define extent of coarse grid BC region
                          even if LWINDO_CH = False to avoid out-of-bounds
                          errors in array BC
05 Jun 2013 - K. Yu       - Define BC boundaries for GEOS_57 0.25 NA grid
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
26 Sep 2013 - R. Yantosca - Removed SEAC4RS C-preprocessor switch
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
01 Apr 2015 - L. Zhang    - IM_W, JM_W are no longer symmetric, but are
                          computed from IO_E and JO_E
06 Feb 2018 - E. Lundgren - Change timestep units from min to sec

```

3.3.13 cleanup_tpcore_bc

Subroutine CLEANUP_TPCORE_BC deallocates all module arrays.

INTERFACE:

```

SUBROUTINE CLEANUP_TPCORE_BC()

```

REVISION HISTORY:

```

04 Mar 2003 - R. Yantosca - Initial version
15 May 2012 - R. Yantosca - Added ProTeX headers

```

3.4 Fortran: Module Interface *tpcore_fvdas_mod.F90*

Overview

Module *Tpcore.Fvdas_Mod* contains routines for the TPCORE transport scheme, as implemented in the GMI model (cf. John Tannahill), based on Lin Rood 1995. The Harvard Atmospheric Chemistry Modeling Group has added modifications to implement the Philip-Cameron Smith pressure fixer for mass conservation. Mass flux diagnostics have also been added.

References

1. Lin, S.-J., and R. B. Rood, 1996: *Multidimensional flux form semi-Lagrangian transport schemes*, *Mon. Wea. Rev.*, **124**, 2046-2070.
2. Lin, S.-J., W. C. Chao, Y. C. Sud, and G. K. Walker, 1994: *A class of the van Leer-type transport schemes and its applications to the moisture transport in a General Circulation Model*, *Mon. Wea. Rev.*, **122**, 1575-1593.

Selecting E/W, N/S and vertical advection options

The flags IORD, JORD, KORD select which transport schemes are used in the E/W, N/S, and vertical directions, respectively. Here is a list of the possible values that IORD, JORD, KORD may be set to (original notes from S-J Lin):

1. 1st order upstream scheme (too diffusive, not a real option; it can be used for debugging purposes; this is THE only known "linear" monotonic advection scheme.).
2. 2nd order van Leer (full monotonicity constraint; see Lin et al 1994, MWR)
3. monotonic PPM* (Collela & Woodward 1984)
4. semi-monotonic PPM (same as 3, but overshoots are allowed)
5. positive-definite PPM (constraint on the subgrid distribution is only strong enough to prevent generation of negative values; both overshoots & undershoots are possible).
6. un-constrained PPM (nearly diffusion free; faster but positivity of the subgrid distribution is not guaranteed. Use this option only when the fields and winds are very smooth.
7. Huynh/Van Leer/Lin full monotonicity constraint. Only KORD can be set to 7 to enable the use of Huynh's 2nd monotonicity constraint for piece-wise parabolic distribution.

Recommended values:

- IORD=JORD=3 for high horizontal resolution.
- KORD=3 or 7

The implicit numerical diffusion decreases as `_ORD` increases. DO NOT use option 4 or 5 for non-positive definite scalars (such as Ertel Potential Vorticity).

In GEOS-Chem we have been using `IORD=3`, `JORD=3`, `KORD=7`. We have tested the OpenMP parallelization with these options. GEOS-Chem users who wish to use different (I,J,K)ORD options should consider doing single-processor vs. multi-processor tests to test the implementation of the parallelization.

GEOS-4 and GEOS-5 Hybrid Grid Definition

For GEOS-4 and GEOS-5 met fields, the pressure at the bottom edge of grid box (I,J,L) is defined as follows:

$$P_{edge}(I, J, L) = A_k(L) + [B_k(L) * P_{surface}(I, J)]$$

where

- $P_{surface}(I,J)$ is the "true" surface pressure at lon,lat (I,J)
- $A_k(L)$ has the same units as surface pressure [hPa]
- $B_k(L)$ is a unitless constant given at level edges

$A_k(L)$ and $B_k(L)$ are supplied to us by GMAO.

REMARKS:

```

Ak(L) and Bk(L) are defined at layer edges.
  ////////////////////////////////////////////////////////////////////
  / \ ----- Model top P=ak(1) ----- ak(1), bk(1)
  |
delp(1) | ..... q(i,j,1) .....
  |
  \ / ----- ak(2), bk(2)
  / \ ----- ak(k), bk(k)
  |
delp(k) | ..... q(i,j,k) .....
  |
  \ / ----- ak(k+1), bk(k+1)
  / \ ----- ak(km), bk(km)
  |
delp(km) | ..... q(i,j,km) .....
  |
  \ / -----Earth's surface P=Psfc ----- ak(km+1), bk(km+1)
  ////////////////////////////////////////////////////////////////////

```

Note: surface pressure can be of any unit (e.g., pascal or mb) as long as it is consistent with the definition of (ak, bk) defined above. Winds (u,v), ps, and q are assumed to be defined at the same points.

The latitudes are given to the initialization routine: `init_tpcore`.

INTERFACE:

```
MODULE Tpcore_FvDas_Mod
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

```
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: Init_Tpcore
```

```
PUBLIC  :: Exit_Tpcore
```

```
PUBLIC  :: Tpcore_FvDas
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: Average_Const_Poles
```

```
PRIVATE :: Set_Cross_Terms
```

```
PRIVATE :: Calc_Vert_Mass_Flux
```

```
PRIVATE :: Set_Jn_Js
```

```
PRIVATE :: Calc_Advec_Cross_Terms
```

```
PRIVATE :: Qckxyz
```

```
PRIVATE :: Set_Lmts
```

```
PRIVATE :: Set_Press_Terms
```

```
PRIVATE :: Calc_Courant
```

```
PRIVATE :: Calc_Divergence
```

```
PRIVATE :: Do_Divergence_Pole_Sum
```

```
PRIVATE :: Do_Cross_Terms_Pole_I2d2
```

```
PRIVATE :: Xadv_Dao2
```

```
PRIVATE :: Yadv_Dao2
```

```
PRIVATE :: Do_Yadv_Pole_I2d2
```

```
PRIVATE :: Do_Yadv_Pole_Sum
```

```
PRIVATE :: Xtp
```

```
PRIVATE :: Xmist
```

```
PRIVATE :: Fxppm
```

```
PRIVATE :: Lmtppm
```

```
PRIVATE :: Ytp
```

```
PRIVATE :: Ymist
```

```
PRIVATE :: Do_Ymist_Pole1_I2d2
```

```
PRIVATE :: Do_Ymist_Pole2_I2d2
```

```
PRIVATE :: Fyppm
```

```
PRIVATE :: Do_Fyppm_Pole_I2d2
```

```
PRIVATE :: Do_Ytp_Pole_Sum
```

```
PRIVATE :: Fzppm
```

```
PRIVATE :: Average_Press_Poles
```

```
!PRIVATE DATA MEMBERS:
```

```
REAL(fp), ALLOCATABLE, SAVE :: dtdx5(:)
```

```

REAL(fp), ALLOCATABLE, SAVE :: dtdy5(:)
REAL(fp), ALLOCATABLE, SAVE :: cosp(:)
REAL(fp), ALLOCATABLE, SAVE :: cose(:)
REAL(fp), ALLOCATABLE, SAVE :: gw(:)
REAL(fp), ALLOCATABLE, SAVE :: DLAT(:)

```

AUTHOR:

Original code from Shian-Jiann Lin, GMAO
 Modified for GMI model by John Tannahill, LLNL (jrt@llnl.gov)
 Implemented into GEOS-Chem by Claire Carouge (ccarouge@seas.harvard.edu)
 ProTeX documentation added by Bob Yantosca (yantosca@seas.harvard.edu)
 OpenMP parallelization added by Bob Yantosca (yantosca@seas.harvard.edu)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from the GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.
05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Added
                        OpenMP parallel loops in various routines (and
                        made some modifications to facilitate OpenMP).
01 Apr 2009 - C. Carouge - Modified OpenMp parallelization and move the
                        loops over vertical levels outside the
                        horizontal transport routines for reducing
                        processing time.
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
21 Nov 2014 - M. Yannetti - Added PRECISION_MOD
19 Jan 2016 - E. Lundgren - Consolidated bpch and netcdf diagnostics code

```

3.4.1 Init_Tpcore

Subroutine Init_Tpcore allocates and initializes all module variables,

INTERFACE:

```

SUBROUTINE Init_Tpcore( IM,      JM,  KM,          JFIRST,      &
                      JLAST,  NG,  MG,          dt,          &
                      ae,     clat, State_Diag, RC          )

```

USES:

```

USE PhysConstants
USE ErrCode_Mod
USE State_Diag_Mod, ONLY : DgnState

```

INPUT PARAMETERS:

```

INTEGER,      INTENT(IN)  :: IM      ! Global E-W dimension
INTEGER,      INTENT(IN)  :: JM      ! Global N-S dimension
INTEGER,      INTENT(IN)  :: KM      ! Vertical dimension
INTEGER,      INTENT(IN)  :: NG      ! large ghost width
INTEGER,      INTENT(IN)  :: MG      ! small ghost width
REAL(fp),    INTENT(IN)  :: dt      ! Time step in seconds
REAL(fp),    INTENT(IN)  :: ae      ! Earth's radius (m)
REAL(fp),    INTENT(IN)  :: clat(JM) ! latitude in radian
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT) :: JFIRST  ! Local first index for N-S axis
INTEGER,      INTENT(OUT) :: JLAST   ! Local last  index for N-S axis
INTEGER,      INTENT(OUT) :: RC      ! Success or failure

```

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.
- 09 Nov 2017 - R. Yantosca - Now add State_Diag, RC as arguments

3.4.2 Exit_Tpcore

Subroutine Exit_Tpcore deallocates all module variables.

INTERFACE:

```
SUBROUTINE Exit_Tpcore
```

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.
- 12 Feb 2015 - E. Lundgren - Add new diagnostic arrays for writing diagnostics ND24, ND25, and ND26 to netcdf.

3.4.3 Tpcore_FvDas

Subroutine `Tpcore_FvDas` takes horizontal winds on sigma (or hybrid sigma-p) surfaces and calculates mass fluxes, and then updates the 3D mixing ratio fields one time step (`tdt`). The basic scheme is a Multi-Dimensional Flux Form Semi-Lagrangian (FFSL) based on the van Leer or PPM (see Lin and Rood, 1995).

INTERFACE:

```

SUBROUTINE Tpcore_FvDas( dt,      ae,      IM,      JM,      KM,      &
                        JFIRST,  JLAST,  ng,      mg,      nq,      &
                        ak,      bk,      u,      v,      ps1,     &
                        ps2,     ps,     q,      iord,    jord,     &
                        kord,    n_adj,  XMASS,  YMASS,  FILL,     &
#if defined( BPCH_DIAG ) || defined( NC_DIAG )
  !%% Adding DiagnArrays for writing diagnostics to netcdf (ewl, 2/12/15).
  !%% MASSFLEW, MASSFLNS, and MASSFLUP are cumulative when BPCH=y.
  !%% They are instantaneous when using NETCDF
                        MASSFLEW, MASSFLNS, MASSFLUP,      &
#endif
                        AREA_M2, ND24, ND25, ND26, State_Diag )

```

USES:

```

! Include files w/ physical constants and met values
USE PhysConstants
USE ErrCode_Mod
USE State_Diag_Mod, ONLY : DgnState

```

INPUT PARAMETERS:

```

! Transport time step [s]
REAL(fp), INTENT(IN)  :: dt

! Earth's radius [m]
REAL(fp), INTENT(IN)  :: ae

! Global E-W, N-S, and vertical dimensions
INTEGER, INTENT(IN)   :: IM
INTEGER, INTENT(IN)   :: JM
INTEGER, INTENT(IN)   :: KM

! Latitude indices for local first box and local last box
! (NOTE: for global grids these are 1 and JM, respectively)
INTEGER, INTENT(IN)   :: JFIRST
INTEGER, INTENT(IN)   :: JLAST

! Primary ghost region
! (NOTE: only required for MPI parallelization; use 0 otherwise)
INTEGER, INTENT(IN)   :: ng

```

```

! Secondary ghost region
! (NOTE: only required for MPI parallelization; use 0 otherwise)
INTEGER, INTENT(IN)    :: mg

! Ghosted latitudes (3 required by PPM)
! (NOTE: only required for MPI parallelization; use 0 otherwise)
INTEGER, INTENT(IN)    :: nq

! Flags to denote E-W, N-S, and vertical transport schemes
INTEGER, INTENT(IN)    :: iord
INTEGER, INTENT(IN)    :: jord
INTEGER, INTENT(IN)    :: kord

! Number of adjustments to air_mass_flux (0 = no adjustment)
INTEGER, INTENT(IN)    :: n_adj

! Ak and Bk coordinates to specify the hybrid grid
! (see the REMARKS section below)
REAL(fp), INTENT(IN)   :: ak(KM+1)
REAL(fp), INTENT(IN)   :: bk(KM+1)

! u-wind (m/s) at mid-time-level (t=t+dt/2)
REAL(fp), INTENT(IN)   :: u(:, :, :)

! E/W and N/S mass fluxes [kg/s]
! (These are computed by the pressure fixer, and passed into TPCORE)
REAL(fp), INTENT(IN)   :: XMASS(:, :, :)
REAL(fp), INTENT(IN)   :: YMASS(:, :, :)

! Grid box surface area for mass flux diag [m2]
REAL(fp), INTENT(IN)   :: AREA_M2(JM)

! Diagnostic flags
INTEGER, INTENT(IN)    :: ND24    ! Turns on E/W    flux diagnostic
INTEGER, INTENT(IN)    :: ND25    ! Turns on N/S    flux diagnostic
INTEGER, INTENT(IN)    :: ND26    ! Turns on up/down flux diagnostic

LOGICAL, INTENT(IN)    :: FILL    ! Fill negatives ?

```

INPUT/OUTPUT PARAMETERS:

```

! V-wind (m/s) at mid-time-level (t=t+dt/2)
REAL(fp), INTENT(INOUT) :: v(:, :, :)

! surface pressure at current time
REAL(fp), INTENT(INOUT) :: ps1(IM, JFIRST:JLAST)

! surface pressure at future time=t+dt

```

```
REAL(fp), INTENT(INOUT) :: ps2(IM, JFIRST:JLAST)
```

```
! Tracer "mixing ratios" [kg tracer/moist air kg]
```

```
REAL(fp), INTENT(INOUT), TARGET :: q(:,:,:,)
```

```
! Diagnostics state object
```

```
TYPE(DgnState), INTENT(INOUT) :: State_Diag
```

```
#if defined( BPCH_DIAG ) || defined( NC_DIAG )
```

```
! E/W, N/S, and up/down diagnostic mass fluxes
```

```
REAL(fp), INTENT(INOUT) :: MASSFLEW(:,:,:,) ! for ND24 diagnostic
```

```
REAL(fp), INTENT(INOUT) :: MASSFLNS(:,:,:,) ! for ND25 diagnostic
```

```
REAL(fp), INTENT(INOUT) :: MASSFLUP(:,:,:,) ! for ND26 diagnostic
```

```
#endif
```

OUTPUT PARAMETERS:

```
! "Predicted" surface pressure [hPa]
```

```
REAL(fp), INTENT(OUT) :: ps(IM, JFIRST:JLAST)
```

AUTHOR:

```
Original code from Shian-Jiann Lin, DAO)
```

```
John Tannahill, LLNL (jrt@llnl.gov)
```

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.
- 01 Apr 2009 - C. Carouge - Modified OpenMp parallelization and move the loops over vertical levels outside the horizontal transport routines for reducing processing time.
- 03 Dec 2009 - C. Carouge - Modify declarations of MASSFLEW, MASSFLNS and MASSFLUP to save memory space.
- 30 May 2013 - S. Farina - For TOMAS, zero out UA and VA variables
- 04 Jun 2013 - R. Yantosca - Use assumed-shape declarations for XMASS, YMASS, U, V, and Q arrays. These arrays are used to pass pointer references, so this may help to reduce the creation of array temporaries, which will reduce memory.
- 5 Jun 2013 - R. Yantosca - Avoid array temporary in call to FZPPM

15 Jun 2015 - E. Lundgren - Now use kg/kg total air as tracer units
(previously v/v)
01 Jul 2015 - E. Lundgren - Set tracer conc to small positive number if
negative at end of advection (occurs at poles)
09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

3.4.4 Average_Const_Poles

Subroutine Average_Const_Poles averages the species concentrations at the Poles when the Polar cap is enlarged. It makes the last two latitudes equal.

INTERFACE:

```

SUBROUTINE Average_Const_Poles( dap ,   dbk,   rel_area, pctm1, const1, &
                                JU1_GL, J2_GL, I2_GL,   I1,   I2,   &
                                JU1,   J2,   ILO,   &
                                IHI,   JUL0,  JHI )

```

INPUT PARAMETERS:

```

! Global latitude indices of the South Pole and North Pole
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Global max longitude index
INTEGER, INTENT(IN)  :: I2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1, I2
INTEGER, INTENT(IN)  :: JU1, J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO, IHI
INTEGER, INTENT(IN)  :: JUL0, JHI

! Pressure difference across layer from (ai * pt) term [hPa]
REAL(fp), INTENT(IN)  :: dap

! Difference in bi across layer - the dSigma term
REAL(fp), INTENT(IN)  :: dbk

! Relative surface area of grid box [fraction]
REAL(fp), INTENT(IN)  :: rel_area(JU1:J2)

! CTM surface pressure at t1 [hPa]
REAL(fp), INTENT(IN)  :: pctm1( ILO:IHI, JUL0:JHI )

```

INPUT/OUTPUT PARAMETERS:

```

! Species concentration, known at zone center [mixing ratio]
REAL(fp), INTENT(INOUT) :: const1( I1:I2, JU1:J2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.

3.4.5 Set_Cross_Terms

Subroutine Set_Cross_Terms sets the cross terms for E-W horizontal advection.

INTERFACE:

```
SUBROUTINE Set_Cross_Terms( crx,   cry,   ua, va, J1P,   J2P,   &
                           I1_GL, I2_GL, JU1_GL, J2_GL, ILO,   &
                           IHI,   JULO, JHI,   I1,   I2,   &
                           JU1,   J2,   CROSS )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)   :: J1P,   J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)   :: I1_GL, I2_GL
INTEGER, INTENT(IN)   :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)   :: I1,   I2
INTEGER, INTENT(IN)   :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)   :: ILO,   IHI
INTEGER, INTENT(IN)   :: JULO,   JHI

! Courant number in E-W direction
REAL(fp), INTENT(IN) :: crx(ILO:IHI, JULO:JHI)
```

```
! Courant number in N-S direction
REAL(fp), INTENT(IN) :: cry(ILO:IHI, JULO:JHI)
```

```
! Logical switch. If CROSS=T then cross-terms will be computed.
LOGICAL, INTENT(IN) :: CROSS
```

OUTPUT PARAMETERS:

```
! Average of Courant numbers from il and il+1
REAL(fp), INTENT(OUT) :: ua(ILO:IHI, JULO:JHI)
```

```
! Average of Courant numbers from ij and ij+1
REAL(fp), INTENT(OUT) :: va(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.6 Calc_Vert_Mass_Flux

Subroutine Calc_Vert_Mass_Flux calculates the vertical mass flux.

INTERFACE:

```
SUBROUTINE Calc_Vert_Mass_Flux( dbk, dps_ctm, dpi, wz, I1, &
                               I2, JU1, J2, K1, K2 )
```

INPUT PARAMETERS:

```
! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN) :: I1, I2
INTEGER, INTENT(IN) :: JU1, J2
INTEGER, INTENT(IN) :: K1, K2
```

```

! Difference in bi across layer - the dSigma term
REAL(fp), INTENT(IN) :: dbk(K1:K2)

! CTM surface pressure tendency; sum over vertical of dpi
! calculated from original mass fluxes [hPa]
REAL(fp), INTENT(IN) :: dps_ctm(I1:I2, JU1:J2)

! Divergence at a grid point; used to calculate vertical motion [mb]
REAL(fp), INTENT(IN) :: dpi(I1:I2, JU1:J2, K1:K2)

```

OUTPUT PARAMETERS:

```

! Large scale mass flux (per time step tdt) in the vertical
! direction as diagnosed from the hydrostatic relationship [hPa]
REAL(fp), INTENT(OUT) :: wz(I1:I2, JU1:J2, K1:K2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel DO loops

```

3.4.7 Set_Jn_Js

Subroutine Set_Jn_Js determines Jn and Js, by looking where Courant number is ≥ 1 .

INTERFACE:

```

SUBROUTINE Set_Jn_Js( jn, js,      crx,  ILO, IHI, JUL0, &
                    JHI, JU1_GL, J2_GL, J1P, J2P, I1,  &
                    I2,  JU1,     J2,    K1,  K2 )

```

INPUT PARAMETERS:

```

! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands

```

```
INTEGER, INTENT(IN)  :: J1P,    J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2
INTEGER, INTENT(IN)  :: K1,    K2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Courant number in E-W direction
REAL(fp), INTENT(IN) :: crx(ILO:IHI, JULO:JHI, K1:K2)
```

OUTPUT PARAMETERS:

```
! Northward of latitude index = jn; Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(OUT) :: jn(K1:K2)

! Southward of latitude index = js; Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(OUT) :: js(K1:K2)
```

AUTHOR:

Original code from Shian-Jiann Lin, (DAO)
John Tannahill, LLNL (jrt@llnl.gov)

REMARKS:

We cannot parallelize this subroutine because there is a CYCLE statement within the outer loop.

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.

3.4.8 Calc_Advec_Cross_Terms

Subroutine Calc_Advec_Cross_Terms calculates the advective cross terms.

INTERFACE:

```

SUBROUTINE Calc_Advec_Cross_Terms( jn,      js,      qq1,  qqu,  qqv,  &
                                   ua,      va,      J1P,  J2P,  I2_GL, &
                                   JU1_GL, J2_GL, ILO,  IHI,  JULO, &
                                   JHI,    I1,    I2,   JU1,  J2,    &
                                   CROSS )

```

INPUT PARAMETERS:

```

! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,    J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  ::          I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,     I2
INTEGER, INTENT(IN)  :: JU1,    J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,    IHI
INTEGER, INTENT(IN)  :: JULO,   JHI

! Northward of latitude index = jn, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(IN)  :: Jn

! Southward of latitude index = js, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(IN)  :: Js

! Species concentration (mixing ratio)
REAL(fp), INTENT(IN)  :: qq1(ILO:IHI, JULO:JHI)

! Average of Courant numbers from il and il+1
REAL(fp), INTENT(IN)  :: ua (ILO:IHI, JULO:JHI)

! Average of Courant numbers from ij and ij+1
REAL(fp), INTENT(IN)  :: va (ILO:IHI, JULO:JHI)

! Logical switch: If CROSS=T then cross-terms are being computed
LOGICAL, INTENT(IN)   :: CROSS

```

OUTPUT PARAMETERS:

```
! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(OUT) :: qqu(ILO:IHI, JULO:JHI)
```

```
! concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(OUT) :: qqv(ILO:IHI, JULO:JHI)
```

AUTHOR:

```
Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)
```

REVISION HISTORY:

```
05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.
05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel do loops.
01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.
```

3.4.9 Qckxyz

Subroutine Qckxyz routine checks for "filling".

INTERFACE:

```
SUBROUTINE Qckxyz( dq1, J1P, J2P,  JU1_GL, J2_GL, &
                  ILO, IHI, JULO, JHI,   I1,   &
                  I2,  JU1, J2,   K1,    K2 )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,   J2P
```

```
! Global min & max latitude (J) indices
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL
```

```
! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2
```

```

INTEGER, INTENT(IN)  :: K1,      K2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,     IHI
INTEGER, INTENT(IN)  :: JULO,    JHI

```

INPUT/OUTPUT PARAMETERS:

```

! Species density [hPa]
REAL(fp), INTENT(INOUT) :: dq1(ILO:IHI, JULO:JHI, K1:K2)

```

AUTHOR:

```

Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)

```

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel DO loops.

```

3.4.10 Set_Lmts

Subroutine Set_Lmts sets ILMT, JLMT, KLMT.

INTERFACE:

```

SUBROUTINE Set_Lmts( ilmt, jlmt, klmt, I2_GL, J2_GL, iord, jord, kord )

```

INPUT PARAMETERS:

```

! Global maximum longitude (I) and longitude (J) indices
INTEGER, INTENT(IN)  :: I2_GL, J2_GL

! Flags to denote E-W, N-S, and vertical transport schemes
! (See REMARKS section of routine Tpcore_FvDas for more info)
INTEGER, INTENT(IN)  :: iord, jord, kord

```

OUTPUT PARAMETERS:

```

! Controls various options in E-W advection
INTEGER, INTENT(OUT) :: ilmt

```

```
! Controls various options in N-S advection
```

```
INTEGER, INTENT(OUT) :: jlmt
```

```
! Controls various options in vertical advection
```

```
INTEGER, INTENT(OUT) :: klmt
```

AUTHOR:

```
Original code from Shian-Jiann Lin, DAO)
```

```
John Tannahill, LLNL (jrt@llnl.gov)
```

REVISION HISTORY:

```
05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.
```

```
05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent.
```

3.4.11 Set_Press_Terms

Subroutine Set_Press_Terms sets the pressure terms: DELP1, DELPM, PU.

INTERFACE:

```
SUBROUTINE Set_Press_Terms( dap,   dbk,   pres1,   pres2,   delp1,   &
                           delpm,   pu,    JU1_GL,   J2_GL,   ILO,     &
                           IHI,    JULO,   JHI,     J1P,    J2P,     &
                           I1,     I2,    JU1,     J2)
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN) :: J1P,   J2P
```

```
! Global min & max latitude (J) indices
INTEGER, INTENT(IN) :: JU1_GL, J2_GL
```

```
! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN) :: I1,     I2
INTEGER, INTENT(IN) :: JU1,    J2
```

```
! Local min & max longitude (I) and latitude (J) indices
```

```
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Pressure difference across layer from (ai * pt) term [hPa]
REAL(fp), INTENT(IN)  :: dap

! Difference in bi across layer - the dSigma term
REAL(fp), INTENT(IN)  :: dbk

! Surface pressure at t1 [hPa]
REAL(fp), INTENT(IN)  :: pres1(ILO:IHI, JULO:JHI)

! Surface pressure at t1+tdt [hPa]
REAL(fp), INTENT(IN)  :: pres2(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! Pressure thickness, the pseudo-density in a
! hydrostatic system at t1 [hPa]
REAL(fp), INTENT(OUT) :: delp1(ILO:IHI, JULO:JHI)

! Pressure thickness, the pseudo-density in a
! hydrostatic system at t1+tdt/2 (approximate) [hPa]
REAL(fp), INTENT(OUT) :: delpm(ILO:IHI, JULO:JHI)

! Pressure at edges in "u" [hPa]
REAL(fp), INTENT(OUT) :: pu(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO)
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.12 Calc_Courant

Subroutine Calc_Courant calculates courant numbers from the horizontal mass fluxes.

INTERFACE:

```

SUBROUTINE Calc_Courant( cose, delpm, pu,      xmass, ymass, crx, cry, &
                        J1P, J2P,  JU1_GL, J2_GL, ILO,  IHI, JULO, &
                        JHI, I1,  I2,      JU1,  J2 )

```

INPUT PARAMETERS:

```

! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,  J2P

! Global min & max latitude (J) indices
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Cosine of grid box edges
REAL(fp), INTENT(IN)  :: cose (JU1_GL:J2_GL)

! Pressure thickness, the pseudo-density in a hydrostatic system
! at t1+tdt/2 (approximate) (mb)
REAL(fp), INTENT(IN)  :: delpm(ILO:IHI, JULO:JHI)

! pressure at edges in "u" (mb)
REAL(fp), INTENT(IN)  :: pu   (iLO:IHI, JULO:JHI)

! horizontal mass flux in E-W and N-S directions [hPa]
REAL(fp), INTENT(IN)  :: xmass(ILO:IHI, JULO:JHI)
REAL(fp), INTENT(IN)  :: ymass(ILO:IHI, JULO:JHI)

```

OUTPUT PARAMETERS:

```

! Courant numbers in E-W and N-S directions
REAL(fp), INTENT(OUT) :: crx(ILO:IHI, JULO:JHI)
REAL(fp), INTENT(OUT) :: cry(ILO:IHI, JULO:JHI)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO)

John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.13 Calc_Divergence

Subroutine Calc_Divergence calculates the divergence.

INTERFACE:

```
SUBROUTINE Calc_Divergence( do_reduction, geofac_pc, geofac, dpi,  &
                             xmass,          ymass,      J1P,   J2P,  &
                             I1_GL,         I2_GL,      JU1_GL, J2_GL, &
                             ILO,           IHI,         JUL0,  JHI,  &
                             I1,           I2,          JU1,   J2  )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,   J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JUL0,  JHI

! Set to F if called on Master or T if called by Slaves
! (NOTE: This is only for MPI parallelization, for OPENMP it should be F)
```

```

LOGICAL, INTENT(IN)  :: do_reduction

! Special geometrical factor (geofac) for Polar cap
REAL(fp) , INTENT(IN)  :: geofac_pc

! Geometrical factor for meridional advection; geofac uses correct
! spherical geometry, and replaces acosp as the meridional geometrical
! factor in TPCORE
REAL(fp) , INTENT(IN)  :: geofac(JU1_GL:J2_GL)

! Horizontal mass flux in E/W and N/S directions [hPa]
REAL(fp) , INTENT(IN)  :: xmass(ILO:IHI, JULO:JHI)
REAL(fp) , INTENT(IN)  :: ymass(ILO:IHI, JULO:JHI)

```

OUTPUT PARAMETERS:

```

! Divergence at a grid point; used to calculate vertical motion [hPa]
REAL(fp), INTENT(OUT) :: dpi(I1:I2, JU1:J2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
 John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

```

3.4.14 Do_Divergence_Pole_Sum

Subroutine Do_Divergence_Pole_Sum sets the divergence at the Poles.

INTERFACE:

```

SUBROUTINE Do_Divergence_Pole_Sum( do_reduction, geofac_pc, dpi, ymass, &
                                   I1_GL,      I2_GL,      J1P, J2P,  &
                                   JU1_GL,     J2_GL,      ILO, IHI,  &
                                   JULO,       JHI,       I1,  I2,   &
                                   JU1,       J2)

```


INPUT PARAMETERS:

```

! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,    J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL,  I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,     I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Set to T if called on Master or F if called by slaves
! NOTE: This seems not to be used here....)
LOGICAL, INTENT(IN)  :: do_reduction

! Special geometrical factor (geofac) for Polar cap
REAL(fp),  INTENT(in)  :: geofac_pc

! Horizontal mass flux in N-S direction [hPa]
REAL(fp),  INTENT(IN)  :: ymass(ILO:IHI, JULO:JHI)

```

OUTPUT PARAMETERS:

```

! Divergence at a grid point; used to calculate vertical motion [hPa]
REAL(fp),  INTENT(OUT) :: dpi(I1:I2, JU1:J2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
 John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                          Yeh with the TPCORE routines from GMI model.
                          This eliminates the polar overshoot in the
                          stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                          Declare all REAL variables as REAL(fp). Also
                          make sure all numerical constants are declared
                          with the "D" double-precision exponent. Added

```

OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.15 Do_Cross_Terms_Pole_I2d2

Subroutine Do_Cross_Terms_Pole_I2d2 sets "va" at the Poles.

INTERFACE:

```
SUBROUTINE Do_Cross_Terms_Pole_I2d2( cry,   va,  I1_GL, I2_GL, JU1_GL, &
                                     J2_GL, J1P, ILO,  IHI,  JULO,  &
                                     JHI,  I1,  I2,   JU1,  J2  )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edge of the South polar cap
! J1P=JU1_GL+1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL,  I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,     I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Courant number in N-S direction
REAL(fp), INTENT(IN) :: cry(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! Average of Courant numbers from ij and ij+1
REAL(fp), INTENT(OUT) :: va(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model.

This eliminates the polar overshoot in the stratosphere.

- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.
- 01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.16 Xadv_Dao2

Subroutine Xadv_Dao2 is the advective form E-W operator for computing the adx (E-W) cross term.

INTERFACE:

```
SUBROUTINE Xadv_Dao2( iad,   jn,   js,   adx,   qqv, &
                     ua,   ILO,   IHI,   JUL0,   JHI, &
                     JU1_GL, J2_GL, J1P, J2P,   I1, &
                     I2,   JU1,   J2)
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,   J2P

! Global min & max latitude (J) indices
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,     I2
INTEGER, INTENT(IN)  :: JU1,    J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,    IHI
INTEGER, INTENT(IN)  :: JUL0,   JHI

! if iad = 1, use 1st order accurate scheme;
! if iad = 2, use 2nd order accurate scheme
INTEGER, INTENT(IN)  :: iad

! Northward of latitude index = jn, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(IN)  :: jn

! southward of latitude index = js, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
```

```

INTEGER, INTENT(IN)  :: js

! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqv(ILO:IHI, JULO:JHI)

! Average of Courant numbers from il and il+1
REAL(fp), INTENT(IN)  :: ua(ILO:IHI, JULO:JHI)

```

OUTPUT PARAMETERS:

```

! Cross term due to E-W advection [mixing ratio]
REAL(fp), INTENT(OUT) :: adx(ILO:IHI, JULO:JHI)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

```

3.4.17 Yadv_Dao2

Subroutine Yadv_Dao2 is the advective form N-S operator for computing the ady (N-S) cross term.

INTERFACE:

```

SUBROUTINE Yadv_Dao2( iad,  ady,  qqu,  va,  I1_GL, &
                    I2_GL, JU1_GL, J2_GL, J1P, J2P, &
                    ILO,  IHI,  JULO,  JHI, I1,  &
                    I2,   JU1,   J2)

```

INPUT PARAMETERS:

```

! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,   J2P

```

```
! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1, I2
INTEGER, INTENT(IN)  :: JU1, J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO, IHI
INTEGER, INTENT(IN)  :: JULO, JHI

! If iad = 1, use 1st order accurate scheme;
! If iad = 2, use 2nd order accurate scheme
INTEGER, INTENT(IN)  :: iad

! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JULO:JHI)

! Average of Courant numbers from ij and ij+1
REAL(fp), INTENT(IN)  :: va(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! Cross term due to N-S advection (mixing ratio)
REAL(fp), INTENT(OUT) :: ady(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```
05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.
05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent.
01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.
```

3.4.18 Do_Yadv_Pole_I2d2

Subroutine Do_Yadv_Pole_I2d2 sets "qquwk" at the Poles.

INTERFACE:

```

SUBROUTINE Do_Yadv_Pole_I2d2 ( qqu, qquwk, I1_GL, I2_GL, JU1_GL, J2_GL, &
                               J1P, ILO, IHI, JULO, JHI, I1, &
                               I2, JU1, J2 )

```

INPUT PARAMETERS:

```

! Global latitude indices at the edges of the South polar cap
! J1P=JU1_GL+1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1, I2
INTEGER, INTENT(IN)  :: JU1, J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO, IHI
INTEGER, INTENT(IN)  :: JULO, JHI

! concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JULO:JHI)

```

OUTPUT PARAMETERS:

```

! qqu working array [mixing ratio]
REAL(fp), INTENT(OUT) :: qquwk(ILO:IHI, JULO-2:JHI+2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared

with the "D" double-precision exponent. Added
 OpenMP parallel DO loops.
 01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.19 Do_Yadv_Pole_Sum

Subroutine Do_Yadv_Pole_Sum sets the cross term due to N-S advection at the Poles.

INTERFACE:

```
SUBROUTINE Do_Yadv_Pole_Sum( ady, I1_GL, I2_GL, JU1_GL, J2_GL, J1P, &
                             ILO, IHI, JULO, JHI, I1, I2, &
                             JU1, J2)
```

INPUT PARAMETERS:

```
! Global latitude index at the edge of the South polar cap
! J1P=JU1_GL+1; for a polar cap of 1 latitude band
! J1P=JU1_GL+2; for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)    :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: I1_GL, I2_GL
INTEGER, INTENT(IN)    :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)    :: I1, I2
INTEGER, INTENT(IN)    :: JU1, J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: ILO, IHI
INTEGER, INTENT(IN)    :: JULO, JHI
```

OUTPUT PARAMETERS:

```
! Cross term due to N-S advection (mixing ratio)
REAL(fp), INTENT(INOUT) :: ady(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
 John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
 Yeh with the TPCORE routines from GMI model.
 This eliminates the polar overshoot in the
 stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops. Also make a logical to test if we are using an extended polar cap.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.20 Xtp

Subroutine Xtp does horizontal advection in the E-W direction.

INTERFACE:

```
SUBROUTINE Xtp( ilmt, jn, js, pu, crx, dq1, qqv, xmass, fx, &
               J1P, J2P, I2_GL, JU1_GL, J2_GL, ILO, IHI, JULO, JHI, &
               I1, I2, JU1, J2, iord )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)    :: J1P, J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: I2_GL
INTEGER, INTENT(IN)    :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)    :: I1, I2
INTEGER, INTENT(IN)    :: JU1, J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: ILO, IHI
INTEGER, INTENT(IN)    :: JULO, JHI

! Controls various options in E-W advection
INTEGER, INTENT(IN)    :: ilmt

! Northward of latitude index = jn, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(IN)    :: jn

! Southward of latitude index = js, Courant numbers could be > 1,
! so use the flux-form semi-Lagrangian scheme
INTEGER, INTENT(IN)    :: js
```



```
! Option for E-W transport scheme. See module header for more info.
INTEGER, INTENT(IN)    :: iord
```

```
! pressure at edges in "u" [hPa]
REAL(fp), INTENT(IN)   :: pu(ILO:IHI, JULO:JHI)
```

```
! Courant number in E-W direction
REAL(fp), INTENT(IN)   :: crx(ILO:IHI, JULO:JHI)
```

```
! Horizontal mass flux in E-W direction [hPa]
REAL(fp), INTENT(IN)   :: xmass(ILO:IHI, JULO:JHI)
```

INPUT/OUTPUT PARAMETERS:

```
! Species density [hPa]
REAL(fp), INTENT(INOUT) :: dq1(ILO:IHI, JULO:JHI)
```

```
! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(INOUT) :: qqv(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! E-W flux [mixing ratio]
REAL(fp), INTENT(OUT)   :: fx(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.
- 01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.21 Xmist

Subroutine Xmist computes the linear tracer slope in the E-W direction. It uses the Lin et. al. 1994 algorithm.

INTERFACE:

```
SUBROUTINE Xmist( dcx, qqv, J1P, J2P, I2_GL, JU1_GL, J2_GL, ILO, IHI, &
                JUL0, JHI, I1, I2, JU1, J2 )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P,    J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  ::      I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JUL0,  JHI

! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqv(-I2/3:I2+I2/3, JUL0:JHI)
```

OUTPUT PARAMETERS:

```
! Slope of concentration distribution in E-W direction [mixing ratio]
REAL(fp), INTENT(OUT) :: dcx(-I2/3:I2+I2/3, JUL0:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.22 Fxppm

Subroutine Fxppm is the 1D "outer" flux form operator based on the Piecewise Parabolic Method (PPM; see also Lin and Rood 1996) for computing the fluxes in the E-W direction.

INTERFACE:

```
SUBROUTINE Fxppm( ij,  ilmt, crx, dcx, fx, qqv,      &
                  ILO, IHI, JULO, JHI, I1,  I2 )
```

INPUT PARAMETERS:

```
! Local min & max longitude (I) and altitude (K) indices
INTEGER, INTENT(IN)    :: I1,    I2
```

```
! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: ILO,    IHI
INTEGER, INTENT(IN)    :: JULO,    JHI
```

```
! Latitude (IJ) and altitude (IK) indices
INTEGER, INTENT(IN)    :: ij
```

```
! Controls various options in E-W advection
INTEGER, INTENT(IN)    :: ilmt
```

```
! Courant number in E-W direction
REAL(fp), INTENT(IN)   :: crx(I1:I2, JULO:JHI)
```

INPUT/OUTPUT PARAMETERS:

```
! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(INOUT) :: qqv(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! Slope of concentration distribution in E-W direction (mixing ratio)
REAL(fp), INTENT(OUT)   :: dcx(ILO:IHI, JULO:JHI)
```

```
! E-W flux [mixing ratio]
REAL(fp), INTENT(OUT)   :: fx(I1:I2, JULO:JHI)
```

AUTHOR:

```
Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)
```

REMARKS:

```
This routine is called from w/in a OpenMP parallel loop fro
```

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Also remove the allocatable arrays, which interfere w/ OpenMP parallelization.
- 01 Apr 2009 - C. Carouge - The input arrays are now 2D only.

3.4.23 Lmtppm

Subroutine Lmtppm enforces the full monotonic, semi-monotonic, or the positive-definite constraint to the sub-grid parabolic distribution of the Piecewise Parabolic Method (PPM).

INTERFACE:

```
SUBROUTINE Lmtppm( lenx, lmt, a6, al, ar, dc, qa )
```

INPUT PARAMETERS:

```
! If 0 => full monotonicity;
! If 1 => semi-monotonic constraint (no undershoots);
! If 2 => positive-definite constraint
INTEGER, INTENT(IN)    :: lmt

! Vector length
INTEGER, INTENT(IN)    :: lenx
```

INPUT/OUTPUT PARAMETERS:

```
! Curvature of the test parabola
REAL(fp), INTENT(INOUT) :: a6(lenx)

! Left edge value of the test parabola
REAL(fp), INTENT(INOUT) :: al(lenx)

! Right edge value of the test parabola
REAL(fp), INTENT(INOUT) :: ar(lenx)

! 0.5 * mismatch
REAL(fp), INTENT(INOUT) :: dc(lenx)

! Cell-averaged value
REAL(fp), INTENT(INOUT) :: qa(lenx)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.

3.4.24 Ytp

Subroutine Ytp does horizontal advection in the N-S direction.

INTERFACE:

```
SUBROUTINE Ytp( jlmt, geofac_pc, geofac, cry, dq1, qqu, qqv, &
                ymass, fy,          J1P,  J2P,  I1_GL, I2_GL, JU1_GL, &
                J2_GL, ilong,       ILO,   IHI,  JUL0,  JHI,   I1,   &
                I2,   JU1,         J2,   jord )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)    :: J1P,   J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: I1_GL,  I2_GL
INTEGER, INTENT(IN)    :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)    :: I1,     I2
INTEGER, INTENT(IN)    :: JU1,    J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: ILO,    IHI
INTEGER, INTENT(IN)    :: JUL0,   JHI

! ???
INTEGER, INTENT(IN)    :: ilong
```

```
! Controls various options in N-S advection
INTEGER, INTENT(IN)    :: jlmt

! N-S transport scheme (see module header for more info)
INTEGER, INTENT(IN)    :: jord

! special geometrical factor (geofac) for Polar cap
REAL(fp), INTENT(IN)   :: geofac_pc

! geometrical factor for meridional advection; geofac uses correct
! spherical geometry, and replaces acosp as the meridional geometrical
! factor in tpcore
REAL(fp), INTENT(IN)   :: geofac(JU1_GL:J2_GL)

! Courant number in N-S direction
REAL(fp), INTENT(IN)   :: cry(ILO:IHI, JULO:JHI)

! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)   :: qqu(ILO:IHI, JULO:JHI)

! Horizontal mass flux in N-S direction [hPa]
REAL(fp), INTENT(IN)   :: ymass(ILO:IHI, JULO:JHI)
```

INPUT/OUTPUT PARAMETERS:

```
! Species density [hPa]
REAL(fp), INTENT(INOUT) :: dq1(ILO:IHI, JULO:JHI)

! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(INOUT) :: qqv(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! N-S flux [mixing ratio]
REAL(fp), INTENT(OUT)   :: fy(ILO:IHI, JULO:JHI+1)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared

with the "D" double-precision exponent. Added
OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.25 Ymist

Subroutine Ymist computes the linear tracer slope in the N-S direction. It uses the Lin et al. 1994 algorithm.

INTERFACE:

```
SUBROUTINE Ymist( id,    dcy, qqu, I1_GL, I2_GL, JU1_GL, &
                  J2_GL, J1P, ILO, IHI,  JUL0,  JHI,    &
                  I1,    I2,  JU1, J2 )
```

INPUT PARAMETERS:

```
! Global latitude index at the edge of the South polar cap
! J1P=JU1_GL+1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JUL0,  JHI

! The "order" of the accuracy in the computed linear "slope"
! (or mismatch, Lin et al. 1994); it is either 2 or 4.
INTEGER, INTENT(IN)  :: id

! Concentration contribution from E-W advection (mixing ratio)
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JUL0:JHI)
```

OUTPUT PARAMETERS:

```
! Slope of concentration distribution in N-S direction [mixing ratio]
REAL(fp), INTENT(OUT) :: dcy(ILO:IHI, JUL0:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.
- 01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.26 Do_Ymist_Pole1_I2d2

Subroutine Do_Ymist_Pole1_I2d2 sets "dcy" at the Poles.

INTERFACE:

```

SUBROUTINE Do_Ymist_Pole1_I2d2( dcy,   qqu, I1_GL, I2_GL, JU1_GL,   &
                                J2_GL, ILO, IHI,   JUL0, JHI,     &
                                I1,    I2,  JU1,   J2 )

```

INPUT PARAMETERS:

```

! Global min & max longitude (I) and latitude (J) indices
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JUL0,  JHI

! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JUL0-2:JHI+2)

```

OUTPUT PARAMETERS:

```

! Slope of concentration distribution in N-S direction [mixing ratio]
REAL(fp), INTENT(OUT) :: dcy(ILO:IHI, JUL0:JHI)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

- 05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.
- 05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops.
- 01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.
-

3.4.27 Do_Ymist_Pole2_I2d2

Subroutine Do_Ymist_Pole2_I2d2 sets "dcy" at the Poles.

INTERFACE:

```
SUBROUTINE Do_Ymist_Pole2_I2d2( dcy,   qqu, I1_GL, I2_GL, JU1_GL, &
                                J2_GL, J1P, ILO,  IHI,  JUL0,  &
                                JHI,  I1,  I2,   JU1,  J2 )
```

INPUT PARAMETERS:

```
! Global latitude index at the edge of the South polar cap
! J1P=JU1_GL+1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)  :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: I1_GL, I2_GL
INTEGER, INTENT(IN)  :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)  :: I1,     I2
INTEGER, INTENT(IN)  :: JU1,    J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,    IHI
INTEGER, INTENT(IN)  :: JUL0,   JHI

! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JUL0-2:JHI+2)
```

OUTPUT PARAMETERS:

```
! Slope of concentration distribution in N-S direction [mixing ratio]
REAL(fp), INTENT(OUT) :: dcy(ILO:IHI, JULO:JHI)
```

AUTHOR:

```
Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)
```

REVISION HISTORY:

```
05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.
05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent. Added
                        OpenMP parallel DO loops.
01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.
```

3.4.28 Fyppm

Subroutine Fyppm is the 1D "outer" flux form operator based on the Piecewise Parabolic Method (PPM; see also Lin and Rood 1996) for computing the fluxes in the N-S direction.

INTERFACE:

```
SUBROUTINE Fyppm( j1mt, cry, dcy, qqu, qqv, j1p, j2p, &
                 i1_gl, i2_gl, ju1_gl, j2_gl, ilong, ilo, ihi, &
                 julio, jhi, i1, i2, ju1, j2 )
```

INPUT PARAMETERS:

```
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN) :: J1P, J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN) :: I1_GL, I2_GL
INTEGER, INTENT(IN) :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN) :: I1, I2
INTEGER, INTENT(IN) :: JU1, J2
```

```
! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! I LONG ??
INTEGER, INTENT(IN)  :: ilong

! Controls various options in N-S advection
INTEGER, INTENT(IN)  :: jlmt

! Courant number in N-S direction
REAL(fp), INTENT(IN)  :: cry(ILO:IHI, JULO:JHI)

! Slope of concentration distribution in N-S direction [mixing ratio]
REAL(fp), INTENT(IN)  :: dcy(ILO:IHI, JULO:JHI)

! Concentration contribution from E-W advection [mixing ratio]
REAL(fp), INTENT(IN)  :: qqu(ILO:IHI, JULO:JHI)
```

OUTPUT PARAMETERS:

```
! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(OUT) :: qqv(ILO:IHI, JULO:JHI)
```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent. Added OpenMP parallel DO loops

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.29 Do_Fyppm_Pole_I2d2

Subroutine Do_Fyppm_Pole_I2d2 sets "al" & "ar" at the Poles.

INTERFACE:

```

SUBROUTINE Do_Fyppm_Pole_I2d2( al,  ar,  I1_GL, I2_GL, JU1_GL, J2_GL, &
                             ILO, IHI, JULO,  JHI,  I1,    I2,    &
                             JU1, J2 )

```

INPUT PARAMETERS:

```

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: I1_GL, I2_GL
INTEGER, INTENT(IN)    :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)    :: I1,    I2
INTEGER, INTENT(IN)    :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)    :: ILO,   IHI
INTEGER, INTENT(IN)    :: JULO,  JHI

```

OUTPUT PARAMETERS:

```

! Left (al) and right (ar) edge values of the test parabola
REAL(fp), INTENT(INOUT) :: al(ILO:IHI, JULO:JHI)
REAL(fp), INTENT(INOUT) :: ar(ILO:IHI, JULO:JHI)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                          Yeh with the TPCORE routines from GMI model.
                          This eliminates the polar overshoot in the
                          stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                          Declare all REAL variables as REAL(fp). Also
                          make sure all numerical constants are declared
                          with the "D" double-precision exponent. Added
                          OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

```

3.4.30 Do_Ytp_Pole_Sum

Subroutine Do_Ytp_Pole_Sum sets "dq1" at the Poles.

INTERFACE:

```

SUBROUTINE Do_Ytp_Pole_Sum( geofac_pc, dq1,   qqv,   fy,  I1_GL,  &
                           I2_GL,   JU1_GL, J2_GL, J1P, J2P,  &
                           ILO,    IHI,   JUL0,  JHI, I1,   &
                           I2,     JU1,   J2 )
!input PARAMETERS:
! Global latitude indices at the edges of the S/N polar caps
! J1P=JU1_GL+1; J2P=J2_GL-1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2; J2P=J2_GL-2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)      :: J1P,   J2P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)      :: I1_GL,  I2_GL
INTEGER, INTENT(IN)      :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)      :: I1,     I2
INTEGER, INTENT(IN)      :: JU1,    J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)      :: ILO,    IHI
INTEGER, INTENT(IN)      :: JUL0,   JHI

! Special geometrical factor (geofac) for Polar cap
REAL(fp), INTENT(IN)     :: geofac_pc

! Concentration contribution from N-S advection [mixing ratio]
REAL(fp), INTENT(IN)     :: qqv(ILO:IHI, JUL0:JHI)

```

INPUT/OUTPUT PARAMETERS:

```

! Species density [hPa]
REAL(fp), INTENT(INOUT) :: dq1(ILO:IHI, JUL0:JHI)

! N-S mass flux [mixing ratio]
REAL(fp), INTENT(INOUT) :: fy (ILO:IHI, JUL0:JHI+1)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
 John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also

make sure all numerical constants are declared
with the "D" double-precision exponent. Added
OpenMP parallel DO loops.

01 Apr 2009 - C. Carouge - Moved the IK loop outside the subroutine.

3.4.31 Fzppm

Subroutine Fzppm is the 1D "outer" flux form operator based on the Piecewise Parabolic Method (PPM; see also Lin and Rood 1996) for computing the fluxes in the vertical direction.

Fzppm was modified by S.-J. Lin, 12/14/98, to allow the use of the KORD=7 (klmt=4) option. KORD=7 enforces the 2nd monotonicity constraint of Huynh (1996). Note that in Huynh's original scheme, two constraints are necessary for the preservation of monotonicity. To use Huynh's algorithm, it was modified as follows. The original PPM is still used to obtain the first guesses for the cell edges, and as such Huynh's 1st constraint is no longer needed. Huynh's median function is also replaced by a simpler yet functionally equivalent in-line algorithm.

INTERFACE:

```
SUBROUTINE Fzppm( klmt,  delp1,  wz,  dq1,  qq1,  fz,      &
                  J1P,   JU1_GL, J2_GL, ILO, IHI, JUL0, JHI,      &
                  ILONG, IVERT, I1,   I2,  JU1, J2,   K1,  K2 )
```

INPUT PARAMETERS:

```
! Global latitude index at the edges of the South polar cap
! J1P=JU1_GL+1 for a polar cap of 1 latitude band
! J1P=JU1_GL+2 for a polar cap of 2 latitude bands
INTEGER, INTENT(IN)   :: J1P

! Global min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)   :: JU1_GL, J2_GL

! Local min & max longitude (I), latitude (J), altitude (K) indices
INTEGER, INTENT(IN)   :: I1,     I2
INTEGER, INTENT(IN)   :: JU1,   J2
INTEGER, INTENT(IN)   :: K1,    K2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)   :: ILO,   IHI
INTEGER, INTENT(IN)   :: JUL0,  JHI

! Dimensions in longitude & altitude ???
INTEGER, INTENT(IN)   :: ilong,  ivert

! Controls various options in vertical advection
```

```

INTEGER, INTENT(IN)    :: klmt

! Pressure thickness, the pseudo-density in a
! hydrostatic system at t1 [hPa]
REAL(fp), INTENT(IN)  :: delp1(ILO:IHI, JULO:JHI, K1:K2)

! Large scale mass flux (per time step tdt) in the vertical
! direction as diagnosed from the hydrostatic relationship [hPa]
REAL(fp), INTENT(IN)  :: wz(I1:I2, JU1:J2, K1:K2)

! Species concentration [mixing ratio]
REAL(fp), INTENT(IN)  :: qq1(:, :, :)

```

INPUT/OUTPUT PARAMETERS:

```

! Species density [hPa]
REAL(fp), INTENT(INOUT) :: dq1(ILO:IHI, JULO:JHI, K1:K2)

```

OUTPUT PARAMETERS:

```

! Vertical flux [mixing ratio]
REAL(fp), INTENT(OUT)  :: fz(ILO:IHI, JULO:JHI, K1:K2)

```

AUTHOR:

Original code from Shian-Jiann Lin, DAO
 John Tannahill, LLNL (jrt@llnl.gov)

REVISION HISTORY:

```

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin
                        Yeh with the TPCORE routines from GMI model.
                        This eliminates the polar overshoot in the
                        stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers.
                        Declare all REAL variables as REAL(fp). Also
                        make sure all numerical constants are declared
                        with the "D" double-precision exponent.

```

3.4.32 Average_Press_Poles

Subroutine Average_Press_Poles averages pressure at the Poles when the Polar cap is enlarged. It makes the last two latitudes equal.

INTERFACE:

```

SUBROUTINE Average_Press_Poles( area_1D, press, I1, I2, JU1, &
                               J2, ILO, IHI, JULO, JHI )

```

INPUT PARAMETERS:

```
! Local min & max longitude (I), latitude (J)
INTEGER, INTENT(IN)  :: I1,    I2
INTEGER, INTENT(IN)  :: JU1,   J2

! Local min & max longitude (I) and latitude (J) indices
INTEGER, INTENT(IN)  :: ILO,   IHI
INTEGER, INTENT(IN)  :: JULO,  JHI

! Surface area of grid box
REAL(fp), INTENT(IN)  :: AREA_1D(JU1:J2)
```

INPUT/OUTPUT PARAMETERS:

```
! Surface pressure [hPa]
REAL(fp), INTENT(INOUT) :: press(ILO:IHI, JULO:JHI)
```

AUTHOR:

Philip Cameron-Smith and John Tannahill, GMI project @ LLNL (2003)
Implemented into GEOS-Chem by Claire Carouge (ccarouge@seas.harvard.edu)

REMARKS:

Subroutine from *pjc_pfix*. Call this one once everything is working fine.

REVISION HISTORY:

05 Dec 2008 - C. Carouge - Replaced TPCORE routines by S-J Lin and Kevin Yeh with the TPCORE routines from GMI model. This eliminates the polar overshoot in the stratosphere.

05 Dec 2008 - R. Yantosca - Updated documentation and added ProTeX headers. Declare all REAL variables as REAL(fp). Also make sure all numerical constants are declared with the "D" double-precision exponent.

4 Convection and scavenging modules

These modules contain routines to perform the convective transport and to scavenge soluble species out of the atmosphere.

4.1 Fortran: Module Interface *convection_mod.F90*

Module CONVECTION_MOD contains routines which select the proper convection code for different met field data sets.

INTERFACE:

MODULE CONVECTION_MOD

USES:

USE PRECISION_MOD ! For GEOS-Chem Precision (fp)

IMPLICIT NONE

PRIVATE

PUBLIC MEMBER FUNCTIONS:

PUBLIC :: DO_CONVECTION

PRIVATE MEMBER FUNCTIONS:

PRIVATE :: DO_CLOUD_CONVECTION

REVISION HISTORY:

- 27 Jan 2004 - R. Yantosca - Initial version
- (1) Contains new updates for GEOS-4/fvDAS convection. Also now references "error_mod.f". Now make F in routine NFCLDMX a 4-D array to avoid memory problems on the Altix. (bmy, 1/27/04)
 - (2) Bug fix: Now pass NTRACE elements of TCVV to FVDAS_CONVECT in routine DO_CONVECTION (bmy, 2/23/04)
 - (3) Now references "logical_mod.f" and "tracer_mod.f" (bmy, 7/20/04)
 - (4) Now also references "ocean_mercury_mod.f" and "tracerid_mod.f" (sas, bmy, 1/19/05)
 - (5) Now added routines DO_GEOS4_CONVECT and DO_GCAP_CONVECT by breaking off code from DO_CONVECTION, in order to implement GCAP convection in a much cleaner way. (swu, bmy, 5/25/05)
 - (6) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (7) Shut off scavenging in shallow convection for GCAP (swu, bmy, 11/1/05)
 - (8) Modified for tagged Hg simulation (cdh, bmy, 1/6/06)
 - (9) Bug fix: now only call ADD_Hg2_WD if LDYNOCEAN=T (phs, 2/8/07)
 - (10) Fix for GEOS-5 met fields in routine NFCLDMX (swu, 8/15/07)
 - (11) Resize DTCSUM array in NFCLDMX to save memory (bmy, 1/31/08)
- 13 Aug 2010 - R. Yantosca - Added ProTeX headers
- 13 Aug 2010 - R. Yantosca - Treat MERRA in the same way as for GEOS-5
- 29 Sep 2010 - R. Yantosca - Added modifications for MERRA
- 05 Oct 2010 - R. Yantosca - Added ND14 and ND38 diagnostics to DO_MERRA_CONVECTION routine
- 16 Aug 2011 - J. Fisher - Minor bug fixes in DO_MERRA_CONVECTION
- 15 Feb 2011 - R. Yantosca - Add modifications for APM from G. Luo
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 02 Mar 2012 - R. Yantosca - Now reference the new grid_mod.F90
- 22 Oct 2012 - R. Yantosca - Now reference Headers/gigc_errcode_mod.F90
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 07 Nov 2014 - M. Yannetti - Added PRECISION_MOD

23 Jun 2015 - E. Lundgren - Convert tracer units from v/v dry air to kg/kg total air for convection

23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer compatible with the FlexChem implementation

06 Jul 2016 - E. Lundgren - Spc units are now kg/kg dry air in convection

29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA and rename routine DO_MERRA_CONVECTION to DO_CLOUD_CONVECTION

6 Nov 2017 - R. Yantosca - Now save equivalent of ND14, ND38 to netCDF

4.1.1 do.convection

Subroutine DO_CONVECTION calls the appropriate convection q driver program for different met field data sets.

INTERFACE:

```

      SUBROUTINE DO_CONVECTION( am_I_Root, Input_Opt, State_Met,
&                               State_Chm, State_Diag, RC           )

```

USES:

```

      USE CMN_SIZE_MOD
      #if defined( BPCH_DIAG )
      USE CMN_DIAG_MOD
      USE DIAG_MOD,      ONLY : CONVFLUP
      USE DIAG_MOD,      ONLY : AD37
      USE DIAG_MOD,      ONLY : AD38
      #endif
      USE ErrCode_Mod
      USE ERROR_MOD,     ONLY : GEOS_CHEM_STOP
      USE GC_GRID_MOD,   ONLY : GET_AREA_M2
      USE Input_Opt_Mod, ONLY : OptInput
      USE PhysConstants
      USE Species_Mod,   ONLY : Species
      USE State_Chm_Mod, ONLY : ChmState
      USE State_Diag_Mod, ONLY : DgnState
      USE State_Met_Mod, ONLY : MetState
      USE TIME_MOD,      ONLY : GET_TS_DYN
      USE TIME_MOD,      ONLY : GET_TS_CONV
      USE UnitConv_Mod
      USE WETSCAV_MOD,   ONLY : COMPUTE_F
      USE WETSCAV_MOD,   ONLY : H2O2s
      USE WETSCAV_MOD,   ONLY : SO2s
      #if defined( USE_TEND )
      USE TENDENCIES_MOD
      USE State_Chm_Mod, ONLY : Ind_

```

```
#endif
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(Optional),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met     ! Meteorology State object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState),  INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState),  INTENT(INOUT)  :: State_Diag    ! Diagnostics State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?
```

REVISION HISTORY:

```
25 May 2005 - S. Wu           - Initial version
08 Feb 2007 - R. Yantosca    - Now reference "CMN_SIZE". Now references
                               CLDMAS, CMFMC, DTRAIN from "dao_mod.f" so that
                               we can pass either GEOS-5 or GEOS-3 meteorology
                               to NFCLDMX.
13 Aug 2010 - R. Yantosca    - Added ProTeX headers
13 Aug 2010 - R. Yantosca    - Treat MERRA in the same way as for GEOS-5
29 Sep 2010 - R. Yantosca    - Now call DO_MERRA_CONVECTION for MERRA met
05 Oct 2010 - R. Yantosca    - Now attach diagnostics to MERRA conv routine
06 Oct 2010 - R. Yantosca    - Parallelized call to DO_MERRA_CONVECTION
15 Oct 2010 - H. Amos        - Now get BXHEIGHT, T from dao_mod.f
15 Oct 2010 - R. Yantosca    - Now get LDYNOCAN from logical_mod.f
15 Oct 2010 - R. Yantosca    - Now get ITS_A_MERCURY_SIM from tracer_mod.f
15 Oct 2010 - R. Yantosca    - Now get IDTHg2, IDTHgP from tracerid_mod.f
15 Oct 2010 - R. Yantosca    - Now get H2O2s, SO2s from wetscav_mod.f
15 Oct 2010 - H. Amos        - Now pass BXHEIGHT, T, to DO_MERRA_CONVECTION
15 Oct 2010 - R. Yantosca    - Now pass H2O2s, SO2s to DO_MERRA_CONVECTION
15 Feb 2011 - R. Yantosca    - Add modifications for APM from G. Luo
29 Aug 2011 - R. Yantosca    - Bug fix: reposition #if defined(APM) statement
09 Feb 2012 - R. Yantosca    - For GEOS-5.7, PFICU and PFLCU fields are
                               defined on level edges. Pass the top edge
                               of each level to DO_MERRA_CONVECTION
28 Feb 2012 - R. Yantosca    - Removed support for GEOS-3
01 Mar 2012 - R. Yantosca    - Now use GET_AREA_M2(I,J,L) from grid_mod.F90
21 Jun 2012 - R. Yantosca    - Now use pointers to pass array slices to routines
09 Nov 2012 - M. Payer       - Replaced all met field arrays with State_Met
                               derived type object
04 Feb 2013 - S. Kim         - Bug fix: H2O2s, SO2s, STT are not in State_Met
25 Mar 2013 - R. Yantosca    - Now accept am_I_Root, Input_Opt, State_Chm, RC
31 May 2013 - R. Yantosca    - Now pass objects to NFCLDMX
```

03 Jun 2013 - R. Yantosca - Bug fix: pass State_Chm to DO_MERRA_CONVECTION
 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
 17 Apr 2014 - R. Yantosca - Speed up MERRA and GEOS-FP convection by
 adding !\$OMP+SCHEDULE(DYNAMIC)
 17 Apr 2014 - R. Yantosca - Test if (ND14>0) and (ND38>0) outside OMP loop
 18 Apr 2014 - R. Yantosca - Now use a pointer to pass a slice of the
 F array to subroutine COMPUTE_F
 18 Apr 2014 - R. Yantosca - Clean up call to DO_MERRA_CONVECTION, remove
 stuff that was leftover from the column code
 18 Apr 2014 - R. Yantosca - Now use proper # of tracers for APM in the
 call to DO_MERRA_CONVECTION (MERRA & GEOS-FP)
 25 Jun 2014 - R. Yantosca - Now pass Input_Opt to COMPUTE_F
 26 Feb 2015 - E. Lundgren - Replace GET_PEDGE with State_Met%PEDGE.
 Remove dependency on pressure_mod
 28 Apr 2015 - E. Lundgren - Consolidate NFCLDMX arguments to remove
 passing State_Met array slice
 15 Jun 2015 - E. Lundgren - Now use kg/kg total air as tracer units
 (previously v/v)
 10 Aug 2015 - E. Lundgren - Incoming tracer units are now kg/kg dry air
 11 Aug 2015 - R. Yantosca - Added support for MERRA2 data
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 06 Jul 2016 - R. Yantosca - Use State_Chm%Map_WetDep to get species ID's
 07 Jul 2016 - R. Yantosca - Now dimension DIAG14 and DIAG38 for the #
 of soluble species (State_Chm%nWetDep)
 07 Jul 2016 - E. Lundgren - Now use spc kg/kg dry instead of kg/kg total
 08 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
 26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
 06 Nov 2017 - R. Yantosca - Bug fix: ND14 should use 1..nAdvect species
 09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.1.2 do_cloud_convection

Subroutine DO_CLOUD_CONVECTION (formerly called NFCLDMX) is S-J Lin's cumulus transport module for 3D GSFC-CTM, modified for the GEOS-Chem model.

INTERFACE:

```

SUBROUTINE DO_CLOUD_CONVECTION( am_I_Root,
&                               Input_Opt,
&                               State_Met,
&                               State_Chm,
&                               I,
&                               J,
&                               AREA_M2,
```

```

&          F,
&          TS_DYN,
&          USE_DIAG14,
&          DIAG14,
&          USE_DIAG38,
&          DIAG38,
&          RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE DEPO_MERCURY_MOD, ONLY : ADD_Hg2_SNOWPACK
USE DEPO_MERCURY_MOD, ONLY : ADD_Hg2_WD
USE DEPO_MERCURY_MOD, ONLY : ADD_HgP_WD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : IT_IS_NAN
USE ERROR_MOD, ONLY : IT_IS_FINITE
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE Species_Mod, ONLY : Species
USE WETSCAV_MOD, ONLY : H2O2s_3D => H2O2s ! [v/v]
USE WETSCAV_MOD, ONLY : SO2s_3D => SO2s ! [v/v]
USE WETSCAV_MOD, ONLY : WASHOUT
USE WETSCAV_MOD, ONLY : LS_K_RAIN
USE WETSCAV_MOD, ONLY : LS_F_PRIME

```

INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
INTEGER, INTENT(IN) :: I, J ! Lon & lat indices
REAL(fp), INTENT(IN) :: AREA_M2 ! Surface area [m2]
REAL(fp), INTENT(IN) :: F(:, :) ! Fraction of soluble species
! for updraft scavenging
! [unitless]. Computed by
! routine COMPUTE_F.

REAL(fp), INTENT(IN) :: TS_DYN ! Dynamic timestep [sec]
LOGICAL, INTENT(IN) :: USE_DIAG14 ! Archive DIAG14?
LOGICAL, INTENT(IN) :: USE_DIAG38 ! Archive DIAG38?

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

OUTPUT PARAMETERS:

```

REAL(fp),      INTENT(OUT)  :: DIAG14(:, :) ! Array for ND14 diagnostic
REAL(fp),      INTENT(OUT)  :: DIAG38(:, :) ! Array for ND38 diagnostic
INTEGER,       INTENT(OUT)  :: RC           ! Return code

```

REMARKS:

Reference:

```

=====
Lin, S.J. "Description of the parameterization of cumulus transport
in the 3D Goddard Chemistry Transport Model, NASA/GSFC, 1996.

```

Unit conversion for BMASS:

```

Ps - Pt (mb) | P2 - P1 | 100 Pa | s^2 | 1 | 1 kg      kg
-----+-----+-----+-----+-----+----- = -----
                | Ps - Pt |   mb   | 9.8 m | Pa | m^2 s^2    m^2

```

NOTE: We are passing I & J down to this routine so that it can call the proper code from "mercury_mod.f". Normally, we wouldn't pass I & J as arguments to columnized code. This prevents rewriting the mercury_mod.f routines ADD_Hg2_

REVISION HISTORY:

```

15 Jul 2009 - R. Yantosca - Columnized and cleaned up.
                    - CLDMAS renamed to CMFMC and DTRN renamed
                      to DTRAIN for consistency w/ GEOS-5.
17 Jul 2009 - R. Yantosca - Now do unit conversion of Q array from
                    [kg] --> [v/v] and vice versa internally
14 Dec 2009 - R. Yantosca - Now remove internal unit conversion, since
                    Q now comes in as [mol/mol] (= [v/v]) from the
                    calling routine.
14 Dec 2009 - R. Yantosca - Remove COEF from the argument list
06 May 2010 - R. Yantosca - Now add IDENT via the argument list
29 Sep 2010 - R. Yantosca - Modified for MERRA met fields
05 Oct 2010 - R. Yantosca - Now pass COEF via the argument list
05 Oct 2010 - R. Yantosca - Attach ND14 and ND38 diagnostics
15 Oct 2010 - H. Amos    - Added BXHEIGHT and T as arguments
15 Oct 2010 - R. Yantosca - Added I, J, H2O2s and SO2s as arguments
15 Oct 2010 - H. Amos    - Added scavenging below cloud base
06 Apr 2011 - M.Fu, H.Amos- Bug fix: make sure washout adheres to the same
                    algorithm as in the wet deposition code.
27 Jul 2011 - R. Yantosca - Declare CLDBASE as INTEGER to avoid PGI errors
16 Aug 2011 - J. Fisher  - Bug fix: use IS_Hg2() and IS_HgP to test if
                    a tracer is Hg2 or HgP (for tagged species)
16 Aug 2011 - J. Fisher  - Now use WETLOSS instead of TO_SUM in the ND38
                    diagnostic below the cloud. Using TO_SUM leads
                    us to over-count the tracer scavenged out of
                    the column.
22 Oct 2012 - R. Yantosca - Now reference Headers/gigc_errcode_mod.F90

```

- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 31 May 2013 - R. Yantosca - Now pass State_Chm to WASHOUT
- 05 Sep 2013 - R. Yantosca - Bug fix: DT is apparently undefined, but still passed to WASHOUT. Use SDT instead. This avoids a floating-point error.
- 18 Apr 2014 - R. Yantosca - Now point to 3-D arrays internally
- 18 Apr 2014 - R. Yantosca - Now also pass N_TRACERS (to facilitate APM)
- 18 Apr 2014 - R. Yantosca - Remove code that we don't need anymore
- 04 Feb 2015 - M. Sulprizio - Fix calculation of WETLOSS for non-aerosol tracers below the cloud base (C. Friedman)
- 20 Apr 2015 - E. Lundgren - Use DELP*100/g instead of AD/area as grid box moist mass per area and remove AD from routine
- 20 May 2015 - M. Sulprizio - Apply bug fixes provided by Viral Shah:
-- Remove F(K,IC) > 0 condition that prevents archiving of deposited mass in DIAG38
-- Add statement that subtracts the wet deposited amount from the atmospheric mass
-- Fix inconsistency in units when TO_SUM is used
- 04 Jun 2015 - E. Lundgren - Adapt Viral Shah bug fixes to moist mixing ratio
- 09 Jun 2015 - R. Yantosca - Now deposit Hg2, HgP to snowpack regardless of whether the dynamic ocean is used
- 15 Jun 2015 - E. Lundgren - Now use kg/kg total air as tracer units not v/v
- 22 Jun 2015 - E. Lundgren - Move QB_NUM calculation to within timestep loop
- 12 Aug 2015 - R. Yantosca - Treat MERRA2 in same way as we do for GEOS-FP
- 14 Sep 2015 - E. Lundgren - Apply bug fixes provided by Viral Shah:
-- Prevent ALPHA > 1 in washout of aerosols
-- Add tracer GAINED to Q before WETLOSS calculation in aerosol washout
- 22 Apr 2016 - R. Yantosca - Now get Is_Hg2 & Is_HgP from species database
- 25 Apr 2016 - R. Yantosca - Now pass Hg category # to ADD_Hg2_* functions
- 28 Apr 2016 - R. Yantosca - Rewrite Is_Hg block to avoid unassociated pointer seg faults
- 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
- 01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
- 05 Jul 2016 - R. Yantosca - Now replace N_TRACERS argument with the State_Chm%nAdvect field
- 06 Jul 2016 - E. Lundgren - Now use kg/kg dry air as spc units, requiring use of DELP_DRY instead of DELP and PEDGE_DRY for avg mixing ratio
- 07 Jul 2016 - R. Yantosca - Bug fix: F and ISOL need to be indexed with the advected species index NA
- 07 Jul 2016 - R. Yantosca - DIAG14 and DIAG38 now are dimensioned with of size State_Chm%nWetDep instead of N_TRACERS
- 20 Sep 2016 - R. Yantosca - Rewrote IF test to avoid Gfortran compiler error
- 24 Aug 2017 - M. Sulprizio - Rename routine from DO_MERRA_CONVECTION to

DO_CLOUD_CONVECTION

06 Nov 2017 - R. Yantosca - Bug fix: ND14 should use 1..nAdvect species

09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.2 Fortran: Module Interface wetscav_mod.F

Module WETSCAV_MOD contains routines and variables used in the wet scavenging of species in cloud updrafts, rainout, and washout.

INTERFACE:

MODULE WETSCAV_MOD

USES:

USE PRECISION_MOD ! For GEOS-Chem Precision (fp)

IMPLICIT NONE

PRIVATE

PUBLIC DATA MEMBERS:

! Save H2O2 and SO2 [v/v] for sulfate chemistry

REAL(fp), PUBLIC, ALLOCATABLE, TARGET :: H2O2s(:, :, :)

REAL(fp), PUBLIC, ALLOCATABLE, TARGET :: SO2s(:, :, :)

PUBLIC MEMBER FUNCTIONS:

PUBLIC :: CLEANUP_WETSCAV

PUBLIC :: COMPUTE_F

PUBLIC :: DO_WETDEP

PUBLIC :: INIT_WETSCAV

PUBLIC :: SETUP_WETSCAV

PUBLIC :: WASHOUT

PUBLIC :: LS_K_RAIN

PUBLIC :: LS_F_PRIME

PRIVATE MEMBER FUNCTIONS:

PRIVATE :: COMPUTE_L2G

PRIVATE :: CONV_F_PRIME

PRIVATE :: E_ICE

PRIVATE :: RAINOUT

PRIVATE :: GET_RAINFRAC

PRIVATE :: SAFETY

PRIVATE :: WASHFRAC_FINE_AEROSOL

PRIVATE :: WASHFRAC_COARSE_AEROSOL

PRIVATE :: WASHFRAC_LIQ_GAS

PRIVATE :: WASHFRAC_HNO3

PRIVATE :: GET_VUD

REMARKS:

References:

- =====
- (1) Liu,H., D.J. Jacob, I. Bey and R.M. Yantosca, "Constraints from 210Pb and 7Be on wet deposition and transport in a global three-dimensional chemical tracer model driven by assimilated meteorological fields", JGR, Vol 106, pp 12109-12128, 2001.
 - (2) D.J. Jacob, H. Liu, C. Mari, and R. M. Yantosca, "Harvard wet deposition scheme for GMI", Harvard Atmospheric Chemistry Modeling Group, March 2000.
 - (3) Chin, M., D.J. Jacob, G.M. Gardner, M.S. Foreman-Fowler, and P.A. Spiro, "A global three-dimensional model of tropospheric sulfate", J. Geophys. Res., 101, 18667-18690, 1996.
 - (4) Balkanski, Y D.J. Jacob, G.M. Gardner, W.C. Graustein, and K.K. Turekian, "Transport and Residence Times of Tropospheric Aerosols from a Global Three-Dimensional Simulation of 210Pb", JGR, Vol 98, (D11) pp 20573-20586, 1993.
 - (5) Giorgi, F, & W.L. Chaimedes, "Rainout Lifetimes of Highly Soluble Aerosols and Gases as Inferred from Simulations With a General Circulation Model", JGR, Vol 86 (D13) pp 14367-14376, 1986.

REVISION HISTORY:

- (1) Now trap allocation errors with routine ALLOC_ERR. (bmy, 7/11/00)
- (2) Moved routine MAKE_QQ here from "dao_mod.f" (bmy, 10/12/00)
- (3) Reordered arguments in INIT_PRECIP (bmy, 10/12/00)
- (4) Updated comments (bmy, 9/4/01)
- (5) Bug fix in MAKE_QQ: BXHEIGHT is sized IIPAR,JJPAR,LLPAR (bmy, 10/4/01)
- (6) Removed obsolete, commented-out code from 10/01 (bmy, 11/26/01)
- (7) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
- (8) Now zero allocatable arrays (bmy, 8/5/02)
- (9) Bug fix: ND39 diagnostic now closes the budget. Also bundled several standalone routines into this module. Now references F90 module "tracerid_mod.f". Also set NSOLMAX=10 since we now have sulfate tracers for wetdep. Now prevent out-of-bounds errors in routine WETDEP. Added GET_WETDEP_NMAX function to return max # of soluble tracers for allocating diagnostic arrays. Added functions GET_WETDEP_NSOL and GET_WETDEP_IDWETD. Now init H2O2s and SO2s to the initial H2O2 and SO2 from STT. Updated comments. (qli, bmy, 1/14/03)
- (10) Improvements for SO2/SO4 scavenging (rjp, bmy, 3/23/03)
- (11) Now references "time_mod.f". Added driver routine DO_WETDEP to remove cumbersome calling sequence from MAIN program. Also declared WETDEP and MAKE_QQ PRIVATE to this module. (bmy, 3/27/03)
- (11) Add parallelization to routine WETDEP (bmy, 3/17/04)
- (12) Added carbon and dust aerosol tracers (rjp, tdf, bmy, 4/5/04)
- (13) Added seasalt aerosol tracers (rjp, bec, bmy, 4/20/04)

- (14) Added secondary organic aerosol tracers (rjp, bmy, 7/13/04)
- (15) Now references "logical_mod.f" and "tracer_mod.f". Now move all internal routines to the module and pass arguments explicitly in order to facilitate parallelization on the Altix. (bmy, 7/20/04)
- (16) Updated for mercury aerosol tracers (eck, bmy, 12/9/04)
- (17) Updated for AS, AHS, LET, NH4aq, SO4aq. Also now pass Hg2 wetdep loss to "ocean_mercury_mod.f". (cas, sas, bmy, 1/20/05)
- (18) Bug fix to avoid numerical blowup in WETDEP. Now use analytical function for E_ICE(T). (bmy, 3/7/05)
- (19) Added SO4s, NITs. Increased NSOLMAX to 31. Also block out parallel loop in WETDEP for SGI MIPS compiler. (bec, bmy, 5/5/05)
- (20) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (21) Bug fixes: do not over-deplete H2O2s. Also include updates for tagged Hg simulation. (dkh, rjp, eck, cdh, bmy, 1/6/06)
- (22) Now wet deposit SOG4, SOA4. Remove unnecessary variables in WETDEP. (dkh, bmy, 5/18/06)
- (23) Bug fixes in COMPUTE_F (bmy, 7/26/06)
- (24) Resize DSTT array in WETDEP to save memory. Added fixes for GEOS-5 wet deposition per Hongyu Liu's suggestions. (bmy, 3/5/08)
- (25) Add wet scavenging of GLYX, MGLY, GLYC, SOAG, SOAM (tmf, 1/7/09)
- (26) Effective Henry's law constant and coefficient from Sander, R, 1999, Compilation of Henry's Law Constants for Inorganic and Organic Species of Potential Importance in Environmental Chemistry.
<http://www.mpch-mainz.mpg.de/~sander/res/henry.html>
(tmf, 1/7/09)
- (27) Remove support for SGI compiler. Bug fix in RAINOUT. (bmy, 7/20/09)
- (28) Update mercury simulation. (ccc, 5/17/10)
- (29) Add LGTMM as condition to output AD39. (ccc, 11/18/09)
- (30) Add snow scavenging, different washout/rainout ratio (wqq, ccc, 7/13/10)
- 13 Aug 2010 - R. Yantosca - Add modifications for MERRA (treat like GEOS-5)
- 16 Sep 2010 - R. Yantosca - Added ProteX headers
- 20 Sep 2010 - H. Amos, R. Yantosca - Implement new algorithms for MERRA
- 08 Oct 2010 - H. Amos - WASHFRAC_LIQ_GAS is now a subroutine
- 08 Oct 2010 - H. Amos - Various other modifications in WETDEP_MERRA
- 01 Aug 2011 - H. Amos - Bug fix for function WASHFRAC_LIQ_GAS
- 01 Aug 2011 - H. Amos - Updated comments
- 09 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x
- 21 Jun 2012 - R. Yantosca - Declare H2O2s, SO2s as TARGETs for pointers
- 23 Apr 2013 - R. Yantosca - Bug fix, eliminate white space from #if block
- 23 Apr 2013 - R. Yantosca - Remove LTOMAS logical, since we now invoke TOMAS with either TOMAS=yes or TOMAS40=yes
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust aerosols (T.D. Fairlie)
- 24 Nov 2014 - M. Yannetti - Added PRECISION_MOD

12 Feb 2015 - C. Keller - Added GET_F and GET_VUD. VUD is now stored in State_Met. In an ESMF environment, it's values are directly taken from the GEOS-5 model. In the traditional GEOS-Chem, the updraft velocity is still calculated in the old manner (5 m/s over water, 10 m/s over land/ice).

23 Jun 2015 - M. Sulprizio- Add impaction scavenging for hydrophobic BC and homogeneous IN removal from Qiaoqiao Wang

08 Jul 2015 - E. Lundgren - Add marine organic aerosols (B.Gantt, M.Johnson)

03 Sep 2015 - R. Yantosca - Remove NSOLMAX, we now get the # of wetdep species from State_Chm%nWetDep

03 Sep 2015 - R. Yantosca - INIT_WETSCAV now just allocates arrays; SETUP_WETSCAV now just sets up quantities

03 Sep 2015 - R. Yantosca - Remove WETDEPID routine, it's not needed

22 Sep 2015 - R. Yantosca - Re-introduce GET_WETDEP_NSOL routine

25 Sep 2015 - R. Yantosca - Use species database to collapse down routines COMPUTE_F, RAINOUT, WASHOUT

16 Jun 2016 - L. Hu - Replaced IDTXXX with Ind_('XXX')

20 Jun 2016 - R. Yantosca - Now define species ID's only in the INIT phase

20 Jun 2016 - R. Yantosca - Renamed species ID's IDTxxxx to id_XXXX

06 Jul 2016 - R. Yantosca - Removed GET_WETDEP_NSOL and GET_WETDEP_IDWETD, these are now replaced by State_Chm fields

17 Mar 2017 - R. Yantosca - Finally remove NSOL and IDWETD; they're unused

24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA and rename routine WETDEP_MERRA to WETDEP (the old WETDEP routine for GEOS-5 met was removed)

13 Nov 2017 - R. Yantosca - Added arrays for State_Diag diagnostics

4.2.1 do_wetdep

Subroutine DO_WETDEP is a driver for the wet deposition code, called from the MAIN program.

INTERFACE:

```

SUBROUTINE DO_WETDEP( am_I_Root, Input_Opt, State_Met,
&                    State_Chm, State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Error_Mod,      ONLY : Debug_MSG
USE Input_Opt_Mod,  ONLY : OptInput
USE PhysConstants
USE State_Chm_Mod,  ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod,  ONLY : MetState

```

```

#if defined( USE_TEND )
    USE TENDENCIES_MOD
#endif
    USE TIME_MOD,          ONLY : GET_TS_DYN
    USE UnitConv_Mod

```

INPUT PARAMETERS:

```

    LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
    TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
    TYPE(MetState), INTENT(IN)    :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

    TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry State object
    TYPE(DgnState), INTENT(INOUT) :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

    INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REVISION HISTORY:

```

27 Mar 2003 - R. Yantosca - Initial version
(1) Now references LPRT from "logical_mod.f" (bmy, 7/20/04)
(2) Don't do rainout/washout for conv precip for GEOS-5 (hyl, bmy, 3/5/08)
13 Aug 2010 - R. Yantosca - Treat GEOS-5 like MERRA
16 Sep 2010 - R. Yantosca - Added ProTeX headers
20 Sep 2010 - R. Yantosca - Rewrote #if block structure for clarity
09 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
02 Jul 2013 - R. Yantosca - Bug fix: add State_Chm to WETDEP calls for
                           met other than MERRA, GEOS-5.7 or GEOS-5
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
08 Aug 2015 - E. Lundgren - Tracer units are now input as [kg/kg]
12 Aug 2015 - R. Yantosca - Add support for MERRA2 meteorology
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
19 Jul 2016 - R. Yantosca - Now bracket tendency calls with #ifdef USE_TEND
09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
09 Nov 2017 - R. Yantosca - Return error condition to calling program

```

4.2.2 make_qq

Subroutine MAKE_QQ computes the large-scale or convective precipitation fields for use with WETDEP

INTERFACE:

```

SUBROUTINE MAKE_QQ( am_I_Root, State_Met, LS, RC )

```

USES:

```
USE CMN_SIZE_mod
USE ErrCode_Mod
USE State_Met_Mod, ONLY : MetState
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root ! Are we on the root CPU
TYPE(MetState),  INTENT(IN)  :: State_Met ! Meteorology State object
LOGICAL,          INTENT(IN)  :: LS        ! =T, denotes large scale precip
                                           ! =F, denotes convective precip
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC          ! Success or failure
```

REMARKS:

Construct QQ and PDOWN directly from met fields.

This only applies to large-scale precip, as the #if defined block in routine DO_WETDEP prevents the wet deposition routines from being called if it is convective precip.

Met fields:

=====

```
DQRLSAN  = 3-D precip production rate (LS+anvil) [kg/kg/s]
PFILSAN  = Dwnwd flux of ice precip    (LS+anvil) [kg/m2/s]
PFLLSAN  = Dwnwd flux of liquid precip (LS+anvil) [kg/m2/s]
REEVAPLS = Evap of precip'ing LS+anvil condensate [kg/kg/s]
```

Unit conversion for QQ:

=====

kg H2O		m ³ H2O		AIRDEN kg air	=	m ³ H2O
kg air * s		1000 kg H2O		m ³ air	=	m ³ air * s

and [m³ H2O/m³ air] = [cm³ H2O/cm³ air] because the same conversion factor from m³ -> cm³ is in both the numerator and the denominator.

Unit conversion for PDOWN:

=====

kg H2O		m ³ H2O		1e6 cm ³		m ²	+
m ² * s		1000 kg H2O		m ³		1e4 cm ²	+
kg ice		m ³ ice		1e6 cm ³		m ²	+

```

m^2 * s | 917 kg ice | m^3 | 1e4 cm2

= [ (PFILSAN/1000) * 100 ] + [ (PFLLSAN/1000) * 100]

```

REMARKS:

The PFILSAN and PFLLSAN fields are defined on level edges. Therefore, we must use L+1 to index them.

REVISION HISTORY:

29 Feb 2000 - H. Liu, R. Yantosca - Initial version

- (1) Now we partition MOISTQ into large-scale and convective parts, using total precipitation PREACC and convective precipitation PRECON (both are vertical integral amounts). The precipitation field at altitudes (PDOWN) is also made (hyl, djj, 10/17/98).
- (2) MAKE_QQ is written in Fixed-Form Fortran 90. (bmy, 4/2/99)!
- (3) AIRDEN, MOISTQ, QQ, and PDOWN are dimensioned (LLPAR,IIPAR,JJPARG) in order to maximize loop efficiency when processing an (I,J) column layer by layer. (bmy, 3/14/00)
- (4) MOISTQ is originally [g H2O/kg air/day], and is converted in READ_A6 to [kg H2O/kg air/s]. (bmy, 3/14/00)
- (5) Now reference PREACC, PRECON from "dao_mod.f" instead of from common block header file "CMN_PRECIP" (bmy, 6/26/00)
- (6) Now pass BXHEIGHT as an argument. Also added to "dao_mod.f". (bmy, 6/26/00)
- (7) Moved from "dao_mod.f" to "wetscav_mod.f". Also made PREACC and PRECON into arguments. (bmy, 10/12/00)
- (8) Updated comments (bmy, 9/4/01)
- (9) BXHEIGHT is now sized (IIPAR,JJPARG,LLPAR) (bmy, 10/4/01)
- (10) Removed obsolete, commented-out code from 10/01 (bmy, 11/26/01)
- (11) Now reference met field arrays directly from "dao_mod.f" (bmy, 11/8/02)

16 Sep 2010 - R. Yantosca - Added ProTeX headers

16 Sep 2010 - R. Yantosca - Compute QQ and PDOWN from MERRA met fields

09 Feb 2012 - R. Yantosca - Added modifications for GEOS-5.7.x met fields

09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object

26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP

06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)

06 Nov 2014 - R. Yantosca - Now use State_Met%MOISTQ(I,J,L)

29 Apr 2015 - E. Lundgren - Now use State_Met%MAIRDEN instead of AIRDEN since AIRDEN is now dry air density and MAIRDEN is moist air density, needed for use with MOISTQ

12 Aug 2015 - R. Yantosca - Add support for MERRA2 meteorology

09 Nov 2017 - R. Yantosca - Move allocations to INIT_WETSCAV

4.2.3 e_ice

Subroutine E_ICE computes Eice(T), the saturation vapor pressure of ice at a given Celsius temperature.

INTERFACE:

```
FUNCTION E_ICE( TK ) RESULT( VALUE )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: TK      ! Temperature [K]
```

RETURN VALUE:

```
REAL(fp)              :: VALUE  ! Saturation vapor pressure [hPa]
```

REMARKS:

Marti & Mauersberber (GRL '93) formulation of saturation vapor pressure of ice [Pa] is: $\log P = A/TK + B$

REVISION HISTORY:

08 Feb 2005 - R. Yantosca - Initial version
 (1) Now use the same analytic function as the Goddard CTM (bmy, 2/8/05)
 16 Sep 2010 - R. Yantosca - Added ProTeX headers

4.2.4 compute_l2g

Subroutine COMPUTE_L2G computes the ratio $L2G = C_{liq} / C_{gas}$, which is the mixing ratio of species in the liquid phase, divided by the mixing ratio of species in the gas phase.

INTERFACE:

```
SUBROUTINE COMPUTE_L2G( K0, CR, pKa, TK, H2OLIQ, L2G )
```

USES:

```
USE Henry_Mod, ONLY : Calc_KH  

USE Henry_Mod, ONLY : Calc_Heff
```

INPUT PARAMETERS:

```
REAL(f8), INTENT(IN) :: K0      ! Henry's solubility constant [M/atm]  

REAL(f8), INTENT(IN) :: CR      ! Henry's volatility constant [K]  

REAL(f8), INTENT(IN) :: pKa     ! Henry's pH correction factor [1]  

REAL(fp), INTENT(IN) :: TK      ! Temperature [K]  

REAL(fp), INTENT(IN) :: H2OLIQ ! Liquid water content [cm3 H2O/cm3 air]
```

OUTPUT PARAMETERS:

```
REAL(fp), INTENT(OUT) :: L2G      ! Cliq/Cgas ratio [1]
```

REMARKS:

The ratio Cliq / Cgas is obtained via Henry's law. The appropriate values of Kstar298 and H298_R must be supplied for each species. (cf Jacob et al 2000, p. 3)

REVISION HISTORY:

```
23 Feb 2000 - R. Yantosca - Initial version
(1 ) Bundled into "wetscav_mod.f" (bmy, 11/8/02)
16 Sep 2010 - R. Yantosca - Added ProTeX headers
10-Jan-2011 - H. Amos - Corrected the units on KStar298 from moles/atm
                    to M/atm
15-May-2013 - F. Paulot - Fix R constant
08 Dec 2015 - R. Yantosca - Now use functions from henry_mod.F
```

4.2.5 compute_f

Subroutine COMPUTE_F computes F, the fraction of soluble species lost by scavenging in convective cloud updrafts.

INTERFACE:

```
SUBROUTINE COMPUTE_F( am_I_Root, N,          F,          ISOL,
&                    Input_Opt, State_Met, State_Chm, RC      )
```

USES:

```
USE CMN_Size_Mod
USE ErrCode_Mod
USE Error_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE Species_Mod,       ONLY : Species
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
#if defined( TOMAS )
USE Tomas_Mod,         ONLY : GetFraction
#endif
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
INTEGER,          INTENT(IN)  :: N          ! Species ID
TYPE(OptInput),  INTENT(IN)  :: Input_Opt  ! Input Options object
TYPE(MetState),  INTENT(IN)  :: State_Met  ! Met State object
```


INPUT/OUTPUT PARAMETERS:

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: ISOL ! Index for ND38 diag
 REAL(fp), INTENT(OUT) :: F(:, :, :) ! Soluble fraction of species
 INTEGER, INTENT(OUT) :: RC ! Success or failure?

REVISION HISTORY:

- 23 Feb 2000 - H. Liu, R. Yantosca - Initial version
- (1) Currently works computes scavenging fractions for either full chemistry simulation (NSRCX == 3) or Rn-Pb-Be chemistry simulation (NSRCX == 1). Set the scavenging fraction to zero for other simulations which do not carry soluble tracers. (bmy, 3/2/00)
 - (2) Need to call INIT_SCAV to initialize the Vud, C_H2O, CLDLIQ, and CLDICE fields once per timestep. (bmy, 2/23/00)
 - (3) For aerosols only: now apply Eq. 2 for all temperatures. Also use the distance between the grid box centers in Eq. 2. Updated comments and made some cosmetic changes (hyl, bmy, 6/18/01)
 - (4) Remove IREF, JREF -- these are obsolete. T is now dimensioned (IIPAR, JJPARG, LLPARG). T(IREF, JREF, L) is now T(I, J, L). (bmy, 9/27/01)
 - (5) Removed obsolete code from 9/01 (bmy, 10/23/01)
 - (6) Fix 2 bugs for aerosol scavenging in Rn-Pb-Be simulation:
 - (a) set F(:, :, 1) = 0 since we don't do any scavenging there.
 - (b) DO L = 2, LLPARG to avoid any subscript range out of bounds errors (rjp, hyl, bmy, 1/10/02)
 - (7) Now set F=0 in the first level for all tracers. Also now compute the distance between grid box centers and use that in Eq. 10 from Jacob et al, 2000 to compute F. (hyl, bmy, 1/24/02)
 - (8) Eliminated obsolete code from 1/02 (bmy, 2/27/02)
 - (9) Now reference T from "dao_mod.f" instead of from "CMN". Also reference BXHEIGHT from "dao_mod.f" instead of from "CMN_NOX". Now bundled into "wetscav_mod.f". Now references IDTHNO3, IDTH2O2, etc, from F90 module "tracerid_mod.f". Added internal routines F_AEROSOL and GET_ISOL. Rewritten so that we don't duplicate code for different chemistry simulations. (bmy, 1/17/03)
 - (10) Now compute F for SO2 in the same way for both fullchem and offline simulations (rjp, bmy, 3/23/03)
 - (11) Added slots for carbon aerosol & dust tracers. Now modified internal routine GET_ISOL so it's not hardwired anymore. (rjp, bmy, 4/5/04)
 - (12) Added slots for sea salt aerosol tracers (rjp, bec, bmy, 4/20/04)
 - (13) Added slots for secondary organic aerosol tracers (rjp, bmy, 7/13/04)
 - (14) Remove reference to CMN, it's not needed. Made internal routine F_AEROSOL a module procedure rather than an internal routine to COMPUTE_F in order to facilitate parallelization on the Altix. Also now pass all arguments explicitly to F_AEROSOL. (bmy, 7/20/04)
 - (15) Now wet scavenge mercury aerosol tracers (eck, bmy, 12/9/04)

- (16) Updated for AS, AHS, LET, NH4aq, SO4aq. Also condensed the IF statement by combining branches for aerosols. (cas, bmy, 12/20/04)
- (17) Updated for SO4s, NITs (bec, bmy, 4/25/05)
- (18) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (19) Bug fix: Now do not over-deplete H2O2s. Also change Henry's law constant for Hg2 to 1.0d+14. Now use functions IS_Hg2 and IS_HgP to determine if a tracer is an Hg2 or HgP tagged tracer. (dkh, rjp, eck, cdh, bmy, 1/6/06)
- (20) Updated for SOG4 and SOA4 (dkh, bmy, 5/18/06)
- (21) Bug fix: now use separate conversion factors for H2O2 and NH3. (havala, bmy, 7/26/06)
- 16 Sep 2010 - R. Yantosca - Added ProTeX headers
- 10-Jan-2011 - H.Amos - Changed Hg2 Henry's law constant from 1.0d14 (no citation) to 1.4d6 M/atm (HgCl2, Lindqvist & Rhode, 1985). Henry's law constant in wetscav_mod.f is now consistent with what's used in mercury_mod.f
- 27 Sep 2011 - H. Amos - remove LHg_WETDasHNO3 logical, it's obsolete
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 06 Mar 2013 - H. Amos - merge C. Friedman's POP code
- 31 May 2013 - R. Yantosca - Now accept State_Chm, and pass it to TOMAS code
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust aerosols (T.D. Fairlie)
- 18 Apr 2014 - R. Yantosca - Now make the F argument an assumed-shape array so that we can pass a pointer array slice to it
- 25 Jun 2014 - R. Yantosca - Now accept Input_Opt via the arg list
- 19 Dec 2014 - M. Sulprizio- Add bug fixes for CONV_H2O2 and CONV_NH3 from Duncan Fairlie. The square root term should be inverted to be consistent with Mari et al (2000).
- 04 Jun 2015 - E. Lundgren - Now accept am_I_Root and RC as arguments
- 23 Jun 2015 - M. Sulprizio- Add impaction scavenging for hydrophobic BC and homogeneous IN removal from Qiaoqiao Wang
- 04 Sep 2015 - R. Yantosca - Use species database to radically simplify the structure of this routine.
- 24 Sep 2015 - E. Lundgren - Convert kg/kg total air <-> kg for TOMAS
- 25 Sep 2015 - R. Yantosca - Move the computation of the Ki rate (Eq. 1 from Jacob et al 2000) into routine COMPUTE_Ki
- 25 Sep 2015 - R. Yantosca - Remove T pointer, it's not needed here
- 30 Sep 2015 - R. Yantosca - Now use ThisSpc%WD_Is_HNO3 and ThisSpc%WD_Is_SO2 to flag the special cases of HNO3 & SO2 wetdep
- 05 Oct 2015 - R. Yantosca - Need to make ThisSpc !\$OMP THREADPRIVATE
- 06 Oct 2015 - R. Yantosca - Add missing variables for TOMAS
- 16 Oct 2015 - E. Lundgren - Consolidate remaining embedded ifelse blocks
- 01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
- 07 Jul 2016 - R. Yantosca - Now set ISOL = -1 if not a wetdep species

07 Jul 2016 - R. Yantosca - Remove pointer nullification from declarations

4.2.6 compute_ki

Subroutine COMPUTE_Ki computes the loss of species by scavenging according to Jacob et al 2000, eq. 1.

INTERFACE:

```
SUBROUTINE COMPUTE_Ki( SpcInfo, C_H2O, CLDICE, CLDLIQ, Kc, T, Ki )
```

USES:

```
USE Species_Mod, ONLY : Species
```

INPUT PARAMETERS:

```
TYPE(Species), INTENT(IN)  :: SpcInfo  ! Species database object
REAL(fp),          INTENT(IN)  :: C_H2O  ! Mixing ratio of H2O [v/v]
REAL(fp),          INTENT(IN)  :: CLDICE  ! Cloud ice mixing ratio
                                           ! [cm3 ice/cm3 air]
REAL(fp),          INTENT(IN)  :: CLDLIQ  ! Cloud liquid water mix ratio
                                           ! [cm3 H2O/cm3 air]
REAL(fp),          INTENT(IN)  :: Kc      ! Rate for conversion of cloud
                                           ! condensate -> precip [1/s]
REAL(fp),          INTENT(IN)  :: T      ! Temperature [K]
```

OUTPUT PARAMETERS:

```
REAL(fp),          INTENT(OUT) :: Ki      ! Loss of species from updraft
                                           ! (cf Eq. 1, Jacob et al, 2000)
```

REMARKS:

This routine centralizes computations that are used in routines COMPUTE_F and RAINOUT.

REVISION HISTORY:

```
25 Sep 2015 - R. Yantosca - Initial version
08 Dec 2015 - R. Yantosca - Make KO, CR, pKa 8-byte variables for
                           compatibility w/ CALC_KH, CALC_HEFF routines
01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
```

4.2.7 f_aerosol

Subroutine F_AEROSOL returns the fraction of aerosol scavenged in updrafts

INTERFACE:

```
SUBROUTINE F_AEROSOL( KC, KcScale, State_Met, F )
```

USES:

```
USE CMN_SIZE_MOD
USE State_Met_Mod,      ONLY: MetState
```

INPUT PARAMETERS:

```
REAL(fp),      INTENT(IN)  :: KC                ! Cloud condensate to
! precipitation rate
! [1/s]
REAL(fp),      INTENT(IN)  :: KcScale(3)        ! Scale factors for Kc
! for 3 temperature
! regimes
TYPE(MetState), INTENT(IN)  :: State_Met        ! Meteorology State
```

OUTPUT PARAMETERS:

```
REAL(fp),      INTENT(OUT) :: F(IIPAR,JJP,LLPAR) ! Fraction of aerosol
! scavenged in
! convective updrafts
```

REVISION HISTORY:

```
07 Nov 2002 - R. Yantosca - Initial version
16 Sep 2010 - R. Yantosca - Added ProTeX headers
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
! derived type object
29 May 2013 - R. Yantosca - Segregate TOMAS-specific code with #ifdefs
10 Nov 2014 - C. Keller   - Now also apply TINY check in ESMF environment.
23 Jun 2015 - M. Sulprizio- Add impaction scavenging for hydrophobic BC and
! homogeneous IN removal from Qiaoqiao Wang
25 Sep 2015 - R. Yantosca - Rewrite this routine to avoid testing on the
! tracer number. Remove the N argument.
```

4.2.8 rainout

Subroutine RAINOUT computes RAINFRAC, the fraction of soluble species lost to rainout events in precipitation.

INTERFACE:

```
SUBROUTINE RAINOUT( I,      J,      L,      N,
&                  K_RAIN, DT,      F,      RAINFRAC,
&                  Input_Opt, State_Met, State_Chm, RC      )
```

USES:

```

USE CMN_Size_Mod
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState

```

INPUT PARAMETERS:

```

INTEGER,          INTENT(IN)  :: I           ! Longitude index
INTEGER,          INTENT(IN)  :: J           ! Latitude index
INTEGER,          INTENT(IN)  :: L           ! Level index
INTEGER,          INTENT(IN)  :: N           ! Species number
REAL(fp),         INTENT(IN)  :: K_RAIN      ! Rainout rate constant [1/s]
REAL(fp),         INTENT(IN)  :: DT         ! Timestep for rainout event [s]
REAL(fp),         INTENT(IN)  :: F           ! Fraction of grid box that is
                                           ! precipitating [unitless]
TYPE(OptInput),  INTENT(IN)  :: Input_Opt   ! Input options
TYPE(MetState),  INTENT(IN)  :: State_Met   ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(IN)  :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

REAL(fp),          INTENT(OUT) :: RAINFRAC   ! Fraction of species lost
                                           ! to rainout [unitless]
INTEGER,           INTENT(OUT) :: RC         ! Success or failure?

```

REVISION HISTORY:

- 28 Feb 2000 - R. Yantosca - Initial version
- (1) Currently works for either full chemistry simulation (NSRCX == 3) or Rn-Pb-Be chemistry simulation (NSRCX == 1). Other simulations do not carry soluble tracer, so set RAINFRAC = 0. (bmy, 2/28/00)
 - (2) Need to call INIT_SCAV to initialize the Vud, C_H2O, CLDLIQ, and CLDICE fields once per dynamic timestep. (bmy, 2/28/00)
 - (3) K_RAIN, the rainout rate constant, and F, the areal fraction of the grid box undergoing precipitation, are computed according to Giorgi & Chaimedes, as described in Jacob et al, 2000.
 - (4) Now no longer suppress scavenging of HNO3 and aerosol below 258K. Updated comments, cosmetic changes. Now set TK = T(I,J,L) since T is now sized (IIPAR,JJP,LLPAR) in "CMN". (djj, hyl, bmy, 1/24/02)
 - (5) Eliminated obsolete code (bmy, 2/27/02)
 - (6) Now reference T from "dao_mod.f". Updated comments. Now bundled into "wetscav_mod.f". Now references "tracerid_mod.f". Also removed reference to CMN since we don't need NSRCX. (bmy, 11/8/02)
 - (7) Now updated for carbon & dust aerosol tracers (rjp, bmy, 4/5/04)

- (8) Now updated for seasalt aerosol tracers (rjp, bec, bmy, 4/20/04)
 - (9) Now updated for secondary aerosol tracers (rjp, bmy, 7/13/04)
 - (10) Now treat rainout of mercury aerosol tracers (eck, bmy, 12/9/04)
 - (11) Updated for AS, AHS, LET, NH4aq, SO4aq. Also condensed the IF statement by grouping blocks together. (cas, bmy, 12/20/04)
 - (12) Updated for SO4s, NITs (bec, bmy, 4/25/05)
 - (13) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (14) Change Henry's law constant for Hg2 to 1.0d+14. Now use functions IS_Hg2 and IS_HgP to determine if the tracer is a tagged Hg0 or HgP tracer. (eck, cdh, bmy, 1/6/06)
 - (15) Updated for SOG4 and SOA4 (dkh, bmy, 5/18/06)
 - (16) For GEOS-5, suppress rainout when T < 258K (hyl, bmy, 3/5/08)
 - (17) Bug fix: need to use separate conversion parameters for H2O2 and NH3. This was the same fix as in COMPUTE_F but until now we had overlooked this. (havala, bmy, 7/20/09)
 - 25 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
 - 16 Sep 2010 - R. Yantosca - Added ProTeX headers
 - 27 Sep 2011 - H. Amos - remove LHg_WETDasHNO3 logical, it's obsolete
 - 09 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA
 - 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
 - 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
 - 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust aerosols (T.D. Fairlie)
 - 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
 - 25 Aug 2014 - M. Sulprizio- Now accept Input_Opt as an argument
 - 19 Dec 2014 - M. Sulprizio- Add bug fixes for CONV_H2O2 and CONV_NH3 from Duncan Fairlie. The square root term should be inverted to be consistent with Mari et al (2000).
 - 23 Jun 2015 - M. Sulprizio- Add impaction scavenging for hydrophobic BC and homogeneous IN removal from Qiaoqiao Wang
 - 08 Jul 2015 - E. Lundgren - Add marine organic aerosols (B.Gantt, M.Johnson)
 - 23 Sep 2015 - R. Yantosca - Use the species database to eliminate many IF blocks to simplify the code logic
 - 25 Sep 2015 - R. Yantosca - Now call routine COMPUTE_Ki to compute the rate of scavenging (Eq. 1, Jacob et al 2000)
 - 25 Sep 2015 - R. Yantosca - Now define THREADPRIVATE pointers
 - 28 Sep 2015 - R. Yantosca - Needed to add special case for HNO3
 - 30 Sep 2015 - R. Yantosca - Now use ThisSpc%WD_Is_HNO3 and ThisSpc%WD_Is_SO2 to flag the special cases of HNO3 & SO2 wetdep
 - 16 Oct 2015 - E. Lundgren - Use ThisSpc%WD_Is_H2SO4 to treat H2SO4 as aerosol for wetdep (for microphysics)
 - 01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
 - 07 Jul 2016 - R. Yantosca - Now remove pointer threadprivate declarations
-

4.2.9 apply_rainout_eff

Subroutine APPLY_RAINOUT_EFF multiplies the rainout fraction computed by RAINOUT with the rainout efficiency for one of 3 temperature ranges: (1) $T \leq 237$ K; (2) $237 < T \leq 258$ K; (3) $T > 258$ K. The rainout efficiencies for each aerosol species are defined in the species database object (i.e. State_Chm

This allows us to apply the impaction scavenging of certain aerosol species (BC, dust, HNO₃) as implemented by Qiaoqiao Wang, while also suppressing rainout for other aerosol species. The prior code achieved this by using a large and confusing IF statement, whose logic was hard to understand.

INTERFACE:

```
SUBROUTINE APPLY_RAINOUT_EFF( TK, SpcInfo, RainFrac )
```

USES:

```
USE Species_Mod, ONLY : Species
```

INPUT PARAMETERS:

```
REAL(fp),      INTENT(IN)      :: TK          ! Temperature [K]
TYPE(Species), INTENT(IN)      :: SpcInfo     ! Species Database object
```

INPUT/OUTPUT PARAMETERS:

```
REAL(fp),      INTENT(INOUT)   :: RainFrac   ! Rainout fraction
```

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Initial version
```

4.2.10 get_rainfrac

Function GET_RAINFRAC computes the fraction of species lost to rainout according to Jacob et al 2000.

INTERFACE:

```
FUNCTION GET_RAINFRAC( K, F, DT ) RESULT( RAINFRAC )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: K          ! Rainout rate constant [1/s]
REAL(fp), INTENT(IN) :: F          ! Timestep for rainout event [s]
REAL(fp), INTENT(IN) :: DT         ! Fraction of grid box that is
                                     ! undergoing precipitation [unitless]
```

RETURN VALUE:

```
REAL(fp)              :: RAINFRAC ! Fraction of species lost to rainout
```

REVISION HISTORY:

08 Nov 2002 - R. Yantosca - Initial version
 (1) Now move internal routines GET_RAINFRAC to the module and pass all arguments explicitly. This facilitates parallelization on the Altix platform (bmy, 7/20/04)
 16 Sep 2010 - R. Yantosca - Added ProTeX headers

4.2.11 washout

Subroutine WASHOUT computes WASHFRAC, the fraction of soluble species lost to washout events in precipitation.

INTERFACE:

```

SUBROUTINE WASHOUT( am_I_Root, I,          J,          L,
&                  N,          BXHEIGHT, TK,          PP,
&                  DT,          F,          H2O2s,      SO2s,
&                  WASHFRAC, KIN,          Input_Opt, State_Met,
&                  State_Chm, RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod,   ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
#if defined( TOMAS )
USE ERROR_MOD
USE TOMAS_MOD,    ONLY : IBINS,    ICOMP
USE UnitConv_Mod
#endif

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)   :: am_I_Root  ! Are we on the root CPU?
INTEGER,          INTENT(IN)   :: I          ! Longitude index
INTEGER,          INTENT(IN)   :: J          ! Latitude index
INTEGER,          INTENT(IN)   :: L          ! Level index
INTEGER,          INTENT(IN)   :: N          ! Species number
REAL(fp),         INTENT(IN)   :: BXHEIGHT  ! Grid box height [m]
REAL(fp),         INTENT(IN)   :: TK        ! Temperature [K]
REAL(fp),         INTENT(IN)   :: PP        ! Precip rate thru bottom
! of grid (I,J,L)
! [cm3 H2O/cm2 air/s]
REAL(fp),         INTENT(IN)   :: DT        ! Timestep [s]
REAL(fp),         INTENT(IN)   :: F        ! Fraction of grid box that

```



```

! is precipitating [1]

TYPE(OptInput), INTENT(IN)    :: Input_Opt  ! Input Options
TYPE(MetState), INTENT(IN)    :: State_Met  ! Meteorology State

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object
REAL(fp),          INTENT(INOUT) :: H2O2s    ! H2O2 [v/v] and SO2 [v/v]
REAL(fp),          INTENT(INOUT) :: SO2s     ! conc's after aqueous rxns
! are applied. These are
! computed in the sulfate
! chemistry module and
! passed here as arguments.

```

OUTPUT PARAMETERS:

```

REAL(fp),          INTENT(OUT)  :: WASHFRAC  ! Fraction of species lost
! to washout [1]
LOGICAL,           INTENT(OUT)  :: KIN       ! =T washout is a
! kinetic process
! =F washout is an
! equilibrium process
INTEGER,           INTENT(OUT)  :: RC        ! Success or failure?

```

REVISION HISTORY:

- 28 Feb 2000 - R. Yantosca - Initial version
- (1) Currently works for either full chemistry simulation (NSRCX == 3) or Rn-Pb-Be chemistry simulation (NSRCX == 1). Other simulations do not carry soluble tracers, so set WASHFRAC = 0.
 - (2) K_WASH, the rainout rate constant, and F, the areal fraction of the grid box undergoing precipitation, are computed according to Giorgi & Chaimedes, as described in Jacob et al, 2000.
 - (3) Washout is only done for T >= 268 K, when the cloud condensate is in the liquid phase.
 - (4) T(I+IO,J+JO,L) is now T(I,J,L). Removed IREF, JREF -- these are obsolete. Updated comments. (bmy, 9/27/01)
 - (5) Removed obsolete commented out code from 9/01 (bmy, 10/24/01)
 - (6) Now reference BXHEIGHT, T from "dao_mod.f". Also remove reference to "CMN_NOX". Updated comments. Now bundled into "wetscav_mod.f". Now also references "tracerid_mod.f". Added internal routines WASHFRAC_AEROSOL and WASHFRAC_LIQ_GAS. Also removed reference to CMN since we don't need to use NSRCX here. (bmy, 11/6/02)
 - (7) Updated for carbon aerosol and dust tracers (rjp, bmy, 4/5/04)
 - (8) Updated for seasalt aerosol tracers (rjp, bec, bmy, 4/20/04)
 - (9) Updated for secondary organic aerosol tracers (rjp, bmy, 7/13/04)
 - (10) Now move internal routines WASHFRAC_AEROSOL and WASHFRAC_LIQ_GAS to the module and pass all arguments explicitly. This facilitates

- parallelization on the Altix platform (bmy, 7/20/04)
- (11) Now handle washout of mercury aerosol tracers (eck, bmy, 12/9/04)
- (13) Updated for AS, AHS, LET, NH4aq, SO4aq. Also condensed the IF statement by grouping blocks together (cas, bmy, 12/20/04)
- (14) Updated for SO4s, NITs (bec, bmy, 4/25/05)
- (15) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (16) Bug fix: Deplete H2O2s the same as SO2s. Also change Henry's law constant for Hg2 to 1.0d+14. Now use functions IS_Hg2 and IS_HgP to determine if a tracer is a tagged Hg0 or HgP tracer. (dkh, rjp, eck, cdh, bmy, 1/6/06)
- (17) Updated for SOG4 and SOA4 (bmy, 5/18/06)
- 16 Sep 2010 - R. Yantosca - Added ProTeX headers
- 30 Sep 2010 - H. Amos - WASHFRAC_LIQ_GAS now a subroutine (was an external function)
- 14 Oct 2010 - H. Amos - Remove dependence on I, J. That means removing I, J as input arguments and adding T, BXHEIGHT, H2O2s, and SO4s and input arguments.
- 16 Aug 2011 - H. Amos - move K_WASH to WASHFRAC_AEROSOL, WASHFRAC_HNO3, and WASHFRAC_LIQ_GAS
- 16 Aug 2011 - H. Amos - Replace logical AER with KIN. Serves the same purpose in the code, but emphasizes to the user that the difference in washout isn't whether or not the tracer is an aerosol, it's whether or not washout is modeled as a kinetic vs equilibrium process.
- 27 Sep 2011 - H. Amos - remove LHg2_WETDashHNO3 logical, it's obsolete
- 20 Jan 2012 - H. Amos - WASHFRAC for aerosol is now either computed by WASHFRAC_FINE_AEROSOL or WASHFRAC_COARSE_AEROSOL
- 31 May 2013 - R. Yantosca - Now accept State_Chm, and pass it to TOMAS code
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust aerosols (T.D. Fairlie)
- 27 Sep 2013 - M. Sulprizio- DST2-DST4 are now considered coarse mode aerosol following the recommendation of T.D. Fairlie. NITd, SO4d, and DALK are similarly updated.
- 25 Aug 2014 - M. Sulprizio- Now accept Input_Opt as an argument
- 12 May 2015 - E. Lundgren - Change tracer units from v/v -> kg for TOMAS if not already in kg (ie. called from convection)
- 04 Jun 2015 - E. Lundgren - Now accept am_I_Root and RC as arguments
- 23 Jun 2015 - E. Lundgren - Adjust 5/12/15 TOMAS bug fix to convert kg/kg total air <-> kg for new convection units
- 08 Jul 2015 - E. Lundgren - Add marine organic aerosols (B.Gantt, M.Johnson)
- 22 Sep 2015 - E. Lundgren - Add kg/m2 <-> kg conversion for TOMAS
- 30 Sep 2015 - R. Yantosca - Now use ThisSpc%WD_Is_HNO3 and ThisSpc%WD_Is_SO2 to flag the special cases of HNO3 & SO2 wetdep
- 16 Oct 2015 - E. Lundgren - Use ThisSpc%WD_Is_H2SO4 to treat H2SO4 as aerosol for wetdep, and consolidate ifelse blocks

27 Jul 2016 - E. Lundgren - Now expect dry mixing ratio not total
 22 Dec 2016 - R. Yantosca - Bug fix: KO, CR, pKa have to be REAL(f8)
 09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.2.12 washfrac_fine_aerosol

Function WASHFRAC_FINE_AEROSOL returns the fraction of soluble aerosol species lost to washout.

INTERFACE:

```
FUNCTION WASHFRAC_FINE_AEROSOL( DT, F, PP, TK )
&      RESULT( WASHFRAC )
```

USES:

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: DT      ! Timestep of washout event [s]
REAL(fp), INTENT(IN) :: F      ! Fraction of grid box that is
                                ! precipitating [unitless]
REAL(fp), INTENT(IN) :: PP     ! Precip rate thru bottom of grid
                                ! box (I,J,L) [cm3 H2O/cm2 air/s]
REAL(fp), INTENT(IN) :: TK     ! Temperature in grid box [K]
```

RETURN VALUE:

```
REAL(fp)              :: WASHFRAC ! Fraction of soluble species
                                ! lost to washout [1]
```

REVISION HISTORY:

08 Nov 2002 - R. Yantosca - Initial version
 (1) WASHFRAC_AEROSOL used to be an internal function to subroutine WASHOUT.
 This caused NaN's in the parallel loop on Altix, so we moved it to
 the module and now pass Iall arguments explicitly (bmy, 7/20/04)
 16 Sep 2010 - R. Yantosca - Added ProTeX headers
 21 Jan 2011 - J. Fisher & Q. Wang - Update to account for time-dependent
 shift in aerosol size distribution that slows washout as a rain
 event proceeds (see e.g. Feng et al., 2007, 2009).
 16 Aug 2011 - H Amos - Remove K_WASH from input list, make a defined
 parameter.
 20 Jan 2012 - H Amos - rename WASHFRAC_FINE_AEROSOL to distinguish
 this function from WASHFRAC_COARSE_AEROSOL
 04 Sep 2013 - R. Yantosca - Bug fix: Prevent div-by-zero if F=0. Because F
 multiplies the whole expression for WASHFRAC,
 WASHFRAC=0 whenever F=0 anyway.

4.2.13 washfrac_coarse_aerosol

Function WASHFRAC_COARSE_AEROSOL returns the fraction of soluble aerosol species lost to washout.

INTERFACE:

```
FUNCTION WASHFRAC_COARSE_AEROSOL( DT, F, PP, TK )
&      RESULT( WASHFRAC )
```

USES:**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: DT      ! Timestep of washout event [s]
REAL(fp), INTENT(IN) :: F      ! Fraction of grid box that is
                                ! precipitating [unitless]
REAL(fp), INTENT(IN) :: PP     ! Precip rate thru bottom of grid
                                ! box (I,J,L) [cm3 H2O/cm2 air/s]
REAL(fp), INTENT(IN) :: TK     ! Temperature in grid box [K]
```

RETURN VALUE:

```
REAL(fp)              :: WASHFRAC ! Fraction of soluble species
                                ! lost to washout
```

REVISION HISTORY:

```
08 Nov 2002 - R. Yantosca - Initial version
(1 ) WASHFRAC_AEROSOL used to be an internal function to subroutine WASHOUT.
      This caused NaN's in the parallel loop on Altix, so we moved it to
      the module and now pass Iall arguments explicitly (bmy, 7/20/04)
16 Sep 2010 - R. Yantosca - Added ProTeX headers
16 Aug 2011 - H Amos      - Remove K_WASH from input list, make a defined
                          parameter.
20 Jan 2012 - H Amos      - WASHFRAC_COARSE_AEROSOL created to handle
                          SALC and DST4
04 Sep 2013 - R. Yantosca - Bug fix: Prevent div-by-zero if F=0. Because F
                          multiplies the whole expression for WASHFRAC,
                          WASHFRAC=0 whenever F=0 anyway.
```

4.2.14 washfrac_size_aerosol

Subroutine WASHFRAC_SIZE_AEROSOL retrieves fraction of soluble aerosol species lost to washout. Size resolved version for TOMAS.

INTERFACE:

```

SUBROUTINE WASHFRAC_SIZE_AEROSOL( DT, F, PP, TK, N, I, J, L,
&                                State_Met, State_Chm, WASHFRAC,
&                                RC )

```

USES:

```

USE ErrCode_Mod
USE ERROR_MOD
USE State_Chm_Mod,      ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TOMAS_MOD,         ONLY : IBINS, GETDP, STRATSCAV

```

INPUT PARAMETERS:

```

REAL(fp),      INTENT(IN)  :: DT      ! Dynamic timestep [s]
REAL(fp),      INTENT(IN)  :: F      ! Fraction of grid box
! that is precipitating
REAL(fp),      INTENT(IN)  :: PP      ! Precip rate thru bottom
! of grid box (I,J,L)
! [cm3 H2O/cm2 air/s]
REAL(fp),      INTENT(IN)  :: TK      ! Temperature [K]
INTEGER,       INTENT(IN)  :: I      ! Longitude index
INTEGER,       INTENT(IN)  :: J      ! Latitude index
INTEGER,       INTENT(IN)  :: L      ! Level index
INTEGER,       INTENT(IN)  :: N      ! Species index

TYPE(MetState), INTENT(IN)  :: State_Met ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

OUTPUT PARAMETERS:

```

REAL(fp),      INTENT(OUT)  :: WASHFRAC ! Fraction of species
! lost to washout [1]
INTEGER,       INTENT(OUT)  :: RC      ! Success or failure?

```

REVISION HISTORY:

```

31 May 2013 - R. Yantosca - Now accept State_Met, State_Chm as arguments
04 Sep 2013 - R. Yantosca - Bug fix: Prevent div-by-zero if F=0. Because F
! multiplies the whole expression for WASHFRAC,
! WASHFRAC=0 whenever F=0 anyway.
24 Sep 2015 - E. Lundgren - Function is now a subroutine and outputs RC

```

4.2.15 washfrac_hno3

Function WASHFRAC_HNO3 returns the fraction of HNO3 species lost to washout.

INTERFACE:

```
FUNCTION WASHFRAC_HNO3( DT, F, PP, TK ) RESULT( WASHFRAC )
```

USES:**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: DT      ! Timestep of washout event [s]
REAL(fp), INTENT(IN) :: F      ! Fraction of grid box that is
                                ! precipitating [unitless]
REAL(fp), INTENT(IN) :: PP     ! Precip rate thru bottom of grid
                                ! box (I,J,L) [cm3 H2O/cm2 air/s]
REAL(fp), INTENT(IN) :: TK     ! Temperature in grid box [K]
```

RETURN VALUE:

```
REAL(fp)              :: WASHFRAC ! Fraction of soluble species
```

REVISION HISTORY:

13 Aug 2011, H Amos: Initial version, modeled after WASHFRAC_AEROSOL. Separate function created to emphasize that the new, updated washout coefficients from Feng et al (2007; 2009) should only be applied to aerosol species. It was a coincidence before that the original washout coefficients for aerosols and HNO3 were the same.

16 Aug 2011, H Amos: Remove K_WASH from input list, now a defined parameter

04 Sep 2013 - R. Yantosca - Bug fix: Prevent div-by-zero if F=0. Because F multiplies the whole expression for WASHFRAC, WASHFRAC=0 whenever F=0 anyway.

4.2.16 washfrac.liq_gas

!Subroutine WASHFRAC_LIQ_GAS returns the fraction of soluble liquid/gas phase species lost to washout.

INTERFACE:

```
SUBROUTINE WASHFRAC_LIQ_GAS( KO, CR, pKa, PP,      DT,
&                            F,  DZ, TK,  WASHFRAC, KIN )
```

INPUT PARAMETERS:

```
REAL(f8), INTENT(IN) :: KO      ! Henry's solubility constant [M/atm]
REAL(f8), INTENT(IN) :: CR      ! Henry's volatility constant [K]
REAL(f8), INTENT(IN) :: pKa     ! Henry's pH correction [1]
REAL(fp), INTENT(IN) :: PP     ! Precip rate thru bottom of the
```

```

! grid box [cm3 H2O/cm2 air/s]
REAL(fp), INTENT(IN)  :: DT      ! Timestep for washout event [s]
REAL(fp), INTENT(IN)  :: F       ! Fraction of grid box that is
! precipitating [unitless]
REAL(fp), INTENT(IN)  :: DZ      ! Height of grid box [cm]
REAL(fp), INTENT(IN)  :: TK      ! Temperature in grid box [K]

```

OUTPUT PARAMETERS:

```

REAL(fp), INTENT(OUT) :: WASHFRAC ! Fraction of species lost to washout
LOGICAL, INTENT(OUT)  :: KIN      ! T = washout is a kinetic process
! F = washout is an equilibrium process

```

REVISION HISTORY:

```

20 Jul 2004 - R. Yantosca - Initial version
(1 ) WASHFRAC_LIQ_GAS used to be an internal function to subroutine WASHOUT.
    This caused NaN's in the parallel loop on Altix, so we moved it to
    the module and now pass all arguments explicitly (bmy, 7/20/04)
16 Sep 2010 - R. Yantosca - Added ProTeX headers
10 Jan 2011 - H. Amos      - Remove AER from the argument list
03 Jun 2011 - H. Amos      - convert from a function to a subroutine and
    add AER to the argument list
16 Aug 2011 - H. Amos      - remove K_WASH from input list, now a defined
    parameter
16 Aug 2011 - H. Amos      - rename AER logical KIN to emphasize that washout
    is either a kinetic or equilibrium process

```

4.2.17 wetdep

Subroutine WETDEP computes the downward mass flux of species due to washout and rainout of aerosols and soluble species in a column. This subroutine implements an algorithm in which the precipitation fields come directly from the met archive.

INTERFACE:

```

SUBROUTINE WETDEP( am_I_Root, Input_Opt, State_Met,
&                 State_Chm, State_Diag, LS,          RC )

```

USES:

```

USE CMN_SIZE_MOD
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD
#endif
USE DEPO_MERCURY_MOD, ONLY : ADD_Hg2_WD
USE DEPO_MERCURY_MOD, ONLY : ADD_HgP_WD
USE DEPO_MERCURY_MOD, ONLY : ADD_Hg2_SNOWPACK
USE ErrCode_Mod

```

```

USE ERROR_MOD
USE Input_Opt_Mod,    ONLY : OptInput
USE State_Chm_Mod,   ONLY : ChmState
USE State_Diag_Mod,  ONLY : DgnState
USE State_Met_Mod,   ONLY : MetState
USE TIME_MOD,        ONLY : GET_TS_DYN
USE Species_Mod,     ONLY : Species
USE UnitConv_Mod,    ONLY : Convert_Spc_Units
    
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
LOGICAL,          INTENT(IN)    :: LS          ! =T for large-scale precip
                                                ! =F for convective precip

TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input options
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object
    
```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT) :: State_Chm    ! Chemistry State object
TYPE(DgnState),  INTENT(INOUT) :: State_Diag    ! Diagnostics State object
    
```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?
    
```

REMARKS:

Precipitation fields:
 =====

Layer	Formation of New Precip	Precipitation falling down	
			===== Top of Atm.
LM	QQ(L,I,J) 	PDOWN(LM,I,J) 	
LM-1	QQ(L,I,J) 	PDOWN(LM-1,I,J) 	
 	
3	QQ(L,I,J) 	PDOWN(3,I,J) 	
2	QQ(L,I,J) 	PDOWN(2,I,J) 	
1	QQ(L,I,J) 	PDOWN(1,I,J) 	

=====V=====V===== Ground

Where:

- (a) New formation forming in grid box (I,J,L) = QQ(L,I,J)
- (b) Precip coming in thru top of layer L = PDOWN(L+1,I,J)
- (c) Precip going out thru bottom of layer L = PDOWN(L, I,J)

Rainout:

=====

Rainout occurs when there is more precipitation in grid box (I,J,L) than in grid box (I,J,L+1). In other words, rainout occurs when the amount of rain falling through the bottom of grid box (I,J,L) is more than the amount of rain coming in through the top of grid box (I,J,L).

Soluble gases/aerosols are incorporated into the raindrops and are completely removed from grid box (I,J,LLPAR). There is no evaporation and "resuspension" of aerosols during a rainout event.

For large-scale (a.k.a. stratiform) precipitation, the first order rate constant for rainout in the grid box (I,J,L=LLPAR) (cf. Eq. 12, Jacob et al, 2000) is given by:

$$K_RAIN = K_MIN + \frac{Q}{L + W} \quad [\text{units: } s^{-1}]$$

and the areal fraction of grid box (I,J,L=LLPAR) that is actually experiencing large-scale precipitation (cf. Eq. 11, Jacob et al, 2000) is given by:

$$F' = \frac{Q}{K_RAIN * (L + W)} \quad [\text{unitless}]$$

Where:

K_MIN = minimum value for K_RAIN
= 1.0e-4 [s⁻¹]

L + W = condensed water content in cloud
= 1.5e-6 [cm³ H₂O/cm³ air]

Q = QQ = rate of precipitation formation
[cm³ H₂O / cm³ air / s]

For convective precipitation, K_RAIN = 5.0e-3 [s⁻¹], and the expression for F' (cf. Eq. 13, Jacob et al, 2000) becomes:

$$F' = \frac{FMAX * Q * \min\left\{\frac{DT}{TAU}, 1.0\right\}}{Q * \min\left\{\frac{DT}{TAU}, 1.0\right\} + FMAX * K_RAIN * (L + W)}$$

Where:

Q = QQ = rate of precipitation formation
[cm³ H₂O/cm³ air/s]

FMAX = maximum value for F'
= 0.3

DT = dynamic time step from the CTM [s]

TAU = duration of rainout event
= 1800 s (30 min)

L + W = condensed water content in cloud
= 2.0e-6 [cm³ H₂O/cm³ air]

K_RAIN and F' are needed to compute the fraction of species in grid box (I,J,L=LLPAR) lost to rainout. This is done in module routine RAINOUT.

Washout:

=====

Washout occurs when we have evaporation (or no precipitation at all) at grid box (I,J,L), but have rain coming down from grid box (I,J,L+1).

REVISION HISTORY:

20 Sep 2010 - R. Yantosca - Initial version, based on WETDEP
 28 Sep 2010 - H. Amos - Now define Q, QDOWN directly from MERRA met
 08 Oct 2010 - R. Yantosca - Adjusted OpenMP do loop
 09-Dec-2010 - H. Amos - Added PDOWN(L+1) > 0 to criterion for IS_WASHOUT
 09-Dec-2010 - H. Amos - SAFETY now prints PDOWN(L+1) instead of PDOWN(L)
 31-Dec-2010 - H. Amos - Clean up code, remove obsolete code
 31-Dec-2010 - H. Amos - Added comments
 27 May 2011 - R. Yantosca - Now pass F_RAINOUT to DO_WASHOUT_ONLY
 25 Aug 2014 - M. Sulprizio - Now accept Input_Opt as an argument
 01 Apr 2015 - L. Zhang - Don't do wetdep in nested-grid buffer zone
 02 Apr 2015 - E. Lundgren - Move tracer unit conversion from kg/kg to kg
 to within this routine
 04 Jun 2015 - E. Lundgren - Now accept am_I_Root and RC as arguments
 09 Jun 2015 - R. Yantosca - Now deposit Hg₂, HgP to snowpack regardless of

whether the dynamic ocean is used

04 Mar 2016 - C. Keller - Added option to scale convective fraction of large-scale precip.

22 Apr 2016 - R. Yantosca - Now get Is_Hg2, Is_HgP from species database

25 Apr 2016 - R. Yantosca - Now get the Hg category # from species database

29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.

30 Jun 2016 - R. Yantosca - Replace STT with Spc and DSTT with DSpc

01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo

05 Jul 2016 - R. Yantosca - Now replace IDWETD with State_Chm%Map_Wetdep

05 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code

05 Aug 2016 - R. Yantosca - Remove reference to Spc, it's not needed

17 Mar 2017 - R. Yantosca - Replace NSOL with State_Chm%nWetDep

24 Aug 2017 - M. Sulprizio- Rename routine from WETDEP_MERRA to WETDEP

28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.2.18 ls_k_rain

Function LS_K_RAIN computes K_RAIN, the first order rainout rate constant for large-scale (a.k.a. stratiform) precipitation.

INTERFACE:

```
FUNCTION LS_K_RAIN( Q ) RESULT( K_RAIN )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: Q          ! Rate of precipitation formation
                                     ! [cm3 H2O/cm3 air/s]
```

RETURN VALUE:

```
REAL(fp)                :: K_RAIN  ! 1st order rainout rate constant [1/s]
```

REVISION HISTORY:

18 Mar 2004 - R. Yantosca - Initial version

(1) Now made into a MODULE routine since we cannot call internal routines from w/in a parallel loop. Updated comments. (bmy, 3/18/04)

16 Sep 2010 - R. Yantosca - Added ProTeX headers

23 Oct 2015 - R. Yantosca - Now use parameter COND_WATER_CONTENT

4.2.19 ls_f_prime

Function LS_F_PRIME computes F', the fraction of the grid box that is precipitating during large scale (a.k.a. stratiform) precipitation.

INTERFACE:

```
FUNCTION LS_F_PRIME( Q, K_RAIN ) RESULT( F_PRIME )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: Q           ! Rate of precipitation formation
                                     ! [cm3 H2O/cm3 air/s]
REAL(fp), INTENT(IN) :: K_RAIN      ! 1st order rainout rate constant [1/s]
```

REMARKS:

```
REAL(fp)                :: F_PRIME ! Fraction of grid box undergoing
                                     ! large-scale precipitation [unitless]
```

REVISION HISTORY:

```
18 Mar 2004 - R. Yantosca - Initial version
(1 ) Now made into a MODULE routine since we cannot call internal routines
      from w/in a parallel loop. Updated comments. (bmy, 3/18/04)
16 Sep 2010 - R. Yantosca - Added ProTeX headers
23 Oct 2015 - R. Yantosca - Now use COND_WATER_CONTENT parameter
```

4.2.20 conv_f_prime

Function CONV_F_PRIME computes F', the fraction of the grid box that is precipitating during convective precipitation.

INTERFACE:

```
FUNCTION CONV_F_PRIME( Q, K_RAIN, DT ) RESULT( F_PRIME )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: Q           ! Rate of precipitation formation
                                     ! [cm3 H2O/cm3 air/s]
REAL(fp), INTENT(IN) :: K_RAIN      ! 1st order rainout rate constant [1/s]
REAL(fp), INTENT(IN) :: DT          ! Wet deposition timestep [s]
```

RETURN VALUE:

```
REAL(fp)                :: F_PRIME ! Frac. of grid box undergoing
                                     ! convective precipitation [unitless]
```

REVISION HISTORY:

18 Mar 2004 - R. Yantosca - Initial version

(1) Now made into a MODULE routine since we cannot call internal routines from w/in a parallel loop. Updated comments. (bmy, 3/18/04)

16 Sep 2010 - R. Yantosca - Added ProTeX headers

4.2.21 do_rainout_only

Subroutine DO_RAINOUT_ONLY removes species by rainout.

INTERFACE:

```

SUBROUTINE DO_RAINOUT_ONLY( am_I_Root,  LS,      I,
&                          J,          L,      IDX,
&                          ERRMSG,    F_RAINOUT, K_RAIN,
&                          DT,        DSpC,   Input_Opt,
&                          State_Met, State_Chm, State_Diag,
&                          RC                               )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD           ! Diagnostic flags
USE DIAG_MOD,             ONLY : AD16           ! ND16 diag array
USE DIAG_MOD,             ONLY : AD17           ! ND17 diag array
USE DIAG_MOD,             ONLY : AD39           ! ND39 diag array
USE DIAG_MOD,             ONLY : CT16           ! ND16 diag counter
USE DIAG_MOD,             ONLY : CT17           ! ND17 diag counter
#endif
USE ErrCode_Mod
USE ERROR_MOD,            ONLY : IT_IS_NAN      ! Test for NaN
USE GET_NDEP_MOD,         ONLY : SOIL_WETDEP    ! Wet deposited species
USE Input_Opt_Mod,        ONLY : OptInput       ! Input options type
USE State_Chm_Mod,        ONLY : ChmState       ! Chemistry State type
USE State_Met_Mod,        ONLY : MetState       ! Met State type
USE State_Diag_Mod,       ONLY : DgnState
#if defined( TOMAS )
USE TOMAS_MOD,            ONLY : IBINS, ICOMP, AQOXID
USE TOMAS_MOD,            ONLY : GETFRACTION
USE DIAG_MOD,             ONLY : AD05
#endif
#endif

```

INPUT PARAMETERS:

```

LOGICAL,                  INTENT(IN)   :: am_I_Root   ! Are we on root CPU?
LOGICAL,                  INTENT(IN)   :: LS          ! =T denotes LS precip
INTEGER,                  INTENT(IN)   :: I           ! Longitude index
INTEGER,                  INTENT(IN)   :: J           ! Latitude index

```

```

INTEGER,          INTENT(IN)    :: L           ! Level index
INTEGER,          INTENT(IN)    :: IDX         ! ND38 index
REAL(fp),        INTENT(IN)    :: F_RAINOUT   ! Fraction of grid box
                                                    ! undergoing rainout
REAL(fp),        INTENT(IN)    :: K_RAIN     ! Rainout constant
REAL(fp),        INTENT(IN)    :: DT         ! Rainout timestep [s]
CHARACTER(LEN=*), INTENT(IN)    :: ERRMSG    ! Error message
TYPE(OptInput),  INTENT(IN)    :: Input_Opt  ! Input options
TYPE(MetState),  INTENT(IN)    :: State_Met  ! Met State object

```

INPUT/OUTPUT PARAMETERS:

```

REAL(fp),          INTENT(INOUT) :: DSpc(:, :, :, :) ! Accumulator array [kg]
TYPE(ChmState),    INTENT(INOUT) :: State_Chm      ! Chemistry State object
TYPE(DgnState),    INTENT(INOUT) :: State_Diag     ! Diags State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)    :: RC           ! Success or failure?

```

REMARKS:

An IF statement in WETDEP decides if this rainout is to be done (and thus if this routine will be called. The criteria for rainout is:

There is rainout if there is new precip formation in the grid box (i.e. DQRLSAN(I,J,L) > 0) and the fraction of the grid box experiencing rainout (i.e. F_RAINOUT) is greater than or equal to the fraction of the grid box directly overhead experiencing precip (i.e. FTOP).

-- Helen Amos (9/10/10)

REVISION HISTORY:

```

16 Sep 2010 - R. Yantosca - Initial version
25 Aug 2014 - M. Sulprizio- Now accept Input_Opt as an argument
22 Sep 2015 - E. Lundgren - Input STT is now in kg/m2 (prev kg)
24 Sep 2015 - E. Lundgren - Now pass am_I_Root and RC as arguments
29 Sep 2015 - E. Lundgren - Now pass State_Chm as argument and use
                           local pointer for STT array [kg/m2]
30 Jun 2016 - R. Yantosca - Replace STT with Spc and DSTT with DSpc
05 Jul 2016 - R. Yantosca - Now replace IDWETD with State_Chm%Map_WetDep
09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
09 Nov 2017 - R. Yantosca - Return error condition to calling program

```

4.2.22 do_washout_only

Subroutine DO_WASHOUT_ONLY removes species by washout.

INTERFACE:

```

SUBROUTINE DO_WASHOUT_ONLY( am_I_Root, LS, I, J, L,
&                          IDX,      ERRMSG, QDOWN,
&                          Q,        F_WASHOUT, F_RAINOUT,
&                          DT,       PDOWN,   DSpC,
&                          Input_Opt, State_Met, State_Chm,
&                          State_Diag, RC,     REEVAP   )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD          ! Diagnostic flags
USE DIAG_MOD,             ONLY : AD05
USE DIAG_MOD,             ONLY : AD16           ! ND16 diag array
USE DIAG_MOD,             ONLY : AD17           ! ND17 diag array
USE DIAG_MOD,             ONLY : AD18           ! ND18 diag array
USE DIAG_MOD,             ONLY : AD39           ! ND39 diag array
USE DIAG_MOD,             ONLY : CT16           ! ND16 diag counter
USE DIAG_MOD,             ONLY : CT17           ! ND17 diag counter
USE DIAG_MOD,             ONLY : CT18           ! ND18 diag counter
#endifif
USE ErrCode_Mod
USE ERROR_MOD
USE GET_NDEP_MOD,        ONLY : SOIL_WETDEP     ! Wet deposited species
USE Input_Opt_Mod,      ONLY : OptInput        ! Input options
USE State_Chm_Mod,      ONLY : ChmState        ! Chemistry State object
USE State_Diag_Mod,     ONLY : DgnState
USE State_Met_Mod,      ONLY : MetState        ! Met State object
#if defined( TOMAS )
USE TOMAS_MOD,          ONLY : IBINS, ICOMP, AQOXID
#endifif

```

INPUT PARAMETERS:

```

LOGICAL,                 INTENT(IN)           :: am_I_Root   ! Are we on root CPU?
LOGICAL,OPTIONAL,        INTENT(IN)           :: REEVAP      ! Do re-evaporation?
LOGICAL,                 INTENT(IN)           :: LS          ! =T denotes LS precip
INTEGER,                 INTENT(IN)           :: I            ! Longitude index
INTEGER,                 INTENT(IN)           :: J            ! Latitude index
INTEGER,                 INTENT(IN)           :: L            ! Level index
INTEGER,                 INTENT(IN)           :: IDX          ! ND38 index
CHARACTER(LEN=*),        INTENT(IN)           :: ERRMSG       ! Error message
REAL(fp),                INTENT(IN)           :: QDOWN        ! Precip leaving thru
! bottom of box (I,J,L)
REAL(fp),                INTENT(IN)           :: Q            ! Precip forming or
! evaporating
! in box (I,J,L)
REAL(fp),                INTENT(IN)           :: F_WASHOUT     ! Fraction of grid box
! undergoing washout

```

```

REAL(fp),          INTENT(IN)    :: F_RAINOUT    ! Fraction of grid box
                                                         ! undergoing rainout
REAL(fp),          INTENT(IN)    :: DT           ! Rainout timestep [s]
REAL(fp),          INTENT(IN)    :: PDOWN(:, :, :) ! Precip
TYPE(OptInput),    INTENT(IN)    :: Input_Opt    ! Input options
TYPE(MetState),    INTENT(IN)    :: State_Met    ! Met State object

```

INPUT/OUTPUT PARAMETERS:

```

REAL(fp),          INTENT(INOUT) :: DSpc(:, :, :, :) ! Accumulator array
                                                         ! [kg/m2]
TYPE(ChmState),    INTENT(INOUT) :: State_Chm    ! Chemistry State object
TYPE(DgnState),    INTENT(INOUT) :: State_Diag   ! Diags State object

```

OUTPUT PARAMETERS:

```

INTEGER,           INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

A fraction ALPHA of the raindrops falling down from grid box (I,J,L+1) to grid box (I,J,L) will evaporate along the way. ALPHA is given by:

$$\begin{aligned}
 \text{ALPHA} &= \frac{\text{precip leaving (I,J,L+1)} - \text{precip leaving (I,J,L)}}{\text{precip leaving (I,J,L+1)}} \\
 &= \frac{-\text{QQ(L,I,J)} * \text{DZ(I,J,L)}}{\text{PDOWN(L+1,I,J)}}
 \end{aligned}$$

We assume that a fraction ALPHA2 = 0.5 * ALPHA of the previously rained-out aerosols and HNO3 coming down from level (I,J,L+1) will evaporate and re-enter the atmosphere in the gas phase in grid box (I,J,L). This process is called "resuspension".

For non-aerosol species, the amount of previously rained out mass coming down from grid box (I,J,L+1) to grid box (I,J,L) is figured into the total mass available for washout in grid box (I,J,L). We therefore do not have to use the fraction ALPHA2 to compute the resuspension.

NOTE from Hongyu Liu about ALPHA (hyl, 2/29/00)

=====
 If our QQ field was perfect, the evaporated amount in grid

box (I,J,L) would be at most the total rain amount coming from above (i.e. PDOWN(I,J,L+1)). But this is not true for the MOISTQ field we are using. Sometimes the evaporation in grid box (I,J,L) can be more than the rain amount from above. The reason is our "evaporation" also includes the effect of cloud detrainment. For now we cannot find a way to distinguish between the two. We then decided to release aerosols in both the detrained air and the evaporated air.

Therefore, we should use this term in the numerator:

$$-QQ(I,J,L) * BXHEIGHT(I,J,L)$$

instead of the term:

$$PDOWN(L+1)-PDOWN(L)$$

Recall that in make_qq.f we have restricted PDOWN to positive values, otherwise, QQ would be equal to PDOWN(L+1)-PDOWN(L).

=====

Update (V. Shah 6/29/16)

For GEOS-FP and MERRA2 met fields we use the following term in the numerator instead:

$$REEVAP(L,I,J) * BXHEIGHT(I,J,L)$$

=====

REVISION HISTORY:

- 16 Sep 2010 - R. Yantosca - Initial version
- 20 Sep 2010 - R. Yantosca - Update definition of ALPHA if we are doing partial re-evaporation.
- 28 Sep 2010 - H. Amos - Now check for NaN's with function IT_IS_NAN
- 31 Dec 2010 - H. Amos - new variable, TK, for temperature
- 26 May 2011 - R. Yantosca - Bug fix: Only apply the error trap for the condition WASHFRAC < 1d-3 for MERRA met
- 25 May 2011 - Q. Wang - new variable, TF, for total precip fraction
- 25 May 2011 - Q. Wang - Also pass F_RAINOUT via the arg list
- 25 May 2011 - Q. Wang - Correct the washfrac to make sure that washout is applied the area of F_washout instead of total area of (F_washout+F_rainout)
- 27 May 2011 - R. Yantosca - Added comments, readjusted IF statements to avoid floating-point problems
- 16 Aug 2011 - H. Amos - Replace logical AER with KIN to emphasize that washout is either modeled as a kinetic process or an equilibrium process
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP

25 Aug 2014 - M. Sulprizio- Now accept Input_Opt as an argument
 04 Feb 2015 - M. Sulprizio- Fix calculation of WETLOSS for non-aerosol tracers (C. Friedman)
 20 May 2015 - M. Sulprizio- Remove WASHFRAC < 1D-3 error trap for MERRA following recommendation by Viral Shah
 04 Jun 2015 - E. Lundgren - Now accept am_I_Root and RC as arguments
 22 Sep 2015 - E. Lundgren - Incoming tracer units are now [kg/m2]
 29 Sep 2015 - E. Lundgren - Now use local pointer for STT array [kg/m2]
 30 Jun 2016 - R. Yantosca - Replace STT with Spc and DSTT with DSpc
 05 Jul 2016 - R. Yantosca - Replace IDWETD with State_Chm%Map_WetDep
 09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
 09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.2.23 do_complete_reevap

Subroutine DO_COMPLETE_REEVAP re-evaporates all of the soluble species back into the atmosphere.

INTERFACE:

```

SUBROUTINE DO_COMPLETE_REEVAP( am_I_Root, LS,          I, J, L,
&                               IDX,          ERRMSG,    DT,
&                               DSpc,         Input_Opt,  State_Met,
&                               State_Chm,    State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD                ! Size parameters
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD                ! Diagnostic flags
USE DIAG_MOD,                   ONLY : AD05
USE DIAG_MOD,                   ONLY : AD16                ! ND16 diag array
USE DIAG_MOD,                   ONLY : AD17                ! ND17 diag array
USE DIAG_MOD,                   ONLY : AD18                ! ND18 diag array
USE DIAG_MOD,                   ONLY : AD39                ! ND39 diag array
USE DIAG_MOD,                   ONLY : CT16                ! ND16 diag counter
USE DIAG_MOD,                   ONLY : CT17                ! ND17 diag counter
USE DIAG_MOD,                   ONLY : CT18                ! ND18 diag counter
#endif
USE ErrCode_Mod
USE Error_Mod,                   ONLY : It_Is_Nan
USE GET_NDEP_MOD,               ONLY : SOIL_WETDEP        ! Wet deposited species
USE Input_Opt_Mod,              ONLY : OptInput
USE State_Chm_Mod,              ONLY : ChmState
USE State_Diag_Mod,             ONLY : DgnState
USE State_Met_Mod,              ONLY : MetState
#if defined( TOMAS )

```

```

      USE TOMAS_MOD,      ONLY : IBINS, ICOMP, AQOXID
#endif

```

INPUT PARAMETERS:

```

      LOGICAL,           INTENT(IN)    :: am_I_Root    ! Are we on root CPU?
      LOGICAL,           INTENT(IN)    :: LS           ! =T denotes LS precip
      INTEGER,          INTENT(IN)    :: I            ! Longitude index
      INTEGER,          INTENT(IN)    :: J            ! Latitude index
      INTEGER,          INTENT(IN)    :: L            ! Level index
      INTEGER,          INTENT(IN)    :: IDX         ! ND38 index
      CHARACTER(LEN=*), INTENT(IN)    :: ERRMSG       ! Error message
      REAL(fp),         INTENT(IN)    :: DT          ! Rainout timestep [s]
      TYPE(OptInput),   INTENT(IN)    :: Input_Opt   ! Input options
      TYPE(MetState),   INTENT(IN)    :: State_Met   ! Meteorology State obj

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(ChmState),   INTENT(INOUT) :: State_Chm    ! Chemistry State object
      TYPE(DgnState),   INTENT(INOUT) :: State_Diag   ! Diagnoss State object

      REAL(fp),         INTENT(INOUT) :: DSpc(:, :, :, :) ! Accumulator array
                                                ! [kg/m2]

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?

```

REVISION HISTORY:

```

16 Sep 2010 - R. Yantosca - Initial version
22 Sep 2015 - E. Lundgren - Incoming tracer units are now kg/m2 (prev kg)
24 Sep 2015 - E. Lundgren - Now pass am_I_Root and RC as arguments and
                           local pointer for STT array [kg/m2]
30 Sep 2015 - E. Lundgren - Now pass Input_Opt for use in TOMAS AQOXID
30 Jun 2016 - R. Yantosca - Replace STT with Spc and DSTT with DSpc
05 Jul 2016 - R. Yantosca - Replace IDWETD with State_Chm%Map_WetDep
09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
09 Nov 2017 - R. Yantosca - Return error condition to calling program

```

4.2.24 do_washout_at_sfc

Subroutine DO_WASHOUT_AT_SFC washes out the species at the surface.

INTERFACE:

```

      SUBROUTINE DO_WASHOUT_AT_SFC( am_I_Root,  LS,      I, J, L,
&                                IDX,        ERRMSG,  QDOWN,
&                                F,          DT,      DSpc,
&                                Input_Opt, State_Met, State_Chm,
&                                State_Diag, RC

```

USES:

```

    USE CMN_SIZE_MOD                ! Size parameters
#if defined( BPCH_DIAG )
    USE CMN_DIAG_MOD                ! Diagnostic flags
    USE DIAG_MOD,                   ONLY : AD16    ! ND16 diag array
    USE DIAG_MOD,                   ONLY : AD17    ! ND17 diag array
    USE DIAG_MOD,                   ONLY : AD18    ! ND18 diag array
    USE DIAG_MOD,                   ONLY : AD39    ! ND39 diag array
    USE DIAG_MOD,                   ONLY : CT16    ! ND16 diag counter
    USE DIAG_MOD,                   ONLY : CT17    ! ND17 diag counter
    USE DIAG_MOD,                   ONLY : CT18    ! ND18 diag counter
#endif
    USE ErrCode_Mod
    USE ERROR_MOD,                   ONLY : IT_IS_NAN ! Test for NaN
    USE GET_NDEP_MOD,                ONLY : SOIL_WETDEP ! Wet deposited species
    USE Input_Opt_Mod,               ONLY : OptInput ! Input options
    USE State_Chm_Mod,               ONLY : ChmState ! Chm State object
    USE State_Diag_Mod,              ONLY : DgnState
    USE State_Met_Mod,               ONLY : MetState ! Met State object

```

INPUT PARAMETERS:

```

    LOGICAL,           INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
    LOGICAL,           INTENT(IN)    :: LS           ! =T denotes LS precip
    INTEGER,           INTENT(IN)    :: I            ! Longitude index
    INTEGER,           INTENT(IN)    :: J            ! Latitude index
    INTEGER,           INTENT(IN)    :: L            ! Level index
    INTEGER,           INTENT(IN)    :: IDX         ! ND38 index
    CHARACTER(LEN=*), INTENT(IN)    :: ERRMSG       ! Error message
    REAL(fp),          INTENT(IN)    :: QDOWN       ! Precip leaving thru
    ! bottom of box (I,J,L)
    REAL(fp),          INTENT(IN)    :: F            ! Fraction of grid box
    ! undergoing precip
    REAL(fp),          INTENT(IN)    :: DT           ! Rainout timestep [s]
    TYPE(OptInput),    INTENT(IN)    :: Input_Opt   ! Input options
    TYPE(MetState),    INTENT(IN)    :: State_Met   ! Met State object

```

INPUT/OUTPUT PARAMETERS:

```

    REAL(fp),          INTENT(INOUT) :: DSpc(:, :, :, :) ! Accumulator array
    ! [kg/m2]
    TYPE(ChmState),    INTENT(INOUT) :: State_Chm   ! Chemistry State object
    TYPE(DgnState),    INTENT(INOUT) :: State_Diag  ! Diags State object

```

OUTPUT PARAMETERS:

```

    INTEGER,           INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

Assume all of the species precipitating down from grid box (I,J,L=2) to grid box (I,J,L=1) gets washed out in grid box (I,J,L=1).

REVISION HISTORY:

16 Sep 2010 - R. Yantosca - Initial version
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
 25 Aug 2014 - M. Sulprizio- Now accept Input_Opt as an argument
 04 Jun 2015 - E. Lundgren - Now accept am_I_Root and RC as arguments
 22 Sep 2015 - E. Lundgren - Tracer units are now kg/m2 (prev kg)
 29 Sep 2015 - E. Lundgren - Now use local pointer for STT array [kg/m2]
 30 Jun 2016 - R. Yantosca - Replace STT with Spc and DSTT with DSpC
 05 Jul 2016 - R. Yantosca - Now replace IDWETD with State_Chm%/Map_Wetdep
 09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
 09 Nov 2017 - R. Yantosca - Return error condition to calling program

4.2.25 safety

Subroutine SAFETY stops the run with debug output and an error message if negative species are found.

INTERFACE:

```

SUBROUTINE SAFETY( I,           J,           L,           N,
&                 A,           LS,         PDOWN,      QQ,
&                 ALPHA,      ALPHA2,     RAINFRAC,   WASHFRAC,
&                 MASS_WASH,  MASS_NOWASH, WETLOSS,    GAINED,
&                 LOST,       DSpC,       Spc,         RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod

```

INPUT PARAMETERS:

```

! Arguments
LOGICAL,          INTENT(IN)  :: LS          !
INTEGER,          INTENT(IN)  :: I           !
INTEGER,          INTENT(IN)  :: J           !
INTEGER,          INTENT(IN)  :: L           !
INTEGER,          INTENT(IN)  :: N           !
CHARACTER(LEN=*), INTENT(IN)  :: A           !
REAL(fp),         INTENT(IN)  :: PDOWN      !
REAL(fp),         INTENT(IN)  :: QQ         !
REAL(fp),         INTENT(IN)  :: ALPHA      !
REAL(fp),         INTENT(IN)  :: ALPHA2    !

```

```

REAL(fp),      INTENT(IN)  :: RAINFRAC      !
REAL(fp),      INTENT(IN)  :: WASHFRAC      !
REAL(fp),      INTENT(IN)  :: MASS_WASH     !
REAL(fp),      INTENT(IN)  :: MASS_NOWASH   !
REAL(fp),      INTENT(IN)  :: WETLOSS      !
REAL(fp),      INTENT(IN)  :: GAINED       !
REAL(fp),      INTENT(IN)  :: LOST         !
REAL(fp),      INTENT(IN)  :: DSpC(LLPAR)   !
REAL(fp),      INTENT(IN)  :: SpC(LLPAR)    !
! OUTPUT PARAMETERS:
INTEGER,       INTENT(OUT) :: RC            ! Success or failure?

```

REVISION HISTORY:

18 Mar 2004 - R. Yantosca - Initial version
(1) Now made into a MODULE routine since we cannot call internal routines
from w/in a parallel loop. Updated comments. (bmy, 3/18/04)
16 Sep 2010 - R. Yantosca - Added ProTeX headers

4.2.26 get_vud

Function GET_VUD returns the vertical updraft velocity in m/s at location I, J, L.

INTERFACE:

```
FUNCTION GET_VUD( State_Met, I, J, L ) RESULT( VUD )
```

USES:

```
USE DAO_MOD,          ONLY : IS_WATER
USE State_Met_Mod,    ONLY : MetState
```

INPUT PARAMETERS:

```
TYPE(MetState), INTENT(IN) :: State_Met  ! Meteorology State object
INTEGER,          INTENT(IN) :: I, J, L   ! Location
```

RETURN VALUE:

```
REAL(fp)          :: VUD    ! Vertical updraft velocity in m/s.
```

REVISION HISTORY:

12 Feb 2015 - C. Keller - Initial version
26 Feb 2015 - E. Lundgren - Remove dependency on pressure_mod since not used.
27 Jun 2016 - C. Keller - Use VUD_DEFINED instead of ESMF_compiler switch.
13 Mar 2017 - C. Keller - Bug fix: make sure VUD_DEFINED is updated after
defining VUD.
01 Jun 2017 - C. Keller - Set VUD_DEFINED only if UPDVVEL has valid values.

4.2.27 get_f

Function GET_F returns the scavenged fraction at location I, J, L and for the given rate constant K.

INTERFACE:

```
FUNCTION GET_F( State_Met, I, J, L, K ) RESULT( F )
```

USES:

```
USE DAO_MOD,           ONLY : IS_WATER
```

```
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

```
INTEGER,           INTENT(IN) :: I, J, L ! Lon, lat, level indices
```

```
REAL(fp),         INTENT(IN) :: K       ! Rate constant
```

RETURN VALUE:

```
REAL(fp)          :: F                  ! Fraction of species scavenged
                                           ! out of the updraft [1]
```

REVISION HISTORY:

12 Feb 2015 - C. Keller - Initial version

26 Feb 2015 - E. Lundgren - Remove dependency on pressure_mod since not used.

25 Jun 2015 - M. Sulprizio- Now pass FF as an optional argument for aerosols

25 Sep 2015 - R. Yantosca - Remove FF; we scale Kc in F_AEROSOL now.

4.2.28 init_wetscav

Subroutine INIT_WETSCAV initializes updraft velocity, cloud liquid water content, cloud ice content, and mixing ratio of water fields, which are used in the wet scavenging routines.

INTERFACE:

```
SUBROUTINE INIT_WETSCAV( am_I_Root, Input_Opt,
&                        State_Chm, State_Diag, RC )
```

USES:

```
USE CMN_Size_Mod
```

```
USE ErrCode_Mod
```

```
USE Input_Opt_Mod,  ONLY : OptInput
```

```
USE Species_Mod,    ONLY : Species
```

```
USE State_Chm_Mod,  ONLY : ChmState
```

```
USE State_Chm_Mod,  ONLY : Ind_
```

```
USE State_Diag_Mod, ONLY : DgnState
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(ChmState),  INTENT(IN)  :: State_Chm     ! Chemistry State object
TYPE(DgnState),  INTENT(IN)  :: State_Diag    ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT) :: RC           ! Success or failure?

```

REVISION HISTORY:

```

03 Sep 2015 - R. Yantosca - Split off from the old INIT_WETSCAV. We now
                        just allocate arrays here. Arrays are assigned
                        values from State_Met in SETUP_WETSCAV.
03 Sep 2015 - R. Yantosca - Also initialize IDWETD here; this makes routine
                        WETDEPID obsolete, so we can remove it.
22 Sep 2015 - R. Yantosca - Bug fix: only define the IDWETD array if
                        wetdep or drydep is turned on. Also only
                        print info if wetdep is turned on.
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
20 Jun 2016 - R. Yantosca - Now define species IDs in the init phase
09 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
09 Nov 2017 - R. Yantosca - Return error condition to calling program

```

4.2.29 setup_wetscav

Subroutine SETUP_WETSCAV initializes updraft velocity, cloud liquid water content, cloud ice content, and mixing ratio of water fields, which are used in the wet scavenging routines.

INTERFACE:

```

SUBROUTINE SETUP_WETSCAV( am_I_Root, Input_Opt,
&                        State_Met, State_Chm, RC )

```

USES:

```

USE CMN_Size_Mod
USE ErrCode_Mod
USE Error_Mod,          ONLY : Alloc_Err
USE Input_Opt_Mod,     ONLY : OptInput
USE PhysConstants,     ONLY : AIRMW
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE UnitConv_Mod

```

INPUT PARAMETERS:


```

LOGICAL,          INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)      :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)      :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT)   :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC           ! Success or failure?

```

REVISION HISTORY:

- 23 Feb 2000 - R. Yantosca - Initial version
- (1) References "e_ice.f" -- routine to compute Eice(T).
 - (2) Vud, CLDLIQ, CLDICE, C_H2O are all independent of tracer, so we can compute them once per timestep, before calling the cloud convection and wet deposition routines.
 - (3) Set C_H2O = 0 below -120 Celsius. E_ICE(T) has a lower limit of -120 Celsius, so temperatures lower than this will cause a stop with an error message. (bmy, 6/15/00)
 - (4) Replace {IJL}GLOB with IIPAR,JJPARG,LLPAR. Also rename PW to P. Remove IREF, JREF, these are obsolete. Now reference IS_WATER from "dao_mod.f" to determine water boxes.
 - (5) Removed obsolete code from 9/01. Updated comments and made cosmetic changes. (bmy, 10/24/01)
 - (6) Now use routine GET_PCENTER from "pressure_mod.f" to compute the pressure at the midpoint of grid box (I,J,L). Also removed P and SIG from the argument list (dsa, bdf, bmy, 8/20/02)
 - (7) Now reference T from "dao_mod.f". Updated comments. Now allocate Vud, C_H2O, CLDLIQ and CLDICE here on the first call. Now references ALLOC_ERR from "error_mod.f". Now set H2O2s and SO2s to the initial values from for the first call to COMPUTE_F . Now call WETDEPID on the first call to initialize the wetdep index array. (bmy, 1/27/03)
 - (8) Now references STT from "tracer_mod.f". Also now we call WETDEPID from "input_mod.f" (bmy, 7/20/04)
 - (9) Now references new function E_ICE, which is an analytic function of Kelvin temperature instead of Celsius. (bmy, 3/7/05)
- 16 Sep 2010 - R. Yantosca - Added ProTeX headers
- 18 Oct 2012 - R. Yantosca - Removed DEVEL from #ifdef statement, now use EXTERNAL_GRID || EXTERNAL_FORCING
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 26 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID_DRY. Remove dependency on pressure_mod.
- 28 Apr 2015 - E. Lundgren - Change PMID_DRY to PMID for Dalton's Law.
- 13 Aug 2015 - E. Lundgren - Incoming tracer units are now kg/kg dry air
- 03 Sep 2015 - R. Yantosca - Now allocate IDWETD

03 Sep 2015 - R. Yantosca - Now move array allocation into INIT_WETSCAV
 22 Sep 2015 - R. Yantosca - Bug fix: now initialize H2O2s and SO2s on
 the first call. This was done in INIT_WETSCAV,
 but that is now called before the restart file
 is read from disk.
 23 Sep 2015 - R. Yantosca - Now accept State_Chm as an argument
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 23 Jun 2016 - R. Yantosca - Remove reference to tracerid_mod.F
 30 Jun 2016 - R. Yantosca - Remove instances of STT and State_Chm%TRACERS
 04 Jul 2016 - M. Yannetti - Replace TCVV with species db MW and phys constant
 05 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code

4.2.30 cleanup_wetscav

Subroutine CLEANUP_WETSCAV deallocates all module arrays.

INTERFACE:

```
SUBROUTINE CLEANUP_WETSCAV( am_I_Root, RC )
```

USES:

```
USE ErrCode_Mod
!INPUT_PARAMETERS:
```

```
LOGICAL, INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

23 Feb 2000 - R. Yantosca - Initial version
 16 Sep 2010 - R. Yantosca - Added ProTeX headers
 03 Sep 2015 - R. Yantosca - Now deallocate IDWETD
 17 Mar 2017 - R. Yantosca - Now remove IDWETD, it's obsolete
 09 Nov 2017 - R. Yantosca - Return error condition to calling program

5 Boundary layer mixing modules

These modules contain routines to perform the boundary layer mixing.

5.1 Fortran: Module Interface mixing_mod.F90

Module mixing_mod.F90 is a wrapper module for the PBL mixing in GEOS-Chem.

INTERFACE:

```
MODULE MIXING_MOD
```

USES:

```
USE PRECISION_MOD
```

```
IMPLICIT NONE
```

```
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: INIT_MIXING
```

```
PUBLIC :: DO_MIXING
```

```
PUBLIC :: DO_TEND
```

PRIVATE MEMBER FUNCTIONS:

REVISION HISTORY:

```
04 Mar 2015 - C. Keller - Initial version.
```

5.1.1 init_mixing

Subroutine INIT_MIXING initialized the pbl mixing wrapper module.

INTERFACE:

```
SUBROUTINE INIT_MIXING ( am_I_Root, Input_Opt, State_Met,      &
                        State_Chm, State_Diag, RC              )
```

USES:

```
USE ErrCode_Mod
```

```
USE ERROR_MOD,      ONLY : ERROR_STOP
```

```
USE Input_Opt_Mod,  ONLY : OptInput
```

```
USE PBL_MIX_MOD,    ONLY : COMPUTE_PBL_HEIGHT
```

```
USE PBL_MIX_MOD,    ONLY : DO_PBL_MIX
```

```
USE State_Met_Mod,  ONLY : MetState
```

```
USE State_Chm_Mod,  ONLY : ChmState
```

```
USE State_Diag_Mod, ONLY : DgnState
```

```
USE VDIFF_MOD,      ONLY : DO_PBL_MIX_2
```

INPUT PARAMETERS:

```
LOGICAL,           INTENT(IN )  :: am_I_Root   ! root CPU?
```

```
TYPE(OptInput),    INTENT(IN )  :: Input_Opt   ! Input opts
```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState),   INTENT(INOUT)  :: State_Met   ! Meteorology State
TYPE(ChmState),   INTENT(INOUT)  :: State_Chm   ! Chemistry State
TYPE(DgnState),   INTENT(INOUT)  :: State_Diag  ! Diagnostics State
INTEGER,          INTENT(INOUT)  :: RC          ! Failure or success

```

!REMARKS

(A) While all dry deposition rates are calculated either in DO_PBL_MIX2 or DO_TEND, settling of aerosols is still computed in the dust/seasalt modules.

REVISION HISTORY:

```

04 Mar 2015 - C. Keller   - Initial version
26 Oct 2016 - R. Yantosca - Now also call COMPUTE_PBL_HEIGHT so that we
                          populate PBL quantities w/ the initial met
09 Mar 2017 - C. Keller   - Do not call COMPUTE_PBL_HEIGHT in ESMF env.
06 Nov 2017 - R. Yantosca - Return error condition to calling program

```

5.1.2 do_mixing

Subroutine DO_MIXING performs the PBL mixing.

INTERFACE:

```

SUBROUTINE DO_MIXING( am_I_Root, Input_Opt,  State_Met,      &
                     State_Chm, State_Diag, RC                )

```

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE PBL_MIX_MOD,       ONLY : DO_PBL_MIX
USE State_Met_Mod,     ONLY : MetState
USE State_Chm_Mod,     ONLY : ChmState
USE State_Diag_Mod,    ONLY : DgnState
USE VDIFFF_MOD,        ONLY : DO_PBL_MIX_2

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN  )  :: am_I_Root   ! root CPU?
TYPE(OptInput),   INTENT(IN  )  :: Input_Opt   ! Input opts

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState),   INTENT(INOUT)  :: State_Met   ! Met state
TYPE(ChmState),   INTENT(INOUT)  :: State_Chm   ! Chemistry State
TYPE(DgnState),   INTENT(INOUT)  :: State_Diag  ! Diagnostics State
INTEGER,          INTENT(INOUT)  :: RC          ! Failure or success

```

!REMARKS

(A) While all dry deposition rates are calculated either in DO_PBL_MIX2 or DO_TEND, settling of aerosols is still computed in the dust/seasalt modules.

REVISION HISTORY:

04 Mar 2015 - C. Keller - Initial version
 12 Aug 2015 - E. Lundgren - Input tracer units are now [kg/kg] and
 are converted to [v/v] for mixing
 30 Sep 2014 - E. Lundgren - Move unit conversion for DO_TEND to DO_TEND
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 08 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
 26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
 28 Sep 2017 - E. Lundgren - Move unit conversions to individual routines
 07 Nov 2017 - R. Yantosca - Now return error condition to calling routine

5.1.3 do.tend

Subroutine DO_TEND adds the species tendencies (dry deposition and emissions) to the species array.

INTERFACE:

```

SUBROUTINE DO_TEND( am_I_Root, Input_Opt, State_Met, State_Chm, &
                   State_Diag, OnlyAbovePBL, RC, DT )

```

USES:

```

USE CMN_SIZE_MOD, ONLY : IIPAR, JJPAR, LLPAR
USE DRYDEP_MOD, ONLY : DEPSAV
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP, SAFE_DIV
USE GET_NDEP_MOD, ONLY : SOIL_DRYDEP
USE HCO_INTERFACE_MOD, ONLY : GetHcoVal, GetHcoDiagn
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_UNDER_PBLTOP
USE PhysConstants, ONLY : AVO
USE Species_Mod, ONLY : Species
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_TS_DYN, GET_TS_CONV, GET_TS_CHEM
USE UnitConv_Mod, ONLY : Convert_Spc_Units
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD, ONLY : ND44
USE DIAG_MOD, ONLY : AD44
#endif
#if defined( USE_TEND )
USE TENDENCIES_MOD
#endif

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN  )           :: am_I_Root    ! root CPU?
TYPE(OptInput),  INTENT(IN  )           :: Input_Opt    ! Input opts
TYPE(MetState),  INTENT(IN  )           :: State_Met     ! Met state
LOGICAL,          INTENT(IN  )           :: OnlyAbovePBL ! Only above PBL?
REAL(fp),        INTENT(IN  ), OPTIONAL :: DT           ! Time step [s]

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT)          :: State_Chm    ! Chemistry state
TYPE(DgnState),  INTENT(INOUT)          :: State_Diag   ! Diags State
INTEGER,          INTENT(INOUT)         :: RC           ! Success/Failure

```

REVISION HISTORY:

```

04 Mar 2015 - C. Keller - Initial version
09 Mar 2015 - R. Yantosca - Bug fix: Use the drydep ID number instead of the
                           tracer number to index the AD44 drydep array
09 Mar 2015 - R. Yantosca - Bug fix: Remove an IF ( L1 == 1 ) block where
                           we define DRYDEPID. This isn't needed here.
10 Apr 2015 - C. Keller - Now exchange PARANOX loss fluxes via HEMCO
                           diagnostics.
12 Jun 2015 - R. Yantosca - Bug fix in SAFE_DIV: the denominator was
                           arranged wrongly. Now corrected.
18 Jun 2015 - C. Keller - Now restrict all emissions to chemistry grid
                           if UCX=false.
30 Sep 2015 - E. Lundgren - Now convert locally to kg/m2 for area-independent
                           compatibility between tracer units and flux
22 Mar 2016 - C. Keller - Bug fix: make sure drydep velocities are written
                           to diagnostics if emissions are zero.
16 Mar 2016 - E. Lundgren - Exclude specialty simulations in restriction of
                           all emissions to chemistry grid if UCX=false
29 Feb 2016 - C. Keller - Make sure PARANOx fluxes are applied to tracers.
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
25 May 2016 - E. Lundgren - Replace input_opt%TRACER_MW_KG with species
                           database field emMW_g (emitted species g/mol)
16 Jun 2016 - C. Miller - Now define species ID's with the Ind_ function
17 Jun 2016 - R. Yantosca - Only define species ID's on the first call
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
                           species ID from State_Chm%Map_Advect.
01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
19 Jul 2016 - R. Yantosca - Now bracket tendency calls with #ifdef USE_TEND
12 Apr 2017 - C. Keller - Bug fix: now allow negative emission fluxes
27 Sep 2017 - E. Lundgren - Apply unit conversion within routine instead
                           of in do_mixing
05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument
10 Oct 2017 - R. Yantosca - Archive drydep fluxes due to mixing for History

```

5.2 Fortran: Module Interface *pbl_mix_mod.F*

Module PBL_MIX_MOD contains routines and variables used to compute the planetary boundary layer (PBL) height and to mix tracers underneath the PBL top.

INTERFACE:

```
MODULE PBL_MIX_MOD
```

USES:

```
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

```
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: CLEANUP_PBL_MIX
PUBLIC  :: DO_PBL_MIX
PUBLIC  :: GET_FRAC_OF_PBL
PUBLIC  :: GET_FRAC_UNDER_PBLTOP
PUBLIC  :: GET_PBL_MAX_L
PUBLIC  :: GET_PBL_TOP_hPa
PUBLIC  :: GET_PBL_TOP_L
PUBLIC  :: GET_PBL_TOP_m
PUBLIC  :: GET_PBL_THICK
PUBLIC  :: INIT_PBL_MIX
PUBLIC  :: COMPUTE_PBL_HEIGHT
```

```
#if defined ( DEVEL )
```

```
  PUBLIC :: PBL_TOP_L, PBL_TOP_m
```

```
#endif
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: TURBDAY
```

REVISION HISTORY:

- 11 Feb 2005 - R. Yantosca - Initial version
- (1) Now modified for GCAP and GEOS-5 met fields (bmy, 5/24/05)
- (2) Remove reference to "CMN" and XTRA2. (bmy, 8/30/05)
- (3) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
- (4) Add INIT_PBL_MIX and COMPUTE_PBL_HEIGHT as PUBLIC routines
(lin, 5/29/09)
- (5) Extend tracers for APM simulation (GanLuo, 2010)
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 28 Feb 2012 - R. Yantosca - Added ProTeX headers
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

19 Nov 2014 - M. Yannetti - Added PRECISION_MOD
 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer
 compatible with the FlexChem implementation

5.2.1 do_pbl_mix

Subroutine DO_PBL_MIX is the driver routine for planetary boundary layer mixing. The PBL layer height and related quantities are always computed. Complete mixing of tracers underneath the PBL top is toggled by the DO_TURBDAY switch.

INTERFACE:

```

SUBROUTINE DO_PBL_MIX( am_I_Root, DO_TURBDAY, Input_Opt,
&                      State_Met, State_Chm, State_Diag, RC )

```

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
USE State_Chm_Mod,     ONLY : ChmState
USE State_Diag_Mod,    ONLY : DgnState
USE TIME_MOD,          ONLY : GET_TS_CONV
#if defined( USE_TEND )
  USE TENDENCIES_MOD
#endif
USE UnitConv_Mod,     ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)    :: am_I_Root    ! Root CPU?
LOGICAL,      INTENT(IN)    :: DO_TURBDAY  ! =T means call TURBDAY
                                           !   for full PBL mixing
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met  ! Meteorology State object
TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
INTEGER,      INTENT(INOUT)  :: RC         ! Return code

```

REVISION HISTORY:

11 Feb 2005 - R. Yantosca - Initial version
 07 Sep 2011 - G. Luo - Add modifications for APM
 28 Feb 2012 - R. Yantosca - Added ProTeX headers
 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
 22 Aug 2014 - R. Yantosca - Now declare State_Met INTENT(INOUT)
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected


```

                                species ID from State_Chm%Map_Advect.
06 Jul 2016 - R. Yantosca - Now pass State_Chm and am_I_Root to TURBDAY
19 Jul 2016 - R. Yantosca - Now bracket tendency calls with #ifdef USE_TEND
08 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
27 Sep 2017 - E. Lundgren - Apply unit conversion within routine instead
                                of in do_mixing
05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument
07 Nov 2017 - R. Yantosca - Now return error conditions to calling program

```

5.2.2 compute_pbl_height

Subroutine COMPUTE_PBL_HEIGHT computes the PBL height and other related quantities.

INTERFACE:

```
SUBROUTINE COMPUTE_PBL_HEIGHT( am_I_Root, State_Met, RC )
```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE ErrCode_Mod
USE PhysConstants         ! Scale height
USE State_Met_Mod,       ONLY : MetState

```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met    ! Meteorology State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?
```

REVISION HISTORY:

```

11 Feb 2005 - R. Yantosca - Initial version
(1 ) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
(2 ) Remove reference to "CMN" and XTRA2 -- they're obsolete. Also do not
     force BLTOP, BLTHIK to minimum values for GEOS-STRAT met fields.
     (bmy, 8/30/05)
(3 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
28 Feb 2012 - R. Yantosca - Added ProTeX headers
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
                             derived type object
22 Aug 2014 - R. Yantosca - Now declare State_Met INTENT(INOUT)
26 Feb 2015 - E. Lundgren - Replace GET_PEDGE with State_Met%PEDGE.
                             Remove dependency on pressure_mod.
06 Nov 2017 - R. Yantosca - Now return error to main program level
08 Jan 2018 - R. Yantosca - Now compute query field State_Met%InPbl

```

5.2.3 turbdlay

! Subroutine TURBDAY executes the GEOS-Chem boundary layer mixing algorithm (full PBL mixing).

INTERFACE:

```

      SUBROUTINE TURBDAY( am_I_root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, RC          )

```

USES:

```

      USE CMN_SIZE_MOD
      #if defined( BPCH_DIAG )
      USE CMN_DIAG_MOD
      USE DIAG_MOD,          ONLY : TURBFLUP
      #endif
      USE ErrCode_Mod
      USE Input_Opt_Mod,    ONLY : OptInput
      USE PhysConstants,   ONLY : AIRMW
      USE State_Chm_Mod,    ONLY : ChmState
      USE State_Met_Mod,    ONLY : MetState
      USE State_Diag_Mod,   ONLY : DgnState
      USE TIME_MOD,        ONLY : GET_TS_CONV

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
      TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options Object
      TYPE(MetState),  INTENT(IN)    :: State_Met     ! Metoerology State object

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
      TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

Original subroutine by Dale Allen, Univ of MD.

REVISION HISTORY:

30 Jan 1998 - I. Bey, R. Yantosca - Initial version

- (1) TURBDAY is written in Fixed-Form Fortran 90. Also use F90 syntax for declarations (bmy, 4/1/99).
- (2) New tracer concentrations are returned in TC.
- (3) PS(I,J) is ACTUAL surface pressure and not Psurface - PTOP
- (4) Change in tracer in kg is now stored in DTC(I,J,L,N). This makes it easier to compute diagnostic quantities. The new mixing ratio is computed as $TC(I,J,L,N) = TC(I,J,L,N) + DTC(I,J,L,N) / AD(I,J,L)$.
- (5) XTRA2(*,*,5) is the height of the PBL in # of layers. So if the PBL top is located in the middle of the 3rd sigma layer at (I,J) the value of XTRA2(I,J,5) would be 2.5. The XTRA2 variable is used by the HCTM drydep subroutines...it really is a historical holdover.
- (6) Restore the following NDxx diagnostics: (a) ND63 : Mass balance (CNVUPP) (b) ND15 : Mass change due to mixing in the boundary layer
- (7) Now pass TCVV and NCONV for the mass flux diagnostics. Also updated comments and cleaned up a few things. (bey, bmy, 11/10/99)
- (8) Remove PTOP and XNUMOL from the arg list. PTOP is now a parameter in "CMN_SIZE". XNUMOL is no longer used in TURBDAY. (bmy, 2/10/00)
- (9) Also removed obsolete ND63 diagnostics and updated comments. (bmy, 4/12/00)
- (10) Now use NTRC instead of NNPAR to dimension variables TC, TCVV, DTC, and DTCSUM (bmy, 10/17/00).
- (11) Removed obsolete code from 10/17/00 (bmy, 12/21/00)
- (12) If the PBL depth is very small (or zero), then assume a PBL depth of 2 mb -- this prevents NaN's from propagating throughout the code. Also updated comments & made cosmetic changes. (bmy, 3/9/01)
- (13) DTCSUM was declared twice but wasn't used. Eliminate declarations to DTCSUM. (bmy, 7/16/01)
- (14) XTRA2(IREF,JREF,5) is now XTRA2(I,J). Also updated comments. Also remove IREF, JREF and some debug output. (bmy, 9/25/01)
- (15) Removed obsolete commented out code from 9/01 (bmy, 10/24/01)
- (16) Now takes in P=PS-PTOP instead of PS. Redimension SIGE to (1:LLPAR+1).
- (17) Renamed PS to PZ so as not to conflict w/ the existing P variable. Now pass P-PTOP thru PZ, in order to ensure that P and AD are consistent w/ each other. Added parallel DO-loops. Updated comments, cosmetic changes. Now print a header to stdout on the first call, to confirm that TURBDAY has been called. (bmy, 4/11/02)
- (18) Now use GET_PEDGE from "pressure_mod.f" to compute the pressure at the bottom edge of grid box (I,J,L). Deleted obsolete code from 4/02. Removed PZ, SIGE from the argument list, since we now compute pressure from GET_PEDGE. (dsa, bdf, bmy, 8/22/02)
- (19) Now reference AD, PBL from "dao_mod.f". Now removed DXYP from the arg list, use GET_AREA_M2 from "grid_mod.f" instead. Now removed NCONV, ALPHA_d, ALPHA_n from the arg list. Now no longer reference SUNCOS. Now set A(:,:)=1 day & nite; we assume full mixing all the time regardless of SUNCOS. Updated comments, cosmetic changes.

(bmy, 2/11/03)

(20) Now can handle PBL field in meters for GEOS-4/fvDAS. Also the atmospheric scale height from CMN_GCTM. (bmy, 6/23/03)

(21) Now bundled into "pbl_mix_mod.f". Broke off the part which computes PBL height and related quantities into COMPUTE_PBL_HEIGHT. (bmy, 2/15/05)

28 Feb 2012 - R. Yantosca - Added ProTeX headers

2 Mar 2012 - R. Yantosca - Remove reference to GET_AREA_M2

09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object

15 Jul 2015 - C. Keller - Added tendencies module

06 Jul 2016 - R. Yantosca - Now pass State_Chm and am_I_Root as args so that we can loop over advected species

25 Jul 2016 - M. Yannetti - Now takes in State_Chm to access spec db

05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument

5.2.4 `get_frac_of_pbl`

Function GET_FRAC_OF_PBL returns the fraction of grid box (I,J,L) that lies within the planetary boundary layer.

INTERFACE:

```
FUNCTION GET_FRAC_OF_PBL( I, J, L ) RESULT( FRAC )
```

USES:

```
USE CMN_SIZE_MOD    ! Size parameters
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J, L    ! Lon, lat, lev indices
```

RETURN VALUE:

```
REAL(fp)           :: FRAC        ! Fraction of box (I,J,L) in the PBL
```

REVISION HISTORY:

11 Feb 2005 - R. Yantosca - Initial version

28 Feb 2012 - R. Yantosca - Added ProTeX headers

5.2.5 `get_frac_under_pbltop`

Function GET_FRAC_UNDER_PBLTOP returns the fraction of grid box (I,J,L) that lies underneath the planetary boundary layer top.

INTERFACE:

```
FUNCTION GET_FRAC_UNDER_PBLTOP( I, J, L ) RESULT( FRAC )
```

USES:

```
USE CMN_SIZE_MOD           ! Size parameters
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J, L  ! Lon, lat, level indices
```

RETURN VALUE:

```
REAL(fp)           :: FRAC      ! Fraction of box (I,J,L) below PBL top
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.6 get_pbl_max_l

Function GET_PBL_MAX_L returns the model level at the highest part of the planetary boundary layer.

INTERFACE:

```
FUNCTION GET_PBL_MAX_L() RESULT( TOP )
```

RETURN VALUE:

```
INTEGER  :: TOP  ! Highest extent of PBL [model levels]
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.7 get_pbl_top_hpa

Function GET_PBL_TOP_hPa returns the planetary boundary layer top [hPa] at a given GEOS-Chem surface location (I,J).

INTERFACE:

```
FUNCTION GET_PBL_TOP_hPa( I, J ) RESULT( TOP )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J  ! Lon and lat indices
```

RETURN VALUE:

```
REAL(fp)          :: TOP    ! PBL top [hPa]
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.8 get_pbl_top_l

Function GET_PBL_TOP_L returns the planetary boundary layer top [model levels] at a given GEOS-Chem surface location (I,J).

INTERFACE:

```
FUNCTION GET_PBL_TOP_L( I, J ) RESULT( TOP )
```

USES:**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I, J    ! Lon and lat indices
```

RETURN VALUE:

```
REAL(fp)          :: TOP    ! PBL top [model levels]
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.9 get_pbl_top_m

Function GET_PBL_TOP_m returns the planetary boundary layer top [m] at a given GEOS-CHEM surface location (I,J).

INTERFACE:

```
FUNCTION GET_PBL_TOP_m( I, J ) RESULT( TOP )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J    ! Lon and lat indices
```

RETURN VALUE:

```
REAL(fp)          :: TOP    ! PBL top [m]
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.10

Function GET_PBL_THICK returns the thickness of the PBL at a given surface location (I,J).

INTERFACE:

```
FUNCTION GET_PBL_THICK( I, J ) RESULT( THICK )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J      ! Lon and lat indices
```

RETURN VALUE:

```
REAL(fp)              :: THICK   ! PBL thickness [hPa]
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.2.11 init_pbl_mix

Subroutine INIT_PBL_MIX allocates and zeroes module arrays

INTERFACE:

```
SUBROUTINE INIT_PBL_MIX( am_I_root, RC )
```

USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC          ! Success or failure?
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
14 Nov 2014 - C. Keller    - Added error trap to prevent second allocation
                        attempt in ESMF environment.
07 Nov 2017 - R. Yantosca - Add am_I_root, RC as arguments so that we
                        can propagate the error to the top level
```

5.2.12 cleanup_pbl_mix

Subroutine CLEANUP_PBL_MIX allocates and zeroes module arrays.

INTERFACE:

```
SUBROUTINE CLEANUP_PBL_MIX
```

REVISION HISTORY:

```
11 Feb 2005 - R. Yantosca - Initial version
28 Feb 2012 - R. Yantosca - Added ProTeX headers
```

5.3 Fortran: Module Interface vdiff_mod.F90

Module VDIFF_MOD includes all routines for the non-local PBL mixing scheme.

INTERFACE:

```
MODULE VDIFF_MOD
```

USES:

```
#if defined( BPCH_DIAG )
  USE CMN_DIAG_MOD, ONLY : ND15, ND44           ! Diagnostics
#endif
  USE CMN_SIZE_MOD, ONLY : IIPAR, JJPARG, LLPARG ! Grid dimensions
  USE CMN_SIZE_MOD, ONLY : plev => LLPARG       ! # of levels
  USE ERROR_MOD, ONLY : DEBUG_MSG             ! Routine for debug output
  USE PhysConstants                             ! Physical constants
  USE PRECISION_MOD                             ! For GEOS-Chem Precision(fp)
  USE VDIFF_PRE_MOD, ONLY : PCNST              ! N_TRACERS
  USE VDIFF_PRE_MOD, ONLY : LPRT               ! Debug print?
  USE VDIFF_PRE_MOD, ONLY : LTURB             ! Do PBL mixing?
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
public :: DO_PBL_MIX_2
!PRIVATE DATA MEMBERS:

save

integer :: plevp

real(fp), parameter ::
```



```

    cpair = 1004.64e+0_fp,      &
    latvap = 2.5104e+06_fp,    &
    rhoh2o = 1.e+3_fp,         &
    tfh2o = 273.16e+0_fp,     &
    rair   = Rd,               &
    rh2o   = Rv,               &
    gravit = g0,               &
    zvir   = rh2o/rair - 1.,   &
    cappa  = Rd/cpair,        &
    r_g    = Rd / g0

! Obsolete variables and variables that are now defined with global params
! (ewl, 1/7/16)
real(fp), parameter ::      &
    rearth = 6.37122e+6_fp,  & ! not used
    cpwv   = 1.81e+3_fp,     & ! not used
    ra     = 1.e+0_fp/rearth, & ! not used
    epsilo = 0.622e+0_fp,    & ! not used
    lattice = 3.336e+5_fp,   & ! not used
    rair   = 287.04e+0_fp,    & ! now use global Rd
    rh2o   = 461.e+0_fp,     & ! now use global Rv
    gravit = 9.80616e+0_fp,   & ! now use global g0
    cappa  = rair/cpair,     & ! now use global Rd
    r_g    = rair / gravit,  & ! now use global Rd and g0

-----

... pbl constants
-----

! These are constants, so use PARAMETER tag
real(fp), parameter ::      &
    betam  = 15.e+0_fp,      & ! constant in wind gradient expression
    betas  = 5.e+0_fp,      & ! constant in surface layer gradient expression
    betah  = 15.e+0_fp,     & ! constant in temperature gradient expression
    fak    = 8.5e+0_fp,     & ! constant in surface temperature excess
    fakn   = 7.2e+0_fp,     & ! constant in turbulent prandtl number
    ricr   = .3e+0_fp,      & ! critical richardson number
    sfrac  = .1e+0_fp,      & ! surface layer fraction of boundary layer
    vk     = .4e+0_fp,      & ! von karmans constant

! These are assigned later, so we can't use the PARAMETER tag
real(fp) ::                  &
    g,                        & ! gravitational acceleration
    onet,                     & ! 1/3 power in wind gradient expression
    ccon,                      & ! fak * sfrac * vk
    binm,                      & ! betam * sfrac
    binh                        ! betah * sfrac

```

```

-----
... constants used in vertical diffusion and pbl
-----
real(fp) :: &
    zkmin          ! minimum kneutral*f(ri)
real(fp), allocatable :: ml2(:) ! mixing lengths squaredB
real(fp), allocatable :: qmincg(:) ! min. constituent concentration
                                ! counter-gradient term

integer :: &
    ntopfl, &      ! top level to which vertical diffusion is applied.
    npbl          ! maximum number of levels in pbl from surface

logical, parameter :: divdiff = .true. , arvdiff = .false.

logical, parameter :: pblh_ar = .true.

logical, parameter :: pbl_mean_drydep = .false. ! use mean concentration
                                                ! within the PBL for
                                                ! calculating drydep fluxes
logical, parameter :: drydep_back_cons = .false. ! backward consistency
                                                ! with previous GEOS-Chem
                                                ! drydep budgets
                                                !-- useless when
                                                ! pbl_mean_drydep=.false.

-----
Diagnostic quantities
-----

LOGICAL :: Do_ND44          = .FALSE. ! Do ND44 bpch diagnostic
LOGICAL :: Archive_DryDepMix = .FALSE. ! Is drydep flux diagnostic turned on?

```

REMARKS:

The non-local PBL mixing routine VDIFF modifies the specific humidity, (State_Met%SPHU) field. Therefore, we must pass State_Met as an argument to DO_PBL_MIX_2 and VDIFFDR with INTENT(INOUT).

Because logical_mod.F and tracer_mod.F have been superseded by Input_Opt, we now use VDIFF_PRE_MOD to supply values

REVISION HISTORY:

(1) This code is modified from mo_vdiff.F90 in MOZART-2.4. (lin, 5/14/09)
07 Oct 2009 - R. Yantosca - Added CVS Id Tag
24 Sep 2010 - J. Lin - Modified ND15 to account for all mixing
processes but not dry deposition and emissions.
17 Dec 2010 - R. Yantosca - Declare constants w/ the PARAMETER attribute

20 Dec 2010 - R. Yantosca - Bug fixes for the parallelization
 02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly
 involve explicitly using "D" exponents
 25 Mar 2011 - R. Yantosca - Corrected bug fixes noted by Jintai Lin
 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.2 met
 22 Jun 2012 - R. Yantosca - Now use pointers to flip arrays in vertical
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 24 Jun 2014 - R. Yantosca - Now get PCNST from vdiff_pre_mod.F90
 24 Nov 2014 - M. Yannetti - Added PRECISION_MOD
 07 Jan 2016 - E. Lundgren - Replace hard-coded physical params w/ global and
 remove unused parameters
 22 Jun 2016 - M. Yannetti - Replace TCVV with species db MW and phys constant
 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

5.3.1 pbinti

Subroutine PBINTI initializes time independent variables of pbl package

INTERFACE:

```
subroutine pbinti( gravx )
```

USES:

```
implicit none
```

INPUT PARAMETERS:

```
real(fp), intent(in) :: gravx      ! acceleration of gravity
```

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly
 involve explicitly using "D" exponents

5.3.2 vdiff

Subroutine vdiff is the driver routine to compute vertical diffusion of momentum, moisture, trace constituents and potential temperature.

INTERFACE:

```
subroutine vdiff( lat,          ip,          uwnd,          vwnd,          &
                 tadv,        pmid,        pint,          rpdel_arg,      &
                 rpdeli_arg, ztodt,        zm_arg,        shflx_arg,      &
                 sflx,        thp_arg,     as2,           pblh_arg,        &
                 kvh_arg,     kvm_arg,     tpert_arg,     qpert_arg,      &
                 cgs_arg,     shp,         wvflx_arg,     plonl,          &
                 Input_Opt,   State_Met,  State_Chm,     State_Diag,     &
                 taux_arg,    tauy_arg,   ustar_arg,     RC              )
```

USES:

```

#if defined( BPCH_DIAG )
  USE DIAG_MOD,          ONLY : TURBFLUP
#endif
  USE ErrCode_Mod
  USE Input_Opt_Mod,     ONLY : OptInput
  USE State_Chm_Mod,     ONLY : ChmState
  USE State_Diag_Mod,    ONLY : DgnState
  USE State_Met_Mod,     ONLY : MetState

  implicit none

```

INPUT PARAMETERS:

```

integer, intent(in) :: lat, ip ! latitude index, long tile index
integer, intent(in) :: plonl ! number of local longitudes
real(fp), intent(in) :: &
  ztodt ! 2 delta-t
real(fp), intent(in) :: &
  uwnd(:, :, :), & ! u wind input
  vwnd(:, :, :), & ! v wind input
  tadv(:, :, :), & ! temperature input
  pmid(:, :, :), & ! midpoint pressures
  pint(:, :, :), & ! interface pressures
  rpdel_arg(:, :, :), & ! 1./pdel (thickness bet interfaces)
  rpdeli_arg(:, :, :), & ! 1./pdeli (thickness bet midpoints)
  zm_arg(:, :, :), & ! midpoint geoptl height above sfc
  shflx_arg(:, :, :), & ! surface sensible heat flux (w/m2)
  sflx(:, :, :), & ! surface constituent flux (kg/m2/s)
  wvflx_arg(:, :, :), & ! water vapor flux (kg/m2/s)
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

real(fp), intent(inout) :: &
  as2(:, :, :, :), & ! moist, tracers after vert. diff
  shp(:, :, :), & ! specific humidity (kg/kg)
  thp_arg(:, :, :), & ! pot temp after vert. diffusion
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

real(fp), optional, intent(inout) :: &
  taux_arg(:, :, :), & ! x surface stress (n)
  tauy_arg(:, :, :), & ! y surface stress (n)
  ustar_arg(:, :, :), & ! surface friction velocity

real(fp), intent(inout) :: pblh_arg(:, :, :), & ! boundary-layer height [m]

```

OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC
real(fp), intent(out) :: &
    kvh_arg(:,:,:), & ! coefficient for heat and tracers
    kvm_arg(:,:,:), & ! coefficient for momentum
    tpert_arg(:,:), & ! convective temperature excess
    qpert_arg(:,:), & ! convective humidity excess
    cgs_arg(:,:,:) ! counter-grad star (cg/flux)

```

REMARKS:

Free atmosphere diffusivities are computed first; then modified by the boundary layer scheme; then passed to individual parameterizations mvdiff, qvdiff.

The free atmosphere diffusivities are based on standard mixing length forms for the neutral diffusivity multiplied by functions of Richardson number. $k = l^2 * |dv/dz| * f(ri)$. The same functions are used for momentum, potential temperature, and constituents.

The stable Richardson num function ($ri > 0$) is taken from Holtslag and Beljaars (1989), ECMWF proceedings. $f = 1 / (1 + 10*ri*(1 + 8*ri))$.

The unstable richardson number function ($ri < 0$) is taken from ccml. $f = \text{sqrt}(1 - 18*ri)$

REVISION HISTORY:

- (1) All arguments are full arrays now. Latitude slices are copied in local variables. (ccc, 11/19/09)
- 24 Sep 2010 - J. Lin - Moved call to ND15 at the end of vdiff.
Modified to account for all mixing processes.
- 02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly involve explicitly using "D" exponents
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 25 Jun 2014 - R. Yantosca - Now accept Input_Opt via the arg list
- 25 Jun 2014 - R. Yantosca - Remove references to tracer_mod.F
- 04 Aug 2016 - M. Yannetti - Now pass State_Chm as argument; replace TCVV with species db MW and phys constant

5.3.3 pbldif

Subroutine PBLDIF computes the atmospheric boundary layer. The nonlocal scheme determines eddy diffusivities based on a diagnosed boundary layer height and a turbulent velocity scale. Also, countergradient effects for heat and moisture, and constituents are included, along with temperature and humidity perturbations which measure the strength of convective thermals in the lower part of the atmospheric boundary layer.

References:

1. Holtslag, A. A. M., and B. A. Boville, 1993: *Local versus nonlocal boundary-layer diffusion in a global climate model*, *J. Clim.*, **6**, 1825-1842.

INTERFACE:

```

subroutine pbldif( th      ,q      ,z      ,u      ,v, &
                  t      ,pmid   ,kvf    ,cflx   ,shflx, &
                  kvm    ,kvh, &
                  cgh    ,cgq    ,cgs    ,pblh   ,tpert, &
                  qpert  ,wvflx  ,cgsh   ,plonl, &
                  taux   ,tauy   ,ustar  )

```

USES:

```

implicit none

```

INPUT PARAMETERS:

```

integer, intent(in) :: &
plonl
real(fp), intent(in) :: &
  th(plonl,plev), &      ! potential temperature [k]
  q(plonl,plev), &      ! specific humidity [kg/kg]
  z(plonl,plev), &      ! height above surface [m]
  u(plonl,plev), &      ! windspeed x-direction [m/s]
  v(plonl,plev), &      ! windspeed y-direction [m/s]
  t(plonl,plev), &      ! temperature (used for density)
  pmid(plonl,plev), &   ! midpoint pressures
  kvf(plonl,plevp), &   ! free atmospheric eddy diffsvty [m2/s]
  cflx(plonl,pcnst), &  ! surface constituent flux (kg/m2/s)
  wvflx(plonl), &      ! water vapor flux (kg/m2/s)
  shflx(plonl)          ! surface heat flux (w/m2)

```

INPUT/OUTPUT PARAMETERS:

```

real(fp), optional, intent(inout) :: &
  taux(plonl), &        ! x surface stress (n)
  tauy(plonl), &        ! y surface stress (n)
  ustar(plonl)          ! surface friction velocity

real(fp), intent(inout) :: pblh(plonl)      ! boundary-layer height [m]

```

OUTPUT PARAMETERS:

```

real(fp), intent(out) :: &
  kvm(plonl,plevp), &   ! eddy diffusivity for momentum [m2/s]
  kvh(plonl,plevp), &   ! eddy diffusivity for heat [m2/s]
  cgh(plonl,plevp), &   ! counter-gradient term for heat [k/m]
  cgq(plonl,plevp,pcnst), & ! counter-gradient term for constituents
  cgsh(plonl,plevp), &  ! counter-gradient term for sh
  cgs(plonl,plevp), &   ! counter-gradient star (cg/flux)
  tpert(plonl), &      ! convective temperature excess
  qpert(plonl)         ! convective humidity excess

```

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly
involve explicitly using "D" exponents

5.3.4 qvdiff

Subroutine QVDIFF solve vertical diffusion eqtn for constituent with explicit srfc flux.

INTERFACE:

```
subroutine qvdiff( ncnst, qm1, qflx, cc, ze, &
                 term, qp1, plonl )
```

USES:

```
implicit none
```

INPUT PARAMETERS:

```
integer, intent(in) :: &
    plonl
integer, intent(in) :: &
    ncnst                ! num of constituents being diffused
real(fp), intent(in) :: &
    qm1(plonl,plev,ncnst), & ! initial constituent
    qflx(plonl,ncnst), &    ! sfc q flux into lowest model level
    cc(plonl,plev), &      ! -lower diag coeff.of tri-diag matrix
    term(plonl,plev)       ! 1./(1. + ca(k) + cc(k) - cc(k)*ze(k-1))
```

INPUT/OUTPUT PARAMETERS:

```
real(fp), intent(inout) :: &
    ze(plonl,plev)       ! term in tri-diag. matrix system
```

OUTPUT PARAMETERS:

```
real(fp), intent(out) :: &
    qp1(plonl,plev,ncnst) ! final constituent
```

REMARKS:

Procedure for solution of the implicit equation follows :
Richtmyer and Morton (1967,pp 198-199)

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly
involve explicitly using "D" exponents

5.3.5 vdiffar

Subroutine VDIFFAR is the driver routine to compute vertical diffusion of trace constituents using archived coefficients for cgs and kvh. This is a gutted version of vdiff.

INTERFACE:

```

SUBROUTINE VDIFFAR( lat    ,tadv , &
                   pmid  ,pint ,rpdel_arg ,rpdeli_arg ,ztodt, &
                   sflx  ,as2  ,kvh_arg   ,cgs_arg   ,plonl )

```

USES:

```

implicit none

```

INPUT PARAMETERS:

```

integer, intent(in) :: lat      ! latitude index
integer, intent(in) :: plonl   ! lon tile dim
real(fp), intent(in) :: &
  ztodt , &                    ! 2 delta-t
  tadv(:, :, :), &             ! temperature input
  pmid(:, :, :), &            ! midpoint pressures
  pint(:, :, :), &           ! interface pressures
  rpdel_arg(:, :, :), &       ! 1./pdel (thickness bet interfaces)
  rpdeli_arg(:, :, :), &     ! 1./pdeli (thickness bet midpoints)
  sflx(:, :, :), &           ! surface constituent flux (kg/m2/s)
  kvh_arg(:, :, :), &        ! coefficient for heat and tracers
  cgs_arg(:, :, :), &        ! counter-grad star (cg/flux)

```

INPUT/OUTPUT PARAMETERS:

```

real(fp), intent(inout) :: &
  as2(:, :, :, :) ! moist, tracers after vert. diff

```

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly involve explicitly using "D" exponents

5.3.6 pbldifar

Subroutine PBLDIFAR is a modified version of pbldif which only calculates cgq given cgs.

INTERFACE:

```

SUBROUTINE PBLDIFAR( t, pmid, cflx, cgs, cgq, plonl )

```

USES:

```

implicit none

```


INPUT PARAMETERS:

```

integer, intent(in) :: &
  plonl
real(fp), intent(in) :: &
  t(plonl,plev), &      ! temperature (used for density)
  pmid(plonl,plev), &   ! midpoint pressures
  cflx(plonl,pcnst), &  ! surface constituent flux (kg/m2/s)
  cgs(plonl,plevp)      ! counter-gradient star (cg/flux)

```

OUTPUT PARAMETERS:

```

real(fp), intent(out) :: &
  cgq(plonl,plevp,pcnst) ! counter-gradient term for constituents

```

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly involve explicitly using "D" exponents

5.3.7 vdinti

Subroutine VDINTI initializes time independent fields for vertical diffusion. Calls initialization routine for boundary layer scheme.

INTERFACE:

```

SUBROUTINE VDINTI( am_I_Root, RC )

```

USES:

```

USE ErrCode_Mod
USE PRESSURE_MOD, ONLY : GET_AP, GET_BP

```

INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root      ! Are we on the root CPU?

```

OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC            ! Success or failure?

```

REVISION HISTORY:

02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly involve explicitly using "D" exponents
07 Nov 2017 - R. Yantosca - Add am_I_root, RC as arguments so that we can propagate the error to the top level

5.3.8 vdiffdr

Subroutine VDIFFDR calculates the vertical diffusion on a latitude slice of data.

1. The dummy argument as2 is in vv. (lin, 06/04/08)
2. TCVV and TRACER_MW_KG assume 12 g/mol for all HCs. Thus, when using them to convert units of HCs to be the inputs for vdiffdr, the converted units are NOT kg/kg for concentrations and kg/m2/s for surface flux. However, since the units for both inputs are consistent, there should not be any problem. (lin, 06/04/08)

INTERFACE:

```

SUBROUTINE VDIFFDR( am_I_Root, Input_Opt, State_Met,           &
                   State_Chm, State_Diag, RC                  )

```

USES:

```

USE DAO_MOD,           ONLY : IS_ICE, IS_LAND
USE DEPO_MERCURY_MOD,  ONLY : ADD_Hg2_DD, ADD_HgP_DD
USE DEPO_MERCURY_MOD,  ONLY : ADD_Hg2_SNOWPACK
#if defined( BPCH_DIAG )
USE DIAG_MOD,          ONLY : AD44
#endif
USE DRYDEP_MOD,        ONLY : DEPSAV
USE ErrCode_Mod
USE GET_NDEP_MOD,      ONLY : SOIL_DRYDEP
USE GLOBAL_CH4_MOD,    ONLY : CH4_EMIS
USE GC_GRID_MOD,       ONLY : GET_AREA_M2
USE Input_Opt_Mod,     ONLY : OptInput
USE MERCURY_MOD,       ONLY : HG_EMIS
USE OCEAN_MERCURY_MOD, ONLY : Fg !hma
USE OCEAN_MERCURY_MOD, ONLY : OMMFP => Fp
USE OCEAN_MERCURY_MOD, ONLY : LHg2HalfAerosol !cdh
USE PBL_MIX_MOD,       ONLY : GET_PBL_TOP_m, COMPUTE_PBL_HEIGHT, &
                           GET_PBL_MAX_L, GET_FRAC_UNDER_PBLTOP

USE Species_Mod,       ONLY : Species
USE State_Chm_Mod,     ONLY : ChmState
USE State_Chm_Mod,     ONLY : Ind_
USE State_Met_Mod,     ONLY : MetState
USE State_Diag_Mod,    ONLY : DgnState
USE TIME_MOD,          ONLY : GET_TS_CONV, GET_TS_EMIS, GET_TS_CHEM
#if defined( USE_TEND )
USE TENDENCIAS_MOD
#endif

! HEMCO update
USE HCO_INTERFACE_MOD, ONLY : GetHcoID, GetHcoVal, GetHcoDiagn

implicit none

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root      ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)      :: Input_Opt      ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)    :: State_Met      ! Meteorology State object
TYPE(ChmState), INTENT(INOUT)    :: State_Chm      ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)    :: State_Diag     ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)    :: RC              ! Success or failure?

```

REMARKS:

- (1) Need to declare the Meteorology State object (State_MET) with INTENT(INOUT). This is because VDIFF will modify the specific humidity field. (bmy, 11/21/12)
- (2) VDIFF also archives drydep fluxes to the soil NOx emissions module (by calling routine SOIL_DRYDEP) and to the ND44 diagnostic.
- (3) As of July 2016, we assume that all of the advected species are listed first in the species database. This is the easiest way to pass a slab to the TPCORE routine. This may change in the future. (bmy, 7/13/16)

REVISION HISTORY:

- (1) Calls to vdiff and vdiffar are now done with full arrays as arguments. (ccc, 11/19/09)
- 04 Jun 2010 - C. Carouge - Updates for mercury simulations with GTMM
- 25 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
- 24 Sep 2010 - J. Lin - Move ND15 to vdiff.
- 21 Dec 2010 - R. Yantosca - Add logical flags for different sim types
- 21 Dec 2010 - R. Yantosca - Now call ITS_A_FULLCHEM_SIM instead of relying on NCS == 0
- 22 Dec 2010 - C. Carouge - Combine array flipping w/ unit conversion to save on operations
- 02 Mar 2011 - R. Yantosca - Bug fixes for PGI compiler: these mostly involve explicitly using "D" exponents
- 26 Apr 2011 - J. Fisher - Use MERRA land fraction information
- 25 Oct 2011 - H. Amos - bring Hg2 gas-particle partitioning code into v9-01-02
- 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.2 in the same way as MERRA
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
- 22 Jun 2012 - R. Yantosca - Now use pointers to flip arrays in vertical
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object

18 Jun 2013 - M. Payer - Add emissions for offline aerosol simulation
 01 Aug 2013 - R. Yantosca - Now pass Input_Opt via the arg list
 01 Aug 2013 - J. Lin - Modified for Rn-Pb-Be simulation
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
 06 Jun 2014 - R. Yantosca - Fix parallelization error in the HEMCO
 modifications: Hold TOPMIX, TEMPBL private
 06 Jun 2014 - R. Yantosca - Wrap some debug printout in #if defined(DEBUG)
 25 Jun 2014 - R. Yantosca - Now get N_MEMBERS from input_mod.F
 16 Oct 2014 - C. Keller - Bug fix: now add deposition rates instead of
 overwriting them.
 26 Feb 2015 - E. Lundgren - Replace GET_PEDGE and GET_PCENTER with
 State_Met%PEDGE and State_Met%PMID.
 Remove dependency on pressure_mod.
 Use virtual temperature in hypsometric eqn.
 10 Apr 2015 - C. Keller - Now exchange PARANOX loss fluxes via HEMCO
 diagnostics.
 25 Jan 2016 - E. Lundgren - Update netcdf drydep flux diagnostic
 22 Apr 2016 - R. Yantosca - Now get Hg info from species database
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 26 May 2016 - E. Lundgren - Replace input_opt TRACER_MW_KG with species
 database field emMW_g (emitted species molec wt)
 16 Jun 2016 - K. Yu - Now define species ID's with the Ind_ function
 17 Jun 2016 - R. Yantosca - Only define species ID's on the first call
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
 13 Jul 2016 - R. Yantosca - Now use NA as loop index for advected species
 and ND as loop index for drydep species
 19 Jul 2016 - R. Yantosca - Now bracket tendency calls with #ifdef USE_TEND
 27 Jul 2016 - R. Yantosca - Bug fix: set nDrydep=0 if drydep is turned off
 04 Aug 2016 - M. Yannetti - Replace TCVV with species db MW and phys constant
 08 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
 20 Oct 2016 - R. Yantosca - Only write eflx/dflx debug on the root CPU
 15 Mar 2017 - C. Keller - Remove variable trc_id (use N instead)
 16 Mar 2017 - R. Yantosca - Remove N_MEMBERS, it's obsolete
 30 Jun 2017 - R. Yantosca - For now, print out eflx/dflx if DEVEL=y
 05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument
 10 Oct 2017 - R. Yantosca - Now archive drydep flux (mixing) for History
 07 Nov 2017 - R. Yantosca - Now accept RC as an argument, so that we can
 pass error codes back to the top level

5.3.9 do_pbl_mix_2

Subroutine DO_PBL_MIX_2 is the driver routine for planetary boundary layer mixing. The PBL layer height and related quantities are always computed. Mixing of tracers underneath the PBL top is toggled by the DO_TURBDAY switch.

INTERFACE:

```

SUBROUTINE DO_PBL_MIX_2( am_I_Root, DO_VDIFF,  Input_Opt, State_Met,      &
                        State_Chm, State_Diag, RC                          )

```

USES:

```

#if defined( BPCH_DIAG )
  USE CMN_DIAG_MOD,      ONLY : ND44
#endif
  USE DAO_MOD,          ONLY : AIRQNT
  USE ERROR_MOD,       ONLY : DEBUG_MSG
  USE ErrCode_Mod
  USE Input_Opt_Mod,   ONLY : OptInput
  USE PBL_MIX_MOD,     ONLY : INIT_PBL_MIX
  USE PBL_MIX_MOD,     ONLY : COMPUTE_PBL_HEIGHT
  USE State_Chm_Mod,   ONLY : ChmState
  USE State_Met_Mod,   ONLY : MetState
  USE State_Diag_Mod,  ONLY : DgnState
  USE TIME_MOD,        ONLY : ITS_TIME_FOR_EMIS
  USE UnitConv_Mod,    ONLY : Convert_Spc_Units

```

```

IMPLICIT NONE

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
LOGICAL,      INTENT(IN)  :: DO_VDIFF    ! Switch which turns on PBL
                                                ! mixing of tracers
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met  ! Meteorology State object
TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag  ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)  :: RC          ! Success or failure?

```

REVISION HISTORY:

```

11 Feb 2005 - R. Yantosca - Initial version
21 Dec 2010 - R. Yantosca - Now only call SETEMIS for fullchem simulations
22 Dec 2010 - R. Yantosca - Bug fix: print debug output only if LPRT=T
05 Mar 2013 - R. Yantosca - Add am_I_root, Input_Opt, RC arguments
05 Mar 2013 - R. Yantosca - Now call SETEMIS with am_I_Root, Input_Opt, RC
05 Mar 2013 - R. Yantosca - Now use Input_Opt%ITS_A_FULLCHEM_SIM
25 Mar 2013 - M. Payer    - Now pass State_Chm object via the arg list
01 Aug 2013 - R. Yantosca - Now pass the Input_Opt object to VDIFFDR

```

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
22 Aug 2014 - R. Yantosca - Renamed DO_TURBDAY to DO_VDIFF for clarity
16 Nov 2015 - E. Lundgren - Update air quantities after VDIFFDR call
since specific humidity is updated
13 Jul 2016 - R. Yantosca - Remove STT, we can point to State_Chm%Species
in the VDIFFDR routine directly
27 Sep 2017 - E. Lundgren - Apply unit conversion within routine instead
of in do_mixing
05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument
7 Nov 2017 - R. Yantosca - Now send error condition back to top level

5.4 Fortran: Module Interface vdiff_pre_mod.F90

Module VDIFF_PRE_MOD contains variables used in VDIFF_MOD.

INTERFACE:

```
MODULE VDIFF_PRE_MOD
```

USES:

```
IMPLICIT NONE
```

```
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Init_VDIFF_PRE  
PUBLIC :: Cleanup_VDIFF_PRE  
PUBLIC :: Set_VDIFF_VALUES
```

PUBLIC DATA MEMBERS:

```
LOGICAL, PUBLIC      :: LPRT           ! Passes LPRT to vdiff_mod  
LOGICAL, PUBLIC      :: LTURB          ! Passes LTURB to vdiff_mod  
INTEGER, PUBLIC      :: PCNST          ! Passes N_TRACERS to vdiff_mod
```

REVISION HISTORY:

01 Jun 2009 - C. Carouge & J. Lin - Initial version
07 Oct 2009 - R. Yantosca - Added CVS Id tag
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
24 Jun 2014 - R. Yantosca - Now add PCNST as a module variable
24 Jun 2014 - R. Yantosca - Add routine SET_VDIFF_VALUES, since we need to
pass N_TRACERS, LPRT, LTURB to vdiff_mod.F90
now that logical_mod.F, tracer_mod.F are gone.
24 Jun 2014 - R. Yantosca - Renamed to vdiff_pre_mod.F90
24 Nov 2014 - M. Yannetti - Added PRECISION_MOD
30 Jun 2016 - M. Sulprizio - Remove NCS, NDRYDEP, ND15, ND44, IIPAR, JJPAR
since they are not used anymore

5.4.1 init_vdiff_pre

Subroutine INIT_VDIFF_PRE allocates all module arrays.

INTERFACE:

```
SUBROUTINE Init_VDIFF_PRE( am_I_Root, RC )
```

USES:

```
USE ErrCode_Mod
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC          ! Success or failure?
```

REMARKS:

Need to add error-checking on the allocation statements, so that we exit the code upon error.

REVISION HISTORY:

```
19 Nov 2012 - R. Yantosca - Added ProTeX headers
24 Jun 2014 - R. Yantosca - Now accept Input_Opt via the arg list
24 Jun 2014 - R. Yantosca - Now allocate EMIS_SAVE to the # of tracers
                        in the simulation (i.e. INIT_OPT)
22 May 2015 - R. Yantosca - Remove variables made obsolete by HEMCO
```

5.4.2 cleanup_vdiff_pre

Subroutine CLEANUP_VDIFF_PRE deallocates all module arrays.

INTERFACE:

```
SUBROUTINE Cleanup_VDIFF_PRE( am_I_Root, RC )
```

USES:

```
USE ErrCode_Mod
```

```
IMPLICIT NONE
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)  :: am_I_Root  ! Are we on the root CPU?
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC          ! Success or failure?
```

REMARKS:**REVISION HISTORY:**

```
19 Nov 2012 - R. Yantosca - Initial version
```

5.4.3 set_vdiff_values

Subroutine SET_VDIFF_VALUES initializes the PCNST value, which is the number of advected species. This is needed in vdiff_mod.F90.

INTERFACE:

```
SUBROUTINE Set_VDIFF_VALUES( am_I_Root, Input_Opt, State_Chm, RC )
```

USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root   ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt   ! Input Options object
TYPE(ChmState),  INTENT(IN)  :: State_Chm   ! Chemistry State object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC          ! Success or failure?
```

REMARKS:

This routine has to be called after routine READ_INPUT_FILE.

REVISION HISTORY:

```
24 Jun 2014 - R. Yantosca - Initial version
```

6 Dry deposition modules

These modules contain routines to remove species from the atmosphere by dry deposition. Related modules compute the land cover that is used by the dry deposition routines.

6.1 Fortran: Module Interface drydep_mod.F

Module DRYDEP_MOD contains variables and routines for the GEOS-Chem dry deposition scheme.

INTERFACE:

```
MODULE DRYDEP_MOD
```

USES:

```

      USE CMN_SIZE_MOD           ! Size parameters
      USE DAO_MOD               ! Met field subroutines
#if defined( BPCH_DIAG )
      USE CMN_DIAG_MOD         ! Diag counters & flags
      USE DIAG_MOD,           ONLY : AD44      ! Diagnostic arrays
-----
      This diagnostic requires some fixes. Comment out for now (mps,3/2/18)
      USE DIAG_MOD,           ONLY : AD44b, AD_RA, AD_ZL
-----
#endif
      USE ERROR_MOD            ! Error handling routines
      USE GC_GRID_MOD,        ONLY : GET_AREA_CM2 ! Grid box surface areas [cm2]
      USE PBL_MIX_MOD         ! Boundary layer quantities
#if defined( TOMAS )
      USE TOMAS_MOD           ! For TOMAS microphysics
#endif
      USE PhysConstants        ! Physical constants
      USE PRECISION_MOD       ! For GEOS-Chem Precision (fp)

      IMPLICIT NONE
      PRIVATE

```

PUBLIC MEMBER FUNCTIONS:

```

      PUBLIC :: CLEANUP_DRYDEP
      PUBLIC :: DO_DRYDEP
      PUBLIC :: INIT_DRYDEP
      PUBLIC :: INIT_WEIGHTSS

```

PUBLIC DATA MEMBERS:

```

      PUBLIC :: DEPNAME
      PUBLIC :: DEPSAV
      PUBLIC :: NUMDEP
      PUBLIC :: NTRAIKD
      PUBLIC :: IDEP,   IRGSS,   IRAC,   IRCLS
      PUBLIC :: IRGSO,  IRLU,    IRI,    IRCLO, DRYCOEFF
      PUBLIC :: NDVZIND ! MSL -> For MPI broadcasting in GCHP

```

REMARKS:

References:

- =====
- (1) Baldocchi, D.D., B.B. Hicks, and P. Camara, "A canopy stomatal resistance model for gaseous deposition to vegetated surfaces", *Atmos. Environ.* 21, 91-101, 1987.
 - (2) Brutsaert, W., "Evaporation into the Atmosphere", Reidel, 1982.
 - (3) Businger, J.A., et al., "Flux-profile relationships in the atmospheric surface layer", *J. Atmos. Sci.*, 28, 181-189, 1971.
 - (4) Dwight, H.B., "Tables of integrals and other mathematical data", MacMillan, 1957.
 - (5) Guenther, A., and 15 others, A global model of natural volatile organic compound emissions, *J. Geophys. Res.*, 100, 8873-8892, 1995.
 - (6) Hicks, B.B., and P.S. Liss, "Transfer of SO₂ and other reactive gases across the air-sea interface", *Tellus*, 28, 348-354, 1976.
 - (7) Jacob, D.J., and S.C. Wofsy, "Budgets of reactive nitrogen, hydrocarbons, and ozone over the Amazon forest during the wet season", *J. Geophys. Res.*, 95, 16737-16754, 1990.
 - (8) Jacob, D.J., et al, "Deposition of ozone to tundra", *J. Geophys. Res.*, 97, 16473-16479, 1992.
 - (9) Levine, I.N., "Physical Chemistry, 3rd ed.", McGraw-Hill, New York, 1988.
 - (10) Munger, J.W., et al, "Atmospheric deposition of reactive nitrogen oxides and ozone in a temperate deciduous forest and a sub-arctic woodland", *J. Geophys. Res.*, in press, 1996.
 - (11) Walcek, C.J., R.A. Brost, J.S. Chang, and M.L. Wesely, "SO₂, sulfate, and HNO₃ deposition velocities computed using regional landuse and meteorological data", *Atmos. Environ.*, 20, 949-964, 1986.
 - (12) Wang, Y.H., paper in preparation, 1996.
 - (13) Wesely, M.L., "Improved parameterizations for surface resistance to gaseous dry deposition in regional-scale numerical models", Environmental Protection Agency Report EPA/600/3-88/025, Research Triangle Park (NC), 1988.
 - (14) Wesely, M. L., Parameterization of surface resistance to gaseous dry deposition in regional-scale numerical models. *Atmos. Environ.*, 23 1293-1304, 1989.
 - (15) Price, H., L. Jaegl, A. Rice, P. Quay, P.C. Novelli, R. Gammon, Global Budget of Molecular Hydrogen and its Deuterium Content: Constraints from Ground Station, Cruise, and Aircraft Observations, submitted to *J. Geophys. Res.*, 2007.
 - (16) Karl, T., Harley, P., Emmons, L., Thornton, B., Guenther, A., Basu, C., Turnipseed, A., and Jardine, K.: Efficient Atmospheric Cleansing of Oxidized Organic Trace Gases by Vegetation, *Science*, 330, 816-819, 10.1126/science.1192534, 2010.

REVISION HISTORY:

- 27 Jan 2003 - R. Yantosca - Moved standalone routines into this module
- (1) Bug fix: Do not assume NO₂ is the 2nd drydep species. This causes a mis-indexing for CANOPYNOX. Now archive ND44 diagnostic in kg for

- Radon runs in routine DRYFLXRnPbBe; convert to kg/s in diag3.f
(bmy, 1/27/03)
- (2) Now references "grid_mod.f" and the new "time_mod.f". Renamed DRYDEP routine to DO_DRYDEP for consistency w/ other drivers called from the MAIN program. (bmy, 2/11/03)
 - (3) Added error check in DRYFLX for SMVGEAR II (bmy, 4/28/03)
 - (4) Added drydep of N2O5. Now added PBLFRAC array, which is the fraction of each level below the PBL top. Also now compute drydep throughout the entire PBL, in order to prevent short-lived species such as HNO3 from being depleted in the shallow GEOS-3 surface layer.
(rjp, bmy, 7/21/03)
 - (5) Bug fix for GEOS-4 in DRYFLXRnPbBe (bmy, 12/2/03)
 - (6) Now made CFRAC, RADIAT local variables in DO_DRYDEP (bmy, 12/9/03)
 - (7) Now enclose AD44 in !\$OMP CRITICAL block for drydep flux (bmy, 3/24/04)
 - (8) Now handle extra carbon & dust tracers (rjp, tdf, bmy, 4/1/04)
 - (9) Added routines AERO_SFCRS1, AERO_SFCRSII. Increased MAXDEP to 25. Now handles extra carbon & dust tracers. (rjp, tdf, bmy, 4/1/04)
 - (10) Increased MAXDEP to 26. Added A_RADI and A_DEN module variables. Other modifications for size-resolved drydep. (rjp, bec, bmy, 4/20/04)
 - (11) Increased MAXDEP to 35 and handle extra SOA tracers (rjp, bmy, 7/13/04)
 - (12) Now references "logical_mod.f", "directory_mod.f", and "tracer_mod.f"
(bmy, 7/20/04)
 - (13) Add Hg2, HgP as drydep tracers (eck, bmy, 12/8/04)
 - (14) Updated for AS, AHS, LET, NH4aq, SO4aq (cas, bmy, 1/6/05)
 - (15) Now references "pbl_mix_mod.f". Removed PBLFRAC array. (bmy, 2/22/05)
 - (16) Now include SO4s, NITs tracers. Now accounts for hygroscopic growth of seasalt aerosols when computing aerodynamic resistances.
(bec, bmy, 4/13/05)
 - (17) Now modified for GEOS-5 and GCAP met fields (bmy, 5/25/05)
 - (18) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (19) Now change Reynold's # criterion from 1 to 0.1 in DEPVEL. Also change Henry's law constant for Hg2. Also increase MAXDEP from 35 to 37. (eck, djj, bmy, 2/1/06)
 - (20) Bug fix in INIT_DRYDEP (bmy, 4/17/06)
 - (21) Now bundle function DIFFG into "drydep_mod.f". Also updated for SOG4 and SOA4 tracers. Bug fix in INIT_DRYDEP. (dkh, bmy, 5/24/06)
 - (22) Fix typo in INIT_DRYDEP (dkh, bmy, 6/23/06)
 - (23) Add H2 and HD as drydep tracers. Added subroutine DRYFLXH2HD for H2HD offline sim (phs, 9/18/07)
 - (24) Extra error check for small RH in AERO_SFCRII (phs, 6/11/08)
 - (25) Added 15 more dry deposition species (tmf, 7/31/08)
 - (26) Modify dry deposition to follow the non-local PBL scheme.
(lin, ccc, 5/29/09)
 - (27) Minor bug fix in mol wts for ALPH, LIMO (bmy, 10/19/09)
 - (28) Change MAXDEP from 50 to 81 (win, 7/14/09)
 - (28a) modified to use Zhang 2001 for all non-size resolved aerosols (hotp)
 - (29) Add aromatics SOA (dkh)
 - (30) Add new species. Some tracers give 2 deposition species: ISOPN-> ISOPNB

and ISOPND. (fp)

(31) Updates for mercury simulation (ccc, 5/17/10)

(32) Add POPs (eck, 9/20/10)

(33) Increase MAXDEP to 51 for dicarbonyls simulation. (ccc, 10/8/10)

01 Aug 2011 - J. Fisher - Set aerosol dry deposition velocity to 0.03 cm/s
over snow and ice based on Nilsson & Rannik, 2001

21 Dec 2011 - M. Payer - Updates for sea salt (jaegle 5/11/11)

22 Dec 2011 - M. Payer - Added ProTeX headers

10 Jan 2012 - M. Payer - Update to use local surface pressure

01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90

26 Mar 2012 - R. Yantosca - Now reference CMN_SIZE_MOD at the top of module

26 Mar 2012 - R. Yantosca - Replace NNTYPE, NNPOLY, NNVEGTYPE w/ the
values NTYPE, NPOLY, NVEGTYPE from CMN_SIZE

26 Mar 2012 - R. Yantosca - Now retire MODIN and RDDRYCF; read drydep inputs
from a netCDF file w/ routine READ_DRYDEP_INPUTS

26 Mar 2012 - R. Yantosca - Reorganize module USE statements for clarity

09 Apr 2012 - R. Yantosca - Now replace IJREG, IJLAND, IJUSE, XYLAI arrays
with IREG, ILAND, IUSE, XLAI.

31 Jul 2012 - R. Yantosca - Modifications for grid-independence

11 Dec 2012 - M. Long - Now call READ_DRYDEP_INPUTS from INIT_DRYDEP

11 Dec 2012 - R. Yantosca - Now call INIT_WEIGHTSS from INIT_DRYDEP

13 Dec 2012 - R. Yantosca - Remove reference to obsolete CMN_DEP_mod.F

26 Feb 2013 - R. Yantosca - Now use Input_Opt fields where possible

13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA +
semivolatile POA simulations (H. Pye)

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust
aerosols (T.D. Fairlie)

29 Jan 2014 - R. Yantosca - Now set MAXDEP=105 for all simulations. For
TOMAS we had MAXDEP=100; this is close enough.

23 Jun 2014 - R. Yantosca - Removed references to logical_mod.F

25 Jul 2014 - R. Yantosca - Removed reference to commsoil_mod.F

12 Nov 2014 - M. Yannetti - Added PRECISION_MOD, changed REAL*8 to REAL(fp)

26 Feb 2015 - E. Lundgren - Replace GET_PEDGE with State_Met%PEDGE

03 Mar 2015 - C. Keller - Disabled DRYFLX. Now done in mixing_mod.F90

15 Jun 2015 - R. Yantosca - Disabled DRYFLXRnPbBe, also done in mixing_mod

01 Oct 2015 - R. Yantosca - Remove DVZ_MINVAL function

01 Oct 2015 - R. Yantosca - Added PRIVATE flag ID_ACET

06 Jan 2016 - E. Lundgren - Use global physical parameters

13 Jul 2016 - R. Yantosca - Remove DRYHg0, DRYHg2, DRYHgP ID flags

13 Jul 2016 - R. Yantosca - Also make NDVZIND a local variable for now

29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

10 Mar 2017 - M. Sulprizio- Add special treatment for ALD2, following what is
done for ACET. Drydep over the oceans is handled
via the HEMCO SeaFlux extension.

16 Mar 2017 - R. Yantosca - Remove references to obsolete vars in Input_Opt

24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA

6.1.1 do_drydep

Subroutine DO_DRYDEP is the driver for the GEOS-CHEM dry deposition scheme. DO_DRYDEP calls DEPVEL to compute deposition velocities [m/s], which are then converted to [cm/s]. Drydep frequencies are also computed. (lwh, gmg, djj, 1989, 1994; bmy, 2/11/03, 5/25/05)

INTERFACE:

```

SUBROUTINE DO_DRYDEP( am_I_Root, Input_Opt, State_Met,
&                    State_Chm, State_Diag, RC          )

```

USES:

```

USE ErrCode_Mod
USE HCO_ERROR_MOD
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_DIAGN_MOD,      ONLY : Diagn_Update
USE Input_Opt_Mod,     ONLY : OptInput
USE Species_Mod,       ONLY : Species
USE State_Chm_Mod,     ONLY : ChmState
USE State_Diag_Mod,    ONLY : DgnState
USE State_Met_Mod,     ONLY : MetState
USE UnitConv_Mod

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

NOTE: Modeled aerosol dry deposition velocities over snow and ice surfaces in the Arctic are much higher than estimated from measured values (e.g., Ibrahim et al. [1983]; Duan et al. [1988]; Nilsson and Rannik [2001]). We will impose a dry deposition velocity of 0.03 cm/s for all aerosols over snow and ice surfaces. (Jenny Fisher, 01 Aug 2011)
References (see full citations above):

- ```

=====
(1) Wesely, M. L., 1989
(2) Jacob, D.J., and S.C. Wofsy, 1990

```

**REVISION HISTORY:**

- (1 ) Remove SUNCOS, USTAR, AZO, OBK from the arg list; now reference these as well as AD and T from "dao\_mod.f". Cleaned up code and updated comments. Now only order tracer numbers into NTRAIN on the first call. Now force double-precision with "D" exponents. Now also reference IDTNOX, IDTOX, etc. from "tracerid\_mod.f". Bundled into "drydep\_mod.f" (bmy, 11/19/02)
  - (2 ) Now make sure that the PBL depth (THIK) is greater than or equal to the thickness of the first layer. Now initialize PBLFRAC array on each call. (rjp, bmy, 7/21/03)
  - (3 ) Now declare CFRAC, RADIAT, AZO, USTAR as local variables, which are returned by METERO. CFRAC and RADIAT have also been deleted from "CMN\_DEP". (bmy, 12/9/03)
  - (4 ) Now use explicit formula for IJLOOP to allow parallelization. Also reference LPRT from "logical\_mod.f" (bmy, 7/20/04)
  - (5 ) Now use routines from "pbl\_mix\_mod.f" to get PBL quantities, instead of re-computing them here. Removed PBLFRAC array. Removed reference to "pressure\_mod.f". Removed reference to header file CMN. Parallelize DO-loops. (bmy, 2/22/05)
  - (6 ) Now define RHB as a local array, which is defined in METERO and then passed to DEPVEL. (bec, bmy, 4/13/05)
  - (7 ) Now dimension AZO for GEOS or GCAP met fields. Remove obsolete variables. (swu, bmy, 5/25/05)
  - (8 ) Remove reference to TRACERID\_MOD, it's not needed (bmy, 10/3/05)
- 01 Aug 2011 - J. Fisher - Set aerosol dry deposition velocity to 0.03 cm/s over snow and ice based on Nilsson & Rannik, 2001
- 15 Aug 2011 - R. Yantosca - Now reference IDTxxx flags from tracerid\_mod.f
- 07 Oct 2011 - R. Yantosca - Rename SUNCOS30 to SUNCOS\_MID, which is the cos(SZA) at the midpt of the chemistry timestep
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 10 Jan 2012 - M. Payer - Added local surface pressure
- 26 Mar 2012 - R. Yantosca - Now read drydep inputs from a netCDF file via routine READ\_DRYDEP\_INPUTS
- 26 Mar 2012 - R. Yantosca - Remove calls to obsolete MODIN, RDDRYCF routines
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 28 Nov 2012 - R. Yantosca - Now make SUNCOS\_MID a local array of size MAXIJ, populated from State\_Met%SUNCOSmid
- 11 Dec 2012 - R. Yantosca - Now do not call READ\_DRYDEP\_INPUTS and INIT\_WEIGHTSS when using the ESMF environment
- 11 Dec 2012 - R. Yantosca - Remove FIRST variable, as we now read inputs from disk in routine INIT\_DRYDEP
- 12 Dec 2012 - R. Yantosca - Now pass State\_Met to DEPVEL
- 26 Feb 2013 - R. Yantosca - Now use Input\_Opt fields where possible. This facilitates connection to the GEOS-5 GCM.

31 May 2013 - R. Yantosca - Now pass Input\_Opt & State\_Chm to DEPVEL  
 15 Jan 2015 - R. Yantosca - Add new netCDF diagnostics structure  
 12 Aug 2015 - E. Lundgren - Tracer units are now [kg/kg]  
 01 Oct 2015 - R. Yantosca - Remove DVZ\_MINVAL; now use species database  
 to set the minimum drydep velocity for aerosols  
 01 Oct 2015 - R. Yantosca - Now use species database to set Vd for aerosols  
 over snow/ice; combine with the prior update to  
 enforce a minimum Vd for aerosol species  
 01 Oct 2015 - R. Yantosca - Remove references to tracerid\_mod.F (except  
 for TOMAS for now)  
 21 Jan 2016 - E. Lundgren - Update netcdf diagnostics for drydep trcr names  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 17 May 2016 - M. Sulprizio- Remove IJLOOP and change dimension of arrays  
 from (MAXIJ) to (IIPAR,JJPAR)  
 18 Jul 2016 - M. Sulprizio- Remove special handling for ISOPN and MVKN for  
 ND44 netCDF diagnostic. Family tracers have been  
 eliminated.  
 13 Jul 2017 - M. Sulprizio- Add scaling of species drydep relative to HNO3,  
 PAN, and ISOPN (from K. Travis, J. Fisher)  
 05 Oct 2017 - R. Yantosca - Now update State\_Diag%DryDepVel diagnostic

### 6.1.2 metero

Subroutine METERO calculates meteorological constants needed for the dry deposition velocity module. (lwh, gmg, djj, 1989, 1994; bmy, 10/3/05)

#### INTERFACE:

```

SUBROUTINE METERO(State_Met, CZ1, TCO, OBK, CFRAC,
& RADIAT, AZO, USTR, ZH, LSNOW,
& RHB, PRESSU, W10, SUNCOS_MID)

```

#### USES:

```

USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### OUTPUT PARAMETERS:

```

LOGICAL, INTENT(OUT) :: LSNOW (IIPAR,JJPAR) ! Flag for denoting snow/ice
REAL(f8), INTENT(OUT) :: CZ1 (IIPAR,JJPAR) ! Midpt ht of 1st model lev
! [m]
REAL(f8), INTENT(OUT) :: TCO (IIPAR,JJPAR) ! Grid box sfc temp [K]
REAL(f8), INTENT(OUT) :: OBK (IIPAR,JJPAR) ! Monin-Obhukov length [m]
REAL(f8), INTENT(OUT) :: CFRAC (IIPAR,JJPAR) ! Column cloud fraction
! [unitless]

```

```

REAL(f8), INTENT(OUT) :: RADIAT(IIPAR,JJPARG) ! Solar radiation @ ground
! [W/m2]
REAL(f8), INTENT(OUT) :: RHB (IIPAR,JJPARG) ! Rel humidity at sfc
! [unitless]
REAL(f8), INTENT(OUT) :: USTR (IIPAR,JJPARG) ! Friction velocity [m/s]
REAL(f8), INTENT(OUT) :: ZH (IIPAR,JJPARG) ! PBL height [m]
REAL(f8), INTENT(OUT) :: PRESSU(IIPAR,JJPARG) ! Local surface press [Pa]
REAL(f8), INTENT(OUT) :: W10 (IIPAR,JJPARG) ! 10 meter windspeed [m/s]
REAL(f8), INTENT(OUT) :: SUNCOS_MID(IIPAR,JJPARG) ! COS(SZA) @ midpt of
! current chem timestep
REAL(f8), INTENT(OUT) :: AZO(IIPAR,JJPARG) ! Roughness heights, by grid box

```

**REMARKS:**

References (see full citations above):

=====

- (1 ) Wesely, M. L., 1989.
- (2 ) Jacob, D.J., and S.C. Wofsy, 1990

**REVISION HISTORY:**

- (1 ) Now reference GET\_PEDGE from "pressure\_mod.f". Now reference T from "dao\_mod.f". Removed obsolete code & comments, and added new documentation header. Now force double precision with "D" exponents. Now compute OBK here as well. Bundled into F90 module "drydep\_mod.f" (bmy, 11/20/02)
  - (2 ) Now reference CLDFRC, RADSWG, ZO, USTAR from "dao\_mod.f". Also now pass CFRAC, RADIAT, AZO, USTR back to the calling routine via the arg list. (bmy, 12/9/03)
  - (3 ) Now use explicit formula for IJLOOP to allow parallelization (bmy, 7/20/04)
  - (4 ) Now compute ZH and LSNOW here instead of w/in DO\_DRYDEP. Parallelize DO-loops. Now use BXHEIGHT from "dao\_mod.f" instead of computing the thickness of the 1st level here. Remove reference to "pressure\_mod.f". Remove reference to T from "dao\_mod.f". Now reference ALBD from "dao\_mod.f" (bmy, 2/22/05)
  - (5 ) Now references RH from "dao\_mod.f". Now passes relative humidity from the surface layer back via RHB argument. (bec, bmy, 4/13/05)
  - (6 ) Now call GET\_OBK from "dao\_mod.f" to get the M-0 length for both GEOS or GCAP met fields. Remove local computation of M-0 length here. Also now dimension AZO appropriately for GCAP or GEOS met fields. Remove obsolete variables. (swu, bmy, 5/25/05)
  - (7 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
  - (8 ) Move XLTMMP function to module MEGANUT\_MOD. (ccc, 11/20/09)
  - (9 ) Add sea level pressure and 10m windspeed as arguments (jaegle 5/11/11)
- 22 Dec 2011 - M. Payer - Added ProTeX headers  
10 Jan 2012 - M. Payer - Added local surface pressure  
09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met



derived type object

28 Nov 2012 - R. Yantosca - Add SUNCOS\_MID to the argument list and populate that with State\_Met%SUNCOSmid

21 Oct 2013 - R. Yantosca - Bug fix: need to hold SP private in OMP loop

25 Jul 2014 - R. Yantosca - Now remove reference to function SFCWINDSQR

17 May 2016 - M. Sulprizio- Remove IJLOOP and change dimension of output arrays from (MAXIJ) to (IIPAR,JJPARG)

---

### 6.1.3 depvel

Subroutine DEPVEL computes the dry deposition velocities using a resistance-in-series model.

#### INTERFACE:

```

SUBROUTINE DEPVEL(am_I_Root, Input_Opt, State_Met, State_Chm,
& RADIAT, TEMP, SUNCOS,
& FO, HSTAR, XMW, AIROSOL,
& USTAR, CZ1, OBK, CFRAC,
& ZH, LSNOW, DVEL, ZO,
& RHB, PRESSU, W10, RC)

```

#### USES:

```

USE Drydep_Toolbox_Mod, ONLY : BioFit
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology state object

REAL(f8), INTENT(IN) :: RADIAT (IIPAR,JJPARG) ! Solar radiation [W/m2]
REAL(f8), INTENT(IN) :: TEMP (IIPAR,JJPARG) ! Temperature [K]
REAL(f8), INTENT(IN) :: SUNCOS (IIPAR,JJPARG) ! Cos of solar zenith angle
LOGICAL, INTENT(IN) :: AIROSOL(NUMDEP) ! =T denotes aerosol species
REAL(f8), INTENT(IN) :: FO (NUMDEP) ! React. factor for oxidation
 ! of biological substances
REAL(f8), INTENT(IN) :: HSTAR (NUMDEP) ! Henry's law constant
REAL(f8), INTENT(IN) :: XMW (NUMDEP) ! Molecular weight [kg/mol]
REAL(f8), INTENT(IN) :: USTAR (IIPAR,JJPARG) ! Friction velocity [m/s]
REAL(f8), INTENT(IN) :: CZ1 (IIPAR,JJPARG) ! Alt @ which Vd is computed

```

```

! [m]
REAL(f8), INTENT(IN) :: OBK (IIPAR,JJPARG) ! Monin-Obhukov length [m]
REAL(f8), INTENT(IN) :: CFRAC (IIPAR,JJPARG) ! Surface cloud fraction
REAL(f8), INTENT(IN) :: ZH (IIPAR,JJPARG) ! Roughness height [m]
REAL(f8), INTENT(IN) :: RHB (IIPAR,JJPARG) ! Relative humidity [%]
REAL(f8), INTENT(IN) :: PRESSU (IIPAR,JJPARG) ! Surface pressure [hPa]
REAL(f8), INTENT(IN) :: W10 (IIPAR,JJPARG) ! Wind speed @ 10m altitude
! [m/s]

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?
REAL(f8), INTENT(OUT) :: DVEL(IIPAR,JJPARG,NUMDEP) ! Drydep velocity [m/s]

```

**REMARKS:**

Need as landtype input for each grid square (I,J); see CMN\_DEP\_mod.F

```

IREG(I,J) - # of landtypes in grid square
ILAND(I,J,LDT) - Land type ID for element LDT =1, IREG(I,J)
 (could be from any source - mapped to deposition
 surface ID in input unit 65)
IJUSE(I,J,LDT) - Fraction ((per mil) of gridbox area occupied by
 land type element LDT

```

Need as leaf area index; see CMN\_DEP\_mod.F

```

XLAI(I,J,LDT) - Leaf Area Index of land type element LDT

```

Need as meteorological input for each grid square(I,J) (passed):

```

RADIAT(I,J) - Solar radiation in W m-2
TEMP(I,J) - Surface air temperature in K
SUNCOS(I,J) - Cosine of solar zenith angle
LSNOW(I,J) - Logical for snow and sea ice
RHB(I,J) - Relative humidity at the surface
PRESSU(I,J) - Local surface pressure
W10(I,J) - 10m wind speed

```

Need as input for each species K (passed):

```

FO(K) - reactivity factor for oxidation of biological substances
HSTAR(K) - Henry's Law constant
XMW(K) - Molecular weight (kg/mole) of species K
 (used to calculate molecular diffusivities)
AIROSOL(K) - LOGICAL flag (T = aerosol species;
 F = gas-phase species)

```

Also need to call the following subroutines to read drydep input data:

```

READ_DRYDEP_INPUTS - (in this module) Reads in Olson land type

```

indices, dry deposition land type indices,  
 default roughness heights, and polynomial  
 coefficients. (This supersedes MODIN, RDDRYCF)  
 COMPUTE\_OLSON\_LANDMAP - (in olson\_landmap\_mod.F90). Reads in the  
 Olson land types at native resolution and re-bins  
 them on-the-fly to the GEOS-Chem grid resolution.  
 (This supersedes RDLAND)  
 "rdlai.f" - reads Leaf Area Indices from files "lai\*\*.global"

Some variables used in the subroutine (passed):

LRGERA(I,J) T -> stable atmosphere; a high aerodynamic resistance  
 (RA=1.E4 m s<sup>-1</sup>) is imposed; else RA is calculated  
 USTAR(I,J) - Friction velocity (m s<sup>-1</sup>)  
 CZ1(I,J) - Altitude (m) at which deposition velocity is computed  
 OBK(I,J) - Monin-Obukhov length (m): set to 1.E5 m under neutral  
 conditions  
 CFRAC(I,J) - Fractional cloud cover  
 ZH(I,J) - Mixing depth (m)

Some variables used in the subroutine:

MAXDEP - the maximum number of species for which the dry  
 deposition calculation is done  
 ZO(LDT) - Roughness height (m) for specific surface type indexed  
 by LDT  
 RSURFC(K,LDT) - Bulk surface resistance (s m<sup>-1</sup>) for species K to  
 surface LDT  
 C1X(K) - Total resistance to deposition (s m<sup>-1</sup>) for species K

Returned:

DVEL(I,J,K) - Deposition velocity (m s<sup>-1</sup>) of species K

References:

=====

Baldocchi, D.D., B.B. Hicks, and P. Camara, A canopy stomatal  
 resistance model for gaseous deposition to vegetated surfaces,  
 Atmos. Environ. 21, 91-101, 1987.  
 Brutsaert, W., Evaporation into the Atmosphere, Reidel, 1982.  
 Businger, J.A., et al., Flux-profile relationships in the atmospheric  
 surface layer, J. Atmos. Sci., 28, 181-189, 1971.  
 Dwight, H.B., Tables of integrals and other mathematical data,  
 MacMillan, 1957.  
 Guenther, A., and 15 others, A global model of natural volatile  
 organic compound emissions, J. Geophys. Res., 100, 8873-8892, 1995.  
 Hicks, B.B., and P.S. Liss, Transfer of SO<sub>2</sub> and other reactive  
 gases across the air-sea interface, Tellus, 28, 348-354, 1976.  
 Jacob, D.J., and S.C. Wofsy, Budgets of reactive nitrogen,  
 hydrocarbons, and ozone over the Amazon forest during the wet season,  
 J. Geophys. Res., 95, 16737-16754, 1990.

- Jacob, D.J., and 9 others, Deposition of ozone to tundra, J. Geophys. Res., 97, 16473-16479, 1992.
- Levine, I.N., Physical Chemistry, 3rd ed., McGraw-Hill, New York, 1988.
- Munger, J.W., and 8 others, Atmospheric deposition of reactive nitrogen oxides and ozone in a temperate deciduous forest and a sub-arctic woodland, J. Geophys. Res., in press, 1996.
- Walcek, C.J., R.A. Brost, J.S. Chang, and M.L. Wesely, SO<sub>2</sub>, sulfate, and HNO<sub>3</sub> deposition velocities computed using regional landuse and meteorological data, Atmos. Environ., 20, 949-964, 1986.
- Wang, Y.H., paper in preparation, 1996.
- Wesely, M.L, Improved parameterizations for surface resistance to gaseous dry deposition in regional-scale numerical models, Environmental Protection Agency Report EPA/600/3-88/025, Research Triangle Park (NC), 1988.
- Wesely, M.L., same title, Atmos. Environ., 23, 1293-1304, 1989.

**REVISION HISTORY:**

- \*\* Contact: D.J. Jacob, Harvard U. (djj@io.harvard.edu)
- \*\* Modularized by G.M. Gardner, Harvard U.
- \*\* Version 3.2: 5/27/97
- \*\* Version 3.2.1: 3/4/99 -- bug fix in expression for RT
- \*\* Version 3.2.2: 3/26/99 -- bug fix: specify a large Ra for aerosols
- \*\* Version 3.2.3: 11/12/99 -- change Reynolds # criterion from 10 to 1  
-- force double precision w/ "D" exponents
- \*\* Version 3.3: 5/8/00 -- bug fixes, cleanup, updated comments.
- \*\* Version 3.4: 1/22/03 -- remove hardwire for CANOPYNOX
- \*\* Version 3.5 7/21/03 -- Remove cap of surface resistance in RLXXX
- \*\* Version 3.6 4/01/04 -- Now do drydep of DUST aerosol tracers
- \*\* Version 3.7 4/20/04 -- Now also do drydep of SEASALT aerosol tracers
- \*\* Version 3.8 4/13/05 -- Accounts for hygroscopic growth of SEASALT  
aerosol tracers. DUST aerosol tracers do  
not grow hygroscopically. Added RHB as  
an input argument.
- \*\* Version 3.9 5/25/05 -- Now restore GISS-specific code for GCAP model
- \*\* Version 3.9.1 11/17/05 -- change Reynolds # criterion from 1 to 0.1
- 11 May 2011 - L. Jaegle - Updated to use actual Sea level pressure instead  
of 1000 hPa  
- Modified to used Slinn & Slinn (1980) over Ocean  
surfaces
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 10 Jan 2012 - M. Payer - Updated to use local surface pressure
- 09 Apr 2012 - R. Yantosca - Remove IJREG, IJLAND, IJUSE, XYLAI arrays and  
replace w/ IREG, ILAND, IUSE, XLAI
- 09 Apr 2012 - R. Yantosca - Remove reference to CMN\_VEL\_mod.F
- 09 Apr 2012 - R. Yantosca - Now use INTENT(IN), INTENT(OUT) for arguments
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when  
running with the traditional driver main.F
- 12 Dec 2012 - R. Yantosca - Now get ILAND, IUSE, IREG from State\_Met

13 Dec 2012 - R. Yantosca - Now get XLAI from State\_Met  
 06 Mar 2013 - H. Amos - Merge C. Friedman's PAH code  
 31 May 2013 - R. Yantosca - Now pass State\_Chm, for TOMAS  
 14 Jun 2013 - R. Yantosca - Now use Input\_Opt%ITS\_A\_POPS\_SIM  
 29 Aug 2013 - R. Yantosca - Bug fix: Skip to the next species if unless  
                   HSTAR>0 and XMW>0, or AIROSOL=t. This avoids  
                   a floating-point invalid condition.  
 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust  
                   aerosols (T.D. Fairlie)  
 28 Jan 2014 - R. Yantosca - For TOMAS, don't hold A\_RADI and A\_DEN PRIVATE  
 19 May 2014 - C. Keller - Now call BIOFIT from drydep\_toolbox\_mod.F\  
 12 Aug 2015 - E. Lundgren - Now accept am\_I\_Root and RC as input args  
 12 Aug 2015 - E. Lundgren - Now pass am\_I\_Root and RC to AERO\_DIADEN  
                   to enable unit conversion in that routine  
 22 Sep 2015 - R. Yantosca - Now use NUMDEP instead of MAXDEP for arrays  
 15 Mar 2016 - C. Keller - Prevent very small numbers of CORR1 and ZOOBK  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 17 May 2016 - M. Sulprizio- Remove IJLOOP and change dimension of arrays  
                   from (MAXIJ) to (IIPAR,JJPARG); Also remove NPTS  
                   input argument

```

Changes from Version 3.2 to Version 3.3: ***
* We now suppress dry deposition over aerodynamically smooth ***
 surfaces. The previous algorithm yielded negative numbers ***
 when u* was very small (due to the logarithm going negative). ***
 See the comments below for more information. ***
* Now eliminate obsolete variables ZLMO and SIH from the code. ***
* Obsolete comments have been updated or removed. ***

Changes from version 3.1 to version 3.2: ***
* In unstable atmospheres with |ZLMO| < ZO, as can happen ***
 occasionally under very low wind conditions with tall canopies, ***
 application of Monin-Obukhov similarity yields negative values ***
 for RA. This was a problem in version 3.1. In fact, ***
 Monin-Obukhov similarity does not apply under such conditions, ***
 so we now set RA to zero and let the boundary ***
 resistance RB define the overall aerodynamic resistance. Since ***
 RB varies inversely with U* it will impose a large aerodynamic ***
 resistance under very low wind conditions. ***
* The range of applicability of stability correction functions ***
 to Monin-Obukhov similarity has been extended to ***
 -2.5 < z/zMO < 1.5, based on Figure 2 of Businger et al. [1971]. ***
 The range used to be -1 < z/zMO < 1 in version 3.1. ***

```

---

### 6.1.4 diffg

Subroutine DIFFG calculates the molecular diffusivity [m<sup>2</sup>/s] in air for a gas X of molecular weight XM [kg] at temperature TK [K] and pressure PRESS [Pa]. (bmy, 5/16/06)

#### INTERFACE:

```
FUNCTION DIFFG(TK, PRESS, XM) RESULT(DIFF_G)
```

#### INPUT PARAMETERS:

```
REAL(f8), INTENT(IN) :: TK ! Temperature [K]
REAL(f8), INTENT(IN) :: PRESS ! Pressure [Pa]
REAL(f8), INTENT(IN) :: XM ! Molecular weight of gas [kg]
```

#### REMARKS:

We specify the molecular weight of air (XMAIR) and the hard-sphere molecular radii of air (RADAIR) and of the diffusing gas (RADX). The molecular radius of air is given in a Table on p. 479 of Levine [1988]. The Table also gives radii for some other molecules. Rather than requesting the user to supply a molecular radius we specify here a generic value of 2.E-10 m for all molecules, which is good enough in terms of calculating the diffusivity as long as molecule is not too big.

#### REVISION HISTORY:

```
(1) Originally was a standalone function; now bundled into drydep_mod.f.
 Also now force REAL(f8) precision with D exponents. Now use F90
 style syntax and updated comments. (bmy, 5/16/06)
22 Dec 2011 - M. Payer - Added ProTeX headers
06 Jan 2016 - E. Lundgren - Now use global physical parameters
```

### 6.1.5 read\_drydep\_inputs

Subroutine READ\_DRYDEP\_INPUTS reads inputs for the dry deposition module corresponding to either the Olson 1992 (GEOS-Chem default) or Olson 2001 (planned replacement for Olson 1992) land map.

#### INTERFACE:

```
SUBROUTINE READ_DRYDEP_INPUTS(am_I_Root, Input_Opt,
& DRYCOEFF, IOLSON, IDEP,
& IWATER, NWATER, IZO,
& IDRYDEP, IRI, IRLU,
& IRAC, IRGSS, IRGSO,
& IRCLS, IRCLO, IVSMAX,
& RC)
```

#### USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

```
! Modules for netCDF read
USE m_netcdf_io_open
USE m_netcdf_io_get_dimlen
USE m_netcdf_io_read
USE m_netcdf_io_readattr
USE m_netcdf_io_close
```

```
include "netcdf.inc"
```

### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

### OUTPUT PARAMETERS:

```
!-----
! DRYCOEFF : Baldocchi polynomial coeffs
! IOLSON : Olson land type indices (+1)
! IDEP : Mapping: Olson ==> drydep ID
! IWATER : Olson types that represent water
! NWATER : Number of Olson types that are water
! IZO : Default ZO (routgness height) for each Olson land type
! IDRYDEP : Dry deposition land type indices
! IRI : RI resistance for drydep
! IRLU : RLU resistance for drydep
! IRAC : RAC resistance for drydep
! IRGSS : RGSS resistance for drydep
! IRGSO : RGSO resistance for drydep
! IRCLS : RCLS resistance for drydep
! IRCLO : RCLO resistance for drydep
! IVSMAX : Max drydep velocity (for aerosol) perr drydep land type
!-----
```

```
REAL(fp), INTENT(OUT) :: DRYCOEFF(NPOLY)
INTEGER, INTENT(OUT) :: IOLSON (NSURFTYPE)
INTEGER, INTENT(OUT) :: IDEP (NSURFTYPE)
INTEGER, INTENT(OUT) :: IWATER (NSURFTYPE)
INTEGER, INTENT(OUT) :: NWATER
INTEGER, INTENT(OUT) :: IZO (NSURFTYPE)
INTEGER, INTENT(OUT) :: IDRYDEP (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRI (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRLU (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRAC (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRGSS (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRGSO (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IRCLS (NDRYDTYPE)
```

```

INTEGER, INTENT(OUT) :: IRCLO (NDRYDTYPE)
INTEGER, INTENT(OUT) :: IVSMAX (NDRYDTYPE)

```

```

! Success or failure flag
INTEGER, INTENT(OUT) :: RC

```

**REMARKS:**

Routine READ\_DRYDEP\_INPUTS replaces routines MODIN (which read the ASCII file "drydep.table") and RDDRYCF (which read the ASCII file "drydep.coef").

READ\_DRYDEP\_INPUTS was generated from the Perl script "ncCodeRead", which is part of the NcdfUtilities package (with subsequent hand-editing).

Assumes that you have:

- (1) A netCDF library (either v3 or v4) installed on your system
- (2) The NcdfUtilities package (from Bob Yantosca) source code

**REVISION HISTORY:**

```

26 Mar 2012 - R. Yantosca - Initial version
03 Feb 2014 - M. Sulprizio- Change the internal resistance for coniferous
 forests to match the internal resistance for
 deciduous forests when using the Olson 2001
 land map (skim, 1/27/14)
18 Dec 2014 - R. Yantosca - Now read DRYCOEFF at REAL*8 precision and
 then cast to flexible precision
13 Mar 2015 - R. Yantosca - Replace DATA_DIR_1x1 w/ CHEM_INPUTS_DIR
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
 now the default.

```

**6.1.6 aero\_sfcrsii**

Function AERO\_SFCSII computes the aerodynamic resistance of seasalt aerosol species according to Zhang et al 2001. We account for hygroscopic growth of the seasalt aerosol particles. (rjp, tdf, bec, bmy, 4/1/04, 6/11/08)

**INTERFACE:**

```

FUNCTION AERO_SFCSII(K, II, PRESS, TEMP, USTAR, RHB,
& W10, Input_Opt) RESULT(RS)

```

**USES:**

```

USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: K ! Drydep species index (range: 1-NUMDEP)
INTEGER, INTENT(IN) :: II ! Surface type index of GEOS-CHEM

```



```

REAL(f8), INTENT(IN) :: PRESS ! Pressure [kPa] (1 mb=100 Pa=0.1 kPa)
REAL(f8), INTENT(IN) :: TEMP ! Temperature [K]
REAL(f8), INTENT(IN) :: USTAR ! Friction velocity [m/s]
REAL(f8), INTENT(IN) :: RHB ! Relative humidity (fraction)
REAL(f8), INTENT(IN) :: W10 ! 10 m windspeed [m/s]
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**RETURN VALUE:**

```

REAL(f8) :: RS ! Surface resistance for particles [s/m]

```

**REMARKS:**

Do computations internally with REAL\*8 (8-byte) floating-point precision, in order to avoid a loss of precision.

**REVISION HISTORY:**

- (1 ) Updated comments. Also now force double precision w/ "D" exponents.  
(bmy, 4/1/04)
  - (2 ) Now limit relative humidity to [tiny(real(f8)),0.99] range for DLOG  
argument (phs, 6/11/08)
  - (3 ) Bug fixes to the Gerber (1985) growth function (jaegle 5/11/11)
  - (4 ) Update growth function to Lewis and Schwartz (2006) and density  
calculation based on Tang et al. (1997) (bec, jaegle 5/11/11)
  - (5 ) Updates of sea salt deposition over water to follow the Slinn & Slinn  
(1980) formulation over water surface. Described in Jaegle et al. (ACP,  
11, 2011) (jaegle 5/11/11)
- 22 Dec 2011 - M. Payer - Added ProTeX headers  
14 Jun 2013 - R. Yantosca - Now pass Input\_Opt via the arg list  
06 Jan 2015 - E. Lundgren - Use global physical parameters

**6.1.7 init\_weightss**

Subroutine INIT\_WEIGHTSS calculates the volume size distribution of sea-salt. This only has to be done once. We assume that sea-salt is the combination of a coarse mode and accumulation model log-normal distribution functions. The resulting arrays are: DMID = diameter of bin and SALT\_V =  $dV/d\ln(D)$  [in  $\mu\text{m}^3$ ]. (jaegle 5/11/11)

**INTERFACE:**

```

SUBROUTINE INIT_WEIGHTSS(Input_Opt)

```

**USES:**

```

USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

TYPE(OptInput), INTENT(IN) :: Input_Opt

```

**REVISION HISTORY:**

11 May 2011 - L. Jaegle - Initial version  
 22 Dec 2011 - M. Payer - Added ProTeX headers  
 14 Jun 2013 - R. Yantosca - Now accept Input\_Opt via the argument list  
 06 Jan 2016 - E. Lundgren - Use global physical parameters

---

**6.1.8 dust\_sfcrsi**

Function DUST\_SFCRSI computes the aerodynamic resistance of dust aerosol species according to Seinfeld et al 96. We do not consider hygroscopic growth of the dust aerosol particles. (rjp, tdf, bmy, bec, 4/1/04, 4/15/05)

**INTERFACE:**

```
FUNCTION DUST_SFCRSI(K, II, PRESS, TEMP, USTAR) RESULT(RS)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: K ! Drydep species (range: 1-NUMDEP)
INTEGER, INTENT(IN) :: II ! Surface type index of GEOS-CHEM
REAL(f8), INTENT(IN) :: PRESS ! Pressure [kPa]
REAL(f8), INTENT(IN) :: TEMP ! Temperature [K]
REAL(f8), INTENT(IN) :: USTAR ! Friction velocity [m/s]
```

**RETURN VALUE:**

```
REAL(f8) :: RS ! Surface resistance for particles [s/m]
```

**REVISION HISTORY:**

(1) Updated comments. Also now force double precision w/ "D" exponents.  
 (bmy, 4/1/04)  
 (2) Renamed to DUST\_SFCRSII, since this will only be used to compute  
 aerodynamic resistance of dust aerosols. (bec, bmy, 4/15/05)  
 22 Dec 2011 - M. Payer - Added ProTeX headers  
 06 Jan 2016 - E. Lundgren - Use global physical parameters

---

**6.1.9 adust\_sfcrsii**

Function ADUST\_SFCRSII computes the aerodynamic resistance of non-size resolved aerosol according to Zhang et al 2001. We do not consider the hygroscopic growth of the aerosol particles. (rjp, tdf, bec, bmy, 4/1/04, 4/15/05)

This routine is used for all aerosols except dust, sulfate, and seasalt (hotp 7/31/09)

**INTERFACE:**

```
FUNCTION ADUST_SFCSII(K, II, PRESS, TEMP, USTAR) RESULT(RS)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: K ! Drydep species index (range: 1-NUMDEP)
INTEGER, INTENT(IN) :: II ! Surface type index of GEOS-CHEM
REAL(f8), INTENT(IN) :: PRESS ! Pressure [kPa] (1 mb = 100 Pa = 0.1 kPa)
REAL(f8), INTENT(IN) :: TEMP ! Temperature [K]
REAL(f8), INTENT(IN) :: USTAR ! Friction velocity [m/s]
```

**RETURN VALUE:**

```
REAL(f8) :: RS ! Surface resistance for particles [s/m]
```

**REVISION HISTORY:**

- (1 ) Updated comments. Also now force double precision w/ "D" exponents.  
(bmy, 4/1/04)
- (2 ) Renamed to DUST\_SFCSII, since this will only be used to compute  
aerodynamic resistance of dust aerosols. (bec, bmy, 4/15/05)
- (3 ) Modified hotp for non size resolved aerosols. This is just DUST\_SFCSII  
renamed and the diameter and density fixed. (hotp 7/12/07)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 06 Jan 2016 - E. Lundgren - Use global physical parameters

**6.1.10 dust\_sfcsii**

Function DUST\_SFCSII computes the aerodynamic resistance of dust aerosol species according to Zhang et al 2001. We do not consider the hygroscopic growth of the aerosol particles. (rjp, tdf, bec, bmy, 4/1/04, 4/15/05)

**INTERFACE:**

```
FUNCTION DUST_SFCSII(K, II, PRESS, TEMP, USTAR, DIAM, DEN)
& RESULT(RS)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: K ! Drydep species index (range: 1-NUMDEP)
INTEGER, INTENT(IN) :: II ! Surface type index of GEOS-CHEM
REAL(f8), INTENT(IN) :: PRESS ! Pressure [kPa]
REAL(f8), INTENT(IN) :: TEMP ! Temperature [K]
REAL(f8), INTENT(IN) :: USTAR ! Friction velocity [m/s]
REAL(f8), INTENT(IN) :: DIAM ! Particle diameter [m]
REAL(f8), INTENT(IN) :: DEN ! Particle density [kg/m3]
```

**RETURN VALUE:**

```
REAL(f8) :: RS ! Surface resistance for particles [s/m]
```

**REVISION HISTORY:**

- (1 ) Updated comments. Also now force double precision w/ "D" exponents.  
(bmy, 4/1/04)
- (2 ) Renamed to DUST\_SFCSII, since this will only be used to compute  
aerodynamic resistance of dust aerosols. (bec, bmy, 4/15/05)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 31 Jan 2014 - R. Yantosca - Now pass DIAM and DEN as arguments so as to  
avoid parallelization errors when using  
the TOMAS microphysics package.
- 06 Jan 2016 - E. Lundgren - Use global physical parameters
- 

**6.1.11 init\_drydep**

Subroutine INIT\_DRYDEP initializes certain variables for the GEOS-CHEM dry deposition subroutines. (bmy, 11/19/02, 10/19/09)

**INTERFACE:**

```

SUBROUTINE INIT_DRYDEP(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)

```

**USES:**

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?!

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure

```

**REMARKS:**

We now know how many drydep species there are before INIT\_DRYDEP is called. This allows us to get rid of MAXDEP. NUMDEP should be equal to State\_Chm%*n*DryDep, otherwise there is an error.

Also note: we need to use the actual molecular weights instead of the emitted molecular weights. These are necessary for the Schmidt # computation.

**REVISION HISTORY:**

- (1 ) Added N2O5 as a drydep tracer, w/ the same drydep velocity as HNO3. Now initialize PBLFRAC array. (rjp, bmy, 7/21/03)
- (2 ) Added extra carbon & dust aerosol tracers (rjp, tdf, bmy, 4/1/04)
- (3 ) Added seasalt aerosol tracers. Now use A\_RADI and A\_DEN to store radius & density of size-resolved tracers. Also added fancy output. (bec, rjp, bmy, 4/26/04)
- (3 ) Now handles extra SOA tracers (rjp, bmy, 7/13/04)
- (4 ) Now references LDRYD from "logical\_mod.f" and N\_TRACERS, SALA\_REEDGE\_um, and SALC\_REEDGE\_um from "tracer\_mod.f" (bmy, 7/20/04)
- (5 ) Included Hg2, HgP tracers (eck, bmy, 12/14/04)
- (6 ) Included AS, AHS, LET, NH4aq, SO4aq tracers (cas, bmy, 1/6/05)
- (7 ) Remove reference to PBLFRAC array -- it's obsolete (bmy, 2/22/05)
- (8 ) Included SO4s, NITs tracers (bec, bmy, 4/13/05)
- (9 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (10) Now set Henry's law constant to 1.0d+14 for Hg2. Now use ID\_Hg2, ID\_HgP, and ID\_Hg\_tot from "tracerid\_mod.f". Bug fix: split up compound IF statements into separate 2 IF statements for ID\_Hg2, ID\_HgP to avoid seg faults. (eck, cdh, bmy, 4/17/06)
- (11) Now also initialize SOG4, SOA4 drydep species. Bug fix: Remove 2nd "IF ( IS\_Hg ) THEN" statement. (dkh, bmy, 5/24/06)
- (12) Bug fix: fix TYPO in IF block for IDTSOA4 (dkh, bmy, 6/23/06)
- (13) Included H2/HD tracers for offline H2-HD sim (phs, 9/18/07)
- (14) Add dicarbonyl chemistry species (tmf, ccc, 3/6/09)
- (15) Minor bug fix: ALPH, LIMO should have molwt = 136.23, not 136 even (bmy, 10/19/09)
- (16) Add TOMAS aerosol NK1-NK30 and H2SO4 to drydep list (win, 7/14/09)
- 15 Dec 2011 - M. Payer - Update OVOC drydep according to Karl et al. 2010 and add drydep for MVK and MACR. (J. Mao)
- 21 Dec 2011 - M. Payer - Add allocation for size distribution of sea salt SALT\_V and DMID (jaegle, 5/11/11)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 14 Mar 2013 - M. Payer - Replace NOx and Ox with NO2 and O3 as part of removal of NOx-Ox partitioning
- 12 Jun 2013 - R. Yantosca - Bug fix: now only copy NUMDEP values to Input\_Opt%NDVZIND and Input\_Opt%DEPNAME
- 14 Jun 2013 - R. Yantosca - Now replace fields from tracer\_mod.F with fields from Input\_Opt
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 29 Aug 2013 - R. Yantosca - Assign XMW=118d-3 to RIP and IEPOX. This now prevents XMW=0e+0\_f8 from being passed to function DIFFG, where it is in the denominator.
- 04 Sep 2013 - R. Yantosca - Improve printout of drydep species
- 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust

```

 aerosols (T.D. Fairlie)
15 Jan 2015 - R. Yantosca - Now save NTRAINND to Input_Opt%NTRAINND
08 Jul 2015 - E. Lundgren - Add marine organic aerosols (B.Gantt, M.Johnson)
22 Sep 2015 - R. Yantosca - Now allocate internal variables that used
 to be of size MAXDEP.
30 Sep 2015 - R. Yantosca - DD_A_Density is renamed to Density
30 Sep 2015 - R. Yantosca - DD_A_Radius is renamed to Radius
14 Dec 2015 - R. Yantosca - Now use actual MW (MW_g) instead of emitted MW
 (EmMW_g) to define the XMW array
20 Jun 2016 - R. Yantosca - Now define a couple of species ID flags here
 as module variables on the first call
18 Jul 2016 - M. Sulprizio- Remove special handling for ISOPN and MVKN.
 Family tracers have been eliminated.
13 Jul 2017 - M. Sulprizio- Add scaling of species drydep relative to HNO3,
 PAN, and ISOPN (from K. Travis, J. Fisher)
05 Oct 2017 - R. Yantosca - Now accept State_Diag as an argument
07 Nov 2017 - R. Yantosca - Now return error condition to calling program

```

### 6.1.12 cleanup\_drydep

Subroutine CLEANUP\_DRYDEP deallocates all module arrays. (bmy, 2/27/03, 2/22/05)

#### INTERFACE:

```
SUBROUTINE CLEANUP_DRYDEP
```

#### REVISION HISTORY:

```

(1) Remove reference to PBLFRAC array; it's obsolete (bmy, 2/22/05)
(2) Added SALT_V and DMID (jaegle, 5/11/11)
22 Dec 2011 - M. Payer - Added ProTeX headers
22 Sep 2015 - R. Yantosca - Now deallocate arrays that were formerly
 sized to MAXDEP.
05 Jul 2018 - H. Lin - Bug fix: deallocate FLAG

```

## 6.2 Fortran: Module Interface modis\_lai\_mod.F90

Module MODIS\_LAI\_MOD reads the MODIS LAI and CHLR data at native resolution and then regrids them to the GEOS-Chem resolution on the fly. CHLR data is only read if marine organic aerosol tracers are enabled. **INTERFACE:**

```
MODULE Modis_Lai_Mod
```

#### USES:

```
USE CMN_SIZE_Mod ! Size parameters
```

```
USE Error_Mod ! Error checking routines
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
USE Mapping_Mod ! Mapping weights & areas
USE Time_Mod ! EXPAND_DATE
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Init_Modis_Lai
PUBLIC :: Read_Modis_Lai
PUBLIC :: Compute_Modis_Lai
PUBLIC :: Find_Lai_Month
PUBLIC :: Cleanup_Modis_Lai
```

#### PRIVATE MEMBER FUNCTIONS:

```
#if defined(ESMF_) || defined(EXTERNAL_GRID) || defined(EXTERNAL_FORCING)
 PUBLIC :: Compute_XLAI_GCHP
#endif
PUBLIC :: Read_Modis
PUBLIC :: Compute_Modis
PRIVATE :: RoundOff
```

#### REMARKS:

- (1) MODIS LAI data resolution is the same as the Olson land map. The Olson 2001 landmap is the default, therefore, you will use MODIS LAI data at 0.25 x 0.25 resolution.  
Note that MODIS CHLR data is only available at 0.25 x 0.25 resolution.
- (2) In HEMCO, MEGAN uses 'offline' MODIS LAI (State\_Met%MODISLAI) which is computed in this module.  
in an ESMF environment, in which case State\_Met%LAI is used instead.
- (3) MODIS LAI arrays and where they are used in GEOS-Chem:
  - (a) State\_Met%XLAI --> dry deposition routine DEPVEL
  - (b) State\_Met%MODISLAI --> MEGAN (if using standard GC); Hg(0) emissions; several diagnostics

#### REVISION HISTORY:

- 03 Apr 2012 - R. Yantosca - Initial version
- 05 Apr 2012 - R. Yantosca - Added descriptive comments
- 09 Apr 2012 - R. Yantosca - Fixed error in ROUNDOFF function that caused numbers to be rounded up incorrectly.
- 09 Apr 2012 - R. Yantosca - Changed variables to REAL(fp)
- 09 Apr 2012 - R. Yantosca - Now set MODIS\_START and MODIS\_END depending on which version of MODIS LAI we are using
- 13 Dec 2012 - R. Yantosca - Remove reference to obsolete CMN\_DEP\_mod.F; XLAI, XLAI2 now are carried in State\_Met
- 23 Jun 2014 - R. Yantosca - Removed references to logical\_mod.F

09 Oct 2014 - C. Keller - Removed GC\_LAI\_PM, GC\_LAI\_CM, GC\_LAI\_NM and MODIS\_LAI\_PM.  
 17 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 07 Jul 2015 - E. Lundgren - Now also read and compute MODIS chlorophyll-a (B. Gantt, M. Johnson). Use separate end years.  
 18 Oct 2016 - E. Lundgren - Move module vars GC\_LAI and GC\_CHLR to State\_Met; rename as MODISLAI and MODISCHLR; make State\_Met vars XLAI2 and XCHLR2 module vars

---

### 6.2.1 compute\_modis\_lai

Subroutine COMPUTE\_MODIS\_LAI is the wrapper routine to compute the daily MODIS leaf area indices for GEOS-Chem directly from the native grid resolution (0.25 x 0.25 or 0.5 x 0.5). If marine organic aerosol tracers are used, then daily MODIS chlorophyll is also computed. This routine is not used in High Performance GEOS-Chem (GCHP).

#### INTERFACE:

```

SUBROUTINE Compute_Modis_Lai(am_I_Root, Input_Opt, State_Met, &
 doy, mm, mapping, &
 wasModisRead, RC

```

#### USES:

```

USE State_Met_Mod, ONLY : MetState
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
INTEGER, INTENT(IN) :: doy ! Day of year
INTEGER, INTENT(IN) :: mm ! Month for LAI data
TYPE(MapWeight), POINTER :: mapping(:, :) ! "fine" -> "coarse" grid map
LOGICAL, INTENT(IN) :: wasModisRead ! Was LAI data just read in?

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REVISION HISTORY:

03 Apr 2012 - R. Yantosca - Initial version  
 05 Apr 2012 - R. Yantosca - Renamed arg "doMonthly" to "wasModisRead"  
 09 Apr 2012 - R. Yantosca - Changed variables to REAL(fp)  
 09 Apr 2012 - R. Yantosca - Now follows same algorithm as rdlai.F for



populating XLAI array

09 Apr 2012 - R. Yantosca - Remove refs to CMN\_VEL\_mod.F and XYLAI array;  
these are now obsolete

17 Apr 2012 - R. Yantosca - Now rename "map" object to "mapping" to avoid  
name confusion w/ an F90 intrinsic function

13 Dec 2012 - R. Yantosca - Add am\_I\_Root, State\_Met, RC arguments

13 Dec 2012 - R. Yantosca - XLAI, XLAI2 are now carried in State\_Met  
instead of in obsolete Headers/CMN\_DEP\_mod.F

23 Jun 2014 - R. Yantosca - Now accept Input\_Opt via the arg list

09 Oct 2014 - C. Keller - Removed GC\_LAI\_PM, GC\_LAI\_CM, GC\_LAI\_NM and  
MODIS\_LAI\_PM.

08 Jul 2015 - E. Lundgren - Now compute LAI and CHLR data. Abstracted  
compute code to new routine Compute\_MODIS

### 6.2.2 compute\_xlai\_gchp

Subroutine COMPUTE\_XLAI\_GCHP computes MODIS-based leaf area indices (LAI) per land type and grid cell. This computation uses offline 0.25x0.25 MODIS LAI/CHLR and Olson landmap data regridded to the cubed sphere. Variables set include State\_Met

#### INTERFACE:

```
SUBROUTINE Compute_XLAI_GCHP(am_I_Root, State_Met, RC)
```

#### USES:

```
USE ErrCode_Mod
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REMARKS:

#### REVISION HISTORY:

18 Oct 2016 - E. Lundgren - Initial version

### 6.2.3 compute\_modis

Subroutine COMPUTE\_MODIS computes either the daily MODIS leaf area indices or the daily chlorophyll for GEOS-Chem directly from the native grid resolution (0.25 x 0.25 or 0.5 x 0.5). The XLAI array (used in the legacy soil NOx and dry deposition routines) are populated accordingly. The XYLAI array is now obsolete and has been replaced by XLAI.

#### INTERFACE:

```

SUBROUTINE Compute_Modis(am_I_Root, ComputeLAI, &
 Input_Opt, State_Met, doy, &
 mm, mapping, wasModisRead, RC)

```

#### USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
LOGICAL, INTENT(IN) :: ComputeLAI
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
INTEGER, INTENT(IN) :: doy ! Day of year
INTEGER, INTENT(IN) :: mm ! Month for LAI data
TYPE(MapWeight), POINTER :: mapping(:, :) ! "fine" -> "coarse" grid map
LOGICAL, INTENT(IN) :: wasModisRead ! Was LAI data just read in?

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REMARKS:

Uses same algorithm as RDISOLAI in the existing lai\_mod.F.

#### REVISION HISTORY:

```

07 Jul 2015 - E. Lundgren - Initial version, contains old Compute_Modis_Lai
 code plus modifications to compute CHLR
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

```

---

### 6.2.4 read\_modis\_lai

Subroutine READ\_MODIS\_LAI is the wrapper routine to read the MODIS LAI from disk in netCDF format for the current month, and for next month. If enabled, MODIS CHLR is also read in the same way as LAI.

#### INTERFACE:

```
SUBROUTINE Read_Modis_Lai(am_I_Root, Input_Opt, yyyy, mm, wasModisRead, RC)
```

#### USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
INTEGER, INTENT(IN) :: yyyy ! Year for LAI data
INTEGER, INTENT(IN) :: mm ! Month for LAI data
```

#### OUTPUT PARAMETERS:

```
LOGICAL, INTENT(OUT) :: wasModisRead ! Was LAI data just read in?
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REVISION HISTORY:

```
03 Apr 2012 - R. Yantosca - Initial version
05 Apr 2012 - R. Yantosca - Renamed arg "doMonthly" to "wasModisRead"
05 Jun 2013 - R. Yantosca - Bug fix, use "mm" for current month index
20 Jun 2014 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC
09 Oct 2014 - C. Keller - MODIS_LAI_PM not needed anymore.
05 Mar 2015 - R. Yantosca - Now read data w/r/t ExtData/CHEM_INPUTS
07 Jul 2015 - E. Lundgren - Generalized to read either LAI or CHLR
08 Jul 2015 - E. Lundgren - Now read LAI and CHLR data. Abstracted
 read code to new routine Read_Modis.
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
 now the default.
15 Mar 2018 - M. Sulprizio- Move nc_dir definition to here from Read_Modis.
 We now use updates MODIS LAI files stored in
 MODIS_LAI_201707/, while CHLR files are still
 read from MODIS_LAI_201204/.
```

### 6.2.5 read\_modis

Subroutine READ\_MODIS reads the MODIS LAI or CHLR from disk (in netCDF format) for the current month, and for next month.

#### INTERFACE:

```

SUBROUTINE Read_Modis(am_I_Root, ReadLAI, nc_tmpl, Input_Opt, &
 yyyy, mm, wasModisRead, RC)

```

**USES:**

```

USE m_netcdf_io_open ! netCDF file open
USE m_netcdf_io_read ! netCDF read
USE m_netcdf_io_readattr ! netCDF attribute reads
USE m_netcdf_io_close ! netCDF file close
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput

```

```

include "netcdf.inc" ! netCDF settings & parameters

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
LOGICAL, INTENT(IN) :: ReadLAI ! T for LAI, F to read CHLR
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
CHARACTER(LEN=255), INTENT(IN) :: nc_tmpl
INTEGER, INTENT(IN) :: yyyy ! Year for LAI data
INTEGER, INTENT(IN) :: mm ! Month for LAI data

```

**OUTPUT PARAMETERS:**

```

LOGICAL, INTENT(OUT) :: wasModisRead ! Was data just read in?
INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

07 Jul 2015 - E. Lundgren - Initial version, containing legacy Read_Modis_Lai
code plus modifications to read CHLR
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
now the default.

```

**6.2.6 find\_lai\_month**

Function FIND\_LAI\_MONTH returns the corresponding LAI month and year for the current calendar date. Note that the LAI data starts at mid-month.

**INTERFACE:**

```

SUBROUTINE Find_Lai_Month(doy, month, year, mm, yyyy)

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: doy ! Current day of year
INTEGER, INTENT(IN) :: month ! Current month
INTEGER, INTENT(IN) :: year ! Current year

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: mm ! Output month for LAI data
INTEGER, INTENT(OUT) :: yyyy ! Output year for LAI data

```

**REVISION HISTORY:**

```

05 Jan 1994 - Y. H. Wang, G.M. Gardner, D. Jacob - Initial version
(1) Updated comments, cosmetic changes (bmy, 4/4/03)
(2) Add the current simulation year as input & the current LAI as output.
 This is necessary for reading in MODIS LAI (mpb,2009).
08 Dec 2009 - R. Yantosca - Added ProTeX headers
03 Apr 2012 - R. Yantosca - Renamed to FIND_LAI_MONTH; made PUBLIC

```

**6.2.7 RoundOff**

Rounds a number X to N decimal places of precision.

**INTERFACE:**

```

FUNCTION RoundOff(X, N) RESULT(Y)

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: X ! Number to be rounded
INTEGER, INTENT(IN) :: N ! Number of decimal places to keep

```

**RETURN VALUE:**

```

REAL(fp) :: Y ! Number rounded to N decimal places

```

**REMARKS:**

The algorithm to round X to N decimal places is as follows:

- (1) Multiply X by 10\*\*(N+1)
- (2) If X < 0, then add -5 to X; otherwise add 5 to X
- (3) Take the integer part of X
- (4) Divide X by 10\*\*(N+1)
- (5) Truncate X to N decimal places: INT( X \* 10\*\*N ) / 10\*\*N

Rounding algorithm from: Hultquist, P.F, "Numerical Methods for Engineers and Computer Scientists", Benjamin/Cummings, Menlo Park CA, 1988, p. 20.

Truncation algorithm from: <http://en.wikipedia.org/wiki/Truncation>

The two algorithms have been merged together for efficiency.

**REVISION HISTORY:**

```

06 Apr 2012 - R. Yantosca - Initial version
09 Apr 2012 - R. Yantosca - Changed all variables & arguments to REAL(fp)

```

### 6.2.8 init\_modis

Subroutine INIT\_MODIS\_LAI initializes and allocates all module variables.

#### INTERFACE:

```
SUBROUTINE Init_Modis_Lai(am_I_Root, Input_Opt, RC)
```

#### USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC !Success or failure?
```

#### REVISION HISTORY:

```
03 Apr 2012 - R. Yantosca - Initial version
03 Feb 2014 - M. Sulprizio- Force last year of MODIS data to 2008. There is
 a large difference in the 2009 file that still
 needs to be investigated (skim, 1/29/14)
23 Jun 2014 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC
08 Jul 2015 - E. Lundgren - New end years to match files from M. Johnson
13 Sep 2017 - M. Sulprizio- Remove Input_Opt%USE_OLSON_2001. Olson 2001 is
 now the default.
15 Mar 2018 - M. Sulprizio- Update MODIS LAI end year to 2011.
```

### 6.2.9 cleanup\_modis\_lai

Subroutine CLEANUP\_MODIS\_LAI deallocates all previously-allocated module variables.

#### INTERFACE:

```
SUBROUTINE Cleanup_Modis_Lai
```

#### REVISION HISTORY:

```
03 Apr 2012 - R. Yantosca - Initial version
```

### 6.3 Fortran: Module Interface olson\_landmap\_mod.F90

Module OLSON\_LANDMAP\_MOD reads the Olson land map and computes the IREG, ILAND, IUUSE, and FRCLND State\_Met arrays.

#### INTERFACE:

```
MODULE Olson_LandMap_Mod
```

#### USES:

```
USE CMN_SIZE_MOD ! Size parameters
USE ERROR_MOD ! Error checking routines
USE GC_GRID_MOD ! Horizontal grid definition
USE MAPPING_MOD ! Mapping weights & areas
USE PhysConstants ! Physical constants
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

```
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
#if defined(ESMF_) || defined(EXTERNAL_GRID) || defined(EXTERNAL_FORCING)
PUBLIC :: Compute_Olson_Landmap_GCHP
#else
PUBLIC :: Init_Olson_Landmap
PUBLIC :: Compute_Olson_Landmap
PUBLIC :: Cleanup_Olson_LandMap
#endif
```

#### REMARKS:

Eloise Marais and the GEOS-Chem Support Team updated the Olson 2001 landcover dataset and corresponding GEOS-Chem modules in 2012. The Olson 2001 landmap superceded the Olson 1992 landmap starting following v9-01-03. The option to use Olson 1992 has been removed from v11-02 and later. The following text is taken from the data processing README:  
 "The Olson 2001 landcover map is at a native resolution of 1km x 1km. I've identified the dominant vegetation types in each 0.25x0.25 degree gridbox and use this as input to GEOS-Chem.

The Olson 2001 landcover map also has 96 vegetation types compared with 74 for Olson 1992. For the most part vegetation types 75-96 are either not dominant vegetation types at 0.25x0.25 degrees or they are crop types that I lump with other similar vegetation types (either crops or mixed forest/field vegetation) so that the Olson 2001 landcover dataset at 0.25x0.25 degrees has 74 vegetation types.

There are also new vegetation types that are defined in the Olson 2001 dataset from 1-74 that were previously listed as "not used" in

drydep.table. These are assigned appropriate deposition ID # and z0 values.

Vegetation types that are listed as "not used" in the updated drydep.table dataset are those that are not dominant at 0.25x0.25, but may be present in the 1kmx1km dataset."

The following table shows the the translation between the Olson 2001 and Olson 1992 land maps:

| Olson 2001<br>LC# Description  | Olson 1992 # in<br>Equivalent | Dry deposition |
|--------------------------------|-------------------------------|----------------|
| 1 Urban                        | 1 2                           |                |
| 2 Low Sparse Grassland         | 2 3                           |                |
| 3 Coniferous Forest            | 3 4                           |                |
| 4 Deciduous Conifer Forest     | 4 5                           |                |
| 5 Deciduous Broadleaf Forest   | 5 6                           |                |
| 6 Evergreen Broadleaf Forests  | 6 7                           |                |
| 7 Tall Grasses and Shrubs      | 7 8                           |                |
| 8 Bare Desert                  | 8 9                           |                |
| 9 Upland Tundra                | 9 10                          |                |
| 10 Irrigated Grassland         | 10 11                         |                |
| 11 Semi Desert                 | 11 12                         |                |
| 12 Glacier Ice                 | 12 13                         |                |
| 13 Wooded Wet Swamp            | 13 14                         |                |
| 14 Inland Water                | 0 1                           |                |
| 15 Sea Water                   | 0 1                           |                |
| 16 Shrub Evergreen             | 16 17                         |                |
| 17 Shrub Deciduous             | 18 19                         |                |
| 18 Mixed Forest and Field      | none present                  |                |
| 19 Evergreen Forest and Fields | 19 20                         |                |
| 20 Cool Rain Forest            | 20 21                         |                |
| 21 Conifer Boreal Forest       | 21 22                         |                |
| 22 Cool Conifer Forest         | 22 23                         |                |
| 23 Cool Mixed Forest           | 23 24                         |                |
| 24 Mixed Forest                | 24 25                         |                |
| 25 Cool Broadleaf Forest       | 25 26                         |                |
| 26 Deciduous Broadleaf Forest  | 26 27                         |                |
| 27 Conifer Forest              | 27 28                         |                |
| 28 Montane Tropical Forests    | 28 29                         |                |
| 29 Seasonal Tropical Forest    | 29 30                         |                |
| 30 Cool Crops and Towns        | 30 31                         |                |
| 31 Crops and Town              | 31 32                         |                |
| 32 Dry Tropical Woods          | 32 33                         |                |
| 33 Tropical Rainforest         | 33 34                         |                |
| 34 Tropical Degraded Forest    | 34 35                         |                |
| 35 Corn and Beans Cropland     | 35 36                         |                |
| 36 Rice Paddy and Field        | 36 37                         |                |
| 37 Hot Irrigated Cropland      | 37 38                         |                |



38 Cool Irrigated Cropland 38 39  
39 Cold Irrigated Cropland none present  
40 Cool Grasses and Shrubs 40 41  
41 Hot and Mild Grasses and Shrubs 41 42  
42 Cold Grassland 42 43  
43 Savanna (Woods) 43 44  
44 Mire, Bog, Fen 44 45  
45 Marsh Wetland 45 46  
46 Mediterranean Scrub 46 47  
47 Dry Woody Scrub 47 48  
48 Dry Evergreen Woods none present  
49 Volcanic Rock none present  
50 Sand Desert none present  
51 Semi Desert Shrubs 51 52  
52 Semi Desert Sage 52 53  
53 Barren Tundra 53 54  
54 Cool Southern Hemisphere Mixed Forests 54 55  
55 Cool Fields and Woods 55 56  
56 Forest and Field 56 57  
57 Cool Forest and Field 57 58  
58 Fields and Woody Savanna 58 59  
59 Succulent and Thorn Scrub 59 60  
60 Small Leaf Mixed Woods 60 61  
61 Deciduous and Mixed Boreal Forest 61 62  
62 Narrow Conifers 62 63  
63 Wooded Tundra 63 64  
64 Heath Scrub 64 65  
65 Coastal Wetland, NW none present  
66 Coastal Wetland, NE none present  
67 Coastal Wetland, SE none present  
68 Coastal Wetland, SW none present  
69 Polar and Alpine Desert 69 70  
70 Glacier Rock none present  
71 Salt Playas none present  
72 Mangrove 72 73  
73 Water and Island Fringe none present  
74 Land, Water, and Shore (see Note 1) none present  
75 Land and Water, Rivers (see Note 1) none present  
76 Crop and Water Mixtures 36 37  
77 Southern Hemisphere Conifers none present  
78 Southern Hemisphere Mixed Forest 32 33  
79 Wet Sclerophytic Forest 26 27  
80 Coastline Fringe none present  
81 Beaches and Dunes none present  
82 Sparse Dunes and Ridges none present  
83 Bare Coastal Dunes none present  
84 Residual Dunes and Beaches none present  
85 Compound Coastlines none present



02 Apr 2012 - R. Yantosca - Moved routine GET\_MAP\_WT to mapping\_mod.F90  
 02 Apr 2012 - R. Yantosca - Now Save mapping info for later use  
 09 Apr 2012 - R. Yantosca - Removed IJREG, IJUSE, IJLAND; these are now  
                   replaced by IREG, IUSE, ILAND arrays  
 09 Apr 2012 - R. Yantosca - Removed reference to CMN\_VEL\_mod.F  
 20 Mar 2014 - R. Yantosca - Speed up Olson computation by skipping boxes  
 24 Jun 2014 - R. Yantosca - Remove references to logical\_mod.F  
 17 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 18 Oct 2016 - E. Lundgren - Add GCHP routine for computing landmap variables  
 02 Nov 2016 - E. Lundgren - Remove N\_OLSON since same as global NSURFTYPE  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90  
 13 Sep 2017 - M. Sulprizio- Remove Input\_Opt%USE\_OLSON\_2001. Olson 2001 is  
                   now the default.

---

### 6.3.1 compute\_olson\_landmap\_gchp

Subroutine COMPUTE\_OLSON\_LANDMAP\_GCHP computes the GEOS-Chem State.Met variables that are dependent on the Olson Landmap, specifically IREG, ILAND, IUSE, and FRCLND.

#### INTERFACE:

```
SUBROUTINE Compute_Olson_Landmap_GCHP(am_I_Root, State_Met, RC)
```

#### USES:

```
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
INTEGER, INTENT(INOUT) :: RC
```

#### REMARKS:

#### REVISION HISTORY:

```
27 Sep 2016 - E. Lundgren - Initial version
```

---

### 6.3.2 init\_olson\_landmap

Subroutine INIT\_OLSON\_LANDMAP reads Olson land map information from disk (in netCDF format).

#### INTERFACE:

```
SUBROUTINE Init_Olson_LandMap(am_I_Root, Input_Opt, RC)
```

**USES:**

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE m_netcdf_io_open
USE m_netcdf_io_read
USE m_netcdf_io_readattr
USE m_netcdf_io_close
```

```
IMPLICIT NONE
```

```
include "netcdf.inc"
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:**

Assumes that you have:

- (1) A netCDF library (either v3 or v4) installed on your system
- (2) The NcdfUtilities package (from Bob Yantosca) source code

**REVISION HISTORY:**

```
13 Mar 2012 - R. Yantosca - Initial version
22 Mar 2012 - R. Yantosca - Also read in surface areas [m2] from file
27 Mar 2012 - R. Yantosca - Now read the "units" attribute of each variable
27 Mar 2012 - R. Yantosca - Now echo file I/O status info to stdout
27 Mar 2012 - R. Yantosca - Now can read Olson 1992 or Olson 2001 land map
29 Nov 2012 - R. Yantosca - Add am_I_Root to the argument list
26 Feb 2013 - M. Long - Now pass DATA_DIR_1x1 via the argument list
24 Jun 2014 - R. Yantosca - Now accept Input_Opt, RC via the arg list
05 Mar 2015 - R. Yantosca - Now read data w/r/t ExtData/CHEM_INPUTS
```

**6.3.3 compute\_olson\_landmap**

Subroutine COMPUTE\_OLSON\_LANDMAP computes the GEOS-Chem arrays IREG, ILAND, IUSE (and corresponding 1-D arrays IJREG, IJLAND, IJUSE) on-the-fly from the Olson Land map file. This routine, which is intended to facilitate the Grid-Independent GEOS-Chem, replaces the old rdland.F, which read from pre-computed "vegtype.global" files.

**INTERFACE:**

```
SUBROUTINE Compute_Olson_LandMap(am_I_Root, mapping, State_Met)
```

**USES:**

```
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(MapWeight), POINTER :: mapping(:,:) ! "fine" -> "coarse" mapping
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

**REMARKS:**

This routine supplies arrays that are required for legacy code routines:

- (1) IREG, ILAND, IUSE are used by the Soil NOx routines
- (2) IJREG, IJLAND, IJUSE are used by the dry deposition routines

**REVISION HISTORY:**

```
13 Mar 2012 - R. Yantosca - Initial version
19 Mar 2012 - R. Yantosca - Reorder ILAND, IUSE, IJLAND, IJUSE to be
 consistent w/ the leaf area indices
19 Mar 2012 - R. Yantosca - Compute the FRCLND array (from CMN_DEP_mod.F)
21 Mar 2012 - R. Yantosca - Now use REAL*4 for computation, to reduce
 roundoff errors at high-resolution
22 Mar 2012 - R. Yantosca - Now get surface area directly from variable
 A_CM2 (read from disk) instead of computing it
02 Apr 2012 - R. Yantosca - Now pass MAP (mapping weight object) via the
 arg list, to save the mapping info for later
09 Apr 2012 - R. Yantosca - Remove IJLOOP variable
09 Apr 2012 - R. Yantosca - Now do not compute IJREG, IJLAND, IJUSE; these
 are replaced by IREG, ILAND, IUSE arrays
17 Apr 2012 - R. Yantosca - Rename "map" object to "mapping" to avoid name
 confusion with an F90 intrinsic function
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
29 Nov 2012 - R. Yantosca - Added am_I_Root argument
12 Dec 2012 - R. Yantosca - Now get IREG, ILAND, IUSE from State_Met
20 Mar 2014 - R. Yantosca - Add shunts in lat & lon to reduce wall time
```

**6.3.4 cleanup\_olson\_landmap**

Subroutine CLEANUP\_OLSON\_LANDMAP deallocates all allocated global module variables.

**INTERFACE:**

```
SUBROUTINE Cleanup_Olson_LandMap(am_I_Root)
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

#### REVISION HISTORY:

```
22 Mar 2012 - R. Yantosca - Initial version
29 Nov 2012 - R. Yantosca - Add am_I_Root as an argument
```

### 6.4 Fortran: Module Interface mapping\_mod.F90

Module MAPPING\_MOD contains a derived-type object to compute and save the mapping weight (i.e. fraction of each "fine" grid box that fits into the "coarse" grid box") and areal mapping (i.e. the area of each "fine" grid box contained within a "coarse" grid box).

#### INTERFACE:

```
MODULE Mapping_Mod
```

#### USES:

```
USE CMN_SIZE_MOD ! Size parameters
USE ERROR_MOD ! Error handling routines
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC DATA MEMBERS:

```
PUBLIC :: MapWeight
TYPE MapWeight
 INTEGER :: count ! # of "fine" boxes per "coarse" box
 INTEGER, POINTER :: II(:) ! Longitude indices, "fine" grid
 INTEGER, POINTER :: JJ(:) ! Latitude indices, "fine" grid
 INTEGER, POINTER :: olson(:) ! Olson land type, "fine" grid
 INTEGER, POINTER :: ordOlson(:) ! Ordering of Olson land types
 REAL*4, POINTER :: area(:) ! Surface areas, "fine" grid
 REAL*4 :: sumarea ! Total surface area, "coarse" grid
END TYPE MapWeight
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Init_Mapping
PUBLIC :: Get_Map_Wt
PUBLIC :: Cleanup_Mapping
```

#### REMARKS:

The mapping weights and areal mapping are initialized when the Olson land map is read from disk (in olson\_landmap\_mod.F90). They are used again when the MODIS leaf area index data is prepared for input into GEOS-Chem's (legacy) dry deposition module.

Also, we do not define the mapping weight object within this module. This allows you to create more than one mapping weight object for different native grids (e.g. 0.5 x 0.5 and 0.25 x 0.25, etc.)

#### REVISION HISTORY:

03 Apr 2012 - R. Yantosca - Initial version  
 05 Apr 2012 - R. Yantosca - Comment out mapwt field of MapWeight type, leave this for future expansion  
 17 Apr 2012 - R. Yantosca - Rename pointer object "map" to "mapping, to remove confusion w/ F90 intrinsic  
 17 Nov 2014 - M. Yannetti - Added PRECISION\_MOD

#### 6.4.1 init\_mapping

Subroutine INIT\_MAPPING allocates and initializes a derived-type object containing grid mapping information.

#### INTERFACE:

```
SUBROUTINE Init_Mapping(am_I_Root, Input_Opt, I_FINE, J_FINE, &
 I_COARSE, J_COARSE, mapping, RC)
```

#### USES:

```
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
INTEGER, INTENT(IN) :: I_FINE ! # of lons on the "fine" grid
INTEGER, INTENT(IN) :: J_FINE ! # of lats on the,"fine" grid
INTEGER, INTENT(IN) :: I_COARSE ! # of lons on the "coarse" grid
INTEGER, INTENT(IN) :: J_COARSE ! # of lats on the "coarse" grid
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(MapWeight), POINTER, INTENT(INOUT) :: mapping(:, :) !"fine" -> "coarse"
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REVISION HISTORY:

03 Apr 2012 - R. Yantosca - Initial version  
 10 Apr 2012 - R. Yantosca - Now add a different # to FINE\_PER\_COARSE depending on which Olson map we are using  
 17 Apr 2012 - R. Yantosca - Rename to "map" to "mapping" to avoid confusion with a F90 intrinsic function  
 17 Apr 2012 - R. Yantosca - Add error check for mapping object  
 18 Apr 2012 - R. Yantosca - Improve error check for sub-fields of mapping object so as not to interfere w/ parallel loop  
 23 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC  
 13 Sep 2017 - M. Sulprizio- Remove Input\_Opt%USE\_OLSON\_2001. Olson 2001 is now the default.

### 6.4.2 get\_map\_wt

Subroutine GET\_MAP\_Wt returns the "mapping weight", that is, the fraction that each "fine" grid box fits into each "coarse" grid box.

#### INTERFACE:

```
SUBROUTINE Get_Map_Wt(xedge_w, xedge_e, xedgeC_w, xedgeC_e, &
 yedge_s, yedge_n, yedgeC_s, yedgeC_n, &
 mapWt
)
```

#### INPUT PARAMETERS:

```
REAL*4, INTENT(IN) :: xedge_w, xedge_e ! Lon edges, fine grid
REAL*4, INTENT(IN) :: xedgeC_w, xedgeC_e ! Lon edges, coarse grid
REAL*4, INTENT(IN) :: yedge_s, yedge_n ! Lat edges, fine grid
REAL*4, INTENT(IN) :: yedgeC_s, yedgeC_n ! Lat edges, coarse grid
REAL*4, INTENT(OUT) :: mapWt ! Mapping weight
```

#### REMARKS:

Follows the algorithm from GAMAP routine ctm\_getweight.pro

#### REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version  
 21 Mar 2012 - R. Yantosca - Typo: set xOverLap to zero if it is out of the range of 0-1. (We had set yOverLap=0 before)  
 21 Mar 2012 - R. Yantosca - Now use REAL\*4 for computations to avoid roundoff errors at hi-res grids  
 03 Apr 2012 - R. Yantosca - Moved from "olson\_landmap\_mod.F90" to here; renamed "Get\_Mapping"



### 6.4.3 cleanup\_mapping

Subroutine CLEANUP\_MAPPING deallocates memory from a derived-type object containing mapping information.

#### INTERFACE:

```
SUBROUTINE Cleanup_Mapping(mapping)
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(MapWeight), POINTER, INTENT(INOUT) :: mapping(:,:)
```

#### REVISION HISTORY:

```
03 Mar 2012 - R. Yantosca - Initial version
17 Apr 2012 - R. Yantosca - Rename to "map" to "mapping to avoid name
 confusion with a F90 intrinsic function
```

---

## 6.5 Fortran: Module Interface *get\_ndepl\_mod.F*

Module GET\_NDEP\_MOD contains routines for computing the accumulated nitrogen dry and wet deposition between emission time steps. These variables are needed for soil NO<sub>x</sub> emission calculations.

This module is basically a simple wrapper module to save out the nitrogen dry and wet deposition rates and pass them to HEMCO for soil NO<sub>x</sub> emission calculation (via *hcoi\_gc\_main\_mod.F90*).

IMPORTANT: Routine RESET\_DEP\_N resets the deposition arrays to zero. It is called in *hcoi\_gc\_main\_mod.F90* after the emission calculations.

#### INTERFACE:

```
MODULE GET_NDEP_MOD
```

#### USES:

```
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC DATA MEMBERS:

```
! Soil NOx deposited N arrays
REAL(fp), PUBLIC, POINTER :: DRY_TOTN (:,:) => NULL() ! Drydep'd N
REAL(fp), PUBLIC, POINTER :: WET_TOTN (:,:) => NULL() ! Wetdep'd N
```

#### PUBLIC MEMBER FUNCTIONS:

```

PUBLIC :: SOIL_DRYDEP
PUBLIC :: SOIL_WETDEP
PUBLIC :: RESET_DEP_N
PUBLIC :: Init_Get_Ndep
PUBLIC :: Cleanup_Get_Ndep

```

**REVISION HISTORY:**

```

23 Oct 2012 - M. Payer - Added ProTeX headers
25 Jul 2014 - R. Yantosca - Moved module variables DEP_RESERVOIR, DRY_TOTN,
 and WET_TOTN here from Headers/comsoil_mod.F90
25 Jul 2014 - R. Yantosca - Add routine INIT_GET_NDEP
25 Jul 2014 - R. Yantosca - Add routine CLEANUP_GET_NDEP
09 Oct 2014 - C. Keller - Removed obsolete routines DEP_RESERVOIR,
 GET_DEP_N, SOURCE_DRYN, and SOURCE_WETN. These
 are now handled in hcox_soilnox_mod.F90.
13 Nov 2014 - M. Yannetti - Added PRECISION_MOD
16 Jun 2016 - C. Miller - Now define species ID flags with Ind_
17 Jun 2016 - R. Yantosca - Add species ID flags as module variables

```

**6.5.1 soil\_drydep**

Subroutine SOIL\_DRYDEP holds dry deposited species [molec/cm<sup>2</sup>/s]. This is called from dry\_dep\_mod.F.

**INTERFACE:**

```

SUBROUTINE SOIL_DRYDEP(I, J, L, NN, TDRYFX)

```

**USES:**

```

USE State_Chm_Mod, only : Ind_

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I ! I
INTEGER, INTENT(IN) :: J ! J
INTEGER, INTENT(IN) :: L ! Level
INTEGER, INTENT(IN) :: NN ! Dry Dep Tracer #
REAL(fp), INTENT(IN) :: TDRYFX ! Dry dep flux [molec/cm2/s]

```

**REVISION HISTORY:**

```

23 Oct 2012 - M. Payer - Added ProTeX headers
14 Mar 2013 - M. Payer - Replace NOx with NO2 as part of removal of
 NOx-Ox partitioning
16 Jun 2016 - C. Miller - Now use

```

### 6.5.2 soil\_wetdep

Subroutine SOIL\_WETDEP holds wet deposited species [molec/cm2/s]. This is called from wetscav\_mod.F.

#### INTERFACE:

```
SUBROUTINE SOIL_WETDEP(I, J, L, NN, TWETFX)
```

#### USES:

```
USE State_Chm_Mod, Only : Ind_
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! I
INTEGER, INTENT(IN) :: J ! J
INTEGER, INTENT(IN) :: L ! Level
INTEGER, INTENT(IN) :: NN ! Wet Dep Tracer #
REAL(fp), INTENT(IN) :: TWETFX ! Wet dep flux [kg/s]
```

#### REVISION HISTORY:

```
23 Oct 2012 - M. Payer - Added ProTeX headers
```

---

### 6.5.3 reset\_dep\_N

Subroutine RESET\_DEP\_N resets the dry and wet deposition arrays and variables so that they can be refilled.

#### INTERFACE:

```
SUBROUTINE RESET_DEP_N()
```

#### REVISION HISTORY:

```
03 Apr 2014 - C. Keller - Initial version
```

---

### 6.5.4 Init\_Get\_Ndep

Routine INIT\_GET\_NDEP allocates all module arrays.

#### INTERFACE:

```
SUBROUTINE Init_Get_Ndep(am_I_Root, RC)
```

#### USES:

```
USE CMN_SIZE_Mod
USE ErrCode_Mod
USE State_Chm_Mod, ONLY : Ind_
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

**REMARKS:**

We now allocate the DEP\_RESERVOIR, DRY\_TOTN, and WET\_TOTN arrays within get\_ndep\_mod.F. These were formerly contained in Headers/commsoil\_mod.F90, which was rendered mostly obsolete by HEMCO.

**REVISION HISTORY:**

```
25 Jul 2014 - R. Yantosca - Initial version
09 Oct 2014 - C. Keller - Removed obsolete variable DEP_RESERVOIR
```

---

**6.5.5 cleanup\_commsoil**

Subroutine CLEANUP\_COMMSOIL deallocates all module arrays.

**INTERFACE:**

```
SUBROUTINE Cleanup_Get_NDep(am_I_Root, RC)
```

**USES:**

```
USE ErrCode_Mod
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Return code
```

**REVISION HISTORY:**

```
25 Jul 2014 - R. Yantosca - Initial version
09 Oct 2014 - C. Keller - Removed obsolete variable DEP_RESERVOIR
```

## 7 Emissions and chemistry modules

These modules contain routines to (1) call HEMCO to compute emissions rates and (2) feed them into the FlexChem/KPP chemistry solver.

---

## 7.1 Fortran: Module Interface emissions\_mod.F90

Module emissions\_mod.F90 is a wrapper module to interface GEOS-Chem and HEMCO. It basically just calls the GEOS-Chem - HEMCO interface routines. For some specialty sims, a few additional steps are required that are also executed here.

### INTERFACE:

```
MODULE Emissions_Mod
```

### USES:

```
IMPLICIT NONE
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Emissions_Init
PUBLIC :: Emissions_Run
PUBLIC :: Emissions_Final
```

### REVISION HISTORY:

```
27 Aug 2014 - C. Keller - Initial version.
20 Jun 2016 - R. Yantosca - Declare species ID flags as module variables
```

---

#### 7.1.1 emissions\_init

Subroutine EMISSIONS\_INIT calls the HEMCO - GEOS-Chem interface initialization routines.

### INTERFACE:

```
SUBROUTINE Emissions_Init(am_I_Root, Input_Opt, State_Met, &
 State_Chm, RC, HcoConfig)
```

### USES:

```
USE ErrCode_Mod
USE HCOI_GC_Main_Mod, ONLY : HCOI_GC_Init
USE HCO_Types_Mod, ONLY : ConfigObj
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
```

### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)) :: am_I_Root ! root CPU?
TYPE(MetState), INTENT(IN)) :: State_Met ! Met state
TYPE(ChmState), INTENT(IN)) :: State_Chm ! Chemistry state
```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input opts
TYPE(ConfigObj), POINTER, OPTIONAL :: HcoConfig ! HEMCO config object
INTEGER, INTENT(INOUT) :: RC ! Failure or success

```

**REVISION HISTORY:**

```

27 Aug 2014 - C. Keller - Initial version
16 Jun 2016 - J. Sheng - Added tracer index retriever
20 Jun 2016 - R. Yantosca - Now define species IDs only in the INIT phase
22 Jan 2018 - R. Yantosca - Return error code to calling routine

```

---

**7.1.2 emissions\_run**

Subroutine EMISSIONS\_RUN calls the HEMCO - GEOS-Chem interface run routines.

**INTERFACE:**

```

SUBROUTINE Emissions_Run(am_I_Root, Input_Opt, State_Met, &
 State_Chm, EmisTime, Phase, RC)

```

**USES:**

```

USE BROMOCARB_MOD, ONLY : SET_BR0
USE BROMOCARB_MOD, ONLY : SET_CH3BR
USE CARBON_MOD, ONLY : EMISSCARBON
USE CO2_MOD, ONLY : EMISSCO2
USE ErrCode_Mod
USE GLOBAL_CH4_MOD, ONLY : EMISSCH4
USE HCOI_GC_MAIN_MOD, ONLY : HCOI_GC_RUN
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
#if defined (TOMAS)
USE CARBON_MOD, ONLY : EMISSCARBONTOMAS !jkodros
USE SULFATE_MOD, ONLY : EMISSSULFATETOMAS !jkodros
#endif

! Setting other surface VMRs
Use sfcVMR_Mod, Only : fixSfcVMR

! Use old mercury code for now (ckeller, 09/23/2014)
USE MERCURY_MOD, ONLY : EMISSMERCURY

! For UCX, use Seb's routines for now
USE UCX_MOD, ONLY : EMISS_BASIC

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! root CPU?
LOGICAL, INTENT(IN) :: EmisTime ! Emissions in this time step
INTEGER, INTENT(IN) :: Phase ! Run phase

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Met state
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry state
TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input opts
INTEGER, INTENT(INOUT) :: RC ! Failure or success

```

**REVISION HISTORY:**

```

27 Aug 2014 - C. Keller - Initial version
13 Nov 2014 - C. Keller - Added EMISSCARBON (for SESQ and POA)
21 Nov 2014 - C. Keller - Added EMISSVOC to prevent VOC build-up
 above tropopause
22 Sep 2016 - R. Yantosca - Don't call EMISSCARBON unless we are doing
 a fullchem or aerosol simulation
26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
22 Jan 2018 - R. Yantosca - Return error code to calling program

```

**7.1.3 emissions\_final**

Subroutine EMISSIONS\_FINAL calls the HEMCO - GEOS-Chem interface finalization routines.

**INTERFACE:**

```

SUBROUTINE Emissions_Final(am_I_Root, Error, RC)

```

**USES:**

```

USE ErrCode_Mod
USE HCOI_GC_Main_Mod, ONLY : HCOI_GC_Final

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! root CPU?
LOGICAL, INTENT(IN) :: Error ! Cleanup arrays after crash?

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

27 Aug 2014 - C. Keller - Initial version

```

## 7.2 Fortran: Module Interface chemistry\_mod.F90

Module CHEMISTRY\_MOD is used to call the proper chemistry subroutine for the various GEOS-Chem simulations.

### INTERFACE:

```
MODULE Chemistry_Mod
```

### USES:

```
USE Precision_Mod ! For GEOS-Chem Precision (fp)
USE Geos_Timers_Mod ! For GEOS-Chem timers (optional)
```

```
IMPLICIT NONE
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: INIT_CHEMISTRY
PUBLIC :: DO_CHEMISTRY
PUBLIC :: RECOMPUTE_OD
```

### PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: CHEM_PASSIVE_SPECIES
```

### REVISION HISTORY:

- (1 ) Bug fix in DO\_CHEMISTRY (bnd, bmy, 4/14/03)
  - (2 ) Now references DEBUG\_MSG from "error\_mod.f" (bmy, 8/7/03)
  - (3 ) Now references "tagged\_ox\_mod.f"(bmy, 8/18/03)
  - (4 ) Now references "Kr85\_mod.f" (jsw, bmy, 8/20/03)
  - (5 ) Bug fix: Now also call OPTDEPTH for GEOS-4 (bmy, 1/27/04)
  - (6 ) Now references "carbon\_mod.f" and "dust\_mod.f" (rjp, tdf, bmy, 4/5/04)
  - (7 ) Now references "seasalt\_mod.f" (rjp, bec, bmy, 4/20/04)
  - (8 ) Now references "logical\_mod.f", "tracer\_mod.f", "diag20\_mod.f", and "diag65\_mod.f", and "aerosol\_mod." (bmy, 7/20/04)
  - (9 ) Now references "mercury\_mod.f" (bmy, 12/7/04)
  - (10) Updated for SO4s, NITs chemistry (bec, bmy, 4/13/05)
  - (11) Now call CHEM\_HCN\_CH3CN from "hcn\_ch3cn\_mod.f". Also remove all references to the obsolete CO-OH param simulation. (xyp, bmy, 6/24/05)
  - (12) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
  - (13) Now call MAKE\_RH from "main.f" (bmy, 3/16/06)
  - (14) Updated for SOA from isoprene (dkh, bmy, 6/1/06)
  - (15) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
  - (16) For now, replace use RPMARES instead of ISORROPIA. (bmy, 4/2/08)
  - (17) Added KPP chemistry driver subroutine (phs,ks,dhk, 09/15/09)
  - (18) Added public member function recompute\_OD (skim, 02/03/11)
- 17 Dec 2009 - R. Yantosca - Added ProTeX headers  
28 Jan 2010 - C. Carouge, R. Yantosca - Modified for ISORROPIA II  
08 Aug 2012 - R. Yantosca - Now align IF statements better



10 Aug 2012 - R. Yantosca - Cosmetic changes  
 25 Mar 2013 - M. Payer - Now pass State\_Chm to several routines  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 19 May 2014 - C. Keller - Added INIT\_CHEMISTRY  
 15 Dec 2014 - M. Yannetti - KPP code is commented out unless compiling KPP  
 08 Jan 2015 - M. Sulprizio- Now restrict KPP to REAL\*8 to allow for KPP code  
                   to compile properly  
 13 Aug 2015 - E. Lundgren - Tracer units are now kg/kg and converted to  
                   kg within DO\_CHEMISTRY  
 03 Nov 2016 - C. Keller - Added wrapper routine for passive tracers.  
 17 Nov 2017 - R. Yantosca - Now in F90 format; added Diag\_OH\_HO2\_01D\_03P

### 7.2.1 do\_chemistry

Subroutine DO\_CHEMISTRY is the driver routine which calls the appropriate chemistry subroutine for the various GEOS-Chem simulations.

#### INTERFACE:

```

SUBROUTINE DO_CHEMISTRY(am_I_Root, Input_Opt, State_Met, &
 State_Chm, State_Diag, RC)

```

#### USES:

```

USE AEROSOL_MOD, ONLY : AEROSOL_CONC
USE AEROSOL_MOD, ONLY : RDAER
USE AEROSOL_MOD, ONLY : SOILDUST
USE C2H6_MOD, ONLY : CHEMC2H6
USE CARBON_MOD, ONLY : CHEMCARBON
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
#endif
USE CMN_SIZE_MOD
USE DUST_MOD, ONLY : CHEMDUST
USE DUST_MOD, ONLY : RDUST_ONLINE
USE ErrCode_Mod
USE ERROR_MOD
USE FlexChem_Mod, ONLY : Do_FlexChem
USE GLOBAL_CH4_MOD, ONLY : CHEMCH4
USE Input_Opt_Mod, ONLY : OptInput
USE ISOROPIAII_MOD, ONLY : DO_ISOROPIAII
USE MERCURY_MOD, ONLY : CHEMMERCURY
USE POPS_MOD, ONLY : CHEMPOPS
USE RnPbBe_MOD, ONLY : CHEMRnPbBe
USE RPMARES_MOD, ONLY : DO_RPMARES
USE SEASALT_MOD, ONLY : CHEMSEASALT
USE SULFATE_MOD, ONLY : CHEMSULFATE
USE State_Chm_Mod, ONLY : ChmState

```

```

USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE STRAT_CHEM_MOD, ONLY : DO_STRAT_CHEM
USE TAGGED_CO_MOD, ONLY : CHEM_TAGGED_CO
USE TAGGED_O3_MOD, ONLY : CHEM_TAGGED_O3
USE TIME_MOD, ONLY : GET_ELAPSED_SEC
USE TIME_MOD, ONLY : GET_TS_CHEM
#if defined(USE_TEND)
 USE TENDENCIES_MOD
#endif
#if defined(TOMAS)
 USE TOMAS_MOD, ONLY : DO_TOMAS !(win, 7/14/09)
#endif
USE UCX_MOD, ONLY : CALC_STRAT_AER
USE UCX_MOD, ONLY : READ_PSC_FILE
USE UCX_MOD, ONLY : WRITE_STATE_PSC
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure

```

**REMARKS:****REVISION HISTORY:**

- (1) Now reference DELP, T from "dao\_mod.f" since we need to pass this to OPTDEPTH for GEOS-1 or GEOS-STRAT met fields (bnd, bmy, 4/14/03)
- (2) Now references DEBUG\_MSG from "error\_mod.f" (bmy, 8/7/03)
- (3) Removed call to CHEM03, it's obsolete. Now calls CHEM\_TAGGED\_OX ! from "tagged\_ox\_mod.f" when NSRCX==6. Now calls Kr85 chemistry if NSRCX == 12 (jsw, bmy, 8/20/03)
- (4) Bug fix: added GEOS-4 to the #if block in the call to OPTDEPTH. (bmy, 1/27/04)
- (5) Now calls CHEMCARBON and CHEMDUST to do carbon aerosol & dust aerosol chemistry (rjp, tdf, bmy, 4/2/04)
- (6) Now calls CHEMSEASALT to do seasalt aerosol chemistry (rjp, bec, bmy, 4/20/04)

- (7 ) Now references "logical\_mod.f" & "tracer\_mod.f". Now references AEROSOL\_CONC, AEROSOL\_RURALBOX, and RDAER from "aerosol\_mod.f". Now includes "CMN\_DIAG" and "comode.h". Also call READER, READCHEM, and INPHOT to initialize the FAST-J arrays so that we can save out ! AOD's to the ND21 diagnostic for offline runs. (bmy, 7/20/04)
  - (8 ) Now call routine CHEMMERCURY from "mercury\_mod.f" for an offline Hg0/Hg2/HgP simulation. (eck, bmy, 12/7/04)
  - (9 ) Now do not call DO\_RPMARES if we are doing an offline aerosol run with crystalline sulfur & aqueous tracers (cas, bmy, 1/7/05)
  - (10) Now use ISORROPIA for aer thermodyn equilibrium if we have seasalt tracers defined, or RPMARES if not. Now call CHEMSEASALT before CHEMSULFATE. Now do aerosol thermodynamic equilibrium before aerosol chemistry for offline aerosol runs. Now also reference CLDF from "dao\_mod.f" (bec, bmy, 4/20/05)
  - (11) Now modified for GCAP met fields. Now call CHEM\_HCN\_CH3CN from "hcn\_ch3cn\_mod.f". Also remove all references to the obsolete CO-OH param simulation. (xyp, bmy, 6/23/05)
  - (12) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
  - (13) Now call MAKE\_RH from "main.f" (bmy, 3/16/06)
  - (14) Removed ISOP\_PRIOR as a local variable (dkh, bmy, 6/1/06)
  - (15) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
  - (16) Now use DRYFLXH2HD and CHEM\_H2\_HD for H2/HD sim (lyj, phs, 9/18/07)
  - (17) Bug fix: now hardwired to use RPMARES since ISORROPIA can return very unphysical values at low RH. Wait for ISORROPIA II. (bmy, 4/2/08)
  - (18) The dry deposition diagnostic (ND44) is done in vdiff\_mod if using non-local PBL (lin, ccc, 5/29/09)
  - (19) Now calls CHEMPOPS from "pops\_mod.f" for an offline POPs simulation (eck, 9/20/10)
- 17 Dec 2009 - R. Yantosca - Added ProTeX headers
- 25 Jan 2010 - R. Yantosca - Now call DO\_TOMAS for TOMAS microphysics
- 28 Jan 2010 - C. Carouge, R. Yantosca - Modified for ISORROPIA II
- 19 Mar 2012 - R. Yantosca - Add C-preprocessor switch to shut off ISORROPIA to facilitate debugging
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument, and pass this down to lower-level chem routines for GIGC
- 08 Aug 2012 - R. Yantosca - Now align IF statements better
- 10 Aug 2012 - R. Yantosca - Cosmetic changes
- 18 Oct 2012 - R. Yantosca - Rename GC\_MET argument to State\_Met
- 18 Oct 2012 - R. Yantosca - Rename CHEM\_STATE argument to State\_Chem
- 19 Oct 2012 - R. Yantosca - Now reference gignc\_state\_chm\_mod.F90
- 19 Oct 2012 - R. Yantosca - Now reference gignc\_state\_met\_mod.F90
- 25 Oct 2012 - R. Yantosca - Add comments for GIGC #ifdefs
- 25 Oct 2012 - R. Yantosca - Add the RC output argument for the GIGC
- 08 Nov 2012 - R. Yantosca - Now pass Input\_Opt argument for the GIGC and use fields of Input\_Opt to replace logicals
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 26 Nov 2012 - R. Yantosca - Now pass Input\_Opt, State\_Chm, RC to routine

DO\_STRAT\_CHEM (in GeosCore/strat\_chem\_mod.F90)

11 Dec 2012 - R. Yantosca - Remove NI, NJ, NL, NCNST arguments; these are now obtained either from CMN\_SIZE\_mod.F or from the Input\_Opt object

05 Mar 2013 - R. Yantosca - Now pass am\_I\_Root, Input\_Opt, RC to DRYFLX

25 Mar 2013 - H. Amos - merged C. Friedman's PAH code into v9-01-03

28 Mar 2013 - S.D. Eastham- Updated to use FAST\_JX\_MOD

31 May 2013 - R. Yantosca - Now pass Input\_Opt, State\_Chm to DO\_TOMAS

19 May 2014 - C. Keller - Removed call for acetone ocean sink - now done in HEMCO.

06 Nov 2014 - M. Yannetti - Added PRECISION\_MOD

08 May 2015 - C. Keller - Added WRITE\_STATE\_PSC.

18 May 2015 - R. Yantosca - Remove DIAG\_STATE\_PSC, that is not used anymore

15 Jun 2015 - R. Yantosca - Removed calls to DRYFLXRnPbBe, that's obsolete

04 Sep 2015 - C. Keller - Added passive tracer call.

17 Mar 2016 - M. Sulprizio- Remove call to OPTDEPTH. The optical depth fields are now saved into State\_Met%OPTD in the routines that read the met fields from disk.

16 May 2016 - M. Sulprizio- Remove call to AEROSOL\_RURALBOX. The FlexChem implementation has rendered the routine obsolete.

16 Jun 2016 - C. Miller - Now use Ind\_function to define species ID's

17 Jun 2016 - R. Yantosca - Now define species ID's only on first call

17 Jun 2016 - R. Yantosca - Now reset first-time flag at end of routine

30 Jun 2016 - R. Yantosca - Remove instances of STT.

19 Jul 2016 - R. Yantosca - Now bracket DO\_TEND calls with #ifdef USE\_TEND

10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code

11 Aug 2016 - R. Yantosca - Clean up calls to error subroutines

09 Mar 2017 - C. Keller - Bug fix: call TEND\_STAGE1 before unit conversion

28 Sep 2017 - E. Lundgren - Simplify unit conversions with wrapper routine

03 Oct 2017 - E. Lundgren - Pass State\_Diag as argument

09 Nov 2017 - R. Yantosca - Reorder arguments for consistency: Input\_Opt, State\_Met, State\_Chm, State\_Diag

20 Nov 2017 - R. Yantosca - Move ND43 diagnostics to flexchem\_mod.F90

03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input\_Opt%LUCX

06 Feb 2018 - E. Lundgren- Change GET\_ELAPSED\_MIN to GET\_ELAPSED\_SEC to match new timestep unit of seconds

## 7.2.2 recompute\_od

Subroutine RECOMPUTE\_OD will update the optical depth values before accumulating or writing the diagnostics.

### INTERFACE:

```

SUBROUTINE RECOMPUTE_OD(am_I_Root, Input_Opt, State_Met, &
 State_Chm, State_Diag, RC)

```

### USES:



**USES:**

```

USE CMN_SIZE_Mod, ONLY : IIPAR, JJPARG, LLPARG
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ind_
USE Time_Mod, ONLY : Get_Ts_Chem

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology state object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry state object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(INOUT) :: RC ! Failure or success

```

**REMARKS:****REVISION HISTORY:**

```

04 Sep 2015 - C. Keller - Initial version
03 Nov 2016 - C. Keller - Moved to chemistry_mod
26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
14 Jul 2017 - E. Lundgren - Remove dependency on passive_species_mod.F90
02 Aug 2017 - R. Yantosca - Turn off debug print unless ND70 is activated
13 Dec 2017 - R. Yantosca - Now apply decay only to those passive species
 with finite atmospheric lifetimes

```

**7.2.4 init\_chemistry**

Subroutine INIT\_CHEMISTRY initializes chemistry variables.

**INTERFACE:**

```

SUBROUTINE Init_Chemistry(am_I_Root, Input_Opt, &
 State_Chm, State_Diag, RC)

```

**USES:**

```

USE ErrCode_Mod
USE FAST_JX_MOD, ONLY : Init_FJX
USE FlexChem_Mod, ONLY : Init_FlexChem

```

```

USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
INTEGER, INTENT(INOUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

19 May 2014 - C. Keller - Initial version (stripped from do_chemistry
 and chemdr.F)
20 Jun 2014 - R. Yantosca - Now pass Input_Opt to INIT_FJX
23 Jun 2016 - R. Yantosca - Remove call to SETTRACE, it's obsolete
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

**7.3 Fortran: Module Interface fast\_jx\_mod.F**

Module FAST\_JX\_MOD contains routines and variables for calculating photolysis rates using the Fast-JX scheme (Prather et al). Current implementation is version 7.0a.

**INTERFACE:**

```

MODULE FAST_JX_MOD

```

**USES:**

```

USE CMN_FJX_MOD
USE PRECISION_MOD !For GEOS-Chem Precision (fp)

```

```

IMPLICIT NONE

```

```

PRIVATE

```

**PUBLIC MEMBER FUNCTIONS:**

```

PUBLIC :: EXITC
PUBLIC :: PHOTO_JX
PUBLIC :: INIT_FJX
PUBLIC :: FAST_JX
PUBLIC :: PHOTRATE_ADJ

```

**PRIVATE MEMBER FUNCTIONS:**

```

PRIVATE :: SOLAR_JX
PRIVATE :: OPMIE
PRIVATE :: MIESCT
PRIVATE :: LEGNDO
PRIVATE :: BLKSLV
PRIVATE :: GEN_ID
PRIVATE :: JRATET
PRIVATE :: X_INTERP
PRIVATE :: SPHERE2
PRIVATE :: EXTRAL
PRIVATE :: RD_PROF_NC
PRIVATE :: RD_XXX
PRIVATE :: RD_AOD
PRIVATE :: RD_MIE
PRIVATE :: RD_JS_JX
PRIVATE :: SET_AER

```

**REVISION HISTORY:**

```

27 Mar 2013 - S. D. Eastham - Initial version (based on original GEOS-Chem
 files and instructions from J. Mao)
28 Mar 2013 - S. D. Eastham - Upgraded to Fast-JX v7.0
20 Feb 2014 - M. Sulprizio - Removed "define.h", this is now obsolete
13 Nov 2014 - M. Yannetti - Added PRECISION_MOD
29 Mar 2016 - R. Yantosca - Add flags for rxns adjusted by PHOTRATE_ADJ
27 Jun 2016 - M. Sulprizio - Remove FJXFUNC and JV_INDEX routines. We now
 obtain the photolysis rates directly from the
 ZPJ array.
29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

```

**7.3.1 fast\_jx**

Subroutine FAST\_JX loops over longitude and latitude, and calls PHOTO\_JX to compute J-Values for each column at every chemistry time-step.

**INTERFACE:**

```

SUBROUTINE FAST_JX(WLAOD, am_I_Root, Input_Opt,
& State_Met, State_Chm, State_Diag, RC)

```

**USES:**

```

USE CMN_SIZE_MOD, ONLY : IIPAR, JJPAR, LLPAR
USE CMN_SIZE_MOD, ONLY : NDUST
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP, ALLOC_ERR
USE ERROR_MOD, ONLY : DEBUG_MSG

```



```

USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH, GET_DAY, GET_DAY_OF_YEAR
USE TIME_MOD, ONLY : GET_TAU, GET_YEAR
USE TOMS_MOD, ONLY : GET_OVERHEAD_03

```

```

IMPLICIT NONE

```

```

=====
Uncomment the appropriate #define statement to denote which of the
available cloud overlap options that you wish to use.

```

```

! Linear overlap

```

```

#define USE_LINEAR_OVERLAP 1

```

```

Approximate random overlap (balance between accuracy & speed)

```

```

#define USE_APPROX_RANDOM_OVERLAP 1

```

```

! Maximum random cloud overlap (most computationally intensive)

```

```

#define USE_MAXIMUM_RANDOM_OVERLAP 1
=====

```

#### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: WLAOD ! AOD calculated how?
 ! (1: 550 nm, 0: 999 nm)
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State obj
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State obj

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State obj

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REMARKS:

```

Parameter to choose cloud overlap algorithm:

```

```

=====
(1) OVERLAP (INTEGER) : 1 - Linear Approximation (used up to v7-04-12)
 2 - Approximate Random Overlap (default)
 3 - Maximum Random Overlap (computation intensive)

```

#### REVISION HISTORY:

- 01 Apr 1998 - P. Murti, R. Martin, R. Yantosca - Initial version
- (1 ) Call this routine EACH chemistry time-step, before solver.
  - (2 ) This routine must know IMAX, JMAX, LMAX.
  - (3 ) Now use new !\$OMP compiler directives for parallelization (bmy, 5/2/00)
  - (4 ) Now reference "cmn\_fj.h" and "jv\_cmn.h" for the aerosol optical depths (bmy, 10/2/00)
  - (5 ) Add OPTDUST as a local variable -- make OPTDUST private for the parallel DO-loop, since it stores 1 column of aerosol optical depth for each dust type (bmy, rvm, 10/2/00)
  - (6 ) For now, LPAR in "cmn\_fj.h" = LGLOBAL in "CMN\_SIZE". Therefore we assume that we are always doing global runs. (bmy, 10/2/00)
  - (7 ) Removed obsolete code from 10/2/00 (bmy, 12/21/00)
  - (8 ) Replace {IJL}GLOB w/ IIPAR,JJPARG,LLPAR everywhere. Also YLMID(NLAT) needs to be referenced by YLMID(NLAT+J0). (bmy, 9/26/01)
  - (9 ) Remove obsolete code from 9/01. Updated comments. (bmy, 10/24/01)
  - (10) Add OPTAER as a local variable, make it private for the parallel DO loop, since it stores 1 column of aerosol optical depths for each aerosol type. Pass OPTAER to PHOT0J via the argument list. Declare OPTAER as PRIVATE for the parallel DO-loop. (rvm, bmy, 2/27/02)
  - (11) Now reference GET\_PEDGE from "pressure\_mod.f", which returns the correct "floating" pressure. (dsa, bdf, bmy, 8/20/02)
  - (12) Now reference T from "dao\_mod.f" (bmy, 9/23/02)
  - (13) Now uses routine GET\_YMID from "grid\_mod.f" to compute grid box latitude. Now make IDAY, MONTH local variables. Now use function GET\_DAY\_OF\_YEAR from "time\_mod.f". Bug fix: now IDAY (as passed to photoj.f) is day of year rather than cumulative days since Jan 1, 1985. (bmy, 2/11/03)
  - (14) Now reference routine GET\_YEAR from "time\_mod.f". Added LASTMONTH as a SAVED variable. Now call READ\_TOMS03 from "toms\_mod.f" at the beginning of a new month (or the first timestep) to read TOMS 03 columns which will be used by "set\_prof.f". Now also reference routine GET\_DAY from "time\_mod.f". Rename IDAY to DAY\_OF\_YR. Pass day of month to PHOT0J. Updated comments, cosmetic changes. (bmy, 7/17/03)
  - (15) Bug fix: PRES needs to be the true surface pressure for GEOS-4, but PS-PTOP for all prior GEOS models. (bmy, 2/6/04)
  - (16) Now account for cloud overlap (Maximum-Random Overlap and Random Overlap) in each column (hyl, phs, bmy, 9/18/07)
  - (17) Now initialize the PJ array here, instead of two layers below in "set\_prof.f". Now no longer pass PRES to "photoj.f". (bmy, 11/29/07)
  - (18) Now switch to approx. random overlap option (hyl, phs, bmy, 10/7/08)
  - (19) Now can handle GEOS-5 reprocessed met data with OPTDEPTH being in-cloud optical depths. (bmy, hyl, 10/24/08)
  - (10) Remove references to IN\_CLOUD\_OD (bmy, 10/15/09)
- 13 Aug 2010 - R. Yantosca - Added ProTeX headers
- 13 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
- 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID(I,J,L) from grid\_mod.F90

06 Mar 2012 - R. Yantosca - Now call GET\_OVERHEAD\_03 to get the total overhead 03 column for FAST-J

30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F

10 Aug 2012 - R. Yantosca - Replace IPAR, JPAR, LPAR w/ IIPAR, JJPAR, LLPAR

27 Mar 2013 - S.D. Eastham- Rolled in FAST\_JX\_MOD

21 Feb 2014 - M. Sulprizio- Replaced all met field arrays with State\_Met derived type object

28 Feb 2014 - M. Sulprizio- Now obtain O3\_CTM directly from CSPEC

05 Mar 2014 - R. Yantosca - Cosmetic changes

06 Nov 2014 - R. Yantosca - Now use State\_Met%CLDF(I,J,L)

06 Nov 2014 - R. Yantosca - Now use State\_Met%OPTD(I,J,L)

26 Feb 2015 - E. Lundgren - Replace GET\_PEDGE with State\_Met%PEDGE. Remove dependency on pressure\_mod and redundant PEDGE pre-preprocessor block if EXTERNAL\_GRID.

09 Apr 2015 - R. Yantosca - Bug fixes: Add missing vars to !\$OMP+PRIVATE clause and error check for JLOOP > 0

11 Aug 2015 - R. Yantosca - MERRA2 behaves the same way as for GEOS-FP

21 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species

17 Jun 2016 - R. Yantosca - Now define id\_03 with the Ind\_ function

03 Oct 2017 - E. Lundgren - Pass State\_Diag as argument

### 7.3.2 blkslv

Subroutine BLKSLV solves the block tri-diagonal system

#### INTERFACE:

```

SUBROUTINE BLKSLV
& (FJ,POMEGA,FZ,ZTAU,ZFLUX,RFL,PM,PMO,FJTOP,FJBOT,ND)

```

#### USES:

#### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: ND
REAL(fp), INTENT(IN) :: POMEGA(M2_,N_,W_),FZ(N_,W_),ZTAU(N_,W_),
& PM(M_,M2_),PMO(M2_),
& RFL(W_),ZFLUX(W_)

```

#### OUTPUT PARAMETERS:

```

REAL(fp), INTENT(OUT) :: FJ(N_,W_),FJTOP(W_),FJBOT(W_)

```

#### REMARKS:

The block tri-diagonal system:

$$A(I)*X(I-1) + B(I)*X(I) + C(I)*X(I+1) = H(I)$$

**REVISION HISTORY:**

27 Mar 2013 - S. D. Eastham - Copied from GEOS-Chem v9-01-03

---

**7.3.3 gen\_id**

Subroutine GEN generates coefficient matrices for the block tri-diagonal system described in BLKSLV.

**INTERFACE:**

```

SUBROUTINE GEN_ID(POMEGA,FZ,ZTAU,ZFLUX,RFL,PM,PMO
& ,B,CC,AA,A,H,C, ND)

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: ND
REAL(fp), INTENT(IN) :: POMEGA(M2_,N_),PM(M_,M2_),PMO(M2_)
REAL(fp), INTENT(IN) :: ZFLUX,RFL
REAL(fp), INTENT(IN),DIMENSION(N_) :: FZ,ZTAU

```

**OUTPUT PARAMETERS:**

```

REAL(fp), INTENT(OUT),DIMENSION(M_,M_,N_) :: B,AA,CC
REAL(fp), INTENT(OUT),DIMENSION(M_,N_) :: A,C,H

```

**REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

**7.3.4 jratet**

Subroutine JRATET calculates temperature-dependent J-rates.

**INTERFACE:**

```

SUBROUTINE JRATET(PPJ,TTJ,FFF,VALJL,LCTM,LCHEM,NJXU)

```

**USES:****INPUT PARAMETERS:**

```

integer, intent(in) :: LCTM,LCHEM,NJXU
real(fp), intent(in) :: PPJ(JXL1_+1),TTJ(JXL1_+1)
real(fp), intent(inout) :: FFF(W_,LCTM)
!OUTPUT VARIABLES:
real(fp), intent(out), dimension(LCTM,NJXU) :: VALJL

```

**REMARKS:**

**REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0  
02 Apr 2013 - S. D. Eastham - Now only assign J-rates on  
levels within the chemistry grid.

---

**7.3.5 x\_interp**

Subroutine X\_INTERP is an up-to-three-point linear interp. function for cross-sections.

**INTERFACE:**

SUBROUTINE X\_INTERP (TINT,XINT,T1,X1,T2,X2,T3,X3,L123)

**USES:**

**INPUT PARAMETERS:**

REAL(fp), INTENT(IN) :: TINT,T1,T2,T3, X1,X2,X3  
INTEGER, INTENT(IN) :: L123  
!OUTPUT VARIABLES:  
REAL(fp), INTENT(OUT) :: XINT

**REMARKS:**

**REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

**7.3.6 sphere2**

Subroutine SPHERE2 is an AMF2.

**INTERFACE:**

SUBROUTINE SPHERE2 (U0,ZHL,AMF2,L1U,LJX1U)

**USES:**

**INPUT PARAMETERS:**

```

 INTEGER, INTENT(IN) :: L1U, LJX1U
 REAL(fp), INTENT(IN) :: U0,ZHL(L1U+1)
!OUTPUT VARIABLES:
 REAL(fp), INTENT(OUT) :: AMF2(2*LJX1U+1,2*LJX1U+1)

```

**REMARKS:**

Quoting from the original:

New v6.2: does AirMassFactors for mid-layer, needed for SZA ~ 90  
 This new AMF2 does each of the half-layers of the CTM separately,  
 whereas the original, based on the pratmo code did the whole layers  
 and thus calculated the ray-path to the CTM layre edges, NOT the middle.  
 Since fast-JX is meant to calculate the intensity at the mid-layer, the  
 solar beam at low sun (interpolated between layer edges) was incorrect.  
 This new model does make some approximations of the geometry of the layers:  
 the CTM layer is split evenly in mass (good) and in height (approx).

Calculation of spherical geometry; derive tangent heights, slant path  
 lengths and air mass factor for each layer. Not called when  
 SZA > 98 degrees. Beyond 90 degrees, include treatment of emergent  
 beam (where tangent height is below altitude J-value desired at).

-----  
 Inputs:

```

 U0 cos(solar zenith angle)
 RAD radius of Earth mean sea level (cm)
 ZHL(L) height (cm) of the bottom edge of CTM level L
 ZZHT scale height (cm) used above top of CTM (ZHL(L_+1))
 L1U dimension of CTM = levels +1 (L+1 = above-CTM level)

```

Outputs:

```

 AMF2(I,J) = air mass factor for CTM level I for sunlight reaching J
 (these are calculated for both layer middle and layer edge)

```

-----

**REVISION HISTORY:**

```

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0
04 Dec 2014 - R. Yantosca - Now use SQRT instead of DSQRT

```

---

**7.3.7 extral**

Subroutine EXTRAL adds sub-layers to thick cloud/aerosol layers using log-spacing for sub-layers of increasing thickness ATAU.

**INTERFACE:**

```

SUBROUTINE EXTRAL (DTAUX,L1X,L2X,NX,JXTRA)

```

**USES:****INPUT PARAMETERS:**

```

 INTEGER, INTENT(IN) :: L1X,L2X !index of cloud/aerosol
 integer, intent(in) :: NX !Mie scattering array size
 real(fp), intent(in) :: DTAUX(L1X) !cloud+3aerosol OD in each layer
!OUTPUT VARIABLES:
 integer, intent(out):: JXTRA(L2X+1) !number of sub-layers to be added

```

**REMARKS:**

DTAUX(L=1:L1X) = Optical Depth in layer L (generally 600 nm OD)  
 This can be just cloud or cloud+aerosol, it is used only to set  
 the number in levels to insert in each layer L  
 Set for log-spacing of tau levels, increasing top-down.

N.B. the TTAU, etc calculated here are NOT used elsewhere

The log-spacing parameters have been tested for convergence and chosen  
 to be within 0.5% for ranges OD=1-500, rreflect=0-100%, mu0=0.1-1.0  
 use of ATAU = 1.18 and min = 0.01, gives at most +135 pts for OD=100  
 ATAU = 1.12 now recommended for more -accurate heating rates (not J's)

**REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

**7.3.8 exitc**

Subroutine EXITC forces an error in GEOS-Chem and quits.

**INTERFACE:**

```

SUBROUTINE EXITC (T_EXIT)

```

**USES:**

```

USE ERROR_MOD, ONLY : ERROR_STOP

```

**INPUT PARAMETERS:**

```

 CHARACTER(LEN=*), INTENT(IN) :: T_EXIT
!OUTPUT VARIABLES:

```

**REMARKS:****REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

### 7.3.9 int\_fjx

Subroutine INIT\_FJX initializes Fast-JX variables.

#### INTERFACE:

```

 SUBROUTINE INIT_FJX(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)

```

#### USES:

```

#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD, ONLY : ND64
#endif
 USE ErrCode_Mod
 USE Input_Opt_Mod, ONLY : OptInput
 USE inquireMod, ONLY : findFreeLUN
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Chm_Mod, ONLY : Ind_
 USE State_Diag_Mod, ONLY : DgnState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Diagnostics State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REVISION HISTORY:

```

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0
05 Mar 2014 - R. Yantosca - Use correct filenames for use w/ ESMF
27 Sep 2017 - M. Sulprizio- Define species IDs for ND22 diagnostic
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
29 Dec 2017 - C. Keller - Print mapping on root CPU only

```

### 7.3.10 rd\_xxx

Subroutine RD\_XXX reads in wavelength bins, solar fluxes, Rayleigh and temperature-dependent cross-sections.

#### INTERFACE:

```

 SUBROUTINE RD_XXX (NUN,NAMFIL,am_I_Root)

```

#### USES:



**INPUT PARAMETERS:**

```

 INTEGER, INTENT(IN) :: NUN
 CHARACTER(*), INTENT(IN) :: NAMFIL
 LOGICAL, INTENT(IN) :: am_I_Root
!OUTPUT VARIABLES:

```

**REMARKS:**

```

NEW v-6.8 now allow 1 to 3 sets of X-sects for T or P
 LQQ = 1, 2, or 3 to determine interpolation with T or P
 IF the temperatures TQQQ are <0, then use as pressure interp (hPa)
 NB - the temperatures and pressures must be increasing
NEW v-6.4 changed to collapse wavelengths & x-sections to Trop-only:
 WX_ = 18 should match the JX_spec.dat wavelengths
 W_ = 12 (Trop-only) or 18 (std) is set in (CMN_FJX.F).
 if W_=12 then drop strat wavels, and drop x-sects (e.g. N2O, ...)
 W_ = 8, reverts to quick fix: fast-J (12-18) plus bin (5) scaled

```

```

NAMFIL Name of spectral data file (FJX_spec.dat) >> j2 for fast-J2
NUN Channel number for reading data file
NJX Number of species to calculate J-values for
NWWW Number of wavelength bins, from 1:NWWW
WBIN Boundaries of wavelength bins
WL Centres of wavelength bins - 'effective wavelength'
FL Solar flux incident on top of atmosphere (cm-2.s-1)
GRAYL Rayleigh parameters (effective cross-section) (cm2)
QO2 O2 cross-sections
QO3 O3 cross-sections
Q1D O3 => O(1D) quantum yield
TQQ Temperature for supplied cross sections
QQQ Supplied cross sections in each wavelength bin (cm2)

```

**REVISION HISTORY:**

```

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0
09 May 2014 - S. D. Eastham - Fixed bug when using <18 wavelengths

```

**7.3.11 rd\_mie**

Subroutine RD\_MIE retrieves aerosol scattering data for FJX.

**INTERFACE:**

```

SUBROUTINE RD_MIE(NUN,NAMFIL,am_I_Root, Input_Opt)

```

**USES:**

```

USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: NUN
CHARACTER(*), INTENT(IN) :: NAMFIL
LOGICAL, INTENT(IN) :: am_I_Root
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

!OUTPUT VARIABLES:

**REMARKS:**

```

NAMFIL Name of scattering data file (e.g., FJX_scat.dat)
NUN Channel number for reading data file
NAA Number of categories for scattering phase functions
QAA Aerosol scattering phase functions
WAA 5 Wavelengths for the supplied phase functions
PAA Phase function: first 8 terms of expansion
RAA Effective radius associated with aerosol type
SAA Single scattering albedo

```

**REVISION HISTORY:**

```

28 Mar 2013 - S. D. Eastham - Adapted from GEOS-Chem v9-1-3
25 Jun 2015 - M. Sulprizio - Add brown carbon optical properties from
 M. Hammer

```

**7.3.12 rd\_aod**

Subroutine RD\_AOD reads aerosol phase functions that are used to scale diagnostic output to an arbitrary wavelength. This facilitates comparing with satellite observations.

**INTERFACE:**

```

SUBROUTINE RD_AOD(NJ1, am_I_Root, Input_Opt)

```

**USES:**

```

USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
IMPLICIT NONE

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: NJ1 ! Unit # of file to open
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?

```

**INPUT/OUTPUT PARAMETERS:****REMARKS:**

The .dat files for each species contain the optical properties at multiple wavelengths to be used in the online calculation of the aerosol optical depth diagnostics.

These properties have been calculated using the same size and optical properties as the FJX\_spec.dat file used for the FAST-J photolysis calculations (which is now redundant for aerosols, the values in the .dat files here are now used). The file currently contains 11 wavelengths for Fast-J and other commonly used wavelengths for satellite and AERONET retrievals. 30 wavelengths follow that map onto RRTMG wavebands for radiative flux calculations (not used if RRTMG is off). A complete set of optical properties from 250-2000 nm for aerosols is available at:

[ftp://ftp.as.harvard.edu/geos-chem/data/aerosol\\_optics/hi\\_spectral\\_res](ftp://ftp.as.harvard.edu/geos-chem/data/aerosol_optics/hi_spectral_res)

-- Colette L. Heald, 05/10/10)

-- David A. Ridley, 05/10/13 (update for new optics files)

**REVISION HISTORY:**

```

10 May 2010 - C. Heald - Initial version
06 Aug 2010 - C. Carouge - Add an error check when opening the file
01 Aug 2012 - R. Yantosca - Now restore NJ1 to INTENT(IN) status
31 Mar 2013 - S. D. Eastham - Added to Fast-JX v7.0 implementation
30 Mar 2015 - M. Sulprizio - Removed NAMFIL argument. Now specify filenames
 for aerosol LUT files in this routine.
02 Apr 2015 - C. Keller - Append '.rc' to species files in ESMF mode.
12 May 2015 - R. Yantosca - Bug fix: For PGI compiler, you need to have
 exactly as many elements in the DATA statement
 as there are elements in the SPECFIL array.
25 Jun 2015 - M. Sulprizio - Add brown carbon optical properties from
 M. Hammer
04 Aug 2015 - M. Long - Removed ".rc" file specifier. Not necessary.
03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input_Opt%LUCX
23 Mar 2018 - E. Lundgren - Pass Input_Opt to Calc_AOD

```

**7.3.13 calc\_aod**

Subroutine CALC\_AOD works out the closest tie points in the optics LUT wavelengths and the coefficients required to calculate the angstrom exponent for interpolating optics to the

requested wavelength. If the wavelength requested matches a standard wavelength in the LUT then we skip the interpolation (DAR 09/2013)

#### INTERFACE:

```
SUBROUTINE CALC_AOD(am_I_Root, Input_Opt)
```

#### USES:

```
USE CMN_FJX_MOD, ONLY : NWVAA, NWVAAO, WVAA
USE CMN_FJX_MOD, ONLY : IWVSELECT
USE CMN_FJX_MOD, ONLY : IRTWVSELECT
USE CMN_FJX_MOD, ONLY : ACOEF_WV, BCOEF_WV, CCOEF_WV
USE CMN_FJX_MOD, ONLY : ACOEF_RTWV, BCOEF_RTWV, CCOEF_RTWV
USE CMN_FJX_MOD, ONLY : NRVREQUIRED, IWRREQUIRED
USE CMN_FJX_MOD, ONLY : NRTWVREQUIRED, IRTWVREQUIRED
USE Input_Opt_Mod, ONLY : OptInput
#if defined(RRTMG)
USE PARRRTM, ONLY : NBNDLW
#endif
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt
```

#### REMARKS:

Now the user is able to select any 3 wavelengths for optics output in the input.geos file we need to be able to interpolate to those wavelengths based on what is available in the optics look-up table.

The standard lookup table currently has values for 11 common wavelengths followed by 30 that are required by RRTMG. Only those required to interpolate to user requested wavelengths are selected from the standard wavelengths. RRTMG wavelengths are not used in the interpolation for AOD output (DAR 10/2013)

UPDATE: because the RT optics output doesnt have access to the standard wavelengths we now calculate two sets of values: one for the ND21 and diag3 outputs that use the standard wavelengths and one for ND72 that interpolates the optics from RRTMG wavelengths. Perhaps a switch needs adding to switch off the RT optics output (and interpolation) if this ends up costing too much and is not used, but it is ideal to have an optics output that matches exactly what RRTMG uses to calculate the fluxes

#### REVISION HISTORY:

18 Jun 2013 - D. Ridley - Initial version  
 15 Dec 2014 - M. Sulprizio- Added ProTeX headers  
 02 Apr 2015 - C. Keller - Added am\_I\_Root argument  
 23 Mar 2018 - E. Lundgren - Added Input\_Opt argument for wavelength vars  
 read in input.geos now stored in Input\_Opt

---

### 7.3.14 rd\_js\_jx

Subroutine RD\_JS\_JX reads in 'FJX.j2j.dat', which defines the mapping of Fast-JX J's (TITLEJX(1:NJX)) onto the CTM reactions. Reaction number JJ, named T\_REACT, uses Fast-JX's T\_FJX (including scaling factor F\_FJX).

#### INTERFACE:

```

SUBROUTINE RD_JS_JX(NUNIT, NAMFIL, TITLEJX,
& NJXX, am_I_Root, Input_Opt, RC)

```

#### USES:

```

USE Charpak_Mod, ONLY : CStrip
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput

```

#### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: NUNIT
INTEGER, INTENT(IN) :: NJXX
CHARACTER(LEN=*), INTENT(IN) :: NAMFIL
CHARACTER(LEN=6), INTENT(IN), DIMENSION(NJXX) :: TITLEJX
LOGICAL, INTENT(IN) :: am_I_Root
TYPE(OptInput), INTENT(IN) :: Input_Opt

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC

```

#### REMARKS:

Now flag special reactions that are to be adjusted for FlexChem later.

#### REVISION HISTORY:

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0  
 31 Mar 2013 - S. D. Eastham - Modified to track number of branches  
 29 Mar 2016 - R. Yantosca - Now flag rxn numbers for use in PHOTRATE\_ADJ  
 27 Jun 2016 - M. Sulprizio - Add RXN\_NO2 flag for PARANOX  
 10 Mar 2017 - C. Keller - Wrapped print out into am\_I\_Root statement  
 19 Dec 2017 - R. Yantosca - Now return error code to calling routine  
 03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input\_Opt%LUCX

---

### 7.3.15 solar\_jx

Subroutine SOLAR\_JX handles solar zenith angles.

#### INTERFACE:

```
SUBROUTINE SOLAR_JX(NDAY,COSSZA,SZA,SOLFX)
```

#### USES:

```
USE PhysConstants,ONLY : PI
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: COSSZA
```

```
INTEGER, INTENT(IN) :: NDAY
```

!OUTPUT VARIABLES:

```
REAL(fp), INTENT(OUT) :: SZA,SOLFX
```

#### REMARKS:

-----  
NDAY = integer day of the year (used for solar lat and declin)

SZA = solar zenith angle in degrees

COSSZA = U0 = cos(SZA)

SOLFX = Solar function  
-----

#### REVISION HISTORY:

28 Mar 2013 - S. D. Eastham - Adapted from Fast-JX v7.0

06 Jan 2016 - E. Lundgren - Use global PI rather than local definition

### 7.3.16 photo\_jx

Subroutine PHOTO\_JX is the core subroutine of Fast-JX. calc J's for a single column atmosphere (aka Indep Colm Atmos or ICA) needs P, T, O3, clds, aersls; adds top-of-atmos layer from climatology needs day-fo-year for sun distance, SZA (not lat or long)

#### INTERFACE:

```
SUBROUTINE PHOTO_JX(U0, REFLB, P_COL,
& T_COL, O3_COL, O3_TOMS,
& AOD999, ODAER_COL, ODMDUST_COL,
& ODCLOUD_COL_IN, ILON, ILAT,
& YLAT, DAY_OF_YR, MONTH,
& DAY, am_I_Root, Input_Opt,
& State_Diag)
```

#### USES:

```

#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD, ONLY : LD22, LD64, ND22, ND64
 USE DIAG_MOD, ONLY : AD22, AD64, LTJV
#endif
 USE Input_Opt_Mod, ONLY : OptInput
 USE State_Diag_Mod, ONLY : DgnState

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: ILO, ILAT
INTEGER, INTENT(IN) :: DAY_OF_YR
INTEGER, INTENT(IN) :: MONTH, DAY
REAL(fp), INTENT(IN) :: YLAT
REAL(fp), INTENT(IN) :: UO, REFLB
REAL(fp), INTENT(IN), DIMENSION(L1_+1) :: P_COL
REAL(fp), INTENT(IN), DIMENSION(L1_) :: T_COL
REAL(fp), INTENT(IN), DIMENSION(L1_) :: O3_COL
REAL(fp), INTENT(IN) :: O3_TOMS
REAL(fp), INTENT(IN), DIMENSION(L_,A_) :: ODAER_COL
REAL(fp), INTENT(IN), DIMENSION(L_,NDUST) :: ODMDUST_COL
REAL(fp), INTENT(IN), DIMENSION(L_) :: ODCLOUD_COL_IN
LOGICAL, INTENT(IN) :: AOD999
LOGICAL, INTENT(IN) :: am_I_Root
TYPE(OptInput), INTENT(IN) :: Input_Opt

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(DgnState), INTENT(INOUT) :: State_Diag

```

**REMARKS:****REVISION HISTORY:**

```

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0
05 Mar 2014 - R. Yantosca - Now only save ND64 with LD64 specified levels
26 Sep 2017 - M. Sulprizio - Add new halogen species to ND22 (L. Zhu)
04 Oct 2017 - E. Lundgren - Populate State_Diag J-Values array for nc diag
03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input_Opt%LUCX

```

**7.3.17 opmie**

Subroutine OPMIE is a core Fast-JX scattering subroutine, specifically for Mie scattering.

**INTERFACE:**

```

SUBROUTINE OPMIE (DTAUX, POME GAX, UO, RFL, AMF2, JXTRA,
& FJACT, FJTOP, FJBOT, FSBOT, FJFLX, FLXD, FLXDO, LU)

```

**USES:****INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: DTAUX(JXL1_,W_),POMEGAX(8,JXL1_,W_)
REAL(fp), INTENT(IN) :: AMF2(2*JXL1_+1,2*JXL1_+1)
REAL(fp), INTENT(IN) :: U0,RFL(W_)
INTEGER, INTENT(IN) :: JXTRA(JXL2_+1), LU
!OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: FJACT(JXL_,W_),FJTOP(W_)
REAL(fp), INTENT(OUT) :: FJBOT(W_),FSBOT(W_)
REAL(fp), INTENT(OUT) :: FJFLX(JXL_,W_),FLXD(JXL1_,W_),FLXDO(W_)

```

**REMARKS:****REVISION HISTORY:**

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

**7.3.18 miesct**

Subroutine MIESCT is an adaptation of the Prather radiative transfer code (mjp, 10/95).

**INTERFACE:**

```
SUBROUTINE MIESCT(FJ,FJT,FJB, POMECA,FZ,ZTAU,ZFLUX,RFL,UO,ND)
```

**USES:****INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: ND
REAL(fp), INTENT(IN) :: POMECA(M2_,N_,W_),FZ(N_,W_),ZTAU(N_,W_)
&
, RFL(W_),UO,ZFLUX(W_)
!OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: FJ(N_,W_),FJT(W_),FJB(W_)

```

**REMARKS:**

Prather, 1974, *Astrophys. J.* 192, 787-792.  
 Solution of inhomogeneous Rayleigh scattering atmosphere.  
 (original Rayleigh w/ polarization)

Cochran and Trafton, 1978, *Ap.J.*, 219, 756-762.  
 Raman scattering in the atmospheres of the major planets.  
 (first use of anisotropic code)

Jacob, Gottlieb and Prather, 1989, *J.Geophys.Res.*, 94, 12975-13002.



Chemistry of a polluted cloudy boundary layer,  
(documentation of extension to anisotropic scattering)

takes atmospheric structure and source terms from std J-code  
ALSO limited to 4 Gauss points, only calculates mean field! (M=1)

## REVISION HISTORY:

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

### 7.3.19 legnd0

Subroutine LEGND0 calculates ordinary Legendre functions of X (real) from  $P[0] = PL(1) = 1$ ,  $P[1] = X, \dots, P[N - 1] = PL(N)$

#### INTERFACE:

SUBROUTINE LEGND0 (X,PL,N)

#### USES:

#### INPUT PARAMETERS:

INTEGER, INTENT(IN) :: N  
REAL(fp), INTENT(IN) :: X  
!OUTPUT VARIABLES:  
REAL(fp), INTENT(OUT) :: PL(N)

#### REMARKS:

## REVISION HISTORY:

28 Mar 2013 - S. D. Eastham - Copied from Fast-JX v7.0

---

### 7.3.20 set\_prof

Subroutine SET\_PROF sets vertical profiles for a given latitude and longitude.

#### INTERFACE:

SUBROUTINE SET\_PROF (YLAT, MONTH, DAY, T\_CTM, P\_CTM,  
& CLDOD, DSTOD, AEROD, O3\_CTM, O3\_TOMS,  
& AERCOL, T\_CLIM, O3\_CLIM, Z\_CLIM, AIR\_CLIM,  
& Input\_Opt )

**USES:**

```

USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AIRMW, AVO, g0, BOLTZ

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: YLAT ! Latitude (degrees)
INTEGER, INTENT(IN) :: MONTH ! Month
INTEGER, INTENT(IN) :: DAY ! Day *of month*
REAL(fp), INTENT(IN) :: T_CTM(L1_) ! CTM temperatures (K)
REAL(fp), INTENT(IN) :: O3_TOMS ! O3 column (DU)
REAL(fp), INTENT(IN) :: P_CTM(L1_+1) ! CTM edge pressures (hPa)
REAL(fp), INTENT(INOUT) :: CLDOD(L_) ! Cloud optical depth
REAL(fp), INTENT(IN) :: DSTOD(L_,NDUST) ! Mineral dust OD
REAL(fp), INTENT(IN) :: AEROD(L_,A_) ! Aerosol OD
REAL(fp), INTENT(IN) :: O3_CTM(L1_) ! CTM ozone (molec/cm3)
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
!OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: AERCOL(A_,L1_) ! Aerosol column
REAL(fp), INTENT(OUT) :: T_CLIM(L1_) ! Clim. temperatures (K)
REAL(fp), INTENT(OUT) :: Z_CLIM(L1_+1) ! Edge altitudes (cm)
REAL(fp), INTENT(OUT) :: O3_CLIM(L1_) ! O3 column depth (#/cm2)
REAL(fp), INTENT(OUT) :: AIR_CLIM(L1_) ! O3 column depth (#/cm2)

```

**REMARKS:****REVISION HISTORY:**

```

30 Mar 2013 - S. D. Eastham - Adapted from J. Mao code
05 Jan 2016 - E. Lundgren - Use global physical parameters

```

**7.3.21 set\_aer**

Subroutine SET\_AER fills out the array MIEDX. Each entry connects a GEOS-Chem aerosol to its Fast-JX counterpart: MIEDX(Fast-JX index) = (GC index)

**INTERFACE:**

```

SUBROUTINE SET_AER(am_I_Root, Input_Opt)

```

**USES:**

```

USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
!OUTPUT VARIABLES:

```

**REMARKS:****REVISION HISTORY:**

31 Mar 2013 - S. D. Eastham - Adapted from J. Mao FJX v6.2 implementation  
 03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input\_Opt%LUCX

---

**7.3.22 rd\_prof\_nc**

Subroutine RAD\_PROF\_NC reads in the reference climatology from a NetCDF file rather than an ASCII .dat.

**INTERFACE:**

```
SUBROUTINE RD_PROF_NC(am_I_Root, Input_Opt, RC)
```

**USES:**

```
! Modules for netCDF read
USE m_netcdf_io_open
USE m_netcdf_io_read
USE m_netcdf_io_readattr
USE m_netcdf_io_close
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:**

This file was automatically generated by the Perl scripts in the NcdfUtilities package (which ships w/ GEOS-Chem) and was subsequently hand-edited.

**REVISION HISTORY:**

19 Apr 2012 - R. Yantosca - Initial version  
 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F  
 10 Apr 2013 - S. D. Eastham - Adapted for Fast-JX v7.0  
 20 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC  
 13 Mar 2015 - R. Yantosca - Replace DATA\_DIR\_1x1 w/ CHEM\_INPUTS\_DIR

---

### 7.3.23 photrate\_adj

Subroutine PHOTRATE\_ADJ adjusts certain photolysis rates for chemistry.

#### INTERFACE:

```

 SUBROUTINE PHOTRATE_ADJ(am_I_root, Input_Opt, State_Diag,
& I, J, L,
& NUMDEN, TEMP, C_H2O,
& FRAC, RC
)

```

#### USES:

```

#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD, ONLY : LD22, ND22
 USE DIAG_MOD, ONLY : AD22, LTJV
#endif
 USE ErrCode_Mod
 USE Input_Opt_Mod, ONLY : OptInput
 USE State_Diag_Mod, ONLY : DgnState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input_Options object
INTEGER, INTENT(IN) :: I, J, L ! Lon, lat, lev indices
REAL(fp), INTENT(IN) :: NUMDEN ! Air # density [molec/m3]
REAL(fp), INTENT(IN) :: TEMP ! Temperature [K]
REAL(fp), INTENT(IN) :: C_H2O ! H2O conc [molec/cm3]
REAL(fp), INTENT(IN) :: FRAC ! Result of SO4_PHOTFRAC,
 ! called from DO_FLEXCHEM

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure

```

#### REMARKS:

NOTE: The netCDF diagnostics are attached in DO\_FLEXCHEM so that we have access to the adjusted rates. Only the bpch diagnostics are updated here.

-- Bob Yantosca, 19 Dec 2017

%%% NOTE: WE SHOULD UPDATE THE COMMENTS TO MAKE SURE THAT WE DO %%%  
 %%% NOT KEEP ANY CONFLICTING OR INCORRECT INFORMATION (bmy, 3/28/16) %%%

#### REVISION HISTORY:

10 Mar 2016 - M. Sulprizio- Added ProTeX header  
 18 Apr 2016 - R. Yantosca - Remove Input\_Opt, State\_Met, State\_Chm args  
 and pass air density & temperature instead  
 18 Apr 2016 - R. Yantosca - Also prevent div-by-zero conditions  
 16 Mar 2017 - M. Sulprizio- Restore P(OH) from O3 photolysis for ND22  
 04 Oct 2017 - E. Lundgren - Pass State\_Diag and populate J-Values array  
 19 Dec 2017 - R. Yantosca - Now use Input\_Opt%LUCX instead of ifdefs

---

#### 7.4 Fortran: Module Interface uvalbedo\_mod.F90

Module UVALBEDO\_MOD contains variables and routines for reading the UV Albedo data. This data is required by the FAST-JX photolysis module. UV albedo data will now be obtained from the HEMCO data structure.

#### INTERFACE:

```
MODULE UValbedo_Mod
```

#### USES:

```
USE Precision_Mod ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: Get_UValbedo
```

#### REMARKS:

References:

```
=====
Herman, J.R and Celarier, E.A., "Earth surface reflectivity climatology
at 340-380 nm from TOMS data", __J. Geophys. Res__, Vol. 102, D23,
pp. 28003-28011, Dec 20, 1997.
```

#### REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Initial version  
 19 Apr 2002 - R. Yantosca - Initial version  
 (1 ) Now read uvalbedo file directly from DATA\_DIR/uvalbedo\_200111  
 subdirectory. (bmy, 4/2/02)  
 (2 ) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and  
 MODULE ROUTINES sections. (bmy, 5/28/02)  
 (3 ) Now references "error\_mod.f" (bmy, 10/15/02)  
 (4 ) Minor modification in READ\_UVALBEDO (bmy, 3/14/03)  
 (5 ) Now references "directory\_mod.f" (bmy, 7/20/04)  
 (6 ) Bug fix for GCAP grid in READ\_UVALBEDO (bmy, 8/16/05)  
 (7 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

24 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte  
12 Jan 2015 - R. Yantosca - Remove CLEANUP\_UVALBEDO routine  
04 Mar 2015 - R. Yantosca - UV albedo data now comes via HEMCO

---

#### 7.4.1 get\_uvalbedo

Copies the UV Albedo data from the HEMCO data structure into the State\_Met derived type object.

##### INTERFACE:

```
SUBROUTINE Get_UValbedo(am_I_root, Input_Opt, State_Met, RC)
```

##### USES:

```
USE ErrCode_Mod
USE Error_Mod, ONLY : Error_Stop
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
```

##### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

##### INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

##### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

##### REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Initial version  
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

---

## 7.5 Fortran: Module Interface *set\_global\_ch4\_mod.F90*

Module SET\_GLOBAL\_CH4 contains variables and routines for reading and applying surface CH4 concentrations from NOAA GMD data (provided by Lee Murray)

##### INTERFACE:

```
MODULE Set_Global_CH4_Mod
```

**USES:**

```
USE PRECISION_MOD
```

```
IMPLICIT NONE
```

**PUBLIC MEMBER FUNCTIONS:**

```
PUBLIC :: Set_CH4
```

```
PUBLIC :: Cleanup_Set_Global_CH4
```

**REVISION HISTORY:**

```
18 Jan 2018 - M. Sulprizio- Initial version
```

---

**7.5.1 set\_ch4**

Subroutine SET\_CH4 copies monthly mean surface CH4 from HEMCO and applies it to CH4 concentrations in State\_Chm

**INTERFACE:**

```
SUBROUTINE Set_CH4(am_I_Root, Input_Opt, State_Met, State_Chm, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
```

```
USE ErrCode_Mod
```

```
USE ERROR_MOD
```

```
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
```

```
USE HCO_Error_Mod
```

```
USE HCO_INTERFACE_MOD, ONLY : HcoState
```

```
USE Input_Opt_Mod, ONLY : OptInput
```

```
USE PBL_MIX_MOD, ONLY : GET_PBL_TOP_L
```

```
USE State_Chm_Mod, ONLY : ChmState, Ind_
```

```
USE State_Met_Mod, ONLY : MetState
```

```
USE UnitConv_Mod, ONLY : Convert_Spc_Units
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
```

```
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

```
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:**

Lee Murray wrote:

The monthly mean surface methane mixing ratios used here were spatially interpolated from NOAA GLOBALVIEW discrete monthly mean flask data using ordinary kriging for 1983-2016 (ch4\_flask\_surface\_2017-07-28.tar.gz). Surface mixing ratios are extended back to 1979 and forward to 2020 via linear extrapolation of the local 1983-1990 and 2011-2016 trends, respectively.

**REVISION HISTORY:**

18 Jan 2018 - M. Sulprizio- Initial version

---

**7.5.2 cleanup\_set\_global\_ch4**

Subroutine CLEANUP\_SET\_GLOBAL\_CH4 deallocates memory from previously allocated module arrays.

**INTERFACE:**

SUBROUTINE Cleanup\_Set\_Global\_CH4

**REVISION HISTORY:**

18 Jan 2018 - M. Sulprizio- Initial version

---

**7.5.3 set\_prof\_o3**

Subroutine SET\_PROF\_O3 sets up atmospheric profiles required by RRTMG in the stratosphere using a doubled version of the level scheme used in the CTM. First pressure and z\* altitude are defined, then O3 and T are taken from the supplied climatology and integrated to the CTM levels (may be overwritten with values directly from the CTM, if desired). This is a stripped down version of SET\_PROF; it does O3 only.

**INTERFACE:**

```

SUBROUTINE SET_PROF_O3(YLAT, MONTH, DAY, T_CTM,
& P_CTM, O3_CTM, O3_TOMS, T_CLIM,
& O3_CLIM, Z_CLIM, AIR_CLIM, Input_Opt)

```

**USES:**

```

USE CMN_FJX_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants ! Physical constants
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)

```

```

IMPLICIT NONE

```



**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: YLAT ! Latitude (degrees)
INTEGER, INTENT(IN) :: MONTH ! Month
INTEGER, INTENT(IN) :: DAY ! Day *of month*
REAL(fp), INTENT(IN) :: T_CTM(L1_) ! CTM temperatures (K)
REAL(fp), INTENT(IN) :: O3_TOMS ! O3 column (DU)
REAL(fp), INTENT(IN) :: P_CTM(L1_+1) ! CTM edge pressures (hPa)
REAL(fp), INTENT(IN) :: O3_CTM(L1_) ! CTM ozone (molec/cm3)
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input options
!OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: T_CLIM(L1_) ! Clim. temperatures (K)
REAL(fp), INTENT(OUT) :: Z_CLIM(L1_+1) ! Edge altitudes (cm)
REAL(fp), INTENT(OUT) :: O3_CLIM(L1_) ! O3 column depth (#/cm2)
REAL(fp), INTENT(OUT) :: AIR_CLIM(L1_) ! O3 column depth (#/cm2)

```

**AUTHOR:**

Oliver Wild & Michael Prather

**REMARKS:****REVISION HISTORY:**

```

01 Jun 1996 - M. Prather & O. Wild - Initial version
(1) Since we parallelize over columns, T, ODCOL, OPTDUST, and OPTAER
 are 1-D vectors. In the original code from Oliver Wild, these were
 3-D arrays. Also P and SA are just scalars since we just pass one
 surface location at a time w/in the parallel loop. (bmy, 9/13/99)
(2) Mineral dust profiles are also constructed (rvm, 06/04/00)
(3) Other aerosol profiles are also constructed (rvm, bmy, 2/27/02)
(4) Added NLON, NLAT, DAY to the arg list. Now weight the O3 column by
 the observed monthly mean EP-TOMS data. Also updated comments and
 added standard GEOS-CHEM documentation header. (mje, bmy, 7/13/03)
(5) We don't need to initialize the PJ array with ETAA and ETAB anymore.
 PJ is now defined in "fast_j.f". Updated comments. (bmy, 10/30/07)
(6) Modified to use GEOS-5 O3 columns when TOMS/SBUV data don't exist,
 i.e. after 2008. (ccc, 7/13/09)
08 Dec 2009 - R. Yantosca - Added ProTeX headers
15 Dec 2014 - M. Sulprizio- Modified for Fast-JX. Now consistent with
 subroutine SET_PROF in fast_jx_mod.F.
20 Mar 2015 - M. Sulprizio- Modified for flexible precision
05 Jan 2016 - E. Lundgren - Use global physical parameters

```

**7.6 Fortran: Module Interface toms\_mod.F**

Module TOMS\_MOD contains variables and routines for reading the TOMS/SBUV O3 column data from disk (for use w/ the FAST-J photolysis routines).

**INTERFACE:**

MODULE TOMS\_MOD

**USES:**USE CMN\_SIZE\_MOD       ! Size parameters  
USE PRECISION\_MOD     ! For GEOS-Chem Precision (fp)IMPLICIT NONE  
PRIVATE**PUBLIC MEMBER FUNCTIONS:**PUBLIC :: INIT\_TOMS  
PUBLIC :: READ\_TOMS  
PUBLIC :: COMPUTE\_OVERHEAD\_O3  
PUBLIC :: GET\_OVERHEAD\_O3  
PUBLIC :: CLEANUP\_TOMS**PUBLIC DATA MEMBERS:**! First & last years for which TOMS/SBUV data is available  
! (update these as new data is added to the archive)  
INTEGER, PUBLIC, PARAMETER :: FIRST\_TOMS\_YEAR = 1979  
INTEGER, PUBLIC, PARAMETER :: LAST\_TOMS\_YEAR = 2010**REMARKS:**

References:

=====  
Version 8 Merged Ozone Data Sets  
Total Ozone Revision 05  
DATA THROUGH: MAR 2009  
LAST MODIFIED: 01 MAY 2009[http://acdb-ext.gsfc.nasa.gov/Data\\_services/merged/index.html](http://acdb-ext.gsfc.nasa.gov/Data_services/merged/index.html)TOMS/SBUV MERGED TOTAL OZONE DATA, Version 8, Revision 5.  
Resolution: 5 x 10 deg.

- \* Includes reprocessed N16 and N17 SBUV/2 data using latest calibration.
- \* OMI data updated from Collection 2 to Collection 3.
- \* New offsets derived based on revised data sets.
- \* 1970-1972 N4 BUV data added with no adjustments. User may wish to apply offset based on Comparisons between BUV and Dobson Measurements.

Responsible NASA official:

Dr. Richard Stolarski (Richard.S.Stolarski@nasa.gov)  
Stacey Frith                   (Stacey.M.Frith@nasa.gov     )

**REVISION HISTORY:**

14 Jul 2003 - R. Yantosca - Initial version  
 (1 ) Now references "directory\_mod.f" (bmy, 7/20/04)  
 (2 ) Now can read files for GEOS or GCAP grids (bmy, 8/16/05)  
 (3 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 (4 ) Now always use 2002 TOMS O3 data for GCAP (swu, bmy, 10/3/06)  
 (5 ) Now reads from TOMS\_200701 directory, w/ updated data (bmy, 2/1/07)  
 (6 ) Now don't replace any tokens in the DATA\_DIR variable (bmy, 12/5/07)  
 (7 ) Latest year of TOMS data is now 2007 (bmy, 1/14/09)  
 01 Dec 2010 - R. Yantosca - Added ProTeX headers  
 06 Mar 2012 - R. Yantosca - Added function GET\_TOTAL\_O3  
 06 Mar 2012 - R. Yantosca - Added parameters FIRST\_TOMS\_YEAR, LAST\_TOMS\_YEAR  
 06 Mar 2012 - R. Yantosca - Updated comments  
 06 Mar 2012 - R. Yantosca - Now make TOMS, DTOMS1, DTOMS2 arrays PRIVATE  
 06 Mar 2012 - R. Yantosca - Add TO3\_DAILY as a PRIVATE module array  
 25 Jun 2012 - S. Kim - Now reads from TOMS\_201203 directory, w/  
                   updated data  
 03 Jul 2012 - R. Yantosca - Restrict reading from TOMS\_201203 directory  
                   to GEOS-5.7.2 met data for the time being.  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 21 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 16 Mar 2015 - R. Yantosca - Retire bpch input; now read TOMS via HEMCO  
 03 Nov 2016 - B. Henderson- Added fix for weird O3 in GEOS-5 simulations

**7.6.1 read\_toms**

Subroutine READ\_TOMS reads in TOMS O3 column data from a binary punch file for the given grid, month and year.

**INTERFACE:**

```
SUBROUTINE READ_TOMS(am_I_Root, Input_Opt, RC)
```

**USES:**

```
USE ErrCode_Mod
USE Error_Mod, ONLY : Error_Stop
USE HCO_Interface_Mod, ONLY : HcoState
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:**

TOMS/SBUV MERGED TOTAL OZONE DATA, Version 8, Revision 5.  
Resolution: 5 x 10 deg.

## Methodology

-----  
FAST-J comes with its own default O3 column climatology (from McPeters 1992 & Nagatani 1991), which is stored in the input file "jv\_atms.dat". These "FAST-J default" O3 columns are used in the computation of the actinic flux and other optical quantities for the FAST-J photolysis.

The TOMS/SBUV O3 columns and 1/2-monthly O3 trends (contained in the TOMS\_200906 directory) are read into GEOS-Chem by routine READ\_TOMS in "toms\_mod.f". Missing values (i.e. locations where there are no data) in the TOMS/SBUV O3 columns are defined by the flag -999.

After being read from disk in routine READ\_TOMS, the TOMS/SBUV O3 data are then passed to the FAST-J routine "set\_prof.F". In "set\_prof.F", a test is done to make sure that the TOMS/SBUV O3 columns and 1/2-monthly trends do not have any missing values for (lat,lon) location for the given month. If so, then the TOMS/SBUV O3 column data is interpolated to the current day and is used to weight the "FAST-J default" O3 column. This essentially "forces" the "FAST-J default" O3 column values to better match the observations, as defined by TOMS/SBUV.

If there are no TOMS/SBUV O3 columns (and 1/2-monthly trends) at a (lat, lon) location for given month, then FAST-J will revert to its own "default" climatology for that location and month. Therefore, the TOMS O3 can be thought of as an "overlay" data -- it is only used if it exists.

Note that there are no TOMS/SBUV O3 columns at the higher latitudes. At these latitudes, the code will revert to using the "FAST-J default" O3 columns.

As of March 2012, we have TOMS/SBUV data for 1979 thru 2008. We will update to the latest TOMS/SBUV data set shortly.

This methodology was originally adopted by Mat Evans.

```
#####
NOTE: BINARY PUNCH INPUT IS BEING PHASED OUT. THIS DATA
WILL EVENTUALLY BE READ IN FROM netCDF FILES VIA HEMCO!
-- Bob Yantosca (05 Mar 2015)
#####
```

**REVISION HISTORY:**

10 Dec 2002 - M. Evans - Initial version

- (1 ) Bundled into "toms\_mod.f" (bmy, 7/14/03)
- (2 ) Now references DATA\_DIR from "directory\_mod.f" (bmy, 7/20/04)
- (3 ) Now can read files for GEOS or GCAP grids (bmy, 8/16/05)
- (4 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (5 ) Now always use 2002 TOMS O3 data for GCAP (swu, bmy, 10/3/06)
- (6 ) Now reads from TOMS\_200701 directory, w/ updated data. Also always use 1979 data prior to 1979 or 2005 data after 2005. (bmy, 2/12/07)
- (7 ) Bug fix: don't include DATA\_DIR in filename, just in case someone's file path has replaceable tokens (e.g. hh, mm, MM etc.) (bmy, 12/5/07)
- (8 ) Latest year of TOMS data is now 2007 (bmy, 1/14/09)
- (9 ) Updated TOMS data in TOMS\_200906. Latest year is 2008. (ccc, 6/15/09)
- 08 Dec 2009 - R. Yantosca - Added ProTeX headers
- 03 Jul 2012 - R. Yantosca - Restrict reading from TOMS\_201203 directory to GEOS-5.7.2 met data for the time being.
- 13 Dec 2013 - M. Sulprizio- Now pass USE\_O3\_FROM\_MET logical flag so that we bypass reading TOMS O3 data when using T03 from the met fields.
- 05 Mar 2014 - M. Sulprizio- Now regrid TOMS O3 from the files on GEOS 1x1 grid (J. Lin)
- 20 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC
- 23 Jun 2014 - R. Yantosca - Now use Input\_Opt%DATA\_DIR\_1x1
- 06 Nov 2014 - R. Yantosca - Replace TRANSFER\_2D with direct casts
- 17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte
- 05 Mar 2015 - R. Yantosca - Now read data w/r/t ExtData/CHEM\_INPUTS

## 7.6.2 compute\_overhead\_o3

Subroutine COMPUTE\_OVERHEAD\_O3 returns the resultant total overhead O3 column for the FAST-J photolysis. This will be one of two options:

1. Default: TOMS/SBUV overhead O3 columns. These will be used by the FAST-J routine set\_prof.F to overwrite the existing FAST-J climatology (cf McPeters & Nagatani 1992). Missing data (i.e. for months & locations where TOMS/SBUV data does not exist) is denoted by the value -999; FAST-J will skip over these points.
2. Overhead O3 columns taken directly from the met fields. These will be returned if the flag USE\_O3\_FROM\_MET is set to TRUE.

### INTERFACE:

```

SUBROUTINE COMPUTE_OVERHEAD_O3(am_I_Root, DAY,
& USE_O3_FROM_MET, T03)

```

### INPUT PARAMETERS:

```

! Root CPU?
LOGICAL, INTENT(IN) :: am_I_Root

```

```
! Day of month
INTEGER, INTENT(IN) :: DAY

! Switch to denote if we should use the default T03
! directly from the met fields
LOGICAL, INTENT(IN) :: USE_O3_FROM_MET

! T03 from the met fields [Dobsons]
REAL(fp), INTENT(IN) :: T03(IIPAR,JJPARG)
```

**REMARKS:**

Reference for the TOMS/SBUV merged O3 columns:

1985 - 2005 are taken from:

[http://code916.gsfc.nasa.gov/Data\\_services/merged/index.html](http://code916.gsfc.nasa.gov/Data_services/merged/index.html)

TOMS/SBUV MERGED TOTAL OZONE DATA, Version 8, Revision 3.  
Resolution: 5 x 10 deg.

Contact person for the merged data product:  
Stacey Hollandsworth Frith ([smh@hyperion.gsfc.nasa.gov](mailto:smh@hyperion.gsfc.nasa.gov))

2006 and 2007 are taken from:

[http://code916.gsfc.nasa.gov/Data\\_services/merged/index.html](http://code916.gsfc.nasa.gov/Data_services/merged/index.html)

Version 8 Merged Ozone Data Sets  
Revision 04  
DATA THROUGH: SEP 2008  
LAST MODIFIED: 20 OCT 2008

Methodology (bmy, 2/12/07)

-----  
FAST-J comes with its own default O3 column climatology (from McPeters 1992 & Nagatani 1991), which is stored in the input file "jv\_atms.dat". These "FAST-J default" O3 columns are used in the computation of the actinic flux and other optical quantities for the FAST-J photolysis.

The TOMS/SBUV O3 columns and 1/2-monthly O3 trends (contained in the TOMS\_200701 directory) are read into GEOS-Chem by routine READ\_TOMS in "toms\_mod.f". Missing values (i.e. locations where there are no data) in the TOMS/SBUV O3 columns are defined by the flag -999.

After being read from disk in routine READ\_TOMS, the TOMS/SBUV O3 data are then passed to the FAST-J routine "set\_prof.f". In

"set\_prof.f", a test is done to make sure that the TOMS/SBUV O3 columns and 1/2-monthly trends do not have any missing values for (lat,lon) location for the given month. If so, then the TOMS/SBUV O3 column data is interpolated to the current day and is used to weight the "FAST-J default" O3 column. This essentially "forces" the "FAST-J default" O3 column values to better match the observations, as defined by TOMS/SBUV.

If there are no TOMS/SBUV O3 columns (and 1/2-monthly trends) at a (lat,lon) location for given month, then FAST-J will revert to its own "default" climatology for that location and month. Therefore, the TOMS O3 can be thought of as an "overlay" data -- it is only used if it exists.

Note that there are no TOMS/SBUV O3 columns at the higher latitudes. At these latitudes, the code will revert to using the "FAST-J default" O3 columns.

As of February 2007, we have TOMS/SBUV data for 1979 thru 2005. 2006 TOMS/SBUV data is incomplete as of this writing. For years 2006 and onward, we use 2005 TOMS O3 columns.

This methodology was originally adopted by Mat Evans. Symeon Koumoutsaris was responsible for creating the downloading and processing the TOMS O3 data files from 1979 thru 2005 in the TOMS\_200701 directory.

## REVISION HISTORY:

06 Mar 2012 - R. Yantosca - Initial version, pulled code out from the FAST-J routine SET\_PROF; based on the GEOS-Chem column code routine  
 22 Oct 2014 - C. Keller - Added am\_I\_Root parameter.

### 7.6.3 get\_overhead\_o3

Function GET\_OVERHEAD\_O3 returns the total overhead O3 column [DU] (which is taken either from TOMS/SBUV or directly from the met fields) at a given surface grid box location (I,J).

#### INTERFACE:

```
FUNCTION GET_OVERHEAD_O3(I, J) RESULT(OVERHEAD_O3)
```

#### INPUT PARAMETERS:

```
INTEGER :: I ! Grid box longitude index
INTEGER :: J ! Grid box latitude index
```

**RETURN VALUE:**

```
REAL(fp) :: OVERHEAD_03 ! Total overhead 03 column [DU]
```

**REVISION HISTORY:**

```
06 Mar 2012 - R. Yantosca - Initial version
```

---

**7.6.4 init\_toms**

Subroutine INIT\_TOMS allocates and zeroes all module arrays.

**INTERFACE:**

```
SUBROUTINE INIT_TOMS(am_I_Root, Input_Opt, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input opts
```

**INPUT/OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Failure or success
```

**REVISION HISTORY:**

```
14 Jul 2003 - R. Yantosca - Initial version
01 Dec 2010 - R. Yantosca - Added ProTeX headers
06 Mar 2012 - R. Yantosca - Now allocate TO3_DAILY
16 Mar 2015 - R. Yantosca - TOMS, DTOMS1, DTOMS2 are now pointers
16 Mar 2015 - R. Yantosca - Add Input_Opt, RC arguments for std interface
03 Nov 2016 - B. Henderson - Add extra fields to fix OH issue for GEOS-5
```

---

**7.6.5 cleanup\_toms**

Subroutine CLEANUP\_TOMS deallocates all module arrays.

**INTERFACE:**



SUBROUTINE CLEANUP\_TOMS

## REVISION HISTORY:

14 Jul 2003 - R. Yantosca - Initial version  
01 Dec 2010 - R. Yantosca - Added ProTeX headers  
06 Mar 2012 - R. Yantosca - Now deallocate T03\_DAILY  
03 Nov 2016 - R. Yantosca - Deallocate and nullify extra fields

---

## 7.7 Fortran: Module Interface flexchem\_mod.F90

Module FlexChem\_Mod contains arrays and routines for the FlexChem chemical solver.

### INTERFACE:

MODULE FlexChem\_Mod

### USES:

USE Precision\_Mod                   ! For GEOS-Chem Precision (fp)

IMPLICIT NONE

PRIVATE

### PUBLIC MEMBER FUNCTIONS:

PUBLIC    :: Do\_FlexChem  
PUBLIC    :: Init\_FlexChem  
PUBLIC    :: Cleanup\_FlexChem

### PRIVATE MEMBER FUNCTIONS:

PRIVATE  :: Diag\_OH\_HO2\_01D\_03P

## REVISION HISTORY:

14 Dec 2015 - M.S. Long - Initial version  
15 Jun 2016 - M. Sulprizio- Remove STTTOCSPEC mapping array. Species and tracers have a 1:1 mapping currently so mapping is not required  
18 Jul 2016 - M. Sulprizio- Remove FAMILIES\_KLUDGE routine. Family tracers have been eliminated.  
24 Aug 2016 - M. Sulprizio- Rename from flexchem\_setup\_mod.F90 to flexchem\_mod.F90  
29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90  
17 Nov 2017 - R. Yantosca - Now call Diag\_OH\_HO2\_01D\_03P, which will let us remove arrays in CMN\_03\_SIZE\_mod.F  
29 Dec 2017 - C. Keller - Make HSAVE\_KPP public (needed for GEOS-5 restart)  
24 Jan 2018 - E. Lundgren - Pass error handling up if RC is GC\_FAILURE

---

### 7.7.1 do\_flexchem

Subroutine Do.FlexChem is the driver subroutine for full chemistry with KPP.

#### INTERFACE:

```

SUBROUTINE Do_FlexChem(am_I_Root, Input_Opt, State_Met, &
 State_Chm, State_Diag, RC)

```

#### USES:

```

USE AEROSOL_MOD, ONLY : SOILDUST, AEROSOL_CONC, RDAER
USE CMN_FJX_MOD
USE CMN_SIZE_MOD, ONLY : IIPAR, JJPARG, LLPARG
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD, ONLY : ND52
USE DIAG_MOD, ONLY : AD65, AD52, ad22
USE DIAG20_MOD, ONLY : DIAG20, POx, LOx
#endif
USE DIAG_OH_MOD, ONLY : DO_DIAG_OH
USE DUST_MOD, ONLY : RDUST_ONLINE, RDUST_OFFLINE
USE ErrCode_Mod
USE ERROR_MOD
USE FAST_JX_MOD, ONLY : PHOTRATE_ADJ, FAST_JX
USE GCKPP_HetRates, ONLY : SET_HET
USE GCKPP_Monitor, ONLY : SPC_NAMES, FAM_NAMES
USE GCKPP_Parameters
USE GCKPP_Integrator, ONLY : INTEGRATE, NHnew
USE GCKPP_Function
USE GCKPP_Model
USE GCKPP_Global
USE GCKPP_Rates, ONLY : UPDATE_RCONST, RCONST
USE GCKPP_Initialize, ONLY : Init_KPP => Initialize
USE GC_GRID_MOD, ONLY : GET_YMID
USE GEOS_Timers_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE PRESSURE_MOD
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE Strat_Chem_Mod, ONLY : SChem_Tend
USE TIME_MOD, ONLY : GET_TS_CHEM
USE TIME_MOD, ONLY : Get_Day
USE TIME_MOD, ONLY : Get_Month
USE TIME_MOD, ONLY : Get_Year
USE UnitConv_Mod, ONLY : Convert_Spc_Units
USE UCX_MOD, ONLY : CALC_STRAT_AER

```

```

 USE UCX_MOD, ONLY : SO4_PHOTFRAC
 USE UCX_MOD, ONLY : UCX_NOX
 USE UCX_MOD, ONLY : UCX_H2SO4PHOT
#if defined(TOMAS)
 USE TOMAS_MOD, ONLY : H2SO4_RATE
#endif

```

**INPUT PARAMETERS:**

```

 LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

 TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
 TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
 TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

 INTEGER, INTENT(OUT) :: RC ! Success or failure

```

**REVISION HISTORY:**

```

14 Dec 2015 - M.S. Long - Initial version
18 Dec 2015 - M. Sulprizio- Add calls to OHSAVE and DO_DIAG_OH
22 Dec 2015 - M. Sulprizio- Set NUMDEN to State_Met%AIRNUMDEN for conversions
 between v/v and molec/cm3
28 Mar 2016 - R. Yantosca - Added several fixes for OpenMP parallelization
30 Mar 2016 - R. Yantosca - Bug fix, now make sure to copy C back into
 State_Chm%Species. Also block out temp diags.
16 May 2016 - M. Sulprizio- Remove call to RURALBOX. The implementation of
 FlexChem has rendered the routine obsolete.
31 May 2016 - E. Lundgren - Use species database MW instead of XNUMOL
06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies and
 NAMEGAS with SpcInfo%Name from species database
14 Jun 2016 - M. Sulprizio- Replace loops over N_TRACERS with loops over
 State_Chm%nSpecies and add checks for Is_Advected
 and Is_Kpp to avoid introducing numerical noise
 by applying unit conversions to non-KPP species
16 Jun 2016 - M. Sulprizio- Now define IDTCH4 locally
20 Jun 2016 - R. Yantosca - Renamed IDTCH4 to id_CH4 for consistency
18 Jul 2016 - M. Sulprizio- Remove calls to FAMILIES_KLUDGE. Family tracers
 have been eliminated. Also simplify code to copy
 to/from STT so that unit conversions to/from
 molec/cm3 are done in the same step as the copy.
02 Aug 2016 - M. Sulprizio- Connect production and loss rates from KPP to
 ND65 diagnostic
16 Aug 2016 - E. Lundgren - Remove all references to tracers, including
 STT and Input_Opt%N_TRACERS, and use routines in

```

```

 unitconv_mod.F for species kg <-> molec/cm3
24 Aug 2016 - M. Sulprizio- Replace CSPECTOKPP with State_Chm%Map_KppSpc
24 Aug 2016 - M. Sulprizio- Move this subroutine to flexchem_mod.F90 and
 rename from FLEX_CHEMDR to Do_FlexChem
22 Sep 2016 - R. Yantosca - Add extra debug printout after FAST_JX
14 Nov 2016 - E. Lundgren - Move UCX calls to after spc conversion to kg
10 Mar 2017 - C. Keller - Make sure ind_CH4 is correctly specified in
 ESMF environment.
30 May 2017 - M. Sulprizio- Add code for stratospheric chemical tendency
 for computing STE in strat_chem_mod.F90
28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine
03 Oct 2017 - E. Lundgren - Pass State_Diag as argument
21 Dec 2017 - R. Yantosca - Add netCDF diagnostics for J-values, prod/loss
03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input_Opt%LUCX
18 Jan 2018 - R. Yantosca - Now do photolysis for all levels, so that
 J-values can be saved up to the atm top

```

## 7.7.2 Diag\_OH\_HO2\_O1D\_O3P

Archives the chemical production of OH, HO2, O1D, O3P.

### INTERFACE:

```

SUBROUTINE Diag_OH_HO2_O1D_O3P(am_I_Root, Input_Opt, State_Met, &
 State_Chm, State_Diag, RC)

```

### USES:

```

USE CMN_SIZE_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
#if defined(BPCH_DIAG)
USE Diag_Mod, ONLY : AD43
USE Diag_Mod, ONLY : LTOH, LTHO2, LT01D, LT03P
#endif

```

### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

### INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

**REMARKS:**

This routine replaces both OHSAVE and DIAGOH. Those routines were needed for SMVGEAR, when we had separate arrays for the non-advected species. But now, all species are stored in State\_Chm%SPECIES, so the various arrays (SAVEOH, SAVEH02, etc.) are no longer necessary. We can now just get values directly from State\_Chm%SPECIES.

Also note: for the netCDF diagnostics, we have removed multiplication by LTOH etc arrays. These are almost always set between 0 and 24.

**REVISION HISTORY:**

06 Jan 2015 - R. Yantosca - Initial version

---

**7.7.3 init\_flexchem**

Subroutine Init\_FlexChem is used to allocate arrays for the KPP solver.

**INTERFACE:**

```
SUBROUTINE Init_FlexChem(am_I_Root, Input_Opt, State_Chm, State_Diag, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Gckpp_Global, ONLY : nReact
USE Gckpp_Monitor, ONLY : Eqn_Names, Fam_Names
USE Gckpp_Parameters, ONLY : nFam
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Diagnostics State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

14 Dec 2015 - M.S. Long - Initial version  
 22 Dec 2015 - M. Sulprizio- Use State\_Met%AIRNUMDEN to convert initial species concentrations from v/v to molec/cm3  
 29 Jan 2016 - M. Sulprizio- Add calls to Register\_Tracer and Register\_Species to populate Tracer\_Name, Tracer\_Id, Species\_Name, and Species\_ID fields in State\_Chm  
 06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State\_Chm%nSpecies and NAMEGAS with SpcInfo%Name from species database  
 06 Jun 2016 - M. Sulprizio- Replace Get\_Indx with Spc\_GetIndx to use the fast-species lookup from the species database  
 06 Jun 2016 - M. Sulprizio- Remove calls to Register\_Tracer and Register\_Species; these routines were made obsolete by the species database  
 14 Jun 2016 - M. Sulprizio- Replace Spc\_GetIndx with Ind\_ (M. Long)  
 25 Jul 2016 - E. Lundgren - Add check that species was not in restart file prior to v/v -> molec/cm3 conversion  
 02 Aug 2016 - E. Lundgren - Move unit conversion of species background values to restart\_mod  
 24 Aug 2016 - M. Sulprizio- Remove CSPECTOKPP array. State\_Chm%Map\_KppSpc is now used instead.  
 20 Sep 2016 - R. Yantosca - Use fixed integer with in WRITE statement  
 03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

#### 7.7.4 cleanup\_flexchem

Subroutine Cleanup\_FlexChem deallocate module variables.

#### INTERFACE:

```
SUBROUTINE Cleanup_FlexChem(am_I_Root, RC)
```

#### USES:

```
USE ErrCode_Mod
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REVISION HISTORY:

24 Aug 2016 - M. Sulprizio- Initial version

## 7.8 Fortran: Module Interface gckpp\_HetRates

FlexChem module for heterogeneous chemistry, via KPP.

### INTERFACE:

```
MODULE GCKPP_HETRATES
```

### USES:

```
USE CMN_FJX_MOD, ONLY : NDUST
USE CMN_FJX_MOD, ONLY : NAER
USE CMN_SIZE_MOD, ONLY : LLSTRAT
USE ERROR_MOD, ONLY : ERROR_STOP
USE ERROR_MOD, ONLY : GEOS_CHEM_STOP
USE ERROR_MOD, ONLY : IS_SAFE_DIV, SAFE_DIV
USE gckpp_Precision
USE gckpp_Parameters
USE gckpp_Global, ONLY : HET
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO, RGASLATM, CONSVAP, RSTARG, PI
USE Precision_Mod, ONLY : fp
```

```
IMPLICIT NONE
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: SET_HET
```

### PRIVATE MEMBER FUNCTIONS:

```
! These functions are used for all mechanisms
PRIVATE :: HetNO3
PRIVATE :: HetNO2
PRIVATE :: HetHO2
PRIVATE :: HetGLYX
PRIVATE :: HetMGLY
PRIVATE :: HetIEPOX
PRIVATE :: HetIMAE
PRIVATE :: HetLVOC
PRIVATE :: HetISOPND
PRIVATE :: HetISOPNB
PRIVATE :: HetMACRN
PRIVATE :: HetMVKN
PRIVATE :: HetR4N2
PRIVATE :: HetISN10G
PRIVATE :: HetDHDN
```

```
PRIVATE :: HetMONITS
PRIVATE :: HetMONITU
PRIVATE :: HetHONIT
PRIVATE :: HetIONITA
PRIVATE :: HetMONITA
PRIVATE :: HetHBr
PRIVATE :: HetN2O5
PRIVATE :: N2O5
PRIVATE :: HO2
```

```
! New iodine heterogeneous chemistry
```

```
PRIVATE :: HETIUptake
PRIVATE :: HETIXCycleSSA
```

```
! These are the new Br/Cl functions from J. Schmidt
```

```
PRIVATE :: HETBrNO3_JS
PRIVATE :: HETClNO3_JS
PRIVATE :: HETHOBr_HBr_JS
PRIVATE :: HETHOBr_HCl_JS
PRIVATE :: HETClNO3_HBr_JS
PRIVATE :: HETO3_HBr_JS
PRIVATE :: HETHOBr_SS_JS
PRIVATE :: HETClNO3_SS_JS
PRIVATE :: HETO3_SS_JS
PRIVATE :: HETHXUptake_JS
PRIVATE :: HETN2O5_SS
```

```
! New subroutines required by the JS functions
```

```
PRIVATE :: Gamma_ClNO3_Br
PRIVATE :: Gamma_O3_Br
PRIVATE :: Gamma_HOBr_X
PRIVATE :: Gamma_HX_Uptake
PRIVATE :: Coth
PRIVATE :: ReactoDiff_Corr
PRIVATE :: Gamma_HOBr_CLD !qjc
PRIVATE :: Gamma_HOBr_AER !qjc
```

```
! These are formerly strat-only reactions extended to take place in the
! troposphere on sulfate aerosol
```

```
PRIVATE :: HETClNO3_HCl
PRIVATE :: HETHOCl_HBr
PRIVATE :: HETHOCl_HCl
PRIVATE :: HETBrNO3_HCl
```

```
! These are subfunctions to calculate rates on/in clouds and SSA
```

```
PRIVATE :: CLD_PARAMS
PRIVATE :: GET_HALIDE_CLDConc
Private :: Get_Halide_SSAConc
```



```

PRIVATE :: COMPUTE_L2G_LOCAL
PRIVATE :: CLD1K_XNO3
PRIVATE :: EPOXUPTK
PRIVATE :: FCRO2HO2
PRIVATE :: FYHORO
PRIVATE :: FYRNO3
PRIVATE :: ARSL1K
PRIVATE :: kIIR1Ltd
PRIVATE :: kIIR1R2L
!PRIVATE DATA MEMBERS:
! Scalars
INTEGER :: NAERO
LOGICAL :: NATSURFACE, PSCBOX, STRATBOX
REAL(fp) :: TEMPK, RELHUM, SUNCOS, SPC_SO4
REAL(fp) :: SPC_NIT, GAMMA_HO2, XTEMP, XDENA
REAL(fp) :: QLIQ, QICE
REAL(fp) :: H_PLUS, MSO4, MNO3, MHSO4
REAL(fp) :: MW_HO2, MW_NO2, MW_NO3
REAL(fp) :: MW_N2O5, MW_GLYX, MW_MGLY
REAL(fp) :: MW_IEPOXA, MW_IEPOXB, MW_IEPOXD
REAL(fp) :: MW_IMAE, MW_LVOC, MW_ISN10G
REAL(fp) :: MW_ISOPND, MW_ISOPNB, MW_MACRN
REAL(fp) :: MW_MVKN, MW_R4N2, MW_DHDN
REAL(fp) :: MW_MONITS, MW_MONITU, MW_HONIT
REAL(fp) :: MW_IONITA, MW_MONITA, MW_BrNO3
REAL(fp) :: MW_HOBr, MW_HBr, MW_ClNO3
REAL(fp) :: MW_HOCl, MW_HI, MW_HOI
REAL(fp) :: MW_I2O2, MW_I2O3, MW_I2O4
REAL(fp) :: MW_IONO, MW_IONO2, MW_HCl
REAL(fp) :: MW_O3
REAL(fp) :: H_KO_O3, H_CR_O3, H_O3_T
REAL(fp) :: H_KO_HOBr, H_CR_HOBr, H_HOBr_T
REAL(fp) :: H_KO_HBr, H_CR_HBr
REAL(fp) :: H_KO_HCl, H_CR_HCl
REAL(fp) :: HS03conc_Cld, SO3conc_Cld, fupdateHOBr

! Arrays
REAL(fp) :: XAREA(25)
REAL(fp) :: XRADI(25)
REAL(fp) :: KHETI_SLA(11)

$OMP THREADPRIVATE(NAERO, NATSURFACE, PSCBOX, STRATBOX)
$OMP THREADPRIVATE(TEMPK, RELHUM, SPC_NIT, SPC_SO4)
$OMP THREADPRIVATE(GAMMA_HO2, XTEMP, XDENA, QLIQ)
$OMP THREADPRIVATE(QICE, XAREA, XRADI)
$OMP THREADPRIVATE(KHETI_SLA, SUNCOS)
$OMP THREADPRIVATE(H_PLUS, MSO4, MNO3, MHSO4)
$OMP THREADPRIVATE(HS03conc_Cld, SO3conc_Cld, fupdateHOBr)

```

**DEFINED PARAMETERS:**

```
REAL(fp), PARAMETER :: HetMinLife = 1.e-3_fp

! Critical RH for uptake of GLYX, MGLYX, and GLYC:
REAL(fp), PARAMETER :: CRITRH = 35.0e+0_fp

! Effective Henry's Law constant of IEPOX for reactive
! uptake to aqueous aerosols (M/atm)
!REAL(fp), PARAMETER :: HSTAR_EPOX = 5.0e+6_fp ! Prior to 3/2/18
REAL(fp), PARAMETER :: HSTAR_EPOX = 1.7e+7_fp

! Conversion factor from atm to bar
REAL(fp), PARAMETER :: con_atm_bar = 1.0e+0_fp/1.01325e+0_fp

! Universal gas constant [bar/(mol/kg)/K]
REAL(fp), PARAMETER :: con_R = RStarG*1.0e-2_fp
```

**REMARKS:**

```
Need
- TOTAREA (previously used for archiving N2O5 hydrolysis in the planeflight
 diagnostic only)
- Air NUM. DENSITY
- TEMPERATURE
- Aerosol Surface Area
- Aerosol Type
- Gamma (XSTKCF; sticking factor)
- ARR
- Species num density (mcl cm-3)
- Continental PBL or no?
- In stratosphere or no?
- Reaction index (e.g. NK1HBr, NK2HBr)
According to S. Eastham, we should also include
cloud and ice area explicitly, in addition to
aerosol area
```

**!REFERENCES:**

```
Eastham et al., Development and evaluation of the unified tropospheric-
stratospheric chemistry extension (UCX) for the global chemistry-transport
model GEOS-Chem, Atmos. Env., doi:10.1016/j.atmosenv.2014.02.001, 2014.
Fisher et al, Organic nitrate chemistry and its implications for nitrogen
budgets in an isoprene- and monoterpene-rich atmosphere: constraints from
aircraft (SEAC4RS) and ground-based (SOAS) observations in the Southeast
US. Atmos. Chem. Phys., 16, 2961-2990, 2016.
Marais et al., Aqueous-phase mechanism for secondary organic aerosol
formation from isoprene: application to the southeast United States and
co-benefit of SO2 emission controls, Atmos. Chem. Phys., 16, 1603-1618,
doi:10.5194/acp-16-1603-2016, 2016.
```

- Parrella et al, Tropospheric bromine chemistry: implications for present and pre-industrial ozone and mercury, *Atmos. Chem. Phys.*, 12, 6,723-6,740, doi:10.5194/acp-12-6723-2012, 2012.
- Schmidt, J., et al., Modelling the observed tropospheric BrO background: Importance of multiphase chemistry & implications for ozone, OH, & mercury, *J Geophys. Res-Atmos.*, 121, 024229, <https://doi.org/10.1002/2015JD024229>, 2016.
- Sherwen, T., et al., Global impacts of tropospheric halogens (Cl, Br, I) on oxidants and composition in GEOS-Chem, *Atmos. Chem. Phys.*, 16, 12239-12271, <https://doi.org/10.5194/acp-16-12239-2016>, 2016.

## REVISION HISTORY:

- 14 Dec 2015 - M. Long - Initial version
- 29 Jan 2016 - M. Sulprizio- Update to include heterogeneous chemistry for UCX mechanism
- 29 Mar 2016 - R. Yantosca - NOTE: SPC\_HBR and SPC\_HOBR are defined for trop-only mechanisms
- 29 Mar 2016 - R. Yantosca - Added ProTeX headers
- 29 Mar 2016 - R. Yantosca - Moved all the UCX-based functions to the end of the module, for clarity
- 01 Apr 2016 - R. Yantosca - Remove many global variables that can be declared locally from the THREADPRIVATEs
- 06 Jun 2016 - M. Sulprizio- Replace Get\_Indx with Spc\_GetIndx to use the fast-species lookup from the species database
- 14 Jun 2016 - M. Sulprizio- Replace Spc\_GetIndx with Ind\_
- 15 Jun 2017 - M. Sulprizio- Add heterogeneous chemistry for isoprene SOA from E. Marais (Marais et al., 2016)
- 14 Jul 2017 - M. Sulprizio- Add heterogeneous chemistry for monoterpenes from J. Fisher (Fisher et al., 2017)
- 24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA
- 15 Nov 2017 - M. Sulprizio- Add modifications for HOBr + S(IV) based on work by Qianjie Chen
- 02 Mar 2018 - M. Sulprizio- Update HSTAR\_EPOX following recommendation from E. Marais to address SOAIE being a factor of 2 lower in v11-02d than in Marais et al. [2016]

---

### 7.8.1 Set\_Het

Main heterogenous chemistry driver routine. Sets up the vector of heterogeneous chemistry rates for the KPP chemistry solver.

#### INTERFACE:

```
SUBROUTINE SET_HET(I, J, L, Input_Opt, State_Chm, State_Met)
```

#### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: I, J, L ! Lon, lat, level indices
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**REMARKS:****REVISION HISTORY:**

```

06 Jan 2015 - R. Yantosca - Initial version
01 Apr 2016 - R. Yantosca - Define many variables locally that don't
 need to be in the THREADPRIVATE statements
01 Apr 2016 - R. Yantosca - Remove KII_KI; we now declare that locally
31 May 2016 - E. Lundgren - Replace Input_Opt%XNUMOL with emMW_g from species
 database (emitted species g/mol)
26 Jul 2017 - M. Sulprizio- Remove hardcoded molecular weights from calls to
 Het* functions and use MW from species database
 instead
03 Jan 2018 - M. Sulprizio- Remove SCF argument. It was apparently added for
 diagnostic purposes and is no longer used. Also
 rename IO,SM,SC to Input_Opt,State_Met,State_Chm
 for consistency with other GEOS-Chem routines.
27 Feb 2018 - M. Sulprizio- Obtain Henry's law parameters from species
 database instead of hardcoding in halogens code

```

**7.8.2 kiir1ltd**

Determine removal rates for both species in an uptake reaction.

**INTERFACE:**

```

FUNCTION kiir1ltd(spcVec, indGas, indEduct, kISource, minLife) &
 RESULT(kII)

```

**INPUT PARAMETERS:**

```

! Rate coefficients
REAL(fp), INTENT(IN) :: spcVec(:)
INTEGER, INTENT(IN) :: indGas
INTEGER, INTENT(IN) :: indEduct
REAL(fp), INTENT(IN) :: kISource
REAL(fp), INTENT(IN), OPTIONAL :: minLife

```

**RETURN VALUE:**

```

REAL(fp) :: kII

```

**REMARKS:****REVISION HISTORY:**

29 Mar 2016 - R. Yantosca - Added ProTeX header  
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally  
01 Apr 2016 - R. Yantosca - Replace KII\_KI with DO\_EDUCT local variable

---

**7.8.3 kiir1r2l**

Determine removal rates for both species in an uptake reaction without assuming which reactant is limiting.

**INTERFACE:**

```
FUNCTION kiir1r2l(spcVec, indGasA, indGasB, kIASource, kIBSource) &
 RESULT(kII)
```

**USES:**

```
!USE Error_Mod, ONLY: Is_Safe_Div
```

**INPUT PARAMETERS:**

```
! Rate coefficients
REAL(fp), INTENT(IN) :: spcVec(:)
INTEGER, INTENT(IN) :: indGasA
INTEGER, INTENT(IN) :: indGasB
REAL(fp), INTENT(IN) :: kIASource
REAL(fp), INTENT(IN) :: kIBSource
```

**RETURN VALUE:**

```
REAL(fp) :: kII
```

**REMARKS:****REVISION HISTORY:**

29 Mar 2016 - R. Yantosca - Added ProTeX header  
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally  
01 Apr 2016 - R. Yantosca - Replace KII\_KI with DO\_EDUCT local variable

---

#### 7.8.4 HetIXCycleSSA

Set the iodine reaction rate on sea salt, assuming a fixed ratio of ICl and IBr (85:15) is produced.

**INTERFACE:**

```
FUNCTION HETIXCycleSSA(A, B, SSalk) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
! Sea salt alkalinity
REAL(fp), INTENT(IN) :: SSalk(2)
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

24 Dec 2016 - S. D. Eastham - Initial version

---

#### 7.8.5 HetIUptake

Set the uptake rate for iodine species.

**INTERFACE:**

```
FUNCTION HETIUptake(A, B, N, Input_Opt) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: A, B ! Rate coefficients
INTEGER, INTENT(IN) :: N ! Which aerosol?
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

24 Dec 2016 - S. D. Eastham - Initial version  
03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input\_Opt%LUCX

---

### 7.8.6 HetNO3

Set the heterogenous chemistry rate for NO3.

#### INTERFACE:

```
FUNCTION HETNO3(A, B) RESULT(HET_NO3)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_NO3
```

#### REMARKS:

#### REVISION HISTORY:

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
```

---

### 7.8.7 HetNO2

Set the heterogenous chemistry rate for NO2.

#### INTERFACE:

```
FUNCTION HETNO2(A, B) RESULT(HET_NO2)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_NO2
```

#### REMARKS:

#### REVISION HISTORY:

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
```

---

### 7.8.8 HetHO2

Set the heterogenous chemistry rate for HO2.

#### INTERFACE:

```
FUNCTION HETHO2(A, B) RESULT(HET_HO2)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_HO2
```

#### REMARKS:

#### REVISION HISTORY:

```
29 Mar 2016 - R. Yantosca - Added ProTeX headers
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
```

---

### 7.8.9 HetHBr

Set the heterogeneous rate for HBr.

#### INTERFACE:

```
FUNCTION HETHBr(A, B) RESULT(HET_HBr)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_HBr
```

#### REMARKS:

#### REVISION HISTORY:

```
29 Mar 2016 - R. Yantosca - Added ProTeX headers
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Define local variable for educt adjustment
```

---



### 7.8.10 HetN2O5

Set heterogenous chemistry rate for N2O5.

#### INTERFACE:

```
FUNCTION HETN2O5(A, B) RESULT(HET_N2O5)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_N2O5
```

#### REMARKS:

#### REVISION HISTORY:

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
```

---

### 7.8.11 HetGLYX

Sets the heterogenous chemistry rate for GLYX.

#### INTERFACE:

```
FUNCTION HETGLYX(A, B) RESULT(HET_GLYX)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_GLYX
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
02 Mar 2018 - M. Sulprizio- Change daytime gamma to 4.4e-3 and nighttime
gamma to 8.0e-6 based on recommendation from E.
Marais to address that SOAGX is a factor of 1.5
lower in v11-02d than in Marais et al. [2016]
```

---

### 7.8.12 HetMGLY

Sets the heterogenous chemistry rate for MGLY.

#### INTERFACE:

```
FUNCTION HETMGLY(A, B) RESULT(HET_MGLY)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MGLY
```

#### REMARKS:

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

---

### 7.8.13 HetIEPOX

Sets the heterogenous chemistry rate for IEPOX.

#### INTERFACE:

```
FUNCTION HETIEPOX(A, B) RESULT(HET_IEPOX)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_IEPOX
```

#### REMARKS:

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

---

### 7.8.14 HetIMAE

Sets the heterogenous chemistry rate for IMAE.

#### INTERFACE:

```
FUNCTION HETIMAE(A, B) RESULT(HET_IMAE)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_IMAE
```

#### REMARKS:

Here use the same values as are read in for IEPOX, but scale down gamma by a factor of 30 to get the value for IMAE. Gamma for the two species are similar under neutral conditions, so use IEPOX gamma when  $[H^+] \leq 8d-5$ . Implemented by (eam, 01/2015) using lab study findings from Riedel et al., EST, 2015.

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

---

### 7.8.15 HetLVOC

Sets the heterogenous chemistry rate for LVOC: condensation of low-volatility ISOPOOH oxidation products.

#### INTERFACE:

```
FUNCTION HETLVOC(A, B) RESULT(HET_LVOC)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_LVOC
```

#### REMARKS:

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

---

### 7.8.16 HetISN1OG

Sets the heterogenous chemistry rate for ISN1OG: uptake of 2nd generation organic nitrates formed from ISOP+NO3 reaction (eam, 02/2015).

#### INTERFACE:

```
FUNCTION HETISN1OG(A, B) RESULT(HET_ISN1OG)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_ISN1OG
```

#### REMARKS:

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

---

### 7.8.17 HetISOPND

Sets the heterogenous chemistry rate for ISOPND.

#### INTERFACE:

```
FUNCTION HETISOPND(A, B) RESULT(HET_ISOPND)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_ISOPND
```

#### REMARKS:

#### REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also has heterogeneous reaction. Remove call to EPOXUPTK here and use gamma value specified in SET\_HET (Fisher et al., 2016).

---

### 7.8.18 HetISOPNB

Sets the heterogenous chemistry rate for ISOPNB.

#### INTERFACE:

```
FUNCTION HETISOPNB(A, B) RESULT(HET_ISOPNB)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_ISOPNB
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also
has heterogeneous reaction. Remove call to
EPOXUPTK here and use gamma value specified in
SET_HET (Fisher et al., 2016).
```

---

### 7.8.19 HetMACRN

Sets the heterogenous chemistry rate for MACRN.

#### INTERFACE:

```
FUNCTION HETMACRN(A, B) RESULT(HET_MACRN)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MACRN
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also
has heterogeneous reaction. Remove call to
EPOXUPTK here and use gamma value specified in
SET_HET (Fisher et al., 2016).
```

---

### 7.8.20 HetMVKN

Sets the heterogenous chemistry rate for MVKN.

#### INTERFACE:

```
FUNCTION HETMVKN(A, B) RESULT(HET_MVKN)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MVKN
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also
has heterogeneous reaction. Remove call to
EPOXUPTK here and use gamma value specified in
SET_HET (Fisher et al., 2016).
```

---

### 7.8.21 HetR4N2

Sets the heterogenous chemistry rate for R4N2.

#### INTERFACE:

```
FUNCTION HETR4N2(A, B) RESULT(HET_R4N2)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_R4N2
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also
has heterogeneous reaction. Remove call to
EPOXUPTK here and use gamma value specified in
SET_HET (Fisher et al., 2016).
```

---

### 7.8.22 HetDHDN

Sets the heterogenous chemistry rate for DHDN.

#### INTERFACE:

```
FUNCTION HETDHDN(A, B) RESULT(HET_DHDN)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_DHDN
```

#### REMARKS:

#### REVISION HISTORY:

```
15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais
14 Jul 2017 - M. Sulprizio- Product has been changed to IONITA, which also
has heterogeneous reaction. Remove call to
EPOXUPTK here and use gamma value specified in
SET_HET (Fisher et al., 2016).
```

---

### 7.8.23 HetMONITS

Sets the heterogenous chemistry rate for MONITS

#### INTERFACE:

```
FUNCTION HETMONITS(A, B) RESULT(HET_MONITS)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MONITS
```

#### REMARKS:

#### REVISION HISTORY:

```
14 Jul 2017 - M. Sulprizio- Initial version based on SEAC4RS code and Fisher
et al. 2016.
```

---

### 7.8.24 HetMONITU

Sets the heterogenous chemistry rate for MONITU

#### INTERFACE:

```
FUNCTION HETMONITU(A, B) RESULT(HET_MONITU)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MONITU
```

#### REMARKS:

#### REVISION HISTORY:

14 Jul 2017 - M. Sulprizio- Initial version based on SEAC4RS code and Fisher et al. 2016.

---

### 7.8.25 HetHONIT

Sets the heterogenous chemistry rate for HONIT

#### INTERFACE:

```
FUNCTION HETHONIT(A, B) RESULT(HET_HONIT)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_HONIT
```

#### REMARKS:

#### REVISION HISTORY:

14 Jul 2017 - M. Sulprizio- Initial version based on SEAC4RS code and Fisher et al. 2016.

---



### 7.8.26 HetIONITA

Sets the heterogenous chemistry rate for IONITA: Aerosol-phase organic nitrate formed from monoterpene precursors.

#### INTERFACE:

```
FUNCTION HETIONITA(A, B) RESULT(HET_IONITA)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_IONITA
```

#### REMARKS:

#### REVISION HISTORY:

14 Jul 2017 - M. Sulprizio- Initial version based on SEAC4RS code and Fisher et al. 2016.

---

### 7.8.27 HetMONITA

Sets the heterogenous chemistry rate for MONITA: Aerosol-phase organic nitrate formed from monoterpene precursors.

#### INTERFACE:

```
FUNCTION HETMONITA(A, B) RESULT(HET_MONITA)
```

#### INPUT PARAMETERS:

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

#### RETURN VALUE:

```
REAL(fp) :: HET_MONITA
```

#### REMARKS:

#### REVISION HISTORY:

14 Jul 2017 - M. Sulprizio- Initial version based on SEAC4RS code and Fisher et al. 2016.

---

**7.8.28 HetN2O5\_SS**

Set heterogenous chemistry rate for N2O5 on sea salt. This reaction follows the N2O5 + Cl- channel, and Cl- is assumed to be in excess.

**INTERFACE:**

```
FUNCTION HETN2O5_SS(A, B) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
```

---

**7.8.29 HetN2O5\_HCl**

Set heterogenous chemistry rate for N2O5(g) + HCl(l,s) in polar stratospheric clouds and on tropospheric sulfate aerosol.

**INTERFACE:**

```
FUNCTION HETN2O5_HCl(A, B, Input_Opt) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: A, B ! Rate coefficients
TYPE(OptionalInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:**

This routine is only activated for UCX-based mechanisms.

**REVISION HISTORY:**

29 Jan 2016 - M. Sulprizio- Initial version, adapted from code previously  
in *calcrate.F*

29 Mar 2016 - R. Yantosca - Added ProTeX header

01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally

01 Apr 2016 - R. Yantosca - Replace KII\_KI with DO\_EDUCT local variable

04 May 2016 - M. Sulprizio- Add fixes for setting rate if not a STRATBOX

24 Dec 2016 - S. D. Eastham - Extended into the troposphere. Also now use the  
standard N2O5 calculation to establish gamma for  
sulfate, rather than relying on a fixed factor.

03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input\_Opt%LUCX

### 7.8.30 HetHXUptake.js

Sets the uptake rate of HCl and HBr on sea salt using Johan Schmidt's updated code.

#### INTERFACE:

```
FUNCTION HETHXUptake_JS(denAir, rAer, AAer, TK, X) RESULT(kISum)
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rAer ! Radius of aerosol (cm)
REAL(fp), INTENT(IN) :: AAer ! Area of aerosol (cm2/cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
INTEGER, INTENT(IN) :: X ! 1: Cl-, 2: Br-
```

#### RETURN VALUE:

```
REAL(fp) :: kISum
```

#### REMARKS:

#### REVISION HISTORY:

29 Mar 2016 - R. Yantosca - Added ProTeX header

01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally

01 Apr 2016 - R. Yantosca - Replace KII\_KI with DO\_EDUCT local variable

22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work

### 7.8.31 HetO3\_SS\_JS

Sets the O3 + Br- (in sea salt) rate using Johan Schmidt's updated code.

#### INTERFACE:

```
FUNCTION HETO3_SS_JS(denAir, rAer, AAer, alkAer, TK, halConc, O3Conc) &
 RESULT(kISum)
```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rAer ! Radius of aerosol (cm)
REAL(fp), INTENT(IN) :: AAer ! Area of aerosol (cm2/cm3)
REAL(fp), INTENT(IN) :: alkAer ! Aerosol alkalinity (?)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: halConc ! Halide concentration (mol/L)
REAL(fp), INTENT(IN) :: O3Conc ! Ozone concentration (#/cm3)

```

**RETURN VALUE:**

```

REAL(fp) :: kISum

```

**REMARKS:****REVISION HISTORY:**

```

29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work

```

**7.8.32 HetClNO3\_SS\_JS**

Sets the ClNO<sub>3</sub> + Br<sup>-</sup> (in sea salt) rate using Johan Schmidt's updated code.

**INTERFACE:**

```

FUNCTION HETClNO3_SS_JS(denAir, rAer, AAer, alkAer, TK, halConc) &
 RESULT(kISum)

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rAer ! Radius of aerosol (cm)
REAL(fp), INTENT(IN) :: AAer ! Area of aerosol (cm2/cm3)
REAL(fp), INTENT(IN) :: alkAer ! Aerosol alkalinity (?)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: halConc ! Halide concentration (mol/L)

```

**RETURN VALUE:**

```

REAL(fp) :: kISum

```

**REMARKS:****REVISION HISTORY:**

```

29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work

```

### 7.8.33 HetHOBBr\_SS\_JS

Sets the HOBr + Br- or Cl- (in sea salt) rate using Johan Schmidt's updated code.

#### INTERFACE:

```

FUNCTION HETHOBr_SS_JS(denAir, rAer, AAer, alkAer, TK, hConc, clConc, &
 brConc, X) &
 RESULT(kISum)

```

#### INPUT PARAMETERS:

```

REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rAer ! Radius of aerosol (cm)
REAL(fp), INTENT(IN) :: AAer ! Area of aerosol (cm2/cm3)
REAL(fp), INTENT(IN) :: alkAer ! Aerosol alkalinity (?)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: hConc ! H+ concentration (mol/L)
REAL(fp), INTENT(IN) :: clConc ! Chloride concentration (mol/L)
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L)
Integer, INTENT(IN) :: X ! 1: Cl-, 2: Br-

```

#### RETURN VALUE:

```

REAL(fp) :: kISum

```

#### REMARKS:

#### REVISION HISTORY:

```

29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work
01 Dec 2017 - Q.J. Chen - Updated to account for Cl- and Br- separately;
 Now calls routine Gamma_HOBr_AER

```

---

### 7.8.34 HetO3\_HBr\_JS

Sets the O3 + Br- rate using Johan Schmidt's updated code.

#### INTERFACE:

```

FUNCTION HETO3_HBr_JS(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, brConc, O3Conc) RESU

```

#### INPUT PARAMETERS:

```

REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)

```

```

REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L)
REAL(fp), INTENT(IN) :: O3Conc ! Ozone concentration (mol/L)

```

**RETURN VALUE:**

```

REAL(fp) :: kISum

```

**REMARKS:****REVISION HISTORY:**

```

29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work

```

**7.8.35 Gamma\_O3\_Br**

Function GAMMA\_O3\_Br calculates reactive uptake coef. for bromide oxidation by O3

**INTERFACE:**

```

FUNCTION GAMMA_O3_Br(Radius, n_air, T, C_Y, C_X_g) RESULT(GAM)

```

**USES:**

```

!OUTPUT PARAMETER:
! Reactive uptake coefficient (unitless)
REAL(fp) :: GAM

```

**INPUT PARAMETERS:**

```

! Radius (cm), n_air (#/cm), and X (1 for Cl and 2 for Br)
REAL(fp), INTENT(IN) :: Radius, n_air
REAL(fp), INTENT(IN) :: T, C_Y, C_X_g

```

**REVISION HISTORY:**

```

24 Sep 2015 - J. Schmidt - Initial version
27 Feb 2018 - M. Sulprizio- Obtain Henry's law parameters from species
 database in SET_HET instead of hardcoding here

```

**7.8.36 HetClNO3\_HBr\_JS**

Sets the ClNO<sub>3</sub> + Br- rate using Johan Schmidt's updated code.

**INTERFACE:**

```
FUNCTION HETClNO3_HBr_JS(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, &
 brConc, Input_Opt) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L)
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
22 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work
03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input_Opt%LUCX
```

**7.8.37 HetClNO3\_JS**

Sets the hydrolysis rate for ClNO<sub>3</sub> using Johan Schmidt's updated code.

**INTERFACE:**

```
FUNCTION HETClNO3_JS(denAir, rLiq, rIce, ALiq, AIce, TK) RESULT(HET_ClNO3)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
```

**RETURN VALUE:**

```
REAL(fp) :: HET_C1N03
```

**REMARKS:****REVISION HISTORY:**

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
16 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work
```

---

**7.8.38 HetHOB<sub>r</sub>HCl<sub>JS</sub>**

Sets the rate of the multiphase reaction HOBr + Cl<sup>-</sup> in sulfate aerosols, on cloud droplets and on PSCs

**INTERFACE:**

```
FUNCTION HETHOBr_HCl_JS(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, &
 hConc_Sul, hConc_LCl, hConc_ICl, clConc, brConc, &
 hso3Conc, so3Conc) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: hConc_Sul ! Sulfate H+ concentration
REAL(fp), INTENT(IN) :: hConc_LCl ! Liquid cloud H+ concentration
REAL(fp), INTENT(IN) :: hConc_ICl ! Ice cloud H+ concentration
REAL(fp), INTENT(IN) :: clConc ! Chloride concentration (mol/L)
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: hso3Conc ! HS03- concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: so3Conc ! S03-- concentration (mol/L), qjc
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**



21 Dec 2016 - S. D. Eastham - Generated code based on Johan Schmidt's work  
 01 Dec 2017 - Q.J. Chen - Updated to account for Br-, HS03-, and S03--;  
 Now calls routine Gamma\_HOBr\_AER

### 7.8.39 HetHOBr\_HBr\_JS

Sets the rate of the multiphase reaction HOBr + Br- in sulfate aerosols, on cloud droplets and on PSCs

#### INTERFACE:

```
FUNCTION HETHOBr_HBr_JS(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, &
 hConc_Sul, hConc_LCl, hConc_ICl, clConc, brConc, &
 hso3Conc, so3Conc) RESULT(kISum)
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: hConc_Sul ! Sulfate H+ concentration
REAL(fp), INTENT(IN) :: hConc_LCl ! Liquid cloud H+ concentration
REAL(fp), INTENT(IN) :: hConc_ICl ! Ice cloud H+ concentration
REAL(fp), INTENT(IN) :: clConc ! Chloride concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L)
REAL(fp), INTENT(IN) :: hso3Conc ! HS03- concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: so3Conc ! S03-- concentration (mol/L), qjc
```

#### RETURN VALUE:

```
REAL(fp) :: kISum
```

#### REMARKS:

#### REVISION HISTORY:

21 Dec 2016 - S. D. Eastham - Generated code based on Johan Schmidt's work  
 01 Dec 2017 - Q.J. Chen - Updated to account for Cl-, HS03-, and S03--;  
 Now calls routine Gamma\_HOBr\_AER

**7.8.40 HetHOB<sub>r</sub>\_HSO<sub>3</sub>**

Sets the rate of the multiphase reaction HOBr + HSO<sub>3</sub><sup>-</sup> in sulfate aerosols, on cloud droplets and on PSCs

**INTERFACE:**

```
FUNCTION HETHOBr_HS03(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, &
 hConc_Sul, hConc_LCl, hConc_ICl, clConc, brConc, &
 hso3Conc, so3Conc) &
 RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: hConc_Sul ! Sulfate H+ concentration
REAL(fp), INTENT(IN) :: hConc_LCl ! Liquid cloud H+ concentration
REAL(fp), INTENT(IN) :: hConc_ICl ! Ice cloud H+ concentration
REAL(fp), INTENT(IN) :: clConc ! Chloride concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: hso3Conc ! HSO3- concentration (mol/L)
REAL(fp), INTENT(IN) :: so3Conc ! SO3-- concentration (mol/L), qjc
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

```
15 Nov 2017 - M. Sulprizio - Generated code based on Qianjie Chen's work
01 Dec 2017 - Q.J. Chen - Updated to account for Cl-, Br-, and SO3--;
 Now calls routine Gamma_HOBr_CLD
 Remove HOBr+S(IV) on sulfate aerosols
```

**7.8.41 HetHOB<sub>r</sub>\_SO<sub>3</sub>**

Sets the rate of the multiphase reaction HOBr + SO<sub>3</sub><sup>-</sup> in sulfate aerosols, on cloud droplets and on PSCs

**INTERFACE:**

```

FUNCTION HETHOBr_S03(denAir, rLiq, rIce, ALiq, AIce, VAir, TK, &
 hConc_Sul, hConc_LCl, hConc_ICl, clConc, brConc, &
 hso3Conc, so3Conc) &
 RESULT(kISum)

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: VAir ! Box volume (cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
REAL(fp), INTENT(IN) :: hConc_Sul ! Sulfate H+ concentration
REAL(fp), INTENT(IN) :: hConc_LCl ! Liquid cloud H+ concentration
REAL(fp), INTENT(IN) :: hConc_ICl ! Ice cloud H+ concentration
REAL(fp), INTENT(IN) :: clConc ! Chloride concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: brConc ! Bromide concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: hso3Conc ! HS03- concentration (mol/L), qjc
REAL(fp), INTENT(IN) :: so3Conc ! S03-- concentration (mol/L)

```

**RETURN VALUE:**

```

REAL(fp) :: kISum

```

**REMARKS:****REVISION HISTORY:**

```

15 Nov 2017 - M. Sulprizio - Generated code based on Qianjie Chen's work
01 Dec 2017 - Q.J. Chen - Updated to account for Cl-, Br-, and HS03-;
 Now calls routine Gamma_HOBr_CLD;
 Remove HOBr+S(IV) on sulfate aerosols

```

**7.8.42 Gamma\_HX\_Uptake**

Function GAMMA\_HX\_uptake calculates mass accomidation coef. for uptake of HX (HCl or HBr)

**INTERFACE:**

```

FUNCTION Gamma_HX_Uptake(Radius, n_air, X, T) RESULT(GAM)
!OUTPUT PARAMETER:
! Reactive uptake coefficient (unitless)
REAL(fp) :: GAM

```

**INPUT PARAMETERS:**



**OUTPUT PARAMETERS:**

```
! Reactive uptake coefficient (unitless)
REAL(fp), INTENT(OUT) :: GAM_HOBr, r_gp
```

**INPUT PARAMETERS:**

```
! Radius (cm), n_air (#/cm), and X (1 for Cl and 2 for Br)
REAL(fp), INTENT(IN) :: Radius ! Radius (cm)
REAL(fp), INTENT(IN) :: n_air ! n_air (#/cm)
INTEGER, INTENT(IN) :: X ! 1=Cl-,2=Br-,3=HSO3-,4=SO3--
REAL(fp), INTENT(IN) :: T ! Temperature (K)
REAL(fp), INTENT(IN) :: C_Y1, C_Y2, C_Y3, C_Y4 ! Concentration (mol/L)
REAL(fp), INTENT(IN) :: C_Hp ! Concentration (mol/L)
```

**REVISION HISTORY:**

```
30 Nov 2017 - Q.J. Chen - Initial version
27 Feb 2018 - M. Sulprizio- Obtain Henry's law parameters from species
 database in SET_HET instead of hardcoding here
```

---

**7.8.45 gamma\_HOBr\_aer**

Returns GAMMA for HOBr in aerosol? (need better description)

**!INTERFACE:**

```
SUBROUTINE GAMMA_HOBr_AER(Radius, n_air, X, T, C_Y1, C_Y2, &
 C_Hp, GAM_HOBr, r_gp)
```

**USES:**

```
!OUTPUT PARAMETER:
! Reactive uptake coefficient (unitless)
REAL(fp), INTENT(OUT) :: GAM_HOBr, r_gp
```

**INPUT PARAMETERS:**

```
! Radius (cm), n_air (#/cm), and X (1 for Cl and 2 for Br)
REAL(fp), INTENT(IN) :: Radius ! Radius (cm)
REAL(fp), INTENT(IN) :: n_air ! n_air (#/cm)
INTEGER, INTENT(IN) :: X ! 1=Cl-,2=Br-,3=HSO3-,4=SO3--
REAL(fp), INTENT(IN) :: T ! Temperature (K)
REAL(fp), INTENT(IN) :: C_Y1, C_Y2 ! Concentration (mol/L)
REAL(fp), INTENT(IN) :: C_Hp ! Concentration (mol/L)
```

**REVISION HISTORY:**

```
30 Nov 2017 - Q.J. Chen - Initial version
27 Feb 2018 - M. Sulprizio- Obtain Henry's law parameters from species
 database in SET_HET instead of hardcoding here
```

---

**7.8.46 Gamma\_ClNO3\_Br**

Function GAMMA\_ClNO3\_Br calculates reactive uptake coef. for bromide oxidation by ClNO3

**INTERFACE:**

```
FUNCTION GAMMA_ClNO3_Br(Radius, n_air, T, C_Y) RESULT(GAM)
```

**USES:**

```
!OUTPUT PARAMETER:
! Reactive uptake coefficient (unitless)
REAL(fp) :: GAM
```

**INPUT PARAMETERS:**

```
! Radius (cm), n_air (#/cm)
REAL(fp), INTENT(IN) :: Radius, n_air
REAL(fp), INTENT(IN) :: T, C_Y
```

**REVISION HISTORY:**

24 Sept 2015 - J. Schmidt - Initial version

---

**7.8.47 COTH**

COTH (Hyperbolic cotangent)  $\text{coth}(x) = \cosh(x)/\sinh(x) = (1 + \exp(-2x))/(1 - \exp(-2x))$

**INTERFACE:**

```
REAL(fp) FUNCTION COTH(X)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: X ! The argument
```

**REVISION HISTORY:**

24 Sept 2015 - J. Schmidt - Initial version

---

**7.8.48 REACTODIFF\_CORR**

REACTODIFF\_CORR Correction =  $\text{COTH}(x) - (1/x)$   $x = \text{radius} / l$  Correction approaches 1 as  $x$  becomes large,  $\text{corr}(x;1000) \approx 1$  Correction approaches  $x/3$  as  $x$  goes towards 0

**INTERFACE:**

```
REAL(fp) FUNCTION REACTODIFF_CORR(radius, l)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: radius, 1 ! [cm] and [cm]
```

**REVISION HISTORY:**

```
14 Oct 2013 - J. Schmidt - Initial version
```

---

**7.8.49 HetBrNO3\_JS**

Sets the hydrolysis rate for BrNO<sub>3</sub> using Johan Schmidt's updated code.

**INTERFACE:**

```
FUNCTION HETBrNO3_JS(denAir, rLiq, rIce, ALiq, AIce, TK) RESULT(HET_BrNO3)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: denAir ! Density of air (#/cm3)
REAL(fp), INTENT(IN) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(IN) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(IN) :: ALiq ! Area of liquid cloud droplets (cm2/cm3)
REAL(fp), INTENT(IN) :: AIce ! Area of ice cloud crystals (cm2/cm3)
REAL(fp), INTENT(IN) :: TK ! Temperature (K)
```

**RETURN VALUE:**

```
REAL(fp) :: HET_BrNO3
```

**REMARKS:****REVISION HISTORY:**

```
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
16 Dec 2016 - S. D. Eastham - Updated code based on Johan Schmidt's work
```

---

**7.8.50 HetClNO3\_HCl**

Sets the heterogenous chemistry rate for ClNO<sub>3</sub>(g) + HCl(l,s) in polar stratospheric clouds and on tropospheric sulfate.

**INTERFACE:**

```
FUNCTION HETClNO3_HCl(A, B) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:**

This routine is only activated for UCX-based mechanisms.

**REVISION HISTORY:**

```
29 Jan 2016 - M. Sulprizio- Initial version, adapted from code previously
 in calcrate.F
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
04 May 2016 - M. Sulprizio- Add fixes for setting rate if not a STRATBOX
```

---

**7.8.51 HetBrNO3.HCl**

Set heterogenous chemistry rate for BrNO<sub>3</sub>(g) + HCl(l,s) in polar stratospheric clouds and on tropospheric sulfate.

**INTERFACE:**

```
FUNCTION HETBrNO3_HCl(A, B) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
! Rate coefficients
REAL(fp), INTENT(IN) :: A, B
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

```
29 Jan 2016 - M. Sulprizio- Initial version, adapted from code previously
 in calcrate.F
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
04 May 2016 - M. Sulprizio- Add fixes for setting rate if not a STRATBOX
24 Dec 2016 - S. D. Eastham - Extended into the troposphere
```

---



**7.8.52 HetHOCl\_HCl**

Set heterogenous chemistry rate for HOCl(g) + HCl(l,s) in polar stratospheric clouds and on sulfate aerosol.

**INTERFACE:**

```
FUNCTION HETHOCl_HCl(A, B, Input_Opt) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: A, B ! Rate coefficients
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:****REVISION HISTORY:**

```
29 Jan 2016 - M. Sulprizio- Initial version, adapted from code previously
 in calcrate.F
29 Mar 2016 - R. Yantosca - Added ProTeX header
01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally
01 Apr 2016 - R. Yantosca - Replace KII_KI with DO_EDUCT local variable
04 May 2016 - M. Sulprizio- Add fixes for setting rate if not a STRATBOX
22 Dec 2016 - S. D. Eastham - Now active for non-UCX mechanisms
03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input_Opt%LUCX
```

---

**7.8.53 HetHOCl\_HBr**

Set heterogenous chemistry rate for HOCl(g) + HBr(l,s) in polar stratospheric clouds and on sulfate aerosol.

**INTERFACE:**

```
FUNCTION HETHOCl_HBr(A, B, Input_Opt) RESULT(kISum)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: A, B ! Rate coefficients
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**RETURN VALUE:**

```
REAL(fp) :: kISum
```

**REMARKS:**

**REVISION HISTORY:**

29 Jan 2016 - M. Sulprizio- Initial version, adapted from code previously  
in *calcrate.F*  
 29 Mar 2016 - R. Yantosca - Added ProTeX header  
 01 Apr 2016 - R. Yantosca - Define N, XSTKCF, ADJUSTEDRATE locally  
 01 Apr 2016 - R. Yantosca - Replace KII\_KI with DO\_EDUCT local variable  
 04 May 2016 - M. Sulprizio- Add fixes for setting rate if not a STRATBOX  
 22 Dec 2016 - S. D. Eastham - Now active for non-UCX mechanisms  
 03 Jan 2018 - M. Sulprizio - Replace UCX CPP switch with Input\_Opt%LUCX

---

**7.8.54 N2O5**

Internal function N2O5 computes the GAMMA sticking factor for N2O5 hydrolysis. (mje, bmy, 8/7/03)

**INTERFACE:**

```
FUNCTION N2O5(AEROTYPE, TEMP, RH) RESULT(GAMMA)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: AEROTYPE ! Denoting aerosol type (cf FAST_JX)
REAL(fp), INTENT(IN) :: TEMP ! Temperature [K]
REAL(fp), INTENT(IN) :: RH ! Relative humidity [1]
```

**RETURN VALUE:**

```
REAL(fp) :: GAMMA
```

**REMARKS:**

Taken from the old SMVGEAR function *calcrate.F*.

**REVISION HISTORY:**

29 Mar 2016 - R. Yantosca - Added ProTeX headers  
 15 Jun 2017 - M. Sulprizio- Move conversion of RH from fraction to % to  
SET\_HET above

---

**7.8.55 HO2**

Function HO2 computes the GAMMA reaction probability for HO2 loss in aerosols based on the recommendation of Thornton, Jaegle, and McNeill, "Assessing Known Pathways For HO2 Loss in Aqueous Atmospheric Aerosols: Regional and Global Impacts on Tropospheric Oxidants" J. Geophys. Res., doi:10.1029/2007JD009236, 2008

**INTERFACE:**

```

FUNCTION HO2(RADIUS, TEMP, DENAIR, &
 SQM, HO2DENS, AEROTYPE, &
 CONTINENTAL_PBL, Input_Opt) &
RESULT(GAMMA)

```

**USES:**

```

USE Input_Opt_Mod, ONLY : OptInput
Use PhysConstants, ONLY : AVO, RGASLATM

```

**INPUT PARAMETERS:**

```

! Arguments
REAL(fp), INTENT(IN) :: RADIUS ! Aerosol radius [cm]
REAL(fp), INTENT(IN) :: TEMP ! Temperature [K]
REAL(fp), INTENT(IN) :: DENAIR ! Air density [molec/cm3]
REAL(fp), INTENT(IN) :: HO2DENS ! HO2 density [molec/cm3]
REAL(fp), INTENT(IN) :: SQM ! Square root of MW [g/mol]
INTEGER, INTENT(IN) :: AEROTYPE ! Aerosol type (cf FAST-JX)
INTEGER, INTENT(IN) :: CONTINENTAL_PBL ! Flag set to 1 if the box
! box is located in the
! continenal boundary
! layer, otherwise 0.
! Also check for ICE/SNOW
! (to disable this at
! high latitudes).
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**RETURN VALUE:**

```

REAL(fp) :: GAMMA ! Reaction probability

```

**REMARKS:**

Taken from the old SMVGEAR routine calcrate.F.  
Gamma(HO2) is a function of aerosol type, radius, temperature.

## References:

- 
- (1) Jacob, D.J., Heterogeneous chemistry and tropospheric ozone, Atmos. Environ., 34, 2131-2159, 2000. [full text (pdf)]
  - (2) J. Mao, Fan, S., Jacob, D. J., and Travis, K. R.: Radical loss in the atmosphere from Cu-Fe redox coupling in aerosols, Atmos. Chem. Phys., 13, 509-519, doi:10.5194/acp-13-509-2013, 2013.

**REVISION HISTORY:**

- 17 May 2013 - M. Payer - Add improved HO2 uptake (J. Mao)
- 22 May 2013 - M. Payer - Added option to read GAMMA\_HO2 from input.geos. Recommended value is 0.2

based on Jacob et al (2000) and Mao et al. (2013).

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

29 Mar 2016 - R. Yantosca - Added ProTeX headers

### 7.8.56 EPOXUPTK

Function EPOXUPTK computes the GAMMA sticking factor for EPOXUPTK hydrolysis to form 2-methyltetrols (AITET). (eam, 2014).

#### INTERFACE:

```
FUNCTION EPOXUPTK(AERAREA, AERRAD, TEMP, SQMW, &
 HENRY, KHPLUS, HPLUS, KNUC, SULF, NITR, &
 KGACID, BISULF, KHYDRO) &
 RESULT(GAMMA)
```

#### USES:

```
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : RGASLTM
USE ERROR_MOD, ONLY : IT_IS_NAN
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: AERRAD ! Aerosol radius [cm]
REAL(fp), INTENT(IN) :: AERAREA ! Aerosol surf. area [cm2/cm3]
REAL(fp), INTENT(IN) :: TEMP ! Temperature [K]
REAL(fp), INTENT(IN) :: SQMW ! Square root of the molecular weight
REAL(fp), INTENT(IN) :: HENRY ! Henry's Law constant [M/atm]
REAL(fp), INTENT(IN) :: KHPLUS ! 1st order rxn rate (acid-catalyzed ring
! opening)
REAL(fp), INTENT(IN) :: HPLUS ! Proton activity [unitless] and [H+] [M]
REAL(fp), INTENT(IN) :: KNUC ! 1st order rxn rate due to specific
! nucleophiles (SO4, NO3)
REAL(fp), INTENT(IN) :: SULF ! Sulfate concentration [M]
REAL(fp), INTENT(IN) :: NITR ! Nitrate concentration [M]
REAL(fp), INTENT(IN) :: KGACID ! 1st order rxn rate due to general acids
! (bisulfate in this case)
REAL(fp), INTENT(IN) :: BISULF ! Bisulfate concentration [M]
REAL(fp), INTENT(IN) :: KHYDRO ! Hydrolysis rate of alkylnitrates [1/s]
```

#### RETURN VALUE:

```
REAL(fp) :: GAMMA ! Reaction probability
```

#### REMARKS:

Calculation is only done for inorganic aqueous phase aerosols

This calculation uses the parameterization of Gaston et al., EST, 2014.

Redistribution of products (e.g. AITET) to yield organosulfates and organonitrates is done in SOA\_CHEMISTRY in carbon\_mod.F.

This is only done for IEPOX and IMAE if it's an SOA simulation

## REVISION HISTORY:

15 Jun 2017 - M. Sulprizio- Initial version based on calcrate.F from E.Marais

## 7.8.57 Cld\_Params

Subroutine CLD\_PARAMS returns ice and liquid cloud parameters based on State\_Met off of cloud particles.

## INTERFACE:

```

SUBROUTINE CLD_PARAMS(I, J, L, DENAIR, &
 VAir, T, QL, QI, State_Met, &
 rLiq, ALiq, VLiq, &
 rIce, AIce, VIce)

```

## USES:

```
USE State_Met_Mod, ONLY : MetState
```

## INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
REAL(fp), INTENT(IN) :: DENAIR ! Density of air [# / cm3]
REAL(fp), INTENT(IN) :: VAir ! Volume of air [cm3]
REAL(fp), INTENT(IN) :: T ! Temperature [K]
REAL(fp), INTENT(IN) :: QL, QI ! Cloud water mixing ratio [kg/kg]
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

## OUTPUT PARAMETERS:

```

REAL(fp), INTENT(OUT) :: rLiq ! Radius of liquid cloud droplets (cm)
REAL(fp), INTENT(OUT) :: rIce ! Radius of ice cloud crystals (cm)
REAL(fp), INTENT(OUT) :: ALiq ! Sfc area of liq. cloud (cm2/cm3)
REAL(fp), INTENT(OUT) :: AIce ! Sfc area of ice cloud (cm2/cm3)
REAL(fp), INTENT(OUT) :: VLiq ! Volume of liq. cloud (cm3/cm3)
REAL(fp), INTENT(OUT) :: VIce ! Volume of ice cloud (cm3/cm3)

```

## REMARKS:

### REVISION HISTORY:

21 Dec 2016 - S. D. Eastham - Adapted from CLD1K\_BrNO3  
24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA

---

### 7.8.58 Get\_Halide\_CldConc

Subroutine GET\_HALIDE\_CLDCONC returns the in-cloud concentration of bromide and chloride (Br- and Cl-).

### INTERFACE:

```
SUBROUTINE GET_HALIDE_CLDCONC(HBr, HCl, VLiq, VIce, VAir, TK, SA_SULF, R_SULF, br_c
```

### USES:

### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: HCl, HBr ! Number density [#/cm3]
REAL(fp), INTENT(IN) :: VAir ! Volume of air [cm3]
REAL(fp), INTENT(IN) :: SA_SULF, R_SULF! Sulfate aerosol surface area (cm2/cm3) and
REAL(fp), INTENT(IN) :: VLiq, VIce ! Volume of the cloud (liq and ice) [cm3]
REAL(fp), INTENT(IN) :: TK ! Air temperature [K]
```

### RETURN VALUE:

```
REAL(fp), INTENT(OUT) :: cl_conc, br_conc ! Liq. phase molar concentration [mol/kg-w
```

### REMARKS:

### REVISION HISTORY:

21 Dec 2016 - S. D. Eastham - Initial version  
27 Feb 2018 - M. Sulprizio - Obtain Henry's law parameters from species  
database in SET\_HET instead of hardcoding here

---

### 7.8.59 Get\_Halide\_SSAConc

Function GET\_HALIDE\_SSACONC calculates concentration of a halide in sea salt aerosol.

### INTERFACE:

```

 SUBROUTINE GET_HALIDE_SSACONC(n_x, surf_area, r_w, conc_x)
!OUTPUT PARAMETER:
 ! concentration of X- in SALX (mol/L)
 REAL(fp) :: conc_x

```

**INPUT PARAMETERS:**

```

 ! n_x = X-(ssa) number density (#/cm3), surf_area = AERO surface area
 ! conc (cm2/cm3), r_w = AERO wet radius (cm)
 REAL(fp), INTENT(IN) :: n_x, surf_area, r_w

```

**REVISION HISTORY:**

25 Jul 2014 - J. Schmidt - Initial version

---

**7.8.60 Compute\_L2G\_Local**

Subroutine COMPUTE\_L2G\_LOCAL is a local copy of the liquid-gas partitioning routine in GEOS-Chem's wetscav\_mod.F file.

**INTERFACE:**

```

 SUBROUTINE COMPUTE_L2G_LOCAL(KO, CR, pKa, TK, H2OLIQ, L2G)

```

**USES:**

```

 USE Henry_Mod, ONLY : Calc_KH
 USE Henry_Mod, ONLY : Calc_Heff

```

**INPUT PARAMETERS:**

```

 REAL(fp), INTENT(IN) :: KO ! Henry's solubility constant [M/atm]
 REAL(fp), INTENT(IN) :: CR ! Henry's volatility constant [K]
 REAL(fp), INTENT(IN) :: pKa ! Henry's pH correction factor [1]
 REAL(fp), INTENT(IN) :: TK ! Temperature [K]
 REAL(fp), INTENT(IN) :: H2OLIQ ! Liquid water content [cm3 H2O/cm3 air]

```

**OUTPUT PARAMETERS:**

```

 REAL(fp), INTENT(OUT) :: L2G ! Cliq/Cgas ratio [1]

```

**REMARKS:**

The ratio Cliq / Cgas is obtained via Henry's law. The appropriate values of Kstar298 and H298\_R must be supplied for each species. (cf Jacob et al 2000, p. 3)

**REVISION HISTORY:**

23 Feb 2000 - R. Yantosca - Initial version  
 (1 ) Bundled into "wetscav\_mod.f" (bmy, 11/8/02)  
 16 Sep 2010 - R. Yantosca - Added ProTeX headers  
 10-Jan-2011 - H. Amos - Corrected the units on KStar298 from moles/atm  
                   to M/atm  
 15-May-2013 - F. Paulot - Fix R constant  
 08 Dec 2015 - R. Yantosca - Now use functions from henry\_mod.F

---

### 7.8.61 CLD1K\_XNO3

Function CLD1K\_XNO3 calculates the rate constant for heterogeneous cycling of XNO3 off of cloud particles.

#### INTERFACE:

```
FUNCTION CLD1K_XNO3(denAir, TK, rLiq, rIce, ALiq, AIce, &
 MX_gmol, AlphaX) RESULT(cld1k)
```

#### USES:

#### INPUT PARAMETERS:

```
REAL(fp), Intent(IN) :: DENAIR ! Density of air [# /cm3]
REAL(fp), Intent(In) :: TK ! Air temperature [K]
REAL(fp), Intent(In) :: rLiq ! Radius of liquid cloud drops [cm]
REAL(fp), Intent(In) :: rIce ! Radius of ice cloud crystals [cm]
REAL(fp), Intent(In) :: ALiq ! Surface area (liquid) [cm2/cm3]
REAL(fp), Intent(In) :: AIce ! Surface area (ice)) [cm2/cm3]
REAL(fp), Intent(IN) :: MX_gmol ! Molecular mass of XNO3 [g/mol]
REAL(fp), Intent(IN) :: AlphaX ! XNO3 accomodation coef [unitless]
```

#### RETURN VALUE:

```
REAL(fp) :: cld1k ! Rate constant for heterogeneous
 ! cycling of BrNO3 off of cloud
 ! particles
```

#### REMARKS:

#### REVISION HISTORY:

27 Feb 2011 - J. Parrella - Initial version  
 22 May 2012 - M. Payer - Added ProTeX headers  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met  
                   derived type object  
 06 Nov 2014 - R. Yantosca - Now use State\_Met%CLDF(I,J,L)

---



### 7.8.62 FcrO2HO2

lfgap, based on saunder 2003 k14

#### INTERFACE:

```
FUNCTION FCRO2HO2(XCARBN) RESULT(FC_RO2HO2)
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: XCARBN
```

#### RETURN VALUE:

```
REAL(fp) :: FC_RO2HO2
```

#### REVISION HISTORY:

24 Jul 2014 - R. Yantosca - Now inlined to calcrate.F

---

### 7.8.63 FyHORO

#### Overview

Function FYHORO returns returns the branching ratio between HOC2H4O oxidation and dissociation: (1)  $\text{HOC2H4} + \text{O2} \rightarrow \text{HO2} + \text{GLYC}$  (2)  $\text{HOC2H4} \rightarrow \text{HO2} + 2\text{CH2O}$

#### References

1. Orlando et al., 1998: *Laboratory and theoretical study of the oxyradicals in the OH- and Cl-initiated oxidation of ethene*, J. Phys. Chem. A, **102**, 8116-8123.
2. Orlando et al., 2003: *The atmospheric chemistry of alkoxy radicals*, Chem. Rev., **103**, 4657-4689.

#### INTERFACE:

```
FUNCTION FYHORO(ZDNUM, TT) RESULT(FY_HORO)
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: ZDNUM ! Air density [molec/cm3]
REAL(fp), INTENT(IN) :: TT ! Temperature [K]
```

#### RETURN VALUE:

```
REAL(fp) :: FY_HORO
```

#### REVISION HISTORY:

(1 ) Branching ratio calculation (tmf, 2/6/05).  
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
25 Jul 2014 - R. Yantosca - Now inlined into calcrate.F

---

### 7.8.64 FyrNO3

Function FYRNO3 returns organic nitrate yields  $YN = RKA/(RKA+RKB)$  from RO2+NO reactions as a function of the number N of carbon atoms.

#### INTERFACE:

```
FUNCTION FYRNO3(XCARBN, ZDNUM, TT) RESULT(FYR_NO3)
```

#### INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: XCARBN ! Number of C atoms in RO2
REAL(fp), INTENT(IN) :: ZDNUM ! Air density [molec/cm3]
REAL(fp), INTENT(IN) :: TT ! Temperature [K]
```

#### RETURN VALUE:

```
REAL(fp) :: FYR_NO3
```

#### REVISION HISTORY:

- (1 ) Original code from Larry Horowitz, Jinyou Liang, Gerry Gardner, and Daniel Jacob circa 1989/1990.
  - (2 ) Updated following Atkinson 1990.
  - (3 ) Change yield from Isoprene Nitrate (ISN2) from 0.44% to 12%, according to Sprengnether et al., 2002. (amf, bmy, 1/7/02)
  - (4 ) Eliminate obsolete code from 1/02 (bmy, 2/27/02)
  - (5 ) Updated comment description of XCARBN (bmy, 6/26/03)
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 25 Jul 2014 - R. Yantosca - Now inlined into calcrate.F
- 

### 7.8.65 Arsl1k

Function ARSL1K calculates the 1st-order loss rate of species on wet aerosol surface.

#### INTERFACE:

```
FUNCTION ARSL1K(AREA, RADIUS, DENAIR, STKCF, STK, SQM) &
 RESULT(ARS_L1K)
```

#### INPUT PARAMETERS:

```
! Surface area of wet aerosols/volume of air [cm2/cm3]
REAL(fp), INTENT(IN) :: AREA

! Radius of wet aerosol [cm], order of 0.01-10 um;
! Note that radius here is Rd, not Ro
REAL(fp), INTENT(IN) :: RADIUS

! Density of air [# /cm3]
```

```

REAL(fp), INTENT(IN) :: DENAIR

! Sticking coefficient [unitless], order of 0.1
REAL(fp), INTENT(IN) :: STKCF

! Square root of temperature [K]
REAL(fp), INTENT(IN) :: STK

! Square root of molecular weight [g/mole]
REAL(fp), INTENT(IN) :: SQM

```

**RETURN VALUE:**

```

REAL(fp) :: ARS_L1K

```

**REMARKS:**

The 1st-order loss rate on wet aerosol (Dentener's Thesis, p. 14) is computed as:

$$\text{ARSL1K [1/s]} = \text{area} / [ \text{radius/dfkg} + 4./(\text{stkcf} * \text{xmms}) ]$$

where XMMS = Mean molecular speed [cm/s] =  $\sqrt{8R*TK/\pi/M}$  for Maxwell  
 DFKG = Gas phase diffusion coeff [cm<sup>2</sup>/s] (order of 0.1)

**REVISION HISTORY:**

```

01 Jul 1994 - lwh, jyl, gmg, djj - Initial version
04 Apr 2003 - R. Yantosca - Updated comments, cosmetic changes
07 Apr 2004 - R. Yantosca - Now return w/ default value if RADIUS is zero
 (i.e. is smaller than a very small number)
03 Dec 2009 - R. Yantosca - Prevent div-by-zero errors by returning the
 default value if any of the args are zero
03 Dec 2009 - R. Yantosca - Added ProTeX Header
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

```

**7.8.66 Check\_Nat**

Subroutine CHECK\_NAT determines whether the solid PSC is composed of ice or NAT (nitric acid trihydrate) (needed for heterogeneous chemistry), or indeed if there is any direct PSC calculation at all. This is important for determining whether to use the JPP or Kirner scheme for ice cloud radii.

**INTERFACE:**

```

SUBROUTINE CHECK_NAT(I, J, L, IS_NAT, IS_PSC, IS_STRAT, &
 Input_Opt, State_Met, State_Chm)

```

**INPUT PARAMETERS:**

```

 INTEGER, INTENT(IN) :: I,J,L ! Grid indices
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
 TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
 TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
!OUTPUT VARIABLES:
 LOGICAL, INTENT(OUT) :: IS_NAT ! Is surface NAT?
 LOGICAL, INTENT(OUT) :: IS_PSC ! Are there solid PSCs?
 LOGICAL, INTENT(OUT) :: IS_STRAT ! Are we in the strat?

```

**REMARKS:**

This routine is only activated for UCX-based mechanisms

**REVISION HISTORY:**

```

17 Apr 2013 - S. D. Eastham - Initial version
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
08 Apr 2015 - R. Yantosca - Remove call to READ_PSC_FILE, this is
 now done from DO_CHEMISTRY (chemistry_mod.F)
29 Jan 2016 - M. Sulprizio - Moved this routine from ucx_mod.F to
 gckpp_HetRates.F90

```

**7.9 Fortran: Module Interface ucx\_mod.F**

Module UCX\_MOD contains routines and variables which are associated with the addition of full stratospheric chemistry to GEOS-Chem (based on the NASA GMI implementation, forming the Unified Chemistry eXtension (UCX)).

**INTERFACE:**

```

MODULE UCX_MOD

```

**USES:**

```

 USE CMN_SIZE_MOD
 USE ERROR_MOD, ONLY : DEBUG_MSG
 USE inquireMod, ONLY : findFreeLUN
 USE PhysConstants ! Physical constants
 USE PRECISION_MOD ! For GEOS-Chem Precision (fp)

#if !defined(ESMF_)
 ! NcdfUtil modules for netCDF I/O
 USE m_netcdf_io_open ! netCDF open
 USE m_netcdf_io_get_dimlen ! netCDF dimension queries
 USE m_netcdf_io_read ! netCDF data reads
 USE m_netcdf_io_close ! netCDF close
#endif

```

```
 IMPLICIT NONE
 #if !defined(ESMF_)
 # include "netcdf.inc"
 #endif
```

```
 PRIVATE
```

#### **PUBLIC DATA MEMBERS:**

```
 CHARACTER(LEN=255), PUBLIC :: NOON_FILE_ROOT ! Directory for noontime data
 CHARACTER(LEN=255), PUBLIC :: MONTREAL_FILE_ROOT ! Root directoy for WHO data
 PUBLIC :: T_STS ! Max temperature of STS formation (K)
 PUBLIC :: NDENS_AER ! See below
```

#### **PUBLIC MEMBER FUNCTIONS:**

```
 PUBLIC :: SET_H2O_TRAC
 PUBLIC :: SET_CLOCK_TRAC
 PUBLIC :: EMISS_BASIC
 PUBLIC :: SETTLE_STRAT_AER
 PUBLIC :: SO4_PHOTFRAC
 PUBLIC :: UCX_NOX
 PUBLIC :: UCX_H2SO4PHOT
 PUBLIC :: CALC_STRAT_AER
 PUBLIC :: GET_STRAT_OPT
 PUBLIC :: KG_STRAT_AER
 PUBLIC :: RHO_STRAT_AER
 PUBLIC :: INIT_UCX
 PUBLIC :: DIAGINIT_UCX
 PUBLIC :: CLEANUP_UCX
 PUBLIC :: READ_PSC_FILE
 PUBLIC :: WRITE_STATE_PSC
PRIVATE MEMBER FUNCTIONS:
 PRIVATE :: READ_SFC
 PRIVATE :: SET_MONTREAL
 PRIVATE :: SET_MONTREAL_NCDF
 PRIVATE :: GET_MONTREAL_NCDF
 PRIVATE :: MONTREAL_SCALEFAC
 PRIVATE :: TERNARY
 PRIVATE :: CARSLAW_DENSITY
 PRIVATE :: CALC_H2SO4_GAS
 PRIVATE :: CALC_SLA_GAMMA
 PRIVATE :: MOLEC_SPEED
 PRIVATE :: SFCMR_INIT
 PRIVATE :: SFCMR_READ
 PRIVATE :: SFCMR_GET
 PRIVATE :: NOXCOEFF_INIT
 PRIVATE :: GET_JJNOX
```

**REVISION HISTORY:**

26 Mar 2013 - S. D. Eastham - Initial version  
 04 Apr 2013 - S. D. Eastham - Rolled several routines into module  
 20 Feb 2014 - M. Sulprizio - Removed "define.h", this is now obsolete  
 24 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 05 Dec 2014 - C. Keller - Added SFCMR object for reading surface mixing ratios to buffer (from ASCII file).  
 Now map NOx coefficients onto grids other than 4x5 or 2x25.  
 14 Jan 2014 - C. Keller - Now read and diagnose STATE\_PSC through HEMCO.  
 08 Apr 2015 - R. Yantosca - Now make READ\_PSC\_FILE a PUBLIC routine  
 08 May 2015 - C. Keller - Added WRITE\_STATE\_PSC  
 05 Jan 2016 - E. Lundgren - Use global physical parameters AVO, PI, AIRMW, RE, RSTARG, and ATM  
 28 Jan 2016 - M. Sulprizio - STATE\_PSC and KHETI\_SLA are now fields in State\_Chm instead of public arrays. These need to be added to the derived type object to be accessed in gckpp\_HetRates.F90.  
 28 Jan 2016 - M. Sulprizio - Move routine CHECK\_NAT to gckpp\_HetRates.F90  
 21 Jun 2016 - R. Yantosca - Now store species ID flags as module variables  
 23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer compatible with the FlexChem implementation  
 18 Jul 2016 - M. Sulprizio - Remove CFCX and HCFCX everywhere and replace with their constituents. Family tracers have been eliminated.  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

**7.9.1 montreal.scalefac**

Function MONTREAL\_SCALEFAC calculates a scaling factor for long-lived species between a future year and a reference year, based on WMO projections.

**INTERFACE:**

```
FUNCTION MONTREAL_SCALEFAC(MS_TRAC,MS_FUTRYR,
& MS_INITYR,MS_TARGMO) RESULT(SCALEFAC)
```

**USES:****INPUT PARAMETERS:**

```
CHARACTER(*) , INTENT(IN) :: MS_TRAC
INTEGER , INTENT(IN) :: MS_FUTRYR ! Future year
INTEGER , INTENT(IN) :: MS_INITYR ! Future year
INTEGER , INTENT(IN) :: MS_TARGMO ! Target month
!OUTPUT VARIABLES:
REAL(fp) :: SCALEFAC ! Scaling factor
```



objects via the arg list

24 Feb 2014 - E. Lundgren - Replace GET\_PCENTER, which retrieves wet air pressure, with State\_Met%PMID\_DRY. Remove dependency on PRESSURE\_MOD.

16 Jun 2016 - S. D. Eastham - Remove references to tracerid\_mod.F

21 Jun 2016 - R. Yantosca - Use the module variables for species ID flags

22 Jun 2016 - M. Yannetti - Replace TCVV with spc db MW and phys constant

07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer

11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code

---

### 7.9.3 get\_noxcoeff

Subroutine GET\_NOXCOEFF reads in O1D and O3P mixing ratios along with NO, NO2, NO3 and N2O J-rates from 2D data, interpolating onto the 3D grid and storing in NOX\_O and NOX\_J.

#### INTERFACE:

```
SUBROUTINE GET_NOXCOEFF(TARG_MONTH, Input_Opt, State_Met)
```

#### USES:

```
USE ERROR_MOD, ONLY : ERROR_STOP
USE ERROR_MOD, ONLY : ALLOC_ERR
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: TARG_MONTH
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### REMARKS:

At some later point we should attempt to rewrite the parallel DO loop so that the loop order is L-J-I. Not sure how easy that is. (bmy, 2/14/14)

#### REVISION HISTORY:

26 Mar 2013 - S. D. Eastham - Initial version

14 Feb 2014 - R. Yantosca - Parallelize main DO loop

21 Feb 2014 - M. Sulprizio - Now pass Input\_Opt and State\_Met objects via the arg list

24 Feb 2014 - E. Lundgren - Replace GET\_PCENTER, which retrieves wet air pressure, with State\_Met%PMID and remove dependency on PRESSURE\_MOD.

---



#### 7.9.4 emiss\_basic

Subroutine EMISS\_BASIC sets surface mixing ratios of N<sub>2</sub>O, OCS and ozone-depleting substances covered by the Montreal protocol.

#### INTERFACE:

```

SUBROUTINE EMISS_BASIC(am_I_Root, Input_Opt, State_Met,
& State_Chm, RC)

```

#### USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH
USE TIME_MOD, ONLY : GET_YEAR

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: RC ! Failure or success

```

#### REMARKS:

#### REVISION HISTORY:

```

28 Mar 2013 - S. D. Eastham - Initial version
11 Feb 2014 - R. Yantosca - Rewrote DO loop to be more efficient
14 Feb 2014 - R. Yantosca - Now parallelize main DO loop
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
10 Sep 2015 - E. Lundgren - Tracer units are now kg/kg dry air (prev v/v)
25 Jul 2016 - M. Yannetti - Pass State_Chm as arg for spc db access;
 Replace TCVV with spc db MW and phys constant
07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer
11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90

```

---

### 7.9.5 read\_sfc

Subroutine READ\_SFC fills out the surface emissions for some species introduced as part of the unified chemistry upgrade.

#### INTERFACE:

```
SUBROUTINE READ_SFC(am_I_Root, Input_Opt)
```

#### USES:

```
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
!OUTPUT VARIABLES:
```

#### REVISION HISTORY:

```
04 Apr 2013 - S. D. Eastham - Initial version
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt object via the arg list
```

---

### 7.9.6 set\_montreal\_ncdf

Subroutine SET\_MONTREAL\_NCDF calls GET\_MONTREAL\_NCDF for the surface mixing ratio for a specific year, and applies it to GRID\_EMIT.

#### INTERFACE:

```
SUBROUTINE SET_MONTREAL_NCDF (TRAC_NAME,TARG_YR_NC,TARG_MO_NC,N)
```

#### USES:

#### INPUT PARAMETERS:

```
CHARACTER(LEN=20), INTENT(IN) :: TRAC_NAME
INTEGER, INTENT(IN) :: TARG_YR_NC
INTEGER, INTENT(IN) :: TARG_MO_NC
INTEGER, INTENT(IN) :: N
```

#### REVISION HISTORY:

```
10 Oct 2013 - S. D. Eastham - Wrapper built from edges of original GET
04 Mar 2014 - S. D. Eastham - Updated to NetCDF
```

---

### 7.9.7 get\_montreal\_ncdf

Function GET\_MONTREAL\_NCDF gets mean surface mixing ratio of a given species based on forecasts from WHO estimates. Values are interpolated between two monthly averages.

#### INTERFACE:

```
FUNCTION GET_MONTREAL_NCDF (TRAC_NAME,TARG_YR_NC,TARG_MO_NC)
& RESULT (MEAN_MIXRATIO)
```

#### USES:

```
USE ERROR_MOD, ONLY : ERROR_STOP, ALLOC_ERR
USE FILE_MOD, ONLY : IOERROR
```

#### INPUT PARAMETERS:

```
REAL(fp) :: MEAN_MIXRATIO
CHARACTER*(*) , INTENT(IN) :: TRAC_NAME
INTEGER, INTENT(IN) :: TARG_YR_NC
INTEGER, INTENT(IN) :: TARG_MO_NC
```

#### REVISION HISTORY:

```
04 Apr 2013 - S. D. Eastham - Initial version
10 Oct 2013 - S. D. Eastham - Split into two routines (GET and SET)
04 Mar 2014 - S. D. Eastham - Added NetCDF compatibility
```

---

### 7.9.8 set\_montreal

Subroutine SET\_MONTREAL calls GET\_MONTREAL for the surface mixing ratio for a specific year, and applies it to GRID\_EMIT.

#### INTERFACE:

```
SUBROUTINE SET_MONTREAL (TRAC_NAME,TARG_LINE,N,Input_Opt)
```

#### USES:

```
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
CHARACTER(LEN=20), INTENT(IN) :: TRAC_NAME
INTEGER, INTENT(IN) :: TARG_LINE
INTEGER, INTENT(IN) :: N
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
```

#### REVISION HISTORY:

```
10 Oct 2013 - S. D. Eastham - Wrapper built from edges of original GET
```

---

### 7.9.9 settle\_strat\_aer

Subroutine SETTLE\_STRAT\_AER performs gravitational settling of stratospheric aerosols. It is copied largely from GRAV\_SETTLING in sulfate\_mod.F. All of this is ignored if APM is active.

#### INTERFACE:

```

SUBROUTINE SETTLE_STRAT_AER(am_I_Root, Input_Opt, State_Met,
& State_Chm, RC)

```

#### USES:

```

USE CMN_FJX_MOD, ONLY : RAA, IND999
USE CMN_SIZE_MOD ! Size parameters
USE ErrCode_Mod
USE ERROR_MOD, ONLY : IT_IS_NAN,ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_ELAPSED_SEC, GET_TS_CHEM

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: RC ! Return code

```

#### REVISION HISTORY:

```

11 Apr 2013 - S. D. Eastham - Initial version
07 Feb 2014 - R. Yantosca - Add missing variables to !$OMP PRIVATE statement
07 Feb 2014 - R. Yantosca - Cosmetic changes
21 Feb 2014 - M. Sulprizio- Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
24 Feb 2014 - E. Lundgren - Replace GET_PCENTER and GET_PEDGE with
 State_Met%PMID and State_Met%PEDGE
24 Feb 2014 - E. Lundgren - Remove dependency on PRESSURE_MOD
24 Feb 2014 - E. Lundgren - Use State_Met%DELP in place of PEDGE difference
08 Apr 2015 - R. Yantosca - Remove call to READ_PSC_FILE, this is
 now done from DO_CHEMISTRY (chemistry_mod.F)
12 May 2016 - M. Sulprizio- Remove 1D arrays that depend on KLOOP. WERADIUS,
 is now a pointer that point to 3D fields in
 in State_Chm.
31 May 2016 - E. Lundgren - Replace Input_Opt%TRACER_MW_G with species
 database emMW_g (emitted species g/mol)
16 Jun 2016 - S.D.Eastham - Remove references to tracerid_mod.F
15 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer

```

---

**7.9.10 calc\_h2so4\_gas**

Subroutine CALC\_H2SO4\_GAS calculates the fraction of strat. SO4 aerosol which can be considered to be gaseous H2SO4.

**INTERFACE:**

```
SUBROUTINE CALC_H2SO4_GAS(Input_Opt, State_Met, State_Chm)
```

**USES:**

```
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
```

**REVISION HISTORY:**

```
11 Apr 2013 - S. D. Eastham - Initial version
07 Feb 2014 - R. Yantosca - Cosmetic changes
14 Feb 2014 - R. Yantosca - Changed parallel DO loop order to L-J-I,
 which is the most efficient configuration
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
24 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID_DRY
24 Feb 2015 - E. Lundgren - Remove dependency on PRESSURE_MOD
26 Mar 2015 - E. Lundgren - Now use moist and dry air partial pressures
31 May 2016 - E. Lundgren - Replace Input_Opt%TRACER_MW_G with species
 database emMW_g (emitted species g/mol)
07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer
```

---

**7.9.11 so4\_photfrac**

FUNCTION SO4\_PHOTFRAC returns the fraction of H2SO4 which is available for photolysis.

**INTERFACE:**

```
REAL(fp) FUNCTION SO4_PHOTFRAC(I,J,L)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I,J,L ! Location indices
! OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: PHOTFRAC ! Gaseous fraction of H2SO4
```

**REVISION HISTORY:**

```
11 Apr 2013 - S. D. Eastham - Initial version
```

---

**7.9.12 calc\_strat\_aer**

Subroutine CALC\_STRAT\_AER calculates aerosol properties stratosphere using the thermodynamic parameterization described in Kirner et al. ('Simulation of polar stratospheric clouds in the chemistry-climate-model EMAC via the submodel PSC', Geosci. Mod. Dev., 4, 169-182, 2011).

**INTERFACE:**

```

SUBROUTINE CALC_STRAT_AER(am_I_Root, Input_Opt,
& State_Met, State_Chm, RC)

```

**USES:**

```

USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE ERROR_MOD, ONLY : ERROR_STOP
USE ERROR_MOD, ONLY : IS_SAFE_DIV
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

13 Apr 2013 - S. D. Eastham - Initial version
06 Feb 2014 - R. Yantosca - Add missing variables to OpenMP loop
14 Feb 2014 - R. Yantosca - Make the L-loop the outermost DO loop
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
06 Nov 2014 - R. Yantosca - Now use State_Met%CLDF(I,J,L)
24 Feb 2014 - E. Lundgren - Replace GET_PCENTER, which retrieves wet
 air pressure, with State_Met%PMID_DRY and
 remove dependency on PRESSURE_MOD.
08 Apr 2015 - R. Yantosca - Remove call to READ_PSC_FILE, this is
 now done from DO_CHEMISTRY (chemistry_mod.F)
31 May 2016 - E. Lundgren - Replace Input_Opt%TRACER_MW_G with emMW_g
 from species database (emitted species g/mol)

```

16 Jun 2016 - S. D. Eastham - Remove references to tracerid\_mod.F  
 21 Jun 2016 - R. Yantosca - Now use module variables for species ID flags  
 07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer  
 11 Aug 2016 - R. Yantosca - Remove temporary tracer removal code

---

### 7.9.13 kg\_strat\_aer

Function KG\_STRAT\_AER returns the calculated mass of a stratospheric aerosol. The routine is essentially just an interface to allow external routines to "see" the arrays.

#### INTERFACE:

```
REAL(fp) FUNCTION KG_STRAT_AER (I,J,L,IAER)
```

#### INPUT PARAMETERS:

```
INTEGER,INTENT(IN) :: I,J,L ! Grid indices
INTEGER,INTENT(IN) :: IAER ! Aerosol index
 ! 1 = SSA (pure H2SO4)
 ! 2 = STS
 ! 3 = Solid PSC
```

#### REVISION HISTORY:

18 Apr 2013 - S. D. Eastham - Initial version

---

### 7.9.14 rho\_strat\_aer

Function RHO\_STRAT\_AER returns the calculated stratospheric aerosol mass density.

#### INTERFACE:

```
REAL(fp) FUNCTION RHO_STRAT_AER (I,J,L,IAER)
```

#### INPUT PARAMETERS:

```
INTEGER,INTENT(IN) :: I,J,L ! Grid indices
INTEGER,INTENT(IN) :: IAER ! Aerosol index:
 ! 1 = Liquid aerosol
 ! 2 = Solid aerosol
```

#### REVISION HISTORY:

18 Apr 2013 - S. D. Eastham - Initial version

---

### 7.9.15 get\_strat\_opt

Subroutine GET\_STRAT\_OPT returns local optical properties for a given stratospheric aerosol. The routine is essentially just an interface to allow external routines to "see" the arrays. However, local aerosol radius is adjusted from liquid to effective radius for aerosol optical depth calculations with liquid aerosols.

#### INTERFACE:

```
SUBROUTINE GET_STRAT_OPT (I,J,L,IAER,RAER,REFF,SAD,XSA)
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J, L ! Grid indices
INTEGER, INTENT(IN) :: IAER ! Aerosol index
 ! 1 = Liquid aerosols
 ! 2 = Solid PSC
```

#### !OUTPUT VARIABLES:

```
REAL(fp), INTENT(OUT) :: REFF ! Effective radius (cm)
REAL(fp), INTENT(OUT) :: RAER ! Physical radius (cm)
REAL(fp), INTENT(OUT) :: SAD ! Surface area density (cm2/cm3)
REAL(fp), INTENT(OUT) :: XSA ! X-S area density (m2/m3)
```

#### REMARKS:

Seb Eastham writes: "I would edit GET\_STRAT\_OPT so that, when SAD is less than some small value (say 1 nm<sup>2</sup>/cm<sup>3</sup>, which is a vanishingly small surface area), it returns SADSTRAT=XSASTRAT=0.d0 and RAER=REFF=0.1d0 for safety's sake. I think that will prevent code blow-up later on."

#### REVISION HISTORY:

```
17 Apr 2013 - S. D. Eastham - Initial version
07 Apr 2015 - R. Yantosca - Add error check to prevent div-by-zero errors
 in other areas of GEOS-Chem caused by low SAD
```

### 7.9.16 ternary

Subroutine TERNARY calculates the composition of SSA/STS aerosols using a parameterization from Carslaw et al. "A Thermodynamic Model of the System HCl-HNO<sub>3</sub>-H<sub>2</sub>SO<sub>4</sub>-H<sub>2</sub>O, Including Solubilities of HBr, from 200 to 328 K". The bulk of this code was taken directly from the Global Modeling Initiative implementation by David Consideine.

#### INTERFACE:

```
SUBROUTINE TERNARY (PCENTER_IN,TCENTER_IN,H2OSUM_IN,H2SO4SUM,
& HNO3SUM,HC1SUM,HOC1SUM,HBRSum,HOBrSUM,
& W_H2SO4,W_H2O,W_HNO3,W_HCl,W_HOCl,W_HBr,W_HOBr,
& HNO3GASFRAC,HC1GASFRAC,HOC1GASFRAC,
& HBrGASFRAC,HOBrGASFRAC,SLA_VOL,SLA_RHO)
```



**USES:**

```

! Temporary - for debug
USE ERROR_MOD, ONLY : IT_IS_NAN,ERROR_STOP ! Test for NaN
USE ERROR_MOD, ONLY : SAFE_EXP,SAFE_DIV, DEBUG_MSG

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: PCENTER_IN ! Pressure (hPa)
REAL(fp), INTENT(IN) :: TCENTER_IN ! Temperature (K)
REAL(fp), INTENT(IN) :: H2OSUM_IN ! Total H2O mixing ratio
REAL(fp), INTENT(IN) :: H2SO4SUM ! Liquid H2SO4 mixing ratio
REAL(fp), INTENT(IN) :: HNO3SUM ! Total HNO3 mixing ratio
REAL(fp), INTENT(IN) :: HCLSUM ! Total HCl mixing ratio
REAL(fp), INTENT(IN) :: HOCLSUM ! Total HOCl mixing ratio
REAL(fp), INTENT(IN) :: HBrSUM ! Total HBr mixing ratio
REAL(fp), INTENT(IN) :: HOBrSUM ! Total HOBr mixing ratio

```

**!OUTPUT VARIABLES:**

```

REAL(fp), INTENT(OUT) :: W_H2SO4 ! kg H2SO4/kg SLA
REAL(fp), INTENT(OUT) :: W_H2O ! kg H2O /kg SLA
REAL(fp), INTENT(OUT) :: W_HNO3 ! kg HNO3 /kg SLA
REAL(fp), INTENT(OUT) :: W_HCl ! kg HCl /kg SLA
REAL(fp), INTENT(OUT) :: W_HOCl ! kg HOCl /kg SLA
REAL(fp), INTENT(OUT) :: W_HBr ! kg HBr /kg SLA
REAL(fp), INTENT(OUT) :: W_HOBr ! kg HOBr /kg SLA
REAL(fp), INTENT(OUT) :: HNO3GASFRAC ! Gas fraction HNO3
REAL(fp), INTENT(OUT) :: HClGASFRAC ! Gas fraction HCl
REAL(fp), INTENT(OUT) :: HOClGASFRAC ! Gas fraction HOCl
REAL(fp), INTENT(OUT) :: HBrGASFRAC ! Gas fraction HBr
REAL(fp), INTENT(OUT) :: HOBrGASFRAC ! Gas fraction HOBr
REAL(fp), INTENT(OUT) :: SLA_VOL ! Aerosol volume (m3/m3)
REAL(fp), INTENT(OUT) :: SLA_RHO ! Aer. mass density (kg/m3)

```

**REVISION HISTORY:**

```

19 Apr 2013 - S. D. Eastham - Initial version
16 Apr 2015 - M. Yannetti - Changed EXP to SAFE_EXP in H_HCL
01 Jan 2016 - E. Lundgren - Calculate R_ATM from global params
29 Dec 2017 - C. Keller - Add error traps to prevent float invalids

```

**7.9.17 carslaw\_density**

Function CARSLAW\_DENSITY determines the density of a sol'n through a relationship from Carslaw et al.. Result is in kg/m3.

**INTERFACE:**

```

REAL(fp) FUNCTION CARSLAW_DENSITY(CS,CN,T)

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: CS ! H2SO4 molality (mol H2SO4/kg solvent)
REAL(fp), INTENT(IN) :: CN ! HNO3 molality (mol HNO3/kg solvent)
REAL(fp), INTENT(IN) :: T ! Temperature (K)

```

**REVISION HISTORY:**

```

19 Apr 2013 - S. D. Eastham - Initial version
29 Dec 2017 - C. Keler - Added error traps for zeros

```

---

**7.9.18 calc\_fallvel**

Function CALC\_FALLVEL calculates the terminal velocity of a solid particle.

**INTERFACE:**

```

FUNCTION CALC_FALLVEL(DENSITY,RADIUS,TCENTER,PCENTER) RESULT(VEL)

```

**USES:**

```

USE ERROR_MOD, ONLY : ERROR_STOP

```

**INPUT PARAMETERS:**

```

REAL(fp),INTENT(IN) :: RADIUS ! Particle radius (cm)
REAL(fp),INTENT(IN) :: DENSITY ! Particle density (kg/m3)
REAL(fp),INTENT(IN) :: TCENTER ! Local temperature (K)
REAL(fp),INTENT(IN) :: PCENTER ! Local pressure (kPa)
!OUTPUT VARIABLES:
REAL(fp) :: VEL ! Fall velocity (m/s)

```

**REMARKS:**

(1) A remark

**REVISION HISTORY:**

```

11 Aug 2012 - S. D. Eastham - Initial version

```

---

**7.9.19 cacl\_sla\_gamma**

Subroutine CALC\_SLA\_GAMMA calculates 11 different sticking coefficients on the surface of local stratospheric liquid aerosols, relevant to each of the 11 reactions listed in Kirner's paper.

**INTERFACE:**

```

SUBROUTINE CALC_SLA_GAMMA(NDENS, T, P, WT_FRC, H2OSUM, HCLSUM,
& HBrSUM, HOBrSUM, ClNO3SUM, BrNO3SUM,
& RHO, ARAD, RXNGAMMA)

```

**USES:**

```
! Temporary - for debug
USE ERROR_MOD, ONLY : IT_IS_NAN,ERROR_STOP ! Test for NaN
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: NDENS ! Air number density (molec/cm3)
REAL(fp), INTENT(IN) :: T ! Temperature (K)
REAL(fp), INTENT(IN) :: P ! Pressure (hPa)
REAL(fp), INTENT(IN) :: WT_FRC ! Weight fraction of H2SO4 (kg/kg)
REAL(fp), INTENT(IN) :: H2OSUM ! H2O mixing ratio
REAL(fp), INTENT(IN) :: HClSUM ! HCl mixing ratio
REAL(fp), INTENT(IN) :: HBrSUM ! HBr mixing ratio
REAL(fp), INTENT(IN) :: HOBrSUM ! HOBr mixing ratio
REAL(fp), INTENT(IN) :: ClNO3SUM ! ClNO3 mixing ratio
REAL(fp), INTENT(IN) :: BrNO3SUM ! BrNO3 mixing ratio
REAL(fp), INTENT(IN) :: RHO ! STS density (g/cm3)
REAL(fp), INTENT(IN) :: ARAD ! SLA radius (cm)
!OUTPUT VARIABLES:
REAL(fp), INTENT(OUT) :: RXNGAMMA(11) ! Premultiplying factors
```

**REVISION HISTORY:**

```
10 Oct 2012 - S. D. Eastham - Initial version
15 Apr 2015 - M. Yannetti - Adjustment to f8 from fp due to overflow concerns
29 Dec 2017 - C. Keller - Avoid div by zero.
```

**7.9.20 molec\_speed**

Function MOLEC\_SPEED calculates the mean velocity of gas phase particles based on temperature and molecular mass.

**INTERFACE:**

```
REAL(fp) FUNCTION MOLEC_SPEED(T,MOLMASS)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: T ! Temperature (K)
REAL(fp), INTENT(IN) :: MOLMASS ! Molecular mass (g/mol)
```

**REVISION HISTORY:**

```
10 Oct 2012 - S. D. Eastham - Initial version
```

**7.9.21 read\_psc\_file**

Subroutine READ\_PSC\_FILE initializes PSC state information from a checkpoint file (binary punch file format).

**INTERFACE:**

```
SUBROUTINE READ_PSC_FILE(am_I_Root, State_Chm, RC)
```

**USES:**

```
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_ERROR_MOD
USE HCO_RESTART_MOD, ONLY : HCO_RestartGet
USE HCO_CLOCK_MOD, ONLY : HcoClock_First
USE HCO_CLOCK_MOD, ONLY : HcoClock_Rewind
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE State_Chm_Mod, ONLY : ChmState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: RC
```

**REVISION HISTORY:**

```
28 Apr 2013 - S. D. Eastham - Initial version, based on READ_CSPEC_FILE
14 Feb 2014 - R. Yantosca - Reorder DO loops for efficiency
16 Apr 2014 - M. Sulprizio - Now get PSC restart file path from Input_Opt
23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN_mod.F
14 Jan 2015 - C. Keller - Now read from HEMCO
 4 Mar 2015 - R. Yantosca - Declare pointer args to HCO_GetPtr w/ REAL(f4)
25 Mar 2015 - C. Keller - Now use HEMCO restart module
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
```

**7.9.22 set\_clock\_trac**

Subroutine SET\_CLOCK\_TRAC sets the clock species mixing ratio within the bottom five grid levels, increasing by a fixed rate of 0.5 ppbv/day

**INTERFACE:**

```
SUBROUTINE SET_CLOCK_TRAC(STEPLEN, State_Chm)
```

**USES:**

```

USE CMN_SIZE_MOD
USE State_Chm_Mod, ONLY : ChmState

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: STEPLEN ! Step length (sec)

```

**OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(OUT) :: State_Chm ! Chemistry State object

```

**REVISION HISTORY:**

```

16 Oct 2013 - S. D. Eastham - Initial version
14 Feb 2014 - R. Yantosca - Reorder DO loop for efficiency
21 Feb 2014 - M. Sulprizio - Now pass State_Chm object via the arg list
06 Feb 2018 - E. Lundgren - Update timestep unit from min to sec

```

---

**7.9.23 set\_h2o\_trac**

Subroutine SET\_H2O\_TRAC sets the H2O species throughout the selected domain (either troposphere only or the full grid).

**INTERFACE:**

```

SUBROUTINE SET_H2O_TRAC (am_I_Root, SETSTRAT, Input_Opt,
& State_Met, State_Chm, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE DAO_MOD, ONLY : AIRQNT
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState, Ind_
USE State_Met_Mod, ONLY : MetState
USE UnitConv_Mod

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
LOGICAL, INTENT(IN) :: SETSTRAT ! Set strat H2O?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
28 Mar 2013 - S. D. Eastham - Initial version
14 Feb 2014 - R. Yantosca - Reordered DO loop for efficiency
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
24 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID and
 remove dependency on PRESSURE_MOD
24 Mar 2015 - E. Lundgren - Change tracer units from kg to kg/kg
29 Apr 2015 - E. Lundgren - Now pass am_I_Root and RC to SET_H2O_TRAC
29 Apr 2015 - E. Lundgren - Now use grid box moist air mass ADMOIST
 with SPHU since AD is now dry air mass
29 Apr 2015 - E. Lundgren - Replace RH calculation method with Nordquist,
 1973 and call AirQnt if moisture vars
 are updated using H2O tracer concentration
28 Oct 2015 - E. Lundgren - Tracer units are now kg/kg dry air
07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer
18 Jul 2016 - E. Lundgren - Remove dependency on grid box mass
11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
24 May 2016 - C. Holmes - Apply H2O from met fields up to the cold point
 tropopause in the tropics (30S-30N) or the
 lapse-rate tropopause, whichever is higher
13 Mar 2017 - C. Keller - Add error trap for id_H2O
12 Nov 2017 - C. Keller - Add option to prescribe trop. H2O up to
 tropopause plus offset.
```

**7.9.24 ucx\_h2so4phot**

Subroutine UCX\_H2SO4PHOT propagates the calculated H2SO4 photolysis (J) rate at the top of the chemistry grid through to the top of the transport grid, approximating H2SO4 photolysis in the mesosphere.

**INTERFACE:**

```
SUBROUTINE UCX_H2SO4PHOT(Input_Opt, State_Met, State_Chm)
```

**USES:**

```
USE CMN_FJX_MOD, ONLY : ZPJ
USE FAST_JX_MOD, ONLY : RXN_H2SO4
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

**INPUT PARAMETERS:**

```

TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**REMARKS:**

(1) A remark

**REVISION HISTORY:**

```

17 Aug 2013 - S. D. Eastham - Initial version
14 Feb 2014 - R. Yantosca - Reorder DO loops for efficiency
21 Feb 2014 - M. Sulprizio - Now pass Input_Opt, State_Met, and State_Chm
 objects via the arg list
31 May 2016 - E. Lundgren - Replace Input_Opt%TRACER_MW_G with species
 database emMW_g (emitted species g/mol)
21 Jun 2016 - R. Yantosca - Remove reference to tracerid_mod.F
27 Jun 2016 - M. Sulprizio - Obtain photolysis rate directly from ZPJ array
 and remove reference to FJXFUNC and obsolete
 SMVGEAR variables like NKS04PHOT, NAMEGAS, etc.
07 Jul 2016 - E. Lundgren - Replace tracer pointer with species pointer

```

**7.9.25 sfcmr\_init**

Subroutine SFCMR\_INIT initializes the surface mixing ratio derived type object. This reads in all the mixing ratios from ASCII files and saves them in the SFCMR object.

**INTERFACE:**

```

SUBROUTINE SFCMR_INIT(am_I_Root, Input_Opt)

```

**USES:**

```

USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
!OUTPUT VARIABLES:

```

**REVISION HISTORY:**

```

05 Dec 2014 - C. Keller - Initial version

```

**7.9.26 sfcmr\_read**

Subroutine SFCMR\_READ reads the surface mixing ratios from ASCII file and stores the values in the next available SFCMR object. The ascii file is expected to be in the following format: C2CL3F3 MIXING RATIO BOUNDARY CONDITIONS FOR WMO-2006 1.0E-12 time CFC113 1959.04 0.99 1959.12 1.00 1959.21 1.01 1959.29 1.02

**INTERFACE:**

```
SUBROUTINE SFCMR_READ(am_I_Root, Input_Opt, iName, iID, iMR)
```

**USES:**

```
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
CHARACTER(LEN=*), INTENT(IN) :: iName ! species name
INTEGER, INTENT(IN) :: iID ! species ID
REAL(fp), OPTIONAL, INTENT(IN) :: iMR ! fixed MR
```

**REVISION HISTORY:**

05 Dec 2014 - C. Keller - Initial version

---

**7.9.27 sfcmr\_get**

Subroutine SFCMR\_GET returns the surface mixing ratio of the given species for the given year and month. The value of the closest available date in the past is taken, and no interpolation between dates is performed.

**INTERFACE:**

```
FUNCTION SFCMR_GET(TRAC_NAME, YEAR, MONTH) RESULT(MR)
```

**USES:****INPUT PARAMETERS:**

```
CHARACTER(LEN=*), INTENT(IN) :: TRAC_NAME ! species name
INTEGER, INTENT(IN) :: YEAR ! desired year
INTEGER, INTENT(IN) :: MONTH ! desired month
!INPUT/OUTPUT VARIABLES:
REAL(fp) :: MR
```

**REVISION HISTORY:**

05 Dec 2014 - C. Keller - Initial version

---



**7.9.28 noxcoeff\_init**

Subroutine NOXCOEFF\_INIT initializes the NOX 2D interpolation values.

**INTERFACE:**

```
SUBROUTINE NOXCOEFF_INIT (am_I_Root, Input_Opt)
```

**USES:**

```
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
!OUTPUT VARIABLES:
```

**REVISION HISTORY:**

05 Dec 2014 - C. Keller - Initial version

---

**7.9.29 get\_jjnox**

Subroutine GET\_JJNOX maps grid box at location IISIM, JJSIM of the simulation grid onto the latitude grid of the NOXCOEFF array. JJNOX can differ from JJSIM if it's not a 4x5 or 2x25 simulation.

This routine simply returns the index of the NOx latitude vector that covers the latitude value of interest. No grid box weighting, etc. is performed.

**INTERFACE:**

```
FUNCTION GET_JJNOX(IISIM, JJSIM) RESULT (JJNOX)
```

**USES:**

```
USE GC_GRID_MOD, ONLY : GET_YMID
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: IISIM ! Latitude index on simulation grid
INTEGER, INTENT(IN) :: JJSIM ! Latitude index on simulation grid
!OUTPUT VARIABLES:
INTEGER :: JJNOX ! Latitude index on NOXCOEFF grid
```

**REVISION HISTORY:**

05 Dec 2014 - C. Keller - Initial version

---

### 7.9.30 init\_ucx

Subroutine INIT\_UCX initializes module arrays.

#### INTERFACE:

```
SUBROUTINE INIT_UCX(am_I_Root, Input_Opt, State_Chm, State_Diag)
```

#### USES:

```
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : IS_SAFE_DIV
USE ERROR_MOD, ONLY : ERROR_STOP
USE GC_GRID_MOD, ONLY : GET_YEDGE
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : PI_180
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_YEAR
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input options
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object
```

#### REVISION HISTORY:

```
04 Apr 2013 - S. D. Eastham - Initial version
21 Jun 2016 - R. Yantosca - Now declare species ID's during INIT phase
21 Jun 2016 - R. Yantosca - Remove reference to tracerid_mod.F
24 Jun 2016 - R. Yantosca - Bug fix: get advected species index for H2
20 Sep 2016 - R. Yantosca - Now avoid using SIND, convert the argument
 to radians and use the SIN function instead
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
```

### 7.9.31 diaginit\_ucx

Subroutine DIAGINIT\_UCX initializes diagnostics containers for the UCX module.

#### INTERFACE:

```
SUBROUTINE DIAGINIT_UCX(am_I_Root, Input_Opt, State_Chm, RC)
```

#### USES:

```
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_RESTART_MOD, ONLY : HCO_RestartDefine
```

```

USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_ERROR_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?!
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object

```

**INPUT/OUTPUT PARAMETERS:**

```

INTEGER, INTENT(INOUT) :: RC ! Success or failure

```

**REVISION HISTORY:**

```

14 Jan 2015 - C. Keller - Initial version
25 Mar 2015 - C. Keller - Now use HEMCO restart module
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

```

---

**7.9.32 write\_state\_psc**

Subroutine WRITE\_STATE\_PSC saves the STATE\_PSC array into diagnostics. This is only of relevance in an ESMF environment, where STATE\_PSC is not automatically written into the HEMCO restart diagnostics but needs to be passed explicitly to the internal state. This should be done on every time step to make sure that replay simulations will always have the most current STATE\_PSC values in the internal state (important for checkpointing!).

**INTERFACE:**

```

SUBROUTINE WRITE_STATE_PSC (am_I_Root, State_Chm, RC)

```

**USES:**

```

USE ErrCode_Mod
USE HCO_ERROR_MOD
USE HCO_RESTART_MOD, ONLY : HCO_RestartWrite
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE State_Chm_Mod, ONLY : ChmState

```

**!INPUT ARGUMENTS:**

```

LOGICAL, INTENT(IN) :: am_I_Root

```

**!INPUT/OUTPUT ARGUMENTS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: RC

```

**REVISION HISTORY:**

```

08 May 2015 - C. Keller - Initial version
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

```

---

### 7.9.33 cleanup\_ucx

Subroutine CLEANUP\_UCX deallocates module variables.

#### INTERFACE:

```
SUBROUTINE CLEANUP_UCX (am_I_Root)

!INPUT ARGUMENTS:
 LOGICAL, INTENT(IN) :: am_I_Root
```

#### REVISION HISTORY:

04 Apr 2013 - S. D. Eastham - Initial version  
13 Apr 2013 - S. D. Eastham - Added PSC arrays

---

## 7.10 Fortran: Module Interface bromocarb\_mod.F

Module BROMOCARB\_MOD contains variables and routines for the GEOS-Chem bromo-carbon simulation.

#### INTERFACE:

```
MODULE BROMOCARB_MOD
```

#### USES:

```
USE PRECISION_MOD

IMPLICIT NONE
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: SET_CH3Br
PUBLIC :: SET_BrO
```

#### REMARKS:

HEMCO has made most of this routine obsolete. HEMCO reads in the data files in netCDF format and tracks them in its data structure.

#### REVISION HISTORY:

13 Aug 2007 - J. Parrella - Initial version  
22 May 2012 - M. Payer - Added ProTeX headers  
27 Aug 2012 - M. Payer - Now parallelize DO loops  
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
24 Jun 2014 - R. Yantosca - Removed code made obsolete by HEMCO  
24 Jun 2014 - R. Yantosca - Removed INIT\_BROMOCARB, CLEANUP\_BROMOCARB  
06 Nov 2015 - M. Yannetti - Added PRECISION\_MOD  
29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

---

**7.10.1 set\_ch3br**

Subroutine SET\_CH3Br set CH3Br Concentrations in the planetary boundary layer. Based on latitude bands (1) 90-55N, (2) 55N-0, (3) 0-55S, (4) 55-90S. Values for setting pbl flux were determined by surface measurements from NOAA 2006 data.

**INTERFACE:**

```

SUBROUTINE SET_CH3Br(am_I_Root, Input_Opt, State_Met,
& State_Chm, RC
)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_UNDER_PBLTOP
USE PhysConstants, ONLY : AIRMW
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

- 1) Spc is converted back and forth between units of [kg] and [v/v]. Placement of the call to SET\_CH3Br in main.f (it's with the emissions) means that it should be in [kg].

**REVISION HISTORY:**

```

12 Feb 2008 - J. Parrella - Initial version
22 May 2012 - M. Payer - Added ProTeX headers
27 Aug 2012 - M. Payer - Added parallel DO loop
28 Aug 2012 - M. Payer - Add error check for CH3Br to avoid OOB error
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object

```

- 24 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Met, State\_Chm. Get fields from these objects.
- 10 Sep 2015 - E. Lundgren - Modification for State\_Chm%TRACERS units now in kg/kg dry air (previously kg/box)
- 10 Sep 2015 - E. Lundgren - Remove passed argument 'unitflag'
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.
- 25 Jul 2016 - M. Yannetti - Replaced TCVV with MW from spec db and phys const

### 7.10.2 set\_bro

Subroutine SET\_BRO sets BrO concentrations in the planetary boundary layer. Based on latitude bands (1) 90-55N, (2) 55N-0, (3) 0-55S, (4) 55-90S. Values for setting pbl flux were determined by surface measurements from NOAA 2006 data.

#### INTERFACE:

```

SUBROUTINE SET_BRO(am_I_Root, Input_Opt, State_Met,
& State_Chm, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
USE DAO_MOD, ONLY : IS_WATER
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_UNDER_PBLTOP
USE PhysConstants, ONLY : AIRMW
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Met State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REMARKS:

- 1) Spc is converted back and forth between units of [kg] and [v/v]. Placement of the call to SET\_Bro in main.f (it's with the emissions) means that it should be in [kg].

## REVISION HISTORY:

- 12 Feb 2008 - J. Parrella - Initial version  
 22 May 2012 - M. Payer - Added ProTeX headers  
 27 Aug 2012 - M. Payer - Added parallel DO loop  
 28 Aug 2012 - M. Payer - Add error check for BrO to avoid OOB error  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object  
 27 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
 24 Jun 2014 - R. Yantosca - Remove IJLOOP, it's not used anymore  
 06 Nov 2014 - M. Yannetti - Changed REAL\*8 to REAL(fp)  
 10 Sep 2015 - E. Lundgren - Modification for State\_Chm%TRACERS units now in kg/kg dry air (previously kg/box)  
 10 Sep 2015 - E. Lundgren - Remove passed argument 'unitflag'  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.  
 25 Jul 2016 - M. Yannetti - Replaced TCVV with MW from spec db and phys const

## 7.11 Fortran: Module Interface *cldice\_hbrhobr\_rxn.F*

Subroutine CLDICE\_HBrHOB\_RXN calculates the rate constants for HBr and HOBr pseudo-reactions with ice.

### INTERFACE:

```

SUBROUTINE CLDICE_HBrHOB_RXN(I, J, L, DENAIR,
& QI, hbr, hobr,
& k_hbr, k_hobr, AREA, State_Met)

```

### USES:

```

USE ERROR_MOD, ONLY : IS_SAFE_DIV, IT_IS_NAN
USE ERROR_MOD, ONLY : GEOS_CHEM_STOP
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
USE State_Met_Mod, ONLY : MetState

```

```

IMPLICIT NONE

```

### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index

```

```

REAL(fp), INTENT(IN) :: DENAIR ! Density of air [# / cm3]
REAL(fp), INTENT(IN) :: QI ! Cloud ice mixing ratio [kg/kg]
REAL(fp), INTENT(IN) :: hbr ! Concentration of HBr [# / cm3]
REAL(fp), INTENT(IN) :: hobr ! Concentration of HOBr [# / cm3]
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**OUTPUT PARAMETERS:**

```

REAL(fp), INTENT(OUT) :: k_hbr ! Rate constant for HBr + ice
 ! pseudo-rxn [cm3/s]
REAL(fp), INTENT(OUT) :: k_hobr ! Rate constant for HOBr + ice
 ! pseudo-rxn [cm3/s]
REAL(fp), INTENT(OUT) :: AREA ! Surface area [cm2/cm3]

```

**REMARKS:**

The rate constant is calculated assuming:

1. A sticking coefficient of 0.1 [JPL 2006], Abbatt [1994], Chai et al. [2000]
2. An effective radius is assumed as a function of (i) temperature and ice water content (IWC). This relationship is taken from Wyser [1998].

\*\* Calculations of a 1st order rate constant are borrowed from the subroutine arsl1k.F. Below are comments from that code:

The 1st-order loss rate on wet aerosol (Dentener's Thesis, p. 14) is computed as:

$$\text{ARSL1K [1/s]} = \text{area} / [\text{radius}/\text{dfkg} + 4./(\text{stkcf} * \text{nu}) ]$$

where nu = Mean molecular speed [cm/s] =  $\sqrt{8R*TK/\pi/M}$  for Maxwell  
DFKG = Gas phase diffusion coeff [cm2/s] (order of 0.1)

**REVISION HISTORY:**

```

16 Jun 2011 - J. Parrella - Initial version
22 May 2012 - M. Payer - Added ProTeX headers
26 Sep 2012 - R. Yantosca - For now, comment out debug print statements
23 Oct 2012 - R. Yantosca - Add better error checks to prevent div-by-zero
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
05 Sep 2013 - R. Yantosca - Now exit if IWC <= 0, this will cause the
 logarithm to choke
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
06 Nov 2014 - R. Yantosca - Now use State_Met%CLDF(I,J,L)
19 Dec 2014 - R. Yantosca - Now perform internal computations w/ REAL*8
11 Aug 2015 - R. Yantosca - MERRA2 behaves the same way as GEOS-FP
04 Jan 2016 - E. Lundgren - Remove grid box area-dependence

```

---



## 7.12 Fortran: Module Interface strat\_chem\_mod.F90

Module STRAT\_CHEM\_MOD contains variables and routines for performing a simple linearized chemistry scheme.

If using a tropchem-based simulation, the simple linearized chemistry here will be applied to the stratosphere using archived 3D monthly production rates and loss frequencies saved from a GEOS-Chem UCX simulation.

If using a UCX-based simulation, the simple linearized chemistry here will be applied to the mesosphere instead using archived 3D monthly climatological production rates and loss frequencies from the GMI combo model.

In the original schem code (schem.F), only the following species were destroyed by photolysis in the stratosphere: PAN, H2O2, ACET, MEK, ALD2, RCHO, MVK, MACR, R4N2, CH2O, N2O5, HNO4, MP and by reaction with OH: ALK4, ISOP, H2O2, ACET, MEK, ALD2, RCHO, MVK, MACR, PMN, R4N2, PRPE, C3H8, CH2O, C2H6, HNO4, MP

The updated code includes at least all of these, and many more. The code is flexible enough to automatically apply the rate to any new species for future simulations that share the name in species\_mod with the UCX or GMI name. (See Documentation on wiki).

The prod rates and loss frequencies are now read via HEMCO. They are stored in a data structure of flexible length (PLVEC). The file containing the prod rates and loss frequencies need to be specified in the HEMCO configuration file for each species of interest. They are then automatically read and remapped onto the simulation grid.

The field names assigned to the production and loss fields are expected to be 'UCX\_PROD\_XXX' and 'UCX\_LOSS\_XXX' (or 'GMI\_PROD\_XXX' and 'GMI\_LOSS\_XXX'), respectively, where XXX is the species name. Production rates must be given in units of v/v/s, and loss frequencies in s<sup>-1</sup>.

The module variable PLMUSTFIND (set below) determines the behavior if no production rates and/or loss frequencies can be found for any of the UCX/GMI species defined in this module. IF PLMUSTFIND is set to TRUE, the code stops with an error if no entry is found. Otherwise, steady-state values are used for all species with no explicitly given values.

The (monthly) OH concentrations are also obtained through HEMCO. The field name must be 'STRAT\_OH', and values must be in v/v.

### INTERFACE:

```
MODULE Strat_Chem_Mod
```

### USES:

```
 for precisions
 USE HCO_Error_Mod
 USE Precision_Mod ! For GEOS-Chem Precision (fp, f4, f8)
```

```
 IMPLICIT NONE
 PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
 PUBLIC :: Init_Strat_Chem
 PUBLIC :: Do_Strat_Chem
 PUBLIC :: Cleanup_Strat_Chem
 PUBLIC :: Calc_STE
```

**PRIVATE MEMBER FUNCTIONS:**

```
PRIVATE :: Set_BryPointers
PRIVATE :: Set_PLVEC
PRIVATE :: Do_Synoz
```

**PUBLIC DATA MEMBERS:**

```
PUBLIC :: SChem_Tend
PUBLIC :: Minit_Is_Set
```

**REMARKS:**

References:

=====

( 1 )

**REVISION HISTORY:**

```
01 Feb 2011 - L. Murray - Initial version
20 Jul 2012 - R. Yantosca - Reorganized declarations for clarity
20 Jul 2012 - R. Yantosca - Correct compilation error in GET_RATES_INTERP
07 Aug 2012 - R. Yantosca - Fix parallelization problem in Bry do loop
05 Oct 2012 - R. Yantosca - Add bug fix for IFORT 12 compiler in CALC_STE
14 Mar 2013 - M. Payer - Replace 0x with 03 as part of removal of NOx-0x
 partitioning
20 Nov 2014 - M. Yannetti - Added PRECISION_MOD
30 Dec 2014 - C. Keller - Now read Bry data through HEMCO
16 Jan 2015 - C. Keller - Now read all prod/loss fields and OH conc.
 through HEMCO.
 4 Mar 2015 - R. Yantosca - Declare pointer args for HCO_GetPtr as REAL(f4)
06 Apr 2016 - C. Keller - Add Minit_Is_Set and SET_MINIT.
03 Oct 2016 - R. Yantosca - Now dynamically allocate BrPtrDay, BrPtrNight
03 Oct 2016 - R. Yantosca - Dynamically allocate BrPtrDay, BrPtrNight
30 May 2017 - M. Sulprizio- SChem_Tend is now public so that it can be
 updated in flexchem_mod.F90 for UCX simulations
24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA
06 Mar 2018 - M. Sulprizio- Update module to use prod/loss rates from UCX
 instead of GMI
23 Mar 2018 - M. Sulprizio- Restore using prod/loss rates from GMI for UCX-
 based simulations. Tropchem-based simulations
 will use the rates from UCX.
```

---

**7.12.1 Do\_Strat\_Chem**

Function DO\_STRAT\_CHEM is the driver routine for computing the simple linearized stratospheric chemistry scheme.

**INTERFACE:**

```

SUBROUTINE DO_STRAT_CHEM(am_I_Root, Input_Opt, &
 State_Met, State_Chm, errCode)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE LINOZ_MOD, ONLY : DO_LINOZ
USE PhysConstants, ONLY : XNUMOLAIR, AIRMW, AVO
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH
USE TIME_MOD, ONLY : TIMESTAMP_STRING
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

```

IMPLICIT NONE

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(INOUT) :: errCode ! Success or failure

```

**REMARKS:****REVISION HISTORY:**

```

01 Feb 2011 - L. Murray - Initial version
18 Jul 2012 - R. Yantosca - For compatibility w/ the GEOS-5/GCM, we cannot
 assume a minimum tropopause level anymore
18 Jul 2012 - R. Yantosca - Make sure I is the innermost DO loop
 wherever expedient
20 Jul 2012 - R. Yantosca - Reorganized declarations for clarity
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
07 Aug 2012 - R. Yantosca - Make BEFORE a local variable for parallel loop
26 Oct 2012 - R. Yantosca - Now pass the Chemistry State object for GIGC
09 Nov 2012 - R. Yantosca - Now pass the Input Options object for GIGC
15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met

```

```

 derived type object
27 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
14 Mar 2013 - M. Payer - Replace Ox with O3 as part of removal of NOx-Ox
 partitioning
18 Mar 2013 - R. Yantosca - Now pass Input_Opt via the arg list
19 Mar 2013 - R. Yantosca - Now only copy Input_Opt%TCVV(1:N_TRACERS)
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
30 Dec 2014 - C. Keller - Now get Bry data through HEMCO.
24 Mar 2015 - E. Lundgren - Replace dependency on tracer_mod with
 CMN_GTCM_MOD for XNUMOLAIR
30 Sep 2015 - E. Lundgren - Now use UNITCONV_MOD for unit conversion
05 Mar 2016 - C. Keller - Allow O3 P/L be done by GMI if both LINOZ and
 SYNOZ are disabled. This is primarily for
 testing/data assimilation applications.
16 Jun 2016 - M. Yannetti - Replaced TRACERID_MOD.\
20 Jun 2016 - R. Yantosca - Now make species ID flags module variables
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
12 Jul 2016 - R. Yantosca - Bug fix: ISBR2 should be held !$OMP PRIVATE
18 Jul 2016 - M. Yannetti - Replaced TCVV with spec db and phys constant
10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
19 Oct 2016 - R. Yantosca - Add routine Set_Init_Conc_Strat_Chem for GCHP
28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine
03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input_Opt%LUCX
17 Jan 2018 - R. Yantosca - Replace GET_TPAUSE_LEVEL w/ State_Met%TropLev
09 Apr 2018 - R. Yantosca - Now use safe division in expression for k

```

### 7.12.2 Set\_BryPointers

Subroutine SET\_BryPointers gets the Bry stratospheric data read by HEMCO. The pointers only need to be established once. Target data is automatically updated through HEMCO.

#### INTERFACE:

```
SUBROUTINE Set_BryPointers(am_I_Root, Input_Opt, State_Chm, State_Met, RC)
```

#### USES:

```

USE ErrCode_Mod
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState

```

```
IMPLICIT NONE
```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorological State object

```

**INPUT/OUTPUT PARAMETERS:**

```

INTEGER, INTENT(INOUT) :: RC ! Success or failure

```

**REVISION HISTORY:**

30 Dec 2014 - C. Keller - Initial version

---

**7.12.3 Set\_Plvec**

Subroutine SET\_PLVEC gets the production and loss terms of all strat chem species from HEMCO. The pointers only need to be established once. Target data is automatically updated through HEMCO.

**INTERFACE:**

```

SUBROUTINE Set_PLVEC(am_I_Root, Input_Opt, State_Chm, State_Met, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

```

IMPLICIT NONE

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorological State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: RC ! Success or failure

```

**REVISION HISTORY:**

16 Jan 2015 - C. Keller - Initial version

---

### 7.12.4 Calc\_Ste

Subroutine CALC\_STE estimates what the stratosphere-to-troposphere exchange flux must have been since the last time it was reset

#### INTERFACE:

```
SUBROUTINE Calc_STE(am_I_Root, Input_Opt, State_Chm, State_Met, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TAU, GET_NYMD, GET_NHMS, EXPAND_DATE
USE UnitConv_Mod, ONLY : Convert_Spc_Units
```

```
IMPLICIT NONE
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REVISION HISTORY:

```
28 Apr 2012 - L. Murray - Initial version
18 Jul 2012 - R. Yantosca - Make sure I is the innermost DO loop
 (wherever expedient)
20 Jul 2012 - R. Yantosca - Reorganized declarations for clarity
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
05 Oct 2012 - R. Yantosca - Bug fix for IFORT 12: extend the #if statement
 to avoid including code for nested-grid sims
25 Mar 2013 - R. Yantosca - Now accept Input_Opt, State_Chm, RC arguments
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
10 Aug 2015 - E. Lundgren - Input tracer concentraton units are now [kg/kg]
25 May 2016 - E. Lundgren - Replace input_opt%TRACER_MW_KG with species
 database field emMW_g (emitted species g/mol)
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
```

species ID from State\_Chm%Map\_Advect.

01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo  
 10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code  
 26 Jun 2017 - R. Yantosca - GC\_ERROR is now contained in errcode\_mod.F90  
 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

---

### 7.12.5 Init\_Strat\_Chem

Subroutine INIT\_STRAT\_CHEM allocates all module arrays. It also opens the necessary rate files.

#### INTERFACE:

```
SUBROUTINE INIT_STRAT_CHEM(am_I_Root, Input_Opt, State_Chm, State_Met, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TAU
USE TIME_MOD, ONLY : GET_NYMD
USE TIME_MOD, ONLY : GET_NHMS
USE TIME_MOD, ONLY : GET_TS_CHEM
```

```
IMPLICIT NONE
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

#### REVISION HISTORY:

01 Feb 2011 - L. Murray - Initial version  
 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when  
 running with the traditional driver main.F  
 26 Oct 2012 - R. Yantosca - Now pass Chemistry State object for GIGC  
 09 Nov 2012 - R. Yantosca - Now pass Input Options object for GIGC  
 05 Nov 2013 - R. Yantosca - Now update tracer flags for tagOx simulation  
 03 Apr 2014 - R. Yantosca - PROD, LOSS, STRAT\_OH, MINIT, SCHEM\_TEND are  
 now REAL\*4, so use 0e0 to initialize  
 11 Aug 2015 - E. Lundgren - Tracer units are now kg/kg and are converted  
 kg for assignment of MInit  
 16 Jun 2016 - M. Yannetti - Replaced TRACERID\_MOD.  
 20 Jun 2016 - R. Yantosca - Now save species ID flags as module variables  
 and only define them in the INIT phase.  
 12 Jul 2016 - R. Yantosca - Now also store advected species ID's  
 11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code  
 20 Sep 2016 - R. Yantosca - Rewrote GMI\_TrName statement for Gfortran  
 26 Jun 2017 - R. Yantosca - GC\_ERROR is now contained in errcode\_mod.F90

### 7.12.6 Set\_Minit

Sets the MINIT array to current values in State\_Chm

#### INTERFACE:

```
SUBROUTINE SET_MINIT(am_I_Root, Input_Opt, State_Met, State_Chm, RC)
```

#### USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

```
IMPLICIT NONE
```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```



**REVISION HISTORY:**

06 Apr 2016 - C. Keller - Initial version: moved outside of  
INIT\_STRAT\_CHEM so that it can be called from  
within DO\_STRAT\_CHEM (for ESMF applications).  
15 Mar 2017 - E. Lundgren - Add unit catch and error handling  
26 Jun 2017 - R. Yantosca - GC\_ERROR is now contained in errcode\_mod.F90  
28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

---

**7.12.7 Cleanup\_Strat\_Chem**

Subroutine CLEANUP\_STRAT\_CHEM deallocates all module arrays.

**INTERFACE:**

SUBROUTINE CLEANUP\_STRAT\_CHEM

**USES:**

IMPLICIT NONE

**REVISION HISTORY:**

01 Feb 2011 - L. Murray - Initial version  
03 Oct 2016 - R. Yantosca - Deallocate BrPtrDay and BrPtrNight  
05 Oct 2016 - R. Yantosca - Now make deallocations more robust

---

**7.12.8 Do\_Synoz**

Subroutine Do\_Synoz establishes the flux boundary condition for Ozone coming down from the stratosphere, using the Synoz algorithm of McLinden et al, 2000.

**INTERFACE:**

SUBROUTINE Do\_Synoz( am\_I\_Root, Input\_Opt, State\_Met, State\_Chm, RC )

**USES:**

USE CMN\_SIZE\_MOD  
USE ErrCode\_Mod  
USE ERROR\_MOD, ONLY : ERROR\_STOP  
USE Input\_Opt\_Mod, ONLY : OptInput  
USE PhysConstants  
USE State\_Chm\_Mod, ONLY : ChmState  
USE State\_Met\_Mod, ONLY : MetState  
USE TIME\_MOD, ONLY : GET\_TS\_CHEM, GET\_YEAR

IMPLICIT NONE

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

Reference:

```

=====
C. A. McLinden, S. Olsen, B. Hannegan, O. Wild, M. J. Prather, and
J. Sundet, "Stratospheric Ozone in 3-D models: A simple chemistry
and the cross-tropopause flux".

```

```

%%
%% NOTE: This SYNOZ scheme is now obsolete, replaced by LINOZ %%
%% We keep this for backwards compatibility w/ older met fields %%
%%

```

**REVISION HISTORY:**

13 Dec 1999 - Q. Li, R. Martin - Initial version

- (1) The parameter Rdg0 from "CMN\_GCTM" = R / g0 = 28.97.
- (2) Pass PW = PS - PTOP to UPBDFLX via "CMN".
- (3) Now pass IORD, JORD, KORD as arguments (bmy, 12/6/00)
- (4) Now compute the proper value of PO3\_vmr that will yield 475 Tg O3/yr for various settings of IORD, JORD, KORD (rvm, bey, bmy, 12/5/00)

```

***** You must use this version of UPBDFLX_03 if you are *****
***** using the Parallel Processor TPCORE v. 7.1 *****

```

- (5) Added to "upbdflex\_mod.f". Also updated comments and made some cosmetic changes. (bmy, 6/28/01)
- (6) Now reference CMN\_SETUP for LSPLIT. Also store strat O3 into tracer #11 for multi-tracer Ox run. (amf, bmy, 7/3/01)
- (7) Removed IREF, JREF -- these are obsolete. Also T(IREF,JREF,L) is now T(I,J,L). (bmy, 9/27/01)
- (8) Also replace PW(I,J) with P(I,J) (bmy, 10/3/01)
- (9) Removed obsolete commented out code from 9/01 (bmy, 10/24/01)
- (10) Removed obsolete commented out code from 7/01 (bmy, 11/26/01)

- (11) Now write file names to stdout (bmy, 4/3/02)
- (12) Replaced all instances of IM with IIPAR and JM with JJPARG, in order to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (13) Now use GET\_PEDGE and GET\_PCENTER from "pressure\_mod.f" to compute the pressure at the bottom edge and center of grid box (I,J,L). Also removed obsolete, commented-out code. Removed G\_SIG and G\_SIGE from the arg list. (dsa, bdf, bmy, 8/21/02)
- (14) Now reference BXHEIGHT and T from "dao\_mod.f". Also reference routine ERROR\_STOP from "error\_mod.f". Now references IDTOX from F90 module "tracerid\_mod.f" instead of from "comtrid.h". (bmy, 11/6/02)
- (15) Now define J30S and J30N for 1x1 nested grid (bmy, 3/11/03)
- (16) Make sure to pass AD via "dao\_mod.f" for GEOS-1 (bnd, bmy, 4/14/03)
- (17) On the first timestep, print how much O3 flux is coming down from the stratosphere in Tg/yr. (mje, bmy, 8/15/03)
- (18) Change O3 flux to 500 Tg/yr for GEOS-3 (mje, bmy, 9/15/03)
- (19) Now calls routine ADD\_STRAT\_POX from "tagged\_ox\_mod.f" in order to pass stratospheric flux of Ox to the proper tagged tracer w/o resorting to hardwiring w/in this routine. (bmy, 8/18/03)
- (20) Add GEOS\_4 to the #if defined block. (bmy, 1/29/04)
- (21) Activated parallel DO-loops. Now made STFLUX a local array in order to facilitate parallelization. (bmy, 4/15/04)
- (22) Removed IORD, JORD, KORD from the arg list. Now reference STT and ITS\_A\_TAGOX\_SIM from "tracer\_mod.f". (bmy, 7/20/04)
- (23) Use an #ifdef block to comment out an EXIT statement from w/in a parallel loop for COMPAQ compiler. COMPAQ seems to have some problems with this. Now supports 1x125 grid. (auvray, bmy, 12/1/04)
- (24) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
- (25) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
- (26) Now set J30S and J30N for GEOS-5 nested grid (yxw, dan, bmy, 11/6/08)
- (27) Remove support for COMPAQ compiler (bmy, 7/8/09)
- (28) Now do not call ADD\_STRAT\_POX for tagged Ox (dbj, bmy, 10/16/09)
- 13 Aug 2010 - R. Yantosca - Treat MERRA like GEOS-5 (bmy, 8/13/10)
- 02 Dec 2010 - R. Yantosca - Added ProTeX headers
- 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.2 in the same way as MERRA
- 10 Feb 2012 - R. Yantosca - Modified for 0.25 x 0.3125 grids
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 28 Apr 2012 - L. Murray - Moved from upbdflex\_mod.F to here, modified to F90, renamed from UPBDFLEX\_O3 to DO\_SYNOZ. Use chem timestep now. Also, removed INIT\_UPBDFLEX, which was last used for GEOS-3.
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 04 Feb 2013 - M. Payer - Replace all JJPARG with values for nested grids since JJPARG is no longer a parameter
- 14 Mar 2013 - M. Payer - Replace Ox with O3 as part of removal of NOx-Ox partitioning
- 25 Mar 2013 - R. Yantosca - Now use explicit numbers for J30S, J30N
- 31 May 2013 - R. Yantosca - Now pass Input\_Opt, RC as arguments

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 26 Sep 2013 - R. Yantosca - Remove SEAC4RS C-preprocessor switch  
 26 Sep 2013 - R. Yantosca - Renamed GEOS\_57 Cpp switch to GEOS\_FP  
 05 Nov 2013 - R. Yantosca - Rename IDTOxStrt to id\_03Strat  
 23 Jan 2014 - M. Sulprizio- Linoz does not call UPBDFLX\_03. Synoz does.  
                   Now uncomment ADD\_STRAT\_POx (jtl,hyl,dbj,11/3/11)  
 26 Feb 2015 - E. Lundgren - Replace GET\_PEDGE and GET\_PCENTER with  
                   State\_Met%PEDGE and State\_Met%PMID. Remove  
                   dependency on pressure\_mod.  
 03 Mar 2015 - E. Lundgren - Use virtual temperature in hypsometric eqn  
 12 Aug 2015 - R. Yantosca - Add placeholder values for 0.5 x 0.625 grids  
 16 Jun 2016 - M. Yannetti - Replaced TRACERID\_MOD.  
 20 Jun 2016 - R. Yantosca - Now make species ID flags module variables  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
                   species ID from State\_Chm%Map\_Advect.  
 12 Jul 2016 - R. Yantosca - Remove references to ADD\_STRAT\_POX

### 7.13 Fortran: Module Interface linoz\_mod.F

Module LINOZ\_MOD contains routines to perform the Linoz stratospheric ozone chemistry.

#### INTERFACE:

```
MODULE LINOZ_MOD
```

#### USES:

```
USE ERROR_MOD, ONLY : DEBUG_MSG ! Routine for debug output
USE PRECISION_MOD ! For GEOS-Chem Precision (fp, f4, f8)
```

```
IMPLICIT NONE
```

```
PRIVATE
```

```
!PRIVATE DATA MEMBERS:
```

```
REAL(fp), ALLOCATABLE :: TLSTT(:, :, :, :)
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: CLEANUP_LINOZ
```

```
PUBLIC :: DO_LINOZ
```

```
PUBLIC :: INIT_LINOZ
```

```
PUBLIC :: LINOZ_READ
```

#### PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: LINOZ_CHEM3
```

```
PRIVATE :: LINOZ_STRATL
```

```
PRIVATE :: LINOZ_STRT2M
```

```
PRIVATE :: LINOZ_SOMLFQ
```

```
PRIVATE :: LINOZ_INTPL
```

**REMARKS:**

LINOZ Climatology:

=====

The LINOZ stratospheric chemistry tables for ozone consist of:

7 tables, each a function of:

- 12 months,
- 18 latitudes (-85 to 85 in 10 deg. increments)
- 25 altitudes ( z\*=10-58 km in 2 km increments)

The 7 data fields are:

- 1- ozone (Logan climatology), v/v
- 2- Temperature climatology, K
- 3- Column ozone climatology, Logan ozone integrated above box, DU
- 4- ozone (P-L) for climatological ozone, v/v/s
- 5- d(P-L) / dO3, 1/s
- 6- d(P-L) / dT, v/v/s/K
- 7- d(P-L) / d(column O3), v/v/s/DU

Implementation notes:

=====

Dylan Jones (dbj@atmos.physics.utoronto.ca) wrote:

Testing this code [in v8-02-04] was more difficult than I thought. I began by trying to compare the output of v8-02-04 with our previous runs with v8-02-01. I accounted for the changes in the transport\_mod.f and I tried to undo the changes in when the diagnostics are archived in v8-02-04, but I was still getting large differences between v8-02-04 and v8-02-01. I finally gave up on this since I may have made a mistake in reverting to the old way of doing the diagnostics in v8-02-04. In the end I took the new linoz code from v8-02-04 and used it in v8-02-01. I ran two GEOS-5 full chemistry simulations for 2007 and the output were consistent over the full year.

I think that it is safe to release [Linoz in v8-02-04]. However, we should acknowledge that it was [only] tested in v8-02-01, since I was not able to assess the quality of the output in v8-02-04.

Bob Yantosca (yantasca@seas.harvard.edu) wrote:

We have also modified the code for use within the GEOS-5 GCM. We now declare the TPARM array as part of the Input\_Opt object. The LINOZ climatology ASCII file is now read on the root CPU and MPI-broadcasted to the non-root CPUs. Also, the INIT\_LINOZ routine is now called not on the first chemistry timestep but rather in the initialization phase at the start of the run. (bmy, 3/18/13)

**REVISION HISTORY:**

23 Mar 2000 - P. Cameron-Smith - Initial version adapted heavily  
from McLinden's original file.

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem

28 May 2009 - D. Jones - Further modifications

18 Nov 2009 - D. Jones - Further modifications

01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90

01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire\_mod.F90

15 Mar 2013 - R. Yantosca - Now use fields from Input\_Opt and made other  
modifications for GIGC interface to GEOS-5 GCM

18 Mar 2013 - R. Yantosca - Comment out STRAT\_INIT, it's not called

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

17 Nov 2014 - M. Yannetti - Added PRECISION\_MOD

19 Oct 2015 - C. Keller - TLSTT is now 4D to work on curvilinear grids

29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

**7.13.1 do\_linoz**

Subroutine DO\_LINOZ is the main driver for the Linoz stratospheric Ozone chemistry package.

**INTERFACE:**

```

SUBROUTINE DO_LINOZ(am_I_Root, Input_Opt,
& State_Met, State_Chm, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH
USE TIME_MOD, ONLY : GET_TS_CHEM

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem  
 10 Jun 2012 - L. Murray - Move call to DO\_LINOZ from transport code to  
                   chemistry code, so the use of DT\_TS\_CHEM is now correct.  
 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when  
                   running with the traditional driver main.F  
 18 Mar 2013 - R. Yantosca - Now accept Input\_Opt, RC as arguments  
 25 Mar 2013 - M. Payer - Now pass State\_Chm object via the arg list

### 7.13.2 linoz\_chem3

Subroutine LINOZ\_CHEM3 applies linearized chemistry based on tables from PRATMO model using climatological T, O3, time of year

#### INTERFACE:

```

SUBROUTINE LINOZ_CHEM3(DTCHEM, am_I_Root, Input_Opt,
& State_Met, State_Chm, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_AREA_CM2
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AIRMW, AVO
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

REAL(fp), INTENT(IN) :: DTCHEM ! Time step [seconds]
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
! OUTPUT PARAMETERS
INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REMARKS:

Replace fields from tracer\_mod.f with fields from Input\_Opt. When we use GEOS-Chem within the GEOS-5 GCM, the fields within Input\_Opt will be read on the root CPU and MPI-broadcasted to all other CPUs.

#### REVISION HISTORY:

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem  
 18 Nov 2009 - D. Jones - For now, set tagged stratospheric  
 tracer to total O3 in the overworld  
 to avoid issues with spin ups  
 08 Feb 2010 - R. Yantosca - Deleted obsolete local variables  
 22 Oct 2010 - R. Yantosca - Added OMP parallel loop  
 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when  
 running with the traditional driver main.F  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met  
 derived type object  
 14 Mar 2013 - M. Payer - Replace O<sub>x</sub> with O<sub>3</sub> as part of removal of NO<sub>x</sub>-O<sub>x</sub>  
 partitioning  
 18 Mar 2013 - R. Yantosca - Now accept Input\_Opt, RC as arguments  
 19 Mar 2013 - R. Yantosca - Now copy Input\_Opt%TCVV(1:N\_TRACERS)  
 25 Mar 2013 - M. Payer - Now pass State\_Chm object via the arg list  
 06 Nov 2013 - R. Yantosca - Now activate the parallel loop. Unit testing  
 revealed that this no longer causes errors.  
 26 Feb 2015 - E. Lundgren - Replace GET\_PCENTER and GET\_PEDGE with  
 State\_Met%PMID and State\_Met%PEDGE. Remove  
 dependency on pressure\_mod.  
 19 Oct 2015 - C. Keller - TLSTT is now 4D to work on curvilinear grids  
 06 Jan 2016 - E. Lundgren - Use global physical parameters  
 16 Jun 2016 - R. Silvern - Now define species ID of O<sub>3</sub> with Ind\_function  
 17 Jun 2016 - R. Yantosca - Only look up species ID of O<sub>3</sub> on first call  
 22 Jun 2016 - M. Yannetti - Replace TCVV with spc db and physical constant  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
 species ID from State\_Chm%Map\_Advect.  
 11 Jul 2016 - R. Yantosca - Bug fix: SPC now points to State\_Chm%Species

### 7.13.3 linoz\_stratl

Subroutine LINOZ\_STRATL performs a monthly fixup of chemistry parameters for the Linoz stratospheric ozone chemistry.

#### INTERFACE:

```
SUBROUTINE LINOZ_STRATL(am_I_Root, Input_Opt, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE PRESSURE_MOD
USE TIME_MOD, ONLY : GET_MONTH
```

#### INPUT PARAMETERS:



```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

Replace size fields NLAT\_LINOZ etc. with fields from Input\_Opt. When we use GEOS-Chem within the GEOS-5 GCM, the fields within Input\_Opt will be read on the root CPU and MPI-broadcasted to all other CPUs.

The LINOZ climatology array is Input\_Opt%LINOZ\_TPARAM(25,18,12,N), which has the following dimensions

- \* 25 layers from 58 km to 10 km by 2 km intervals
- \* 18 latitudes (85S, 75S, ...85N)
- \* 12 months
- \* N fields (currently N=7)

**REVISION HISTORY:**

- 24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID(I,J,L) from grid\_mod.F90
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 18 Mar 2013 - R. Yantosca - Accept Input\_Opt, RC arguments. Use fields from Input\_Opt to facilitate GC/GEOS-5 interface
- 18 Mar 2013 - R. Yantosca - Cosmetic changes, updated comments
- 19 Oct 2015 - C. Keller - TLSTT is now 4D to work on curvilinear grids

**7.13.4 linoz\_strt2m**

Subroutine LINOZ\_STRT2M interpolates quantities from the LINOZ vertical grid to the GEOS-Chem vertical grid. It also computes the 1st & 2nd moments of the distribution.

**INTERFACE:**

```

SUBROUTINE LINOZ_STRT2M(am_I_Root, Input_Opt, NSTRT,
& STRTX, POL, STRTOL,
& STRT1L, STRT2L, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

!-----
! am_I_Root : Are we on the root CPU?
! Input_Opt : Input Options object
! NSTRT : # of levels in the GEOS-Chem grid (= LLPAR)
! STRTX : Quantity on the LINOZ vertical grid
! (i.e. fields #1-7 of the LINOZ climatology)
! POL : Pressure edges on the GEOS-Chem grid
!-----
LOGICAL, INTENT(IN) :: am_I_Root
TYPE(Input_Opt), INTENT(IN) :: Input_Opt
INTEGER, INTENT(IN) :: NSTRT
REAL(fp), INTENT(IN) :: STRTX(Input_Opt%LINOZ_NLEVELS)
REAL(fp), INTENT(IN) :: POL(LLPAR+1)

```

**OUTPUT PARAMETERS:**

```

!-----
! STRTOL : 0th moment of distribution, on GEOS-Chem grid edges
! STRT1L : 1st moment of distribution, on GEOS-Chem grid edges
! STRT2L : 2nd moment of distribution, on GEOS-Chem grid edges
! RC : Success or failure?
!-----
REAL(fp), INTENT(OUT) :: STRTOL(LLPAR+1)
REAL(fp), INTENT(OUT) :: STRT1L(LLPAR+1)
REAL(fp), INTENT(OUT) :: STRT2L(LLPAR+1)
INTEGER, INTENT(OUT) :: RC

```

**REMARKS:**

Comments from Chris McLinden to Peter Cameron-Smith:

```
=====
CALL SOMLFQ(P1,P2,F0,F1,F2,PS,F,NL)
```

- P1,P2 are the pressure EDGES for the CTM layer onto which the coefficients will be mapped. [P1>P2 I believe {PJC}]
- F0,F1,F2 are the CTM layer vertical moments determined in SOMLFQ
- PS are the pressure layer edges of the original [ie Linoz] grid
- F is the column of coefficients (on the original grid); note F is flipped relative to STRTX and since the coefficients begin at z\*=10, F(1)=F(2)=...=F(5)=0
- NL is 30; size of F()

The box model calculations were performed at z\*=10km, 12km, ... and so these would represent the centres with the corresponding edges at 9,11km ; 11,13km; ...

PS() represents the edges (although PS(1) is set to 1000mb).

The first few values are:

```

PS(1)=1000
PS(2)=874.947105 (note PS(2) is not quite 1000 exp(-1/16) as the
PS(3)=656.117767 the average pressure is used - not the pressure

```

```

PS(4)=492.018914 at the average z*)
PS(5)=368.96213
PS(6)=276.68257
PS(7)=207.48266
...
PS(30)=0.276682568
PS(31)=0.0

F(1) spans PS(1)-PS(2)
F(2) spans PS(2)-PS(3)
...
F(30) spans PS(30)-PS(31)

```

**REVISION HISTORY:**

```

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem
18 Mar 2013 - R. Yantosca - Now pass am_I_root, Input_Opt, RC arguments
18 Mar 2013 - R. Yantosca - Rearrange argument list so that inputs are
 all listed before outputs
18 Mar 2013 - R. Yantosca - Now make NL a local variable and set its
 value from Input_Opt%LINOZ_NLEVELS

```

**7.13.5 linoz\_somlfq**

subroutine LINOZ\_SOMLFQ calculates loss freq moments from a set of loss frequencies at std  $z^*$ , given a CTM model interval pressure range:  $P1$  ;  $P2$  (decreasing up)

**INTERFACE:**

```

SUBROUTINE LINOZ_SOMLFQ(P1,P2,F0,F1,F2,PS,F,NL)

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: NL
REAL(fp), INTENT(IN) :: F(NL)
REAL(fp), INTENT(IN) :: PS(NL+1)
REAL(fp), INTENT(IN) :: P1
REAL(fp), INTENT(IN) :: P2

```

**OUTPUT PARAMETERS:**

```

REAL(fp), INTENT(OUT) :: F0
REAL(fp), INTENT(OUT) :: F1
REAL(fp), INTENT(OUT) :: F2

```

**REMARKS:**

```

The pressure levels BETWEEN z^* values are:
 PS(i) > PS(i+1) bounds $z^*(i)$

```

```

NL: z* levels, ==> PS(NL+1) = 0 (extrapolate chemical loss to top)
 Z1 = 16.DO*LOG10(1000.DO/P1)
 Z2 = 16.DO*LOG10(1000.DO/P2)

```

```

The MOMENTS for a square-wave or 'bar': F(x)=f0 b<=x<=c, =0.0 else
 S0 = f0 (x) [from x=b to x=c]
 S1 = 3 f0 (x^2 - x) [from x=b to x=c]
 S2 = 5 f0 (2x^3 - 3x^2 + x) [from x=b to x=c]

```

## REVISION HISTORY:

```

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem
19 Mar 2013 - R. Yantosca - P1, P2 are now declared as INTENT(IN)

```

### 7.13.6 linoz\_read

Subroutine LINOZ\_READ reads the input data file for the Linoz stratospheric ozone chemistry.

## INTERFACE:

```

SUBROUTINE LINOZ_READ(am_I_Root, Input_Opt, RC)

```

## USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
USE InquireMod, ONLY : findFreeLun

```

## INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?

```

## INPUT/OUTPUT PARAMETERS:

```

TYPE(OptInput), INTENT(INOUT) :: Input_Opt ! Input Options object

```

## OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

## REMARKS:

```

LINOZ_READ is called from "main.f" at the start of the simulation.
LINOZ_READ will also call INIT_LINOZ to initialize the arrays.

```

## REVISION HISTORY:

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem  
 16 Oct 2009 - R. Yantosca - Now use IU\_FILE instead of IU\_LINOZ  
 16 Oct 2009 - R. Yantosca - Read file from DATA\_DIR\_1x1  
 01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire\_mod.F90  
 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block  
 15 Mar 2013 - R. Yantosca - Now call INIT\_LINOZ from GIGC\_Init\_Extra  
 5 Mar 2015 - R. Yantosca - Now read data relative to ExtData/CHEM\_INPUTS

---

### 7.13.7 linoz\_intpl

Subroutine LINOZ\_INTPL does some kind of interpolation.

#### INTERFACE:

```
SUBROUTINE LINOZ_INTPL(KE, IE, ND, NE, XI, XN, YI, YN)
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: KE
INTEGER, INTENT(IN) :: IE
INTEGER, INTENT(IN) :: ND
INTEGER, INTENT(IN) :: NE
REAL(fp), INTENT(IN) :: XI(IE)
REAL(fp), INTENT(IN) :: XN(ND)
REAL(fp), INTENT(IN) :: YI(KE, IE)
```

#### OUTPUT PARAMETERS:

```
REAL(fp), INTENT(OUT) :: YN(KE, ND)
```

#### REVISION HISTORY:

24 Jun 2003 - B. Field & D. Jones - Further updates for GEOS-Chem

---

### 7.13.8 init\_linoz

Subroutine INIT\_LINOZ allocates and zeroes the module arrays used in the Linoz stratospheric ozone algorithm.

#### INTERFACE:

```
SUBROUTINE INIT_LINOZ(am_I_Root, Input_Opt, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE Input_Opt_Mod, ONLY : OptInput
```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

16 Oct 2009 - R. Yantosca - Initial version
18 Mar 2013 - R. Yantosca - Remove TPARM array since that is now carried
 within the Input_Opt object.
18 Mar 2013 - R. Yantosca - Accept am_I_Root, Input_Opt, RC arguments
14 Mar 2013 - M. Payer - Replace 0x with 03 for full-chemistry simulation
19 Oct 2015 - C. Keller - TLSTT is now 4D to work on curvilinear grids

```

---

**7.13.9 cleanup\_linoz**

Subroutine CLEANUP\_LINOZ deallocates all module arrays.

**INTERFACE:**

```

SUBROUTINE CLEANUP_LINOZ

```

**REVISION HISTORY:**

```

16 Oct 2009 - R. Yantosca - Initial version

```

**8 Aerosol modules**

These modules contain routines to perform chemistry or removal of the various aerosol species.

---

**8.1 Fortran: Module Interface aerosol\_mod.F**

Module AEROSOL\_MOD contains variables and routines for computing optical properties for aerosols which are needed for both the FAST-J photolysis and ND21 optical depth diagnostics. (bmy, 7/20/04, 2/10/09)

**INTERFACE:**

```

MODULE AEROSOL_MOD

```

**USES:**

```
USE PRECISION_MOD
```

```
IMPLICIT NONE
```

```
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: AEROSOL_CONC
```

```
PUBLIC :: CLEANUP_AEROSOL
```

```
PUBLIC :: INIT_AEROSOL
```

```
PUBLIC :: RDAER
```

### PUBLIC DATA MEMBERS:

```
!=====
! BCPI : Hydrophilic black carbon aerosol [kg/m3]
! BCPO : Hydrophobic black carbon aerosol [kg/m3]
! OCPI : Hydrophilic organic carbon aerosol [kg/m3]
! OCPO : Hydrophobic organic carbon aerosol [kg/m3]
! OCPISOA : Hydrophilic OC + SOA aerosol [kg/m3]
! SALA : Accumulation mode seasalt aerosol [kg/m3]
! SALC : Coarse mode seasalt aerosol [kg/m3]
! SO4_NH4_NIT : Lumped SO4-NH4-NIT aerosol [kg/m3]
! SO4 : Sulfate aerosol [kg/m3]
! NH4 : Ammonium aerosol [kg/m3]
! NIT : Inorganic nitrate aerosol [kg/m3]
! SOILDUST : Mineral dust aerosol from soils [kg/m3]
! SLA : Stratospheric liquid aerosol [kg/m3]
! SPA : Stratospheric particulate aerosol [kg/m3]
! TSOA : Terpene SOA [kg/m3]
! ISOA : Isoprene SOA (VBS scheme) [kg/m3]
! ASOA : Aromatic + IVOC SOA [kg/m3]
! OPOA : Aerosol product of SVOC oxidation [kg/m3]
! SOAGX : SOA product of GLYX [kg/m3]
! SOAMG : SOA product of MYLY [kg/m3]
! PM25 : Particulate matter < 2.5 um [kg/m3]
! ISOAAQ : Isoprene SOA (aqueous formation) [kg/m3]
! SOAS : Simple SOA [kg/m3]
! OCFPOA : OM/OC for POA [unitless]
! OCFPOA : OM/OC for OPOA, OCPI, OCPO [unitless]
!=====
REAL(fp), ALLOCATABLE, PUBLIC :: BCPI(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: BCPO(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OCPI(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OCPO(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OCPISOA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SALA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SALC(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SO4_NH4_NIT(:,:,:)

```

```

REAL(fp), ALLOCATABLE, PUBLIC :: SO4(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: NH4(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: NIT(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: FRAC_SNA(:,:,:,)
REAL(fp), ALLOCATABLE, PUBLIC :: SOILDUST(:,:,:,)
REAL(fp), ALLOCATABLE, PUBLIC :: SLA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SPA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: TSOA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: ISOA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: ASOA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OPOA(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SOAGX(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SOAMG(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: PM25(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: ISOAAQ(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: SOAS(:,:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OCFPOA(:,:)
REAL(fp), ALLOCATABLE, PUBLIC :: OCFPOA(:,:)

```

! Growth factors

```

REAL(fp), PUBLIC :: SIA_GROWTH
REAL(fp), PUBLIC :: ORG_GROWTH
REAL(fp), PUBLIC :: SSA_GROWTH

```

#### DEFINED PARAMETERS:

! For SOAGX, assume the total aerosol mass/glyoxal mass = 1.d0  
! for now (tmf, 1/7/09)

```

REAL(fp), PARAMETER, PUBLIC :: OCFG = 1.e+0_fp

```

! For SOAMG, assume the total aerosol mass/methylglyoxal mass = 1.d0  
! for now (tmf, 1/7/09)

```

REAL(fp), PARAMETER, PUBLIC :: OCFM = 1.e+0_fp

```

#### REMARKS:

References:

=====

- (1 ) Pye, H.O.T., and J.H. Seinfeld, "A global perspective on aerosol from low-volatility organic compounds", Atmos. Chem. & Phys., Vol 10, pp 4377-4401, 2010.
- (2 ) Philip, S., R.V. Martin, J.R. Pierce, J.L. Jimenez, Q. Zhang, M.R. Canagaratna, D.V. Spracklen, C.R. Nowlan, L.N. Lamsal, M.J. Cooper, and N.A. Krotkov, "Spatially and seasonally resolved estimate of the ratio of global organic mass to organic carbon", Atmospheric Environment, 87, 34-40, doi:10.1016/j.atmosenv.2013.11.065, 2014

#### REVISION HISTORY:

- (1 ) Added AEROSOL\_RURALBOX routine (bmy, 9/28/04)



- (2 ) Now convert ABSHUM from absolute humidity to relative humidity in AEROSOL\_RURALBOX, using the same algorithm as in "gasconc.f".  
(bmy, 1/27/05)
  - (3 ) Now references "tropopause\_mod.f" (bmy, 8/22/05)
  - (4 ) Now add contribution of SOA4 into Hydrophilic OC (dkh, bmy, 5/18/06)
  - (5 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
  - (6 ) Add support for variable tropopause (bdf, phs, 9/14/06)
  - (7 ) Now set OCF=2.1 in AEROSOL\_CONC for consistency w/ carbon\_mod.f  
(tmf, 2/10/09)
  - (8 ) Add WTAREA and WERADIUS for dicarbonyl SOA production.  
WTAREA is the same as TAREA, but excludes dry dust, BCPO and OCPO;  
use same units as TAREA.  
WERADIUS is same as ERADIUS, but excludes dry dust, BCPO and OCPO;  
use same units as ERADIUS. (tmf, 3/2/09)
  - (9 ) Add SOAG and SOAM species. (tmf, ccc, 3/2/09)
  - (10) Modify AOD output to wavelength specified in jv\_spec\_aod.dat  
(clh, 05/07/10)
- 22 Dec 2011 - M. Payer - Added ProTeX headers  
 05 Mar 2013 - R. Yantosca - Now make INIT\_AEROSOL a public routine  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

### 8.1.1 aerosol\_conc

Subroutine AEROSOL\_CONC computes aerosol concentrations in kg/m<sup>3</sup> from the tracer mass in kg in the Species array. These are needed to compute optical properties for photolysis, for the optical depth diagnostics, and for the SOA concentration diagnostics. (bmy, 7/20/04, 2/10/09)

#### INTERFACE:

```

SUBROUTINE AEROSOL_CONC(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_FJX_MOD, ONLY : REAA
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE HCO_Error_Mod
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE UCX_MOD, ONLY : KG_STRAT_AER
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

```

 USE TIME_MOD, ONLY : GET_MONTH
#if defined(TOMAS)
 USE TOMAS_MOD, ONLY : IBINS
#endif

```

**INPUT PARAMETERS:**

```

 LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
 TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

 TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
 TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

 INTEGER, INTENT(OUT) :: RC ! Success or failure

```

**REMARKS:**

This code was originally included in "chemdr.f", but the same computation also needs to be done for offline aerosol simulations. Therefore, we have split this code off into a separate subroutine which can be called by both fullchem and offline aerosol simulations.

**REVISION HISTORY:**

```

(1) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(2) Now add contribution from SOA4 into Hydrophilic OC (dkh, bmy, 5/18/06)
(3) Now set OCF=2.1 to be consistent w/ "carbon_mod.f" (tmf, 2/10/09)
22 Dec 2011 - M. Payer - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
13 Nov 2012 - R. Yantosca - Now pass am_I_Root, Input_Opt, RC as arguments
15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
05 Mar 2013 - R. Yantosca - Remove call to INIT_AEROSOL, this is now done
 in the initialization stage
25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
13 Aug 2013 - M. Sulprizio - Add modifications for updated SOA and SOA +
 semivolatile POA simulations (H. Pye)
17 Jun 2014 - J. Pierce - Now allow bulk dust in TOMAS simulations
07 Apr 2015 - R. Yantosca - Add check to prevent div-by-zero errors
25 Jun 2015 - E. Lundgren - Use L. Zhang dust size distribution params for
 mineral dust aerosols (non-TOMAS)
13 Aug 2015 - E. Lundgren - Convert tracer units to kg if input units are
 mixing ratio
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code

```

18 Nov 2016 - M. Sulprizio- Move calculation of SOA and PM2.5 concentrations here from diag42\_mod.F  
 22 Nov 2016 - M. Sulprizio- Add online calculation of aerosol growth factors at 35% RH  
 07 Feb 2017 - M. Sulprizio- Apply STP correction factor to PM2.5  
 07 Sep 2017 - M. Sulprizio- Add simple SOA to PM2.5 calculation  
 28 Sep 2017 - E. Lundgren - Simplify unit conversion with wrapper routine  
 13 Dec 2017 - R. Yantosca - Now accept State\_Diag as an argument  
 05 Jan 2018 - M. Sulprizio- Use simple SOA by default in calculation of PM2.5 and OCPISOA (used in hetchem and AOD)

### 8.1.2 rdaer

Subroutine RDAER reads global aerosol concentrations as determined by Mian Chin. Calculates optical depth at each level for "set\_prof.f". Also calculates surface area for heterogeneous chemistry. It uses aerosol parameters in FAST-J input file "jv\_spec.dat" for these calculations. (rvm, rjp, tdf, bmy, 11/04/01, 7/20/04)

#### INTERFACE:

```

SUBROUTINE RDAER(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC,
& MONTH, YEAR, ODSWITCH)

```

#### USES:

```

USE CMN_FJX_MOD
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : CONSVAP
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : ITS_A_NEW_MONTH
USE TIME_MOD, ONLY : SYSTEM_TIMESTAMP
USE TRANSFER_MOD, ONLY : TRANSFER_3D
USE UCX_MOD, ONLY : GET_STRAT_OPT
USE UCX_MOD, ONLY : NDENS_AER
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
 USE DIAG_MOD, ONLY : AD21
#endif
#if defined(NC_DIAG)
 USE Species_Mod, ONLY : Species
#endif

```

IMPLICIT NONE

### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on root CPU?
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State
INTEGER, OPTIONAL :: MONTH ! # of current month
INTEGER, OPTIONAL :: YEAR ! 4-digit year
INTEGER, OPTIONAL :: ODSWITCH ! Logical indicator
 ! = 0: AOD computed
 ! at 999 nm
 ! = 1: AOD computed
 ! at wavelength set
 ! in Radiation Menu
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

### REVISION HISTORY:

- (1 ) At the point in which "rdaer.f" is called, ABSHUM is actually absolute humidity and not relative humidity (rvm, bmy, 2/28/02)
- (2 ) Now force double-precision arithmetic by using the "D" exponent. (bmy, 2/28/02)
- (3 ) At present aerosol growth is capped at 90% RH. The data in jv\_spec.dat could be used to allow a particle to grow to 99% RH if desired. (rvm, 3/15/02)
- (4 ) Bug fix: TEMP2 needs to be sized (IIPAR,JJPAR,LLPAR) (bmy, 5/30/02)
- (5 ) Now reference BXHEIGHT from "dao\_mod.f". Also references ERROR\_STOP from "error\_mod.f". Delete local declaration of TIME, since that is also declared w/in comode.h -- this causes compile-time errors on the ALPHA platform. (gcc, bmy, 11/6/02)
- (6 ) Now use the online SO4, NH4, NIT aerosol, taken from the STT array, and passed via SO4\_NH4\_NIT argument if sulfate chemistry is turned on. Otherwise, read monthly mean sulfate from disk. (rjp, bmy, 3/23/03)
- (7 ) Now call READ\_BPCH2 with QUIET=.TRUE., which prevents info from being printed to stdout. Also made cosmetic changes. (bmy, 3/27/03)
- (8 ) Add BCPI, BCPO, OCPI, OCPO to the arg list. Bug fix: for online sulfate & carbon aerosol tracers, now make sure these get updated every timestep. Now references "time\_mod.f". Now echo info about which online/offline aerosols we are using. Updated comments. (bmy, 4/9/04)
- (9 ) Add SALA, SALC to the arg list (rjp, bec, bmy, 4/20/04)
- (10) Now references DATA\_DIR from "directory\_mod.f". Now references LSULF,

- LCARB, LSSALT from "logical\_mod.f". Added minor bug fix for conducting the appropriate scaling for optical depth for ND21 diagnostic. Now make MONTH and YEAR optional arguments. Now bundled into "aerosol\_mod.f". (rvm, aad, clh, bmy, 7/20/04)
- (11) Now remove FWET from extinction efficiency computation (avd, 8/3/10)
- (12) Include third input argument to determine the wavelength at which the AOD should be computed. This will set the optical properties that are used for the calculation of the AOD. The ND21 diagnostic should only be updated when WAVELENGTH = 1. (skim, 02/03/11)
- 09 Mar 2011 - R. Yantosca - Set MSDENS(2) = 1800 for APM (G. Luo)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 13 Nov 2012 - R. Yantosca - Now pass Input\_Opt, RC arguments for GIGC
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 27 Mar 2013 - S.D. Eastham- Upgraded from Fast-J to Fast-JX
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte
- 05 Mar 2015 - R. Yantosca - Remove DO\_READ\_DATA option. We can no longer have binary punch input in HPC environments.
- 31 Mar 2015 - R. Yantosca - Bug fix: Declare LINTERP as !\$OMP PRIVATE
- 07 Apr 2015 - R. Yantosca - Bug fix: Add error check to JLOOP do-loop so that we cycle if JLOOP <= 0
- 24 Jun 2015 - M. Sulprizio- Update organic aerosol density MSDENS(3) from 1800 to 1300 kg/m3 (E. Marais, M. Hammer)
- 15 Oct 2015 - C. Keller - Empty ODAER before refilling.
- 12 May 2016 - M. Sulprizio- Remove 1D arrays that depend on JLOOP. ABSHUM is now a 3D field in State\_Met. ERADIUS, TAREA, WERADIUS, WTAREA are now pointers that point to 3D fields in State\_Chm.
- 16 May 2016 - M. Sulprizio- Remove JLOOP entirely and loop over LLPAR, JJPARG, IIPARG instead.
- 28 Jun 2016 - M. Sulprizio- Make IOUT a local variable to remove dependence on comode\_loop\_mod.F
- 22 Nov 2016 - M. Sulprizio- Now define aerosol densities in species database and use SpcData%Info%Density to populate MSDENS
- 03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument
- 17 Jan 2018 - E. Lundgren - Implement netcdf cloud diagnostics (eq to ND21)
- 17 Jan 2018 - E. Lundgren - Fix bpch diagnostic indexing for aer aq volume and SOA AOD from aq isoprene (34->58; 35->59)
- 11 May 2018 - R. Yantosca - Rename LSOA to Is\_ComplexSOA for clarity

---

### 8.1.3 init\_aerosol

Subroutine INIT\_AEROSOL allocates and zeroes module arrays

**INTERFACE:**

```
SUBROUTINE Init_Aerosol(am_I_Root, Input_Opt, State_Diag, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object
```

**INPUT/OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
20 Jul 2004 - R. Yantosca - Initial version
22 Dec 2011 - M. Payer - Added ProTeX headers
05 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC arguments
17 Jun 2016 - R. Yantosca - Now use locally-defined species ID flags,
13 Dec 2017 - R. Yantosca - Now accept State_Diag as an argument
```

---

**8.1.4 cleanup\_aerosol**

Subroutine CLEANUP\_AEROSOL deallocates all module arrays (bmy, 7/20/04)

**INTERFACE:**

```
SUBROUTINE CLEANUP_AEROSOL
```

**REVISION HISTORY:**

```
22 Dec 2011 - M. Payer - Added ProTeX headers
```

---

**8.2 Fortran: Module Interface carbon\_mod.F**

Module CARBON\_MOD contains arrays and routines for performing a carbonaceous aerosol simulation. Original code taken from Mian Chin's GOCART model and modified accordingly. (rjp, bmy, 4/2/04, 6/30/10)

**INTERFACE:**

```
MODULE CARBON_MOD
```

**USES:**

```
USE AEROSOL_MOD, ONLY : OCFPOA, OCFPOA
USE HCO_ERROR_MOD ! For HEMCO error reporting
USE PhysConstants ! Physical constants
USE PRECISION_MOD ! For GEOS-Chem Precisions
```

```
IMPLICIT NONE
PRIVATE
```

**PUBLIC MEMBER FUNCTIONS:**

```
PUBLIC :: CHEMCARBON
PUBLIC :: EMISSCARBON
PUBLIC :: CLEANUP_CARBON
PUBLIC :: INIT_CARBON
#if defined (TOMAS)
PUBLIC :: EMISSCARBONTOMAS
#endif
```

**PUBLIC DATA MEMBERS:**

```
! SOAupdate: for branching ratio diagnostic (hotp 5/24/10)
PUBLIC :: BETANOSAVE

! ORVC_SESQ needs to be public so that it can be added as a
! restart variable to GEOS-5. (ckeller, 10/10/17)
PUBLIC :: ORVC_SESQ

! Make BIOG_SESQ array public so that it can be used by HEMCO
! (ckeller, 05/19/14)
PUBLIC :: BIOG_SESQ
```

**REMARKS:**

```
4 Aerosol species : Organic and Black carbon
 : hydrophilic (soluble) and hydrophobic of each
```

For secondary organic aerosol (SOA) simulation original code developed by Chung and Seinfeld [2002] and Hong Liao from John Seinfeld's group at Caltech was taken and further modified accordingly (rjp, bmy, 7/15/04)

SOAupdate: Traditional SOA simulation updated by hotp 7/2010  
New code treats semivolatile or nonvolatile POA, aerosol from IVOCs, and has updated biogenic SOA  
For more details on the updated SOA/POA simulation, see comments in SOA\_CHEMISTRY, Pye and Seinfeld ACP 2010, Pye et al. in prep for ACP 2010  
Note that modifications were made throughout the code for SOAupdate

## References:

- =====
- (1 ) Bond, T.C., E. Bhardwaj, R. Dong, R. Jogani, S. Jung, C. Roden, D.G. Streets, and N.M. Trautmann, "Historical emissions of black and organic carbon aerosol from energy-related combustion, 1850-2000", *Global Biogeochem. Cycles*, 21, GB2018, doi:10.1029/2006GB002840, 2007.
  - (2 ) Chan, A.W.H., K.E. Kautzman, P.S. Chhabra, J.D. Surratt, M.N. Chan, J.D. Crouse, A. Kurten, P.O. Wennberg, R.C. Flagan, and J.H. Seinfeld, "Secondary organic aerosol formation from photooxidation of naphthlene and alkyl naphthalenes: implications for oxidation of intermediate volatility organic compounds (IVOCs)", *Atmos. Chem. Phys.*, Vol 9, 3049-3060, doi:10.5194/acp-9-3049-2009, 2009.
  - (3 ) Chung, S.H., and J.H. Seinfeld. "Global distribution and climate forcing of carbonaceous aerosols", *J. Geophys. Res.*, Vol 107(D19), 4407, doi:10.1029/2001JD001397, 2002.
  - (4 ) Grieshop, A.P., J.M. Logue, N.M. Donahue, and A.L. Robinson, "Laboratory investigation of photochemical oxidation of organic aerosol from wood fires 1: Measurement and simulation of organic aerosol evolution", *Atmos. Chem. Phys.*, Vol 9, 1263-1277, doi:10.5194/acp-9-1263-2009, 2009.
  - (5 ) Griffin, R.J., D.R. Cocker, R.C. Flagan, and J.H. Seinfeld, "Organic aerosol formation from the oxidation of biogenic hydrocarbons", *J. Geophys. Res.*, 104(D3), 3555-3567, 1999.
  - (6 ) Henze, D.K., and J.H. Seinfeld, "Global secondary organic aerosol from isoprene oxidation", *Geophys. Res. Lett.*, Vol 33, L09812, doi:10.1029/2006GL025976, 2006.
  - (7 ) Henze, D.K., J.H. Seinfeld, N.L. Ng, J.H. Kroll, T.-M. Fu, D.J. Jacob, and C.L. Heald, "Global modeling of secondary organic aerosol formation from aromatic hydrocarbons: high vs. low-yield pathways", *Atmos. Chem. Phys.*, Vol 8, 2405-2420, doi:10.5194/acp-8-2405-2008, 2008.
  - (8 ) Kroll, J.H., N.L. Ng, S.M. Murphy, R.C. Flagan, and J.H. Seinfeld, "Secondary organic aerosol formation from isoprene photooxidation", *Environ. Sci. Technol.*, Vol 40, 1869-1877, doi:10.1021/Es0524301, 2006.
  - (9 ) Liao, H., D.K. Henze, J.H. Seinfeld, S.L. Wu, and L.J. Mickley, "Biogenic secondary aerosol over the United States: Comparison of climatological simulations with observations", *J. Geophys. Res.* Vol 112, D06201, doi:10.1029/2006JD007813, 2007.
  - (10) Ng, N.L., P.S. Chhabra, A.W.H. Chan, J.D. Surratt, J.H. Kroll, A.J. Kwan, D.C. McCabe, P.O. Wennberg, A. Sorooshian, S.M. Murphy, N.F. Dalleska, R.C. Flagan, and J.H. Seinfeld, "Effect of NOx level on secondary organic aerosol (SOA) formation from the photooxidation of terpenes", *Atmos. Chem. Phys.*, Vol 7, 5159-5174, doi:10.5194/acp-7-5195-2007, 2007a.
  - (11) Ng, N.L., J.H. Kroll, A.W.H. Chan, P.S. Chhabra, R.C. Flagan, and J.H. Seinfeld, "Secondary organic aerosol formation from m-xylene, toluene, and benzene", *Atmos. Chem. Phys.*, Vol 7, 3909-3922,



- doi:10.5194/acp-7-3909-2007, 2007b.
- (12) Ng, N.L., A.J. Kwan, J.D. Surratt, A.W.H. Chan, P.S. Chhabra, A. Sorooshian, H.O.T. Pye, J.D. Crouse, P.O. Wennberg, R.C. Flagan, and J.H. Seinfeld, "Secondary organic aerosol (SOA) formation from reaction of isoprene with nitrate radicals (NO<sub>3</sub>)", *Atmos. Chem. Phys.*, Vol 8, 4117-4140, doi:10.5194/acp-8-4117-2008, 2008.
- (13) Pye, H.O.T., and J.H. Seinfeld, "A global perspective on aerosol from low-volatility organic compounds", *Atmos. Chem. Phys.*, Vol 10, 4377-4401, doi:10.5194/acp-10-4377-2010, 2010.
- (14) Pye, H.O.T., A.W.H. Chan, M.P. Barkley, and J.H. Seinfeld, "Global modeling of organic aerosol: The importance of reactive nitrogen (NO<sub>x</sub> and NO<sub>3</sub>)", *Atmos. Chem. Phys.*, Vol 10, 11261-11276, doi:10.5194/acp-10-11261-2010, 2010.
- (15) Shilling, J.E., Q. Chen, S.M. King, T. Rosenoern, J.H. Kroll, D.R. Worsnop, K.A. McKinney, S.T. Martin, "Particle mass yield in secondary organic aerosol formed by the dark ozonolysis of  $\alpha$ -pinene", *Atmos Chem Phys*, Vol 8, 2073-2088, doi: 10.5194/acp-8-2073-2008, 2008.
- (16) Shrivastava, M.K., E.M. Lipsky, C.O. Stanier, A.L. Robinson, "Modeling semivolatile organic mass emissions from combustion systems", *Environ. Sci. Technol.*, Vol 40, 2671-2677, doi:10.1021/ES0522231, 2006.
- (17) Zhang, J.Y., K.E.H. Hartz, S.N. Pandis, N.M. Donahue, "Secondary organic aerosol formation from limonene ozonolysis: Homogeneous and heterogeneous influences as a function of NO<sub>x</sub>", *J. Phys. Chem. A*, Vol 110, 11053-11063, doi:10.1021/Jp06286f, 2006.

Base Year is 2000. More at <http://www.hiwater.org>

#### REVISION HISTORY:

- (1 ) Added code from the Caltech group for SOA chemistry (rjp, bmy, 7/15/04)
- (2 ) Now references "directory\_mod.f", "logical\_mod.f", "tracer\_mod.f". (bmy, 7/20/04)
- (3 ) Now read data from carbon\_200411/ subdir of DATA\_DIR. Also added some extra debug output. Now read T. Bond yearly emissions as default, but overwrite N. America with the monthly Cooke/RJP emissions. Added module variables I1\_NA, I2\_NA, J1\_NA, J2\_NA. (rjp, bmy, 12/1/04)
- (4 ) Now can read seasonal or interannual BCPO, OCPO biomass emissions. Also parallelize loop in OHNO3TIME. (rjp, bmy, 1/18/05)
- (5 ) Now references "pbl\_mix\_mod.f". Bug fix: now make sure only to save up to LD07 levels for the ND07 diagnostic in SOA\_LUMP. (bmy, 3/4/05)
- (6 ) Now can read data for both GEOS and GCAP grids (bmy, 8/16/05)
- (7 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (8 ) Now references "megan\_mod.f". Also now references XNUMOL and XNUMOLAIR from "tracer\_mod.f" (tmf, bmy, 10/25/05)
- (9 ) Bug fix for GCAP in BIOGENIC\_OC (bmy, 4/11/06)
- (10) Updated for SOA production from ISOP (dkh, bmy, 5/22/06)
- (11) Updated for IPCC future emission scale factors. Also added function GET\_DOH to return ISOP that has reacted w/ OH. (swu, dkh, bmy, 6/1/06)

- (12) Now add SOG condensation onto SO4, NH4, NIT (rjp, bmy, 8/3/06)
- (13) Minor fix for 20 carbon tracers. (phs, 9/14/06)
- (14) Now remove reading of biomass emissions from "carbon\_mod.f", since they are better done in gc\_biomass\_mod.f. This will allow us to standardize treatment of GFED2 or default BB emissions. Also applied a typo fix in SOA\_LUMP. (tmf, bmy, 10/16/06)
- (15) Prevent seg fault error in BIOMASS\_CARB\_GEOS (bmy, 11/3/06)
- (16) Corrected typos in SOA\_LUMP. Now also save GPROD and APROD to disk for each new diagnostic interval. (dkh, tmv, havala, bmy, 2/6/07)
- (17) Modifications for 0.5 x 0.666 nested grids (yxw, dan, bmy, 11/6/08)
- (18) Now account for various GFED2 products (yc, phs, 12/23/08)
- (19) Now add future scaling to BIOMASS\_CARB\_GEOS (hotp, swu, 2/19/09)
- (20) Added SOA production from dicarbonyls (tmf, 3/2/09)
- (21) Bugfix: cleanup ORVC\_TERP and ORVC\_SESQ (tmf, 3/2/09)
- (22) Replace USE\_MONTHLY\_BIOB with USE\_BOND\_BIOBURN, since this hardwired flag is a switc b/w annual Bond biomass burning emissions, and default GC source, which can be monthly/8 days/3hr. Implement changes for reading new Bond files (eml, phs, 5/18/09)
- (23) Add option for non-local PBL scheme (lin, 06/09/08)
- (24) Now added NESTED\_EU grid. Updated formulation of SOG condensation onto OC aerosol, according to recommendations of Aerosol Working Group. (amv, clh, bmy, 12/21/09)
- (25) Bug fix for EMIS\_SAVE in EMITHIGH (bmy, 1/11/10)
- (26) Modifications for TOMAS (win, bmy, 1/25/10)
- (27) Bug fix: call SOA\_PARA\_INIT (ensberg, bmy, 6/30/10)
- (28) Modified to include GFED3 (psk, 1/5/11)
- 01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90
- 30 Jul 2012 - R. Yantosca - Modifications for grid-independence
- 28 Nov 2012 - R. Yantosca - Replace SUNCOS array with State\_Met%SUNCOS and SUNCOS\_MID array with State\_Met%SUNCOSmid
- 04 Mar 2013 - R. Yantosca - Now call INIT\_CARBON from the init stage which facilitates connection to GEOS-5 GCM
- 05 Mar 2013 - R. Yantosca - Remove reference to LNL PBL from logical\_mod.F and replace with Input\_Opt%LNL PBL
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 20 Jun 2014 - R. Yantosca - Removed obsolete code; we now use HEMCO
- 23 Sep 2014 - M. Sulprizio- Get OH, NO3, and O3 fields for offline aerosol simulation from HEMCO
- 06 Nov 2014 - M. Yannetti - Added PRECISION\_MOD
- 13 Nov 2014 - C. Keller - Added EMISSCARBON to make sure that SESQ and POA emissions are properly passed from HEMCO.
- 05 Jan 2016 - E. Lundgren - Use global physical parameters (AIRMW and AVO)
- 14 Jan 2016 - M. Sulprizio- POAEMISS(:, :, :, 1) now represents POG1 and POAEMISS(:, :, :, 2) now represents POG2
- 17 Jun 2016 - J.D.Maaasakkers - Now use Ind\_ to define species ID's

- 17 Jun 2016 - R. Yantosca - Now declare all species ID's as module variables, and initialize them at startup
- 28 Jun 2016 - M. Sulprizio- Add species ID's for L\*R02N/H and LISOP\* to replace uses of IL\*R02N/H and ILISOP\* from comode\_loop\_mod.F
- 24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA; Also remove functions GET\_VCLDF and GET\_LWC, they're no longer needed

### 8.2.1 chemcarbon

Subroutine CHEMCARBON is the interface between the GEOS-Chem main program and the carbon aerosol chemistry routines that calculates dry deposition, chemical conversion between hydrophilic and hydrophobic, and SOA production.

#### INTERFACE:

```

SUBROUTINE CHEMCARBON(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_O3_MOD, ONLY : SAVEOA
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : ITS_A_NEW_MONTH
USE TIME_MOD, ONLY : GET_TS_CHEM
#if defined(TOMAS)
USE TOMAS_MOD, ONLY : SOACOND, IBINS !(win, 1/25/10)
USE TOMAS_MOD, ONLY : CHECKMN !(sfarina)
USE PRESSURE_MOD, ONLY : GET_PCENTER
#endif

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:****REVISION HISTORY:**

- 01 Apr 1994 - R. Park - Initial version
- (1 ) Added code from the Caltech group for SOA chemistry. Also now reference "global\_oh\_mod.f", "global\_o3\_mod.f", "global\_no3\_mod.f". (rjp, bmy, 7/8/04)
- (2 ) Now reference LSOA and LEMIS from CMN\_SETUP. Now only call OHNO3TIME if it hasn't been done before w/in EMISSCARBON. (rjp, bmy, 7/15/04)
- (3 ) Now reference LSOA, LEMIS, LPRT from "logical\_mod.f". Now reference STT and ITS\_AN\_AEROSOL\_SIM from "tracer\_mod.f" (bmy, 7/20/04)
- (4 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (5 ) Now updated for SOA production from ISOP. (dkh, bmy, 6/1/06)
- (6 ) Bug fix for aerosol sim w/ 20 tracers (phs, 9/14/06)
- (7 ) Add subroutine call AGING\_CARB for converting H-phobic 30-bin EC or OC to H-philic EC or OC. (win, 1/25/10)
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 15 Nov 2012 - R. Yantosca - Added ProTeX headers
- 04 Mar 2013 - R. Yantosca - Remove call to INIT CARBON
- 04 Mar 2013 - R. Yantosca - Now pass Input\_Opt to SOA\_CHEMISTRY
- 25 Mar 2013 - M. Payer - Now pass State\_Chm object via the arg list
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET\_GLOBAL\_OH routine
- 23 Sep 2014 - M. Sulprizio- Get OH, NO3, and O3 fields for offline aerosol simulation from HEMCO
- 16 Jun 2016 - J.D.Maasackers- Now define species ID's with Ind\_ function
- 23 Jun 2016 - R. Yantosca - Add more error checks to make sure each id\_\* species flag is defined before using it
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.
- 10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
- 14 Oct 2016 - S. Farina - Added 1-day e-folding "chemistry" for SOA Simplified and SOA Precursor
- 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument
-

## 8.2.2 chem\_bcpo

Subroutine CHEM\_BCPO converts hydrophobic BC to hydrophilic BC and calculates the dry deposition of hydrophobic BC.

### INTERFACE:

```
SUBROUTINE CHEM_BCPO(am_I_Root, Input_Opt, State_Diag, TC, RC)
```

### USES:

```
USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : AD07_BC
#endif
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

### INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diags State
REAL(fp), INTENT(INOUT) :: TC(IIPAR,JJP,LLPAR) ! H-phobic BC [kg]
```

### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success?
```

### REMARKS:

Drydep is now applied in mixing\_mod.F90.

### REVISION HISTORY:

```
01 Apr 2004 - R. Park - Initial version
(1) Remove reference to "CMN", it's obsolete (bmy, 7/20/04)
(2) Replace PBLFRAC from "drydep_mod.f" with GET_FRAC_UNDER_PBLTOP
 from "pbl_mix_mod.f" (bmy, 2/17/05)
(3) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
(4) Add option for non-local PBL scheme (lin, 06/09/08)
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
26 Nov 2012 - R. Yantosca - Added ProTeX headers
05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNL PBL
19 Mar 2013 - R. Yantosca - Now copy Input_Opt%XNUMOL(1:N_TRACERS)
```

04 Mar 2015 - C. Keller - Now do dry dep in mixing\_mod.F90  
12 Jun 2015 - R. Yantosca - Remove orphaned ND44 drydep variables  
12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing\_mod.F90,  
so we can greatly collapse this code  
23 Sep 2015 - R. Yantosca - Remove reference to DRYBCP0, it's obsolete  
02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

---

### 8.2.3 chem\_bcpi

Subroutine CHEM\_BCPI calculates dry deposition of hydrophilic BC.

#### INTERFACE:

```
SUBROUTINE CHEM_BCPI(am_I_Root, Input_Opt, State_Diag, TC, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Diag_Mod, ONLY : DgnState
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diags State
REAL(fp), INTENT(INOUT) :: TC(IIPAR,JJP,LLPAR) ! H-philic BC [kg]
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success?
```

#### REMARKS:

Drydep is now applied in mixing\_mod.F90.

#### REVISION HISTORY:

01 Apr 2004 - R. Park - Initial version  
(1 ) Remove reference to "CMN", it's obsolete (bmy, 7/20/04)  
(2 ) Replace PBLFRAC from "drydep\_mod.f" with GET\_FRAC\_UNDER\_PBLTOP from  
"pbl\_mix\_mod.f" (bmy, 2/17/05)  
(3 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)  
01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments  
26 Nov 2012 - R. Yantosca - Added ProTeX Headers  
05 Mar 2013 - R. Yantosca - Now use Input\_Opt%LNLPL

19 Mar 2013 - R. Yantosca - Now copy Input\_Opt%XNUMOL(1:N\_TRACERS)  
04 Mar 2015 - C. Keller - Now do dry dep in mixing\_mod.F90  
12 Jun 2015 - R. Yantosca - Remove orphaned ND44 drydep variables  
12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing\_mod.F90,  
so we can greatly collapse this code  
23 Sep 2015 - R. Yantosca - Remove reference to DRYBCPI, it's obsolete  
02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

---

## 8.2.4 chem\_ocpo

Subroutine CHEM\_OCPO converts hydrophobic OC to hydrophilic OC and calculates the dry deposition of hydrophobic OC.

### INTERFACE:

```
SUBROUTINE CHEM_OCPO(am_I_Root, Input_Opt, State_Diag, TC, RC)
```

### USES:

```
USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : ADO7_OC
#endif
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

### INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diags State
REAL(fp), INTENT(INOUT) :: TC(IIPAR,JJPAR,LLPAR) ! H-phobic OC [kg]
```

### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success?
```

### REMARKS:

### REVISION HISTORY:

01 Apr 2004 - R. Park - Initial version  
(1) Remove reference to "CMN", it's obsolete (bmy, 7/20/04)  
(2) Replace PBLFRAC from "drydep\_mod.f" with GET\_FRAC\_UNDER\_PBLTOP from

"pbl\_mix\_mod.f" (bmy, 2/17/05)  
 (3 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)  
 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
 14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments  
 26 Nov 2012 - R. Yantosca - Added ProTeX headers  
 05 Mar 2013 - R. Yantosca - Now use Input\_Opt%LNLPLBL  
 19 Mar 2013 - R. Yantosca - Now copy Input\_Opt%XNUMOL(1:N\_TRACERS)  
 04 Mar 2015 - C. Keller - Now do dry dep in mixing\_mod.F90  
 12 Jun 2015 - R. Yantosca - Remove orphaned ND44 drydep variables  
 23 Sep 2015 - R. Yantosca - Remove reference to DRYOCPO, it's obsolete  
 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

---

### 8.2.5 chem\_ocpi

Subroutine CHEM\_BCPI calculates dry deposition of hydrophilic OC.

#### INTERFACE:

```
SUBROUTINE CHEM_OCPI(am_I_Root, Input_Opt, State_Diag, TC, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Diag_Mod, ONLY : DgnState
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diags State
REAL(fp), INTENT(INOUT) :: TC(IIPAR,JJP,LLPAR) ! H-philic OC [kg]
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success?
```

#### REMARKS:

#### REVISION HISTORY:

01 Apr 2004 - R. Park - Initial version  
 (1 ) Remove reference to "CMN", it's obsolete (bmy, 7/20/04)  
 (2 ) Replace PBLFRAC from "drydep\_mod.f" with GET\_FRAC\_UNDER\_PBLTOP from  
 "pbl\_mix\_mod.f" (bmy, 2/17/05)



(3 ) Bug fix: add BL\_FRAC to the PRIVATE list (mak, bmy, 10/3/05)  
 (4 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)  
 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
 14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments  
 26 Nov 2012 - R. Yantosca - Added ProTeX headers  
 05 Mar 2013 - R. Yantosca - Now use Input\_Opt%LNLPLBL  
 19 Mar 2013 - R. Yantosca - Now copy Input\_Opt%XNUMOL(1:N\_TRACERS)  
 04 Mar 2015 - C. Keller - Now do dry dep in mixing\_mod.F90  
 12 Jun 2015 - R. Yantosca - Remove orphaned ND44 drydep variables  
 23 Sep 2015 - R. Yantosca - Remove reference to DRYOCPI, it's obsolete  
 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

---

### 8.2.6 aging\_carb

Subroutine AGING\_CARB converts the size-resolved hydrophobic EC or OC to hydrophilic EC or OC with an assumed e-folding time. (win, 9/11/07)

#### INTERFACE:

```
SUBROUTINE AGING_CARB(MIL, MOB)
```

#### USES:

```
USE CMN_SIZE_MOD
USE TIME_MOD, ONLY : GET_TS_CHEM ! [=] second
USE TOMAS_MOD, ONLY : IBINS
```

#### INPUT/OUTPUT PARAMETERS:

```
REAL(fp), INTENT(INOUT) :: MIL(IIPAR,JJPARG,LLPAR, IBINS)
REAL(fp), INTENT(INOUT) :: MOB(IIPAR,JJPARG,LLPAR, IBINS)
```

#### REMARKS:

20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 14 Oct 2016 - S. Farina - Vectorized loop

---

### 8.2.7 soa\_chemistry

Subroutine SOA\_CHEMISTRY performs SOA formation. This code is from the Caltech group (Hong Liao, Serena Chung, et al) and was modified for GEOS-CHEM. (rjp, bmy, 7/8/04, 12/21/09)

#### INTERFACE:

```
SUBROUTINE SOA_CHEMISTRY(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)
```

#### USES:

```

USE CMN_O3_MOD
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

Procedure:

=====

- (1 ) Read in NO3, O3, OH in CHEM\_SOA
- (2 ) Scales these fields using OHNO3TIME in sulfate\_mod.f (see GET\_OH)
- (3 ) Calculate reaction rates (Serena's OCHEMPARAETER)
- (4 ) CALCULATE DELHC
- (5 ) get TOM gas products
- (6 ) equilibrium calculation

As of 5/20/10: Havala's New formulation

```

%%
% FOR SEMIVOLATILE POA and IVOC (aka SOA_SVPOA) simulations: %
%%
GEOS-Chem treats formation of aerosol from 11 parent hydrocarbons
and oxidation by OH, O3, and NO3:

```

The parent hydrocarbons are lumped into 5 semivolatile systems:

- TSOA/G: the lumped semivolatile oxidation products of  
monoterpene and sesquiterpene oxidation
- ISOA/G: the lumped semivolatile oxidation products of isoprene ox
- ASOA/G: the lumped semivolatile (and nonvolatile) products of  
benzene, toluene, xylene, and naphthalene (IVOC surrogate)  
oxidation

POA/G : the lumped primary semivolatile emissions  
 OPOA/G: the lumped products of primary semivolatile oxidation

| Parent HC   | Oxidized by   | Products         |
|-------------|---------------|------------------|
| MTPA        | OH, O3, NO3   | TSOA/G0-3        |
| LIMO        | OH, O3, NO3   | TSOA/G1-3        |
| MTP0        | OH, O3, NO3   | TSOA/G0-3        |
| SESQ        | OH, O3, NO3   | TSOA/G1-3        |
| ISOP        | OH, NO3       | ISOA/G1-3        |
| BENZ        | OH, (+NO,HO2) | ASOAN, ASOA/G1-3 |
| TOLU        | OH, (+NO,HO2) | ASOAN, ASOA/G1-3 |
| XYLE        | OH, (+NO,HO2) | ASOAN, ASOA/G1-3 |
| SVOC/POA    | OH            | POA/G1-2         |
| O-SVOC/OPOA | OH            | OPOA/G1-2        |
| NAP         | OH, (+NO,HO2) | ASOAN, ASOA/G1-3 |

Species that must be defined in input.geos (in addition to standard full chem species) (34 additional):

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| TSOA1 | TSOG1 | ISOA1 | ISOG1 | ASOA1 | ASOG1 |
| TSOA2 | TSOG2 | ISOA2 | ISOG2 | ASOA2 | ASOG2 |
| TSOA3 | TSOG3 | ISOA3 | ISOG3 | ASOA3 | ASOG3 |
| ASOAN | TSOA0 | TSOG0 |       |       |       |
| BENZ  | TOLU  | XYLE  | MTPA  | LIMO  | MTP0  |
| NAP   |       |       |       |       |       |
| POA1  | POG1  | POA2  | POG2  |       |       |
| OPOA1 | OPOG1 | OPOA2 | OPOG2 |       |       |

The following should NOT be defined for semivol POA: OCPI, OCPO

```

%%
% FOR NON-VOLATILE TRADITIONAL POA (aka SOA) simulations: %
%%
GEOS-Chem treats formation of aerosol from 8 parent hydrocarbons
and oxidation by OH, O3, and NO3:

```

Two non-volatile, traditional primary OC species exist:  
 OCPO: hydrophobic POA  
 OCPI: hydrophillic POA

The parent hydrocarbons are lumped into 3 semivolatile systems:  
 TSOA/G: the lumped semivolatile oxidation products of  
           monoterpene and sesquiterpene oxidation  
 ISOA/G: the lumped semivolatile oxidation products of isoprene ox  
 ASOA/G: the lumped semivolatile (and nonvolatile) products of  
           benzene, toluene, and xylene oxidation

| Parent HC | Oxidized by | Products |
|-----------|-------------|----------|
|-----------|-------------|----------|

```

=====
MTPA OH, O3, NO3 TSOA/G0-3
LIMO OH, O3, NO3 TSOA/G1-3
MTPO OH, O3, NO3 TSOA/G0-3
SESQ OH, O3, NO3 TSOA/G1-3
ISOP OH, NO3 ISOA/G1-3
BENZ OH, (+NO,HO2) ASOAN, ASOA/G1-3
TOLU OH, (+NO,HO2) ASOAN, ASOA/G1-3
XYLE OH, (+NO,HO2) ASOAN, ASOA/G1-3
=====

```

Species that must be defined in input.geos (in addition to standard full chem species) (25 additional):

```

TSOA1 TSOG1 ISOA1 ISOG1 ASOA1 ASOG1
TSOA2 TSOG2 ISOA2 ISOG2 ASOA2 ASOG2
TSOA3 TSOG3 ISOA3 ISOG3 ASOA3 ASOG3
ASOAN TSOA0 TSOG0
BENZ TOLU XYLE MTPA LIMO MTPO

```

The following should NOT be defined for traditional POA:

NAP, POA/G OPOA/G

References (see above for full citations):

=====

Monoterpenes and sesquiterpenes:

Experimental Data:

Griffin et al. 1999 JGR (sesquiterps low NOx)  
 Shilling et al. 2008 ACP (a-pinene ozonolysis for MTPO/MTPA)  
 Zhang et al. 2006 JPhysChemA (limonene ozonolysis)  
 Ng et al. 2007 ACP (data for NOx effect on sesq aerosol)

Modeling:

Chung and Seinfeld 2002 JGR (original formulation in GEOS-Chem)  
 Liao et al. 2007 JGR (comparison to measurements)  
 Pye et al. in prep 2010 (new lumping scheme, NOx effect)

Isoprene

Kroll et al. 2006 ES&T (low NOx experiments)  
 Ng et al. 2008 ACP (isoprene + NO3 experiments)  
 Henze et al. 2006 GRL (low NOx isoprene modeling in GEOS-Chem)  
 Pye et al. in prep 2010 (new lumping scheme and isop+no3 modeling)

Aromatics: benz, tolu, xyle

Ng et al. 2007 ACP (experiments)  
 Henze et al. 2008 ACP (global modeling)

POA/OPOA

Shrivastava et al. 2006 ES&T (POA experiments)  
 Grieshop et al. 2009 ACP (POA/SVOC oxidation experiments)  
 Pye and Seinfeld 2010 ACP (global modeling)

IVOC/Naphthalene

Chan et al. 2009 ACP (experiments)  
 Pye and Seinfeld 2010 ACP (global modeling)



**RETURN VALUE:**

```
REAL(fp) :: SOA_MASS
```

**REMARKS:**

This version does NOT assume that the gas and aerosol phases are in equilibrium before chemistry; therefore, gas phase concentrations are needed explicitly. The gas and aerosol phases are assumed to be in equilibrium after chemistry.

Note: Unlike FUNCTION SOA, this function assumes no reactions. It only considers the partitioning of existing products of VOC oxidation.

```
HC_JHC + OXID_IOXID - >
 alpha(1,IOXID,JHC) [SOAprod_gas(1,IOXID,JHC)+SOAprod(1,IOXID,JHC)]+
 alpha(2,IOXID,JHC) [SOAprod_gas(2,IOXID,JHC)+SOAprod(2,IOXID,JHC)]
```

```
SOAprod_gas(IPR,IOXID,JHC) <--> SOAprod(IPR,IOXID,JHC)
 (aerosol phase)
```

w/ equilibrium partitioning:

$$\text{SOAprod\_gas}(\text{IPR},\text{IOXID},\text{JHC}) = \frac{\text{SOAprod}(\text{IPR},\text{IOXID},\text{JHC})}{\text{Kom}(\text{IPR},\text{IOXID},\text{JHC})}$$

**NOTES:**

- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers

**8.2.9 zeroin**

Function ZEROIN computes a zero of the function f(x) in the interval ax,bx.

**INTERFACE:**

```

FUNCTION ZEROIN(AX,BX,TOL,MPOC,AEROSOL,GAS,KOM) RESULT(MNEW)
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: ax
REAL(fp), INTENT(IN) :: bx
REAL(fp), INTENT(IN) :: tol
REAL(fp), INTENT(IN) :: Mpoc
REAL(fp), INTENT(IN) :: Aerosol(MPROD,MSV)
REAL(fp), INTENT(IN) :: Gas(MPROD,MSV)
REAL(fp), INTENT(IN) :: Kom(MPROD,MSV)
```

**RETURN VALUE:**

REAL(fp)                    :: MNEW

**REMARKS:**

NOTE: This function may be problematic -- it uses GOTO's, which are not good for parallelization. (bmy, 7/8/04)

shc I got this code from <http://www.netlib.org>

a zero of the function f(x) is computed in the interval ax,bx .

input..

ax     left endpoint of initial interval  
 bx     right endpoint of initial interval  
 f     function subprogram which evaluates f(x) for any x in  
       the interval ax,bx  
 tol    desired length of the interval of uncertainty of the  
       final result ( .ge. 0.0e+0\_fp)

output..

zeroin abscissa approximating a zero of f in the interval ax,bx

it is assumed that f(ax) and f(bx) have opposite signs without a check. zeroin returns a zero x in the given interval ax,bx to within a tolerance  $4 * \text{macheps} * \text{abs}(x) + \text{tol}$ , where macheps is the relative machine precision.

this function subprogram is a slightly modified translation of the algol 60 procedure zero given in richard brent, algorithms for minimization without derivatives, prentice - hall, inc. (1973).

**REVISION HISTORY:**

- (1 ) Change dabs to ABS and dsign to SIGN, in order to avoid conflicts with intrinsic function names on the PGI compiler. (bmy, 12/2/04)
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers

**8.2.10 rtbis**

Function RTBIS finds the root of the function SOA\_EQUIL via the bisection method. Original algorithm from "Numerical Recipes" by Press et al, Cambridge UP, 1986. Modified for inclusion into GEOS-CHEM. (bmy, 7/8/04)

**INTERFACE:**

```

 FUNCTION RTBIS(X1, X2, XACC,
& MPOC, AEROSOL, GAS, KOM) RESULT(ROOT)

```

**USES:**

```

 USE ERROR_MOD, ONLY : ERROR_STOP

```

**INPUT PARAMETERS:**

```

 REAL(fp), INTENT(IN) :: X1 ! Endpoint #1
 REAL(fp), INTENT(IN) :: X2 ! Endpoint #2
 REAL(fp), INTENT(IN) :: XACC ! Desired accuracy of solution
 REAL(fp), INTENT(IN) :: MPOC ! POA mass [ug/m3]
 REAL(fp), INTENT(IN) :: AEROSOL(MPROD,MSV) ! Aerosol concentration [ug/m3]
 REAL(fp), INTENT(IN) :: GAS(MPROD,MSV) ! Gas-phase concentration [ug/m3]
 REAL(fp), INTENT(IN) :: KOM(MPROD,MSV) ! Equilibrium gas-aerosol
 ! partition coeff. [m3/ug]

```

**RETURN VALUE:**

```

 REAL(fp) :: ROOT

```

**REVISION HISTORY:**

```

 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA +
 semivolatile POA simulations (H. Pye)
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers

```

**8.2.11 soa\_para**

Subroutine SOA\_PARA gives mass-based stoichiometric coefficients for semi-volatile products from the oxidation of hydrocarbons. It calculates secondary organic aerosol yield parameters. Temperature effects are included. Original code from the CALTECH group and modified for inclusion to GEOS-CHEM. (rjp, bmy, 7/8/04, 6/30/08)

**INTERFACE:**

```

 SUBROUTINE SOA_PARA(TEMP, KO3, KOH, KNO3, KOM,
& II, JJ, LL, KRO2NO, KRO2HO2, State_Met)

```

**USES:**

```

 USE State_Met_Mod, ONLY : MetState

```

**INPUT PARAMETERS:**

```

 ! Arguments
 INTEGER, INTENT(IN) :: II ! Longitude index
 INTEGER, INTENT(IN) :: JJ ! Latitude index
 INTEGER, INTENT(IN) :: LL ! Altitude index
 REAL(fp), INTENT(IN) :: TEMP ! Temperature [k]
 TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```



**OUTPUT PARAMETERS:**

```

REAL(fp), INTENT(OUT) :: KO3(MHC) ! Rxn rate for HC oxidation
 ! by O3 [cm3/molec/s]
REAL(fp), INTENT(OUT) :: KOH(MHC) ! Rxn rate for HC oxidation
 ! by OH [cm3/molec/s]
REAL(fp), INTENT(OUT) :: KNO3(MHC) ! Rxn rate for HC oxidation
 ! by NO3 [cm3/molec/s]
REAL(fp), INTENT(OUT) :: KOM(MPROD,MSV) ! Equilibrium gas-aerosol
 ! partition coeff [m3/ug]

! RO2+NO,HO2 rate constants (hotp 5/7/10)
REAL(fp), INTENT(OUT) :: KRO2NO ! RO2+NO rate constant
REAL(fp), INTENT(OUT) :: KRO2HO2 ! RO2+HO2 rate constant

```

**REMARKS:**

## References:

=====

PHOTO-OXIDATION RATE CONSTANTS OF ORGANICS come from:

- (1 ) Atkinson, et al., Int. J. Chem.Kinet., 27: 941-955 (1995)
- (2 ) Shu and Atkinson, JGR 100: 7275-7281 (1995)
- (3 ) Atkinson, J. Phys. Chem. Ref. Data 26: 215-290 (1997)
- (4 ) Some are reproduced in Table 1 of Griffin, et al., JGR 104: 3555-3567
- (5 ) Chung and Seinfeld (2002)

ACTIVATION ENERGIES come from:

- (6 ) Atkinson, R. (1994) Gas-Phase Tropospheric Chemistry of Organic Compounds. J. Phys. Chem. Ref. Data, Monograph No.2, 1-216.
- (7 ) They are also reproduced in Tables B.9 and B.10 of Seinfeld and Pandis (1988).

PARAMETERS FOR ISOPRENE:

- (8 ) Kroll et al., GRL, 109, L18808 (2005)
- (9 ) Kroll et al., Environ Sci Tech, in press (2006)
- (10) Henze and Seinfeld, GRL, submitted (2006)

**REVISION HISTORY:**

- (1 ) Now use temporary variables TMP1, TMP2, TMP3 to pre-store the values of exponential terms outside of DO-loops (bmy, 7/8/04)
  - (2 ) Add parameters for isoprene. Now include grid cell location in subroutine arguments. Define a reference temperature at 295. Now use ITS\_IN\_THE\_TROP to determine if we are in a tropospheric grid box. Now pass II, JJ, LL via the argument list. (dkh, bmy, 5/22/06)
  - (3 ) Corrected confusing documentation. (clh, bmy, 6/30/08)
  - (4 ) Add paramters for aromtics. Add high NOx low NOx index to every parameter, NNOX (dkh, 10/29/06)
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met

derived type object

13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA +  
semivolatile POA simulations (H. Pye)

20 Aug 2013 - M. Sulprizio- Added ProTeX headers

---

### 8.2.12 soa\_para\_init

Subroutine SOA\_PARA\_INIT initializes the ALPHAS and KOMS, the latter at their reference temperature. It is faster to define these seperately as it only needs to be done once. (dkh, 11/12/06)

#### INTERFACE:

```
SUBROUTINE SOA_PARA_INIT(Input_Opt)
```

#### USES:

```
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

#### REMARKS:

NOTE: REFT for KOM\_REF depends on hydrocarbon.

#### REVISION HISTORY:

20 Aug 2013 - M. Sulprizio- Added ProTeX headers

---

### 8.2.13 chem\_nvoc

Subroutine CHEM\_NVOC computes the oxidation of Hydrocarbon by O3, OH, and NO3. This comes from the Caltech group (Hong Liao, Serena Chung, et al) and was incorporated into GEOS-CHEM. (rjp, bmy, 7/6/04,6/1/06)

#### INTERFACE:

```
SUBROUTINE CHEM_NVOC(I, J, L,
& KO3, KOH, KNO3,
& GMO, KNO, KHO2,
& Input_Opt, State_Met, State_Chm,
& State_Diag, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
```

```

 USE DIAG_MOD, ONLY : ADO7_HC
#endif
 USE ErrCode_Mod
 USE Input_Opt_Mod, ONLY : OptInput
 USE State_Met_Mod, ONLY : MetState
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Diag_Mod, ONLY : DgnState
 USE TIME_MOD, ONLY : GET_TS_CHEM, GET_MONTH

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
REAL(fp), INTENT(IN) :: KO3(MHC) ! Rxn rate for HC oxidation
 ! by O3 [cm3/molec/s]
REAL(fp), INTENT(IN) :: KOH(MHC) ! Rxn rate for HC oxidation
 ! by OH [cm3/molec/s]
REAL(fp), INTENT(IN) :: KNO3(MHC) ! Rxn rate for HC oxidation
 ! by NO3 [cm3/molec/s]
! RO2+NO, RO2+HO2 rate constants (hotp 5/7/10)
REAL(fp), INTENT(IN) :: KNO ! RO2+NO rate constant
REAL(fp), INTENT(IN) :: KHO2 ! RO2+HO2 rate constant
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

REAL(fp), INTENT(INOUT) :: GMO(MPROD,MSV)! Gas mass for HCs and
 ! oxidation products [kg]
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

```

SVOCs should immediately partition upon emission
SVOCs also react in the gas-phase
If SVOCs were emitted before reactions, we wouldn't know how
much to put in each phase
H.O.T. Pye decided to emit them after the existing SVOCs
react in the gas-phase. Thus the order of operations is:
 SVOC + OH in gas-phase
 SVOC emission (added to gas-phase GMO)
 partitioning
 dry dep
 wet dep
 etc

```



**REMARKS:**

NOTE: GPROD and APROD are mass ratios of individual oxidation products of gas/aerosol to the sum of all.

**REVISION HISTORY:**

(1 ) Now references STT from "tracer\_mod.f" (bmy, 7/20/04)  
 (2 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 (3 ) Updated for SOG4, SOA4 (bmy, 5/22/06)  
 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.

**8.2.15 soa\_lump**

Subroutine SOA\_LUMP returns the organic gas and aerosol back to the STT array. (rjp, bmy, 7/7/04, 2/6/07)

**INTERFACE:**

```
SUBROUTINE SOA_LUMP(I, J, L, GMO, AMO, State_Chm, State_Diag)
```

**USES:**

```
USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : ADO7_HC
#endif
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
REAL(fp), INTENT(IN) :: GMO(MPROD,MSV) ! Gas mass for HCs and
 ! oxidation products [kg]
REAL(fp), INTENT(IN) :: AMO(MPROD,MSV) ! Aer mass for HCs and
 ! oxidation products [kg]
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State obj
```

**REVISION HISTORY:**

- (1 ) Now references STT from "tracer\_mod.f" (bmy, 7/20/04)
- (2 ) Bug fix: make sure L <= LD07 before saving into AD07 array, or else we will get an out-of-bounds error. (bmy, 3/4/05)
- (3 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (4 ) Updated for SOG4, SOA4 (dkh, bmy, 5/22/06)
- (5 ) Typo fix: GPROD should be APROD in a couple places (tmf, bmy, 10/16/06)
- (6 ) Bug fix: For SOA4, GPROD and APROD should have default values of 0.5, instead of 1.0 (dkh, bmy, 2/6/07)
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.
- 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

**8.2.16 emitsgc**

Subroutine EMITSGC calculates sub-grid coagulation for the size distribution of emission. (win, 10/6/07)

**INTERFACE:**

```

SUBROUTINE EMITSGC(EMISMASS, CTYPE,
& Input_Opt, State_Met, State_Chm)

```

**USES:**

```

USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD ! ND59
USE DIAG_MOD, ONLY : AD59_ECIL, AD59_ECOB
USE DIAG_MOD, ONLY : AD59_OCIL, AD59_OCOB
USE DIAG_MOD, ONLY : AD59_NUMB
#endif
USE ERROR_MOD, ONLY : IT_IS_NAN
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_MAX_L
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TOMAS_MOD, ONLY : IBINS, AVGMASS, ICOMP, IDIAG
USE TOMAS_MOD, ONLY : SRTECIL, SRTECOB, SRTOCIL
USE TOMAS_MOD, ONLY : SRTOCOB, SRTS04, SRTNH4
USE TOMAS_MOD, ONLY : SRTH20, MNFIX
USE TOMAS_MOD, ONLY : SUBGRIDCOAG
USE TOMAS_MOD, ONLY : SGCTSCALE
USE TOMAS_MOD, ONLY : NH4BULKTOBIN

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: EMISMASS(IIPAR, JJPAR, IBINS)
INTEGER, INTENT(IN) :: CTYPE ! 1 = EC and 2 = OC
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**REVISION HISTORY:**

```

09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
20 Aug 2013 - M. Sulprizio- Added ProTeX headers
26 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID and
 remove dependency on pressure_mod.
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.

```

**8.2.17 scalecarb**

Function SCALECARB split the carbonaceous emission from each source into the TOMAS aerosol size bins using different mass distribution for fossil fuel and biomass burning+biofuel. The mass size distributions are different for EC and OC. (win, 9/4/07)

**INTERFACE:**

```

FUNCTION SCALECARB(BULKEMIS, STYPE, CTYPE) RESULT(tomasdist)

```

**USES:**

```

USE CMN_SIZE_MOD
USE TOMAS_MOD, ONLY : IBINS
USE TOMAS_MOD, ONLY : OCSCALE30, ECSCALE30
USE TOMAS_MOD, ONLY : OCSCALE100, ECSCALE100

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: BULKEMIS(IIPAR, JJPAR)
INTEGER, INTENT(IN) :: STYPE, CTYPE

```

**RETURN VALUE:**

```

REAL(fp) :: tomasdist(IIPAR, JJPAR, IBINS)

```

**REMARKS:**

```

STYPE (source type): 1 = Fossil fule
 2 = Biofuel
 3 = Biomass burning

```

```

CTYPE (carbon type): 1 = EC
 2 = OC

```

Array ECSCALE30 and OCSCALE100 specify how mass is distributed into bins for a 30 nm number peak and a 100 nm peak. Similary for OC size split.

This function is adapted from emisOCbond.f and emisBCbond.f by Jeff Pierce (Jan, 2007) used in GISS GCM-II'. Introduced to GEOS-Chem by Win T.(9/4/07)

## REVISION HISTORY:

20 Aug 2013 - M. Sulprizio- Added ProTeX headers

---

### 8.2.18 emisscarbon

Subroutine EMISSCARBON is the emissions routine for the carbon module. All carbon emissions, incl. SESQ and SVOC, are calculated through HEMCO and this module simply makes sure that the SESQ and SVOC emissions (if defined) are properly passed to the internal arrays.

## INTERFACE:

```

SUBROUTINE EMISSCARBON(am_I_Root, Input_Opt,
& State_Met, RC)

```

## USES:

```

USE CMN_SIZE_MOD, ONLY : IIPAR, JJPAR
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE HCO_INTERFACE_MOD, ONLY : HcoState, GetHcoID, GetHcoVal
USE HCO_ERROR_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_PBL_MAX_L
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS

```

## INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

## INPUT/OUTPUT PARAMETERS:

```

INTEGER, INTENT(INOUT) :: RC ! Failure?

```

## REMARKS:



SVOC emissions are expected to be fully calculated by HEMCO, i.e. its emissions need be specified in the HEMCO configuration file. In the original code, the emissions were calculated by scaling anthropogenic, biofuel and biomass burning OC emissions. The same behavior can be achieved in HEMCO by assigning the desired SVOC species name to the given source type, e.g.:

```
0 BOND_ANTH_POG1 Bond_fossil.nc OC 2000/1-12/1/0 C xy kg/m2/s POG1 74 1 1
```

All POG1 emissions (anthropogenic + biofuel + biomass burning) will go into POAEMISS(:, :, :, 1) and all POG2 emissions will go into POAEMISS(:, :, :, 2). SVOC emissions are assigned to POG1 and POG2 in HEMCO using a ratio of 0.49:0.51. We no longer separate anthropogenic from biofuel and biomass burning since this appears to have been done only for debugging purposes. Routine CHEM\_NVOC handles passing POAEMISS to the two gas-phase semivolatile species in the GMO array.

IMPORTANT: The SVOC emissions scale factor should be applied through HEMCO. In the example above, scale factor 74 represents the scale factor POGSCAL. The SCALING\_POG1 scale factor is applied to the GFED biomass burning extensions. The two scale factors should be set to the same value in the HEMCO configuration file. The recommended value is 1.27.

## REVISION HISTORY:

```
11 Nov 2014 - C. Keller - Initial version
11 Sep 2015 - E. Lundgren - Remove State_Chm from arguments since not used
14 Jan 2016 - M. Sulprizio- Emit SVOC emissions as POG1 and POG2 (not POA1
 and POG1) in HEMCO to better reflect that these
 emissions are added to the gas-phase species
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
```

### 8.2.19 emisscarbontomas

Subroutine emisscarbontomas scales BULK HEMCO emissions into TOMAS arrays. Only use for TOMAS simulations. This is essential a re-write of the TOMAS portions of the v9 emisscarbon (JKodros 6/2/15)

## INTERFACE:

```
SUBROUTINE EMISSCARBONTOMAS(am_I_Root, Input_Opt,
& State_Met, State_Chm, RC)
```

## USES:

```
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
#endif
 USE CMN_SIZE_MOD
 USE ErrCode_Mod
 USE ERROR_MOD
```

```

 USE GC_GRID_MOD, ONLY : GET_XOFFSET, GET_YOFFSET
 USE Input_Opt_Mod, ONLY : OptInput
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Met_Mod, ONLY : MetState
 USE UnitConv_Mod, ONLY : Convert_Spc_Units
TOMAS DIAGNs:
 USE PRESSURE_MOD, ONLY : GET_PCENTER
 USE TOMAS_MOD, ONLY : IBINS, AVGMASS, SOACOND
 USE TOMAS_MOD, ONLY : ICOMP, IDIAG
 USE TOMAS_MOD, ONLY : CHECKMN

 ! HEMCO update
 USE HCO_INTERFACE_MOD, ONLY : HcoState, GetHcoDiagn

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

12 Jun 2015 - J. Kodros - Initial version
12 Jun 2015 - R. Yantosca - Bug fix: also add reference to DEBUG_MSG
10 Jul 2015 - R. Yantosca - Fixed typo in ProTeX header
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
25 Jun 2017 - S. Farina - take 3d emis from hemco, squash it to surface, and
 use subgrid coag or emithigh2 to spread thru pbl.
 real 3d emis might happen next...

```

**8.2.20 emithigh2**

Subroutine EMITHIGH2 mixes species completely from the surface to the PBL top. This is a copy of subroutine EMITHIGH modified to work with 30-bin EC and OC mass and also aerosol number. (win, 9/4/07)

**INTERFACE:**

```

SUBROUTINE EMITHIGH2(BCSRC, OCSRC, NUMBSRC,
& Input_Opt, State_Chm)

```

**USES:**

```

USE CMN_SIZE_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_MAX_L
USE State_Chm_Mod, ONLY : ChmState
USE TOMAS_MOD, ONLY : IBINS

```

**INPUT PARAMETERS:**

```

REAL(fp), INTENT(IN) :: BCSRC(IIPAR,JJPARG,IBINS, 2) !Total BC [kg]
REAL(fp), INTENT(IN) :: OCSRC(IIPAR,JJPARG,IBINS, 2) !Total OC [kg]
REAL(fp), INTENT(IN) :: NUMBSRC(IIPAR,JJPARG,IBINS)
TYPE(OptInput), INTENT(IN) :: Input_Opt !Input Options

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**REVISION HISTORY:**

- (1 ) Now also mix ALPH, LIMO, ALCO tracers (rjp, bmy, 7/8/04)
- (2 ) Now reference STT from "tracer\_mod.f" (bmy, 7/20/04)
- (3 ) Remove references to "dao\_mod.f", "pressure\_mod.f", and "error\_mod.f".  
       Rewrote for computational expediency using routines from  
       "pbl\_mix\_mod.f". (bmy, 2/17/05)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
               species ID from State\_Chm%Map\_Advect.

**8.2.21 ohno3time**

Subroutine OHNO3TIME computes the sum of cosine of the solar zenith angle over a 24 hour day, as well as the total length of daylight. This is needed to scale the offline OH and NO3 concentrations. (rjp, bmy, 12/16/02, 1/18/05)

**INTERFACE:**

```

SUBROUTINE OHNO3TIME

```

**USES:**

```

USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID_R
USE TIME_MOD, ONLY : GET_NHMSb, GET_ELAPSED_SEC
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_DAY_OF_YEAR, GET_GMT

```

**REVISION HISTORY:**

- (1 ) Copy code from COSSZA directly for now, so that we don't get NaN values. Figure this out later (rjp, bmy, 1/10/03)
- (2 ) Now replace XMID(I) with routine GET\_XMID from "grid\_mod.f".  
Now replace RLAT(J) with routine GET\_YMID\_R from "grid\_mod.f".  
Removed NTIME, NHMSb from the arg list. Now use GET\_NHMSb, GET\_ELAPSED\_SEC, GET\_TS\_CHEM, GET\_DAY\_OF\_YEAR, GET\_GMT from "time\_mod.f". (bmy, 3/27/03)
- (3 ) Now store the peak SUNCOS value for each surface grid box (I,J) in the COSZM array. (rjp, bmy, 3/30/04)
- (4 ) Also added parallel loop over grid boxes (bmy, 1/18/05)
- 01 Mar 2012 - R. Yantosca - Now use GET\_XMID(I,J,L) from grid\_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID\_R(I,J,L) from grid\_mod.F90
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 16 May 2016 - M. Sulprizio- Remove IJLOOP and change SUNTMP array dimensions from (MAXIJ) to (IIPAR,JJPAR)

### 8.2.22 get\_oh

Function GET\_OH returns OH from State\_Chm (coupled runs) or monthly mean OH (for offline runs). Imposes a diurnal variation on OH for offline simulations. (bmy, 7/9/04)

#### INTERFACE:

```
FUNCTION GET_OH(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(OH_MOLEC_CM3)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: OH_MOLEC_CM3
```

#### REVISION HISTORY:

(1 ) We assume SETTRACE has been called to define IDOH (bmy, 11/1/02)  
 (2 ) Now use function GET\_TS\_CHEM from "time\_mod.f" (bmy, 3/27/03)  
 (3 ) Now reference inquiry functions from "tracer\_mod.f" (bmy, 7/20/04)  
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
 28 Nov 2012 - R. Yantosca - Add State\_Met to the argument list  
 4 Mar 2013 - R. Yantosca - Add Input\_Opt to the argument list  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 23 Sep 2014 - M. Sulprizio- Now get OH for offline aerosol sim from HEMCO  
 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species  
 07 Sep 2016 - M. Sulprizio- Bug fix: Convert State\_Chm%Species from kg to  
 molec/cm3

---

### 8.2.23 get\_no3

Function GET\_NO3 returns NO3 from State\_Chm coupled runs) or monthly mean OH (for offline runs). For offline runs, the concentration of NO3 is set to zero during the day. (rjp, bmy, 12/16/02, 7/20/04)

#### INTERFACE:

```
FUNCTION GET_NO3(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(NO3_MOLEC_CM3)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: NO3_MOLEC_CM3
```

#### REVISION HISTORY:

(1 ) Now references ERROR\_STOP from "error\_mod.f". We also assume that SETTRACE has been called to define IDNO3. Now also set NO3 to zero during the day. (rjp, bmy, 12/16/02)

(2 ) Now reference inquiry functions from "tracer\_mod.f" (bmy, 7/20/04)  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met  
                   derived type object  
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
 04 Mar 2013 - R. Yantosca - Add Input\_Opt to the argument list  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN\_MOD  
 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally  
 23 Sep 2014 - M. Sulprizio- Now get NO3 for offline aerosol sims from HEMCO  
 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species  
 28 Jul 2016 - R. Yantosca - Now convert NO3 from kg to molec/cm3  
 07 Sep 2016 - M. Sulprizio- Get MolecRatio and NO3\_MW\_kg from species DB

### 8.2.24 get\_o3

Function GET\_O3 returns monthly mean O3 for offline sulfate aerosol simulations. (bmy, 12/16/02, 7/20/04)

#### INTERFACE:

```
FUNCTION GET_O3(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(O3_MOLEC_CM3)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: O3_MOLEC_CM3
```

#### REVISION HISTORY:

(1 ) We assume SETTRACE has been called to define IDO3. (bmy, 12/16/02)  
 (2 ) Now reference inquiry functions from "tracer\_mod.f" (bmy, 7/20/04)

(3 ) Now reference XNUMOLAIR from "tracer\_mod.f" (bmy, 10/20/05)  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met  
                   derived type object  
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
 04 Mar 2013 - R. Yantosca - Add Input\_Opt to the argument list  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally  
 18 Sep 2014 - M. Sulprizio- Now get O3 for offline aerosol sim from HEMCO  
 24 Mar 2015 - E. Lundgren - Replace dependency on tracer\_mod with  
                   CMN\_GTCM\_MOD for XNUMOLAIR  
 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species  
 06 Jan 2016 - E. Lundgren - Use global physical parameter AVO  
 28 Jul 2016 - R. Yantosca - Now convert O3 from kg to molec/cm3  
 07 Sep 2016 - M. Sulprizio- Get MolecRatio and NO3\_MW\_kg from species DB

### 8.2.25 get\_daro2

Function GET\_DARO2 returns the amount of aromatic peroxy radical that reacted with HO2 or NO during the last chemistry timestep. (dkh, 11/10/06)

#### INTERFACE:

```
FUNCTION GET_DARO2(I, J, L, NOX, JHC, Input_Opt,
& State_Chm, State_Met)
& RESULT(DARO2)
```

#### USES:

```
USE CMN_O3_MOD
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
INTEGER, INTENT(IN) :: NOX
INTEGER, INTENT(IN) :: JHC
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: DAR02
```

## REVISION HISTORY:

```
04 Mar 2013 - R. Yantosca - Add Input_Opt to the argument list
13 Aug 2013 - M. Sulprizio- Add NAP for SOA + semivolatle POA (H. Pye)
20 Aug 2013 - M. Sulprizio- Added ProTeX headers
21 Dec 2015 - M. Sulprizio- Get grid box volume directly from State_Met
22 Dec 2015 - M. Sulprizio- Replace CSPEC with State_Met%Species
17 May 2016 - R. Yantosca - Now get MolecRatio from species database
31 May 2016 - E. Lundgren - Now get molecular weight from species database
07 Sep 2016 - M. Sulprizio- Bug fix: Convert State_Chm%Species from kg LAR02
 to kg AROM
```

### 8.2.26 get\_doh

Function GET\_DOH returns the amount of isoprene [kg] that has reacted with OH during the last chemistry time step. (dkh, bmy, 6/01/06)

## INTERFACE:

```
FUNCTION GET_DOH(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(DOH)
```

## USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

## INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

## RETURN VALUE:

```
REAL(fp) :: DOH
```

## REVISION HISTORY:



04 Mar 2013 - R. Yantosca - Now use fields from Input\_Opt object  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 21 Dec 2015 - M. Sulprizio- Get grid box volume directly from State\_Met  
 17 May 2016 - R. Yantosca - Now get MolecRatio from species database  
 31 May 2016 - E. Lundgren - Now get molecular weight from species database  
 07 Sep 2016 - M. Sulprizio- Bug fix: Convert State\_Chm%Species from kg ISOP0H  
 to kg ISOP

---

## 8.2.27 check\_eqlb

Subroutine CHECK\_EQLB makes sure aerosols are at equilibrium (checks SOA=SOG\*KOM\*Mo).  
 Called inside SOA\_SVPOA\_CHEMISTRY I, J, L loop after SOA\_SVPOA\_LUMP. Created  
 by Havala Pye (5/18/10).

### INTERFACE:

```

SUBROUTINE CHECK_EQLB(I, J, L, KOMIJL, CONVFAC, MSOACHEM,
& LOW, TOL, ASOANGAS, ASOANAER, OCPIOCPO,
& State_Chm)

```

### USES:

```

USE State_Chm_Mod, ONLY : ChmState

```

### INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
REAL(fp), INTENT(IN) :: KOMIJL(MPROD,MSV) ! KOM at grid box (adj T)
REAL(fp), INTENT(IN) :: CONVFAC ! Conversion factor kg to ug/m3
REAL(fp), INTENT(IN) :: OCPIOCPO ! POA mass [ug/m3]

! Arguments for debugging
REAL(fp), INTENT(IN) :: MSOACHEM ! MNEW from calling prog
REAL(fp), INTENT(IN) :: LOW ! Lower bound on soln
REAL(fp), INTENT(IN) :: TOL ! Tolerance on soln
REAL(fp), INTENT(IN) :: ASOANGAS ! Gas phase ASOAN (should =0)
REAL(fp), INTENT(IN) :: ASOANAER ! Aer phase ASOAN [ug/m3]

TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object

```

### REMARKS:

Note: There are some deviations from equilibrium due to the fact  
 that ASOAN is supposed to be nonvolatile, but is modeled with a KOM of  
 $10^6$ . An adjustment is made in SOA\_CHEMISTRY to force all ASOAN to  
 the aerosol phase. This was found to lead to error up to  $1e-5$  ug/m3

in Mo. This error is small, but the effects can be investigated here if you're interested!

As of 6/2010, KOM for ASOAN was increased and the error in Mo reduced.

### REVISION HISTORY:

- (1) Updated for TSOA and ISOA (hotp 5/24/10)
- (2) Add OCPIOCPD and remove NOX (hotp 6/9/10)
- (3) Add TSOAO (hotp 6/12/10)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.

### 8.2.28 save\_oaginit

Subroutine SAVE\_OAGINIT saves total SOA+SOG before partitioning for diagnostic purposes. Units are the same as the STT array ([kg] or [kgC per box]). created hotp 5/17/10

#### INTERFACE:

```
SUBROUTINE SAVE_OAGINIT(State_Chm)
```

#### USES:

```
USE CMN_SIZE_MOD
USE State_Chm_Mod, ONLY : ChmState
```

#### INPUT PARAMETERS:

```
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
```

### REVISION HISTORY:

- (1) added TSOA and ISOA (hotp 5/24/10)
- (2) OAGINITSAVE dimensions changes from (I,J,L,NOx,NPROD,JSV) to (I,J,L,NPROD,JSV)
- (3) Add compatibility with non-vol sim (hotp 6/7/10)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.

### 8.2.29 check\_mb

Subroutine CHECK\_MB checks total SOA+SOG mass balance for diagnostic/debugging purposes. Units are the same as the STT array ([kg] or [kgC per box]). Routine also prints helpful budget info. Created by Havala Pye (5/18/10).

#### INTERFACE:

```
SUBROUTINE CHECK_MB(am_I_Root, Input_Opt, State_Met, State_Chm)
```

**USES:**

```
USE CMN_SIZE_MOD
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
```

**REVISION HISTORY:**

- (1) added monoterpene, sesq, isoprene SOA (hotp 5/24/10)
- (2) updated OAGINITSAVE dimensions (hotp 5/24/10)
- (3) keeps track and prints to screen amount of parent HC reacted  
with each oxidant cumulative (hotp 5/24/10)
- (4) Add non-volatile compatability (hotp 6/9/10)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
species ID from State\_Chm%Map\_Advect.

**8.2.30 get\_no**

Function GET\_NO returns NO from State\_Chm(for coupled runs). (hotp 5/7/2010)

**INTERFACE:**

```
FUNCTION GET_NO(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(NO_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
!RETURN VALUE
REAL(fp) :: NO_MOLEC_CM3 ! NO conc [molec/cm3]
```

**REVISION HISTORY:**

(1 ) We assume SETTRACE has been called to define IDNO (bmy, 11/1/02)  
 (3 ) Now reference inquiry functions from "tracer\_mod.f" (bmy, 7/20/04)  
 (4 ) Based on GET\_OH (hotp 5/7/2010)  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 25 Jun 2014 - R. Yantosca - Now accept Input\_Opt via the arg list  
 25 Jun 2014 - R. Yantosca - Remove references to tracer\_mod.F  
 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species  
 28 Jul 2016 - R. Yantosca - Now convert NO from kg to molec/cm3  
 07 Sep 2016 - M. Sulprizio- Get MolecRatio and NO3\_MW\_kg from species DB

**8.2.31 get\_ho2**

Function GET\_HO2 returns HO2 from State\_Chm(for coupled runs). Created by Havalala Pye (5/7/2010).

**INTERFACE:**

```
FUNCTION GET_HO2(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(HO2_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
!RETURN VALUE
REAL(fp) :: HO2_MOLEC_CM3 ! HO2 conc [molec/cm3]
```

**REVISION HISTORY:**

(1 ) We assume SETTRACE has been called to define IDHO2 (bmy, 11/1/02)  
 (3 ) Now reference inquiry functions from "tracer\_mod.f" (bmy, 7/20/04)  
 (4 ) Based on GET\_OH (hotp 5/6/2010)  
 20 Aug 2013 - M. Sulprizio- Added ProTeX headers  
 25 Jun 2014 - R. Yantosca - Now accept Input\_Opt via the arg list  
 25 Jun 2014 - R. Yantosca - Remove references to tracer\_mod.F  
 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species

### 8.2.32 get\_isopno3

Modification of GET\_DOH that returns the amount of isoprene [kgC] that has reacted with NO3 during the last chemistry time step. (hotp 5/22/10)

#### INTERFACE:

```
FUNCTION GET_ISOPNO3(I, J, L, Input_Opt, State_Chm, State_Met)
& RESULT(ISOPNO3)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Altitude index
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
!RETURN VALUE
REAL(fp) :: ISOPNO3 ! ISOP replaced w/ NO3 [kg C]
```

#### REVISION HISTORY:

- (1) IDLISOPNO3 is declared in tracerid\_mod.f and initialized by SETTRACE in tracerid\_mod (called in chemdr). Before each chemistry call, CSPEC(JLOOP,IDLISOPNO3) is zeroed so that the CSPEC array only stores the parent HC reacted during that timestep. (hotp 6/1/10)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
- 25 Jun 2014 - R. Yantosca - Now accept Input\_Opt via the arg list
- 25 Jun 2014 - R. Yantosca - Remove references to tracer\_mod.F
- 21 Dec 2015 - M. Sulprizio- Get grid box volume directly from State\_Met
- 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Met%Species
- 17 May 2016 - R. Yantosca - Now get MolecRatio from species database
- 31 May 2016 - E. Lundgren - Now get molecular weight from species database
- 07 Sep 2016 - M. Sulprizio- Bug fix: Convert State\_Chm%Species from kg ISOPOH to kg ISOP

### 8.2.33 init\_carbon

Subroutine INIT\_CARBOON initializes all module arrays. (rjp, bmy, 4/1/04, 12/19/09)

#### INTERFACE:

```
SUBROUTINE INIT_CARBON(am_I_Root, Input_Opt, State_Diag, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR, ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
#if defined(TOMAS)
USE TOMAS_MOD, ONLY : IBINS
#endif
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object
```

**INPUT/OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure
```

**REVISION HISTORY:**

- ( 1 ) Also added arrays for secondary organic aerosols (rjp, bmy, 7/8/04)
  - ( 2 ) Remove reference to CMN, it's obsolete (bmy, 7/20/04)
  - ( 3 ) Now reference LSOA from "logical\_mod.f" not CMN\_SETUP. Now call GET\_BOUNDING\_BOX from "grid\_mod.f" to compute the indices I1\_NA, I2\_NA, J1\_NA, J2\_NA which define the N. America region. (bmy, 12/1/04)
  - ( 4 ) Now call READ\_GPROD\_APROD to read GPROD & APROD from disk. (tmf, havala, bmy, 2/6/07)
  - ( 5 ) Now set I1\_NA, I2\_NA, J1\_NA, J2\_NA appropriately for both 1 x 1 and 0.5 x 0.666 nested grids (yxw, dan, bmy, 11/6/08)
  - ( 6 ) Now set parameters for NESTED\_EU grid (amv, bmy, 12/19/09)
  - 14 Jan 2011 - R. Yantosca - If we are using GEOS-5 or MERRA met, then get the cloud fraction directly from the met fields.
  - 01 Mar 2012 - R. Yantosca - Now use GET\_BOUNDING\_BOX from grid\_mod.F90
  - 04 Mar 2013 - R. Yantosca - Now take am\_I\_Root, Input\_Opt, RC as arguments
  - 04 Mar 2013 - R. Yantosca - Now search for drydep flags here
  - 05 Mar 2013 - R. Yantosca - Now use Input\_Opt%LSOA
  - 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
  - 20 Aug 2013 - M. Sulprizio- Added ProTeX headers
  - 26 Sep 2013 - R. Yantosca - Removed SEAC4RS Cpp switch, this is supplanted by NESTED\_NA
  - 26 Sep 2013 - R. Yantosca - Renamed GEOS\_57 Cpp switch to GEOS\_FP
  - 11 Aug 2015 - R. Yantosca - Add support for MERRA2 data
  - 23 Sep 2015 - R. Yantosca - Remove reference to DRY\* flags, they're obsolete
  - 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument
-

### 8.2.34 cleanup\_carbon

Subroutine CLEANUP\_CARBON deallocates all module arrays (rjp, bmy, 4/1/04, 7/8/04)

#### INTERFACE:

```
SUBROUTINE CLEANUP_CARBON
```

#### REVISION HISTORY:

- (1 ) Now deallocate arrays for secondary organic aerosols (rjp, bmy, 7/8/04)
- 13 Aug 2013 - M. Sulprizio- Add modifications for updated SOA and SOA + semivolatile POA simulations (H. Pye)
- 20 Aug 2013 - M. Sulprizio- Added ProTeX headers

## 8.3 Fortran: Module Interface dust\_mod.F

Module DUST\_MOD contains routines for computing dust aerosol emissions, chemistry, and optical depths.

#### INTERFACE:

```
MODULE DUST_MOD
```

#### USES:

```
USE inquireMod, ONLY : findFreeLUN
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: CHEMDUST
#if defined(TOMAS)
PUBLIC :: SETTLEDUST
#endif
PUBLIC :: RDUST_ONLINE
PUBLIC :: RDUST_OFFLINE
PUBLIC :: GET_DUST_ALK
PUBLIC :: INIT_DUST
PUBLIC :: CLEANUP_DUST
```

#### PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: DRY_SETTLING
```

#### REVISION HISTORY:

30 Mar 2004 - T. D. Fairlie - Initial version  
 (1 ) Bug fix in SRC\_DUST\_DEAD (bmy, 4/14/04)  
 (2 ) Now references "logical\_mod.f", "directory\_mod.f", and "tracer\_mod.f"  
       Added comments. (bmy, 7/2/04)  
 (3 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 (4 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)  
 (5 ) Bug fix in snow height computation (bmy, 11/18/05)  
 (6 ) Now only do drydep if LDRYD=T (bmy, 5/23/06)  
 (7 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)  
 (8 ) Updated output print statement in SRC\_DUST\_DEAD (bmy, 1/23/07)  
 (9 ) Modifications for GEOS-5 (bmy, 1/24/07)  
 (10) Modified to archive only hydrophilic aerosol/aqueous dust surface area  
       (excluding BCPO and OCP0) for aqueous chemistry calculations  
       Dust surfaces are considered aqueous only when RH > 35% (tmf, 3/6/09)  
 (11) Add AOD output for all dust size bins (clh, 5/7/10)  
 (12) Modify AOD output to wavelength specified in jv\_spec\_aod.dat  
       (clh, 05/07/10)

25 Aug 2010 - R. Yantosca - Added ProTeX headers  
 03 Sep 2010 - R. Yantosca - Bug fix in SRC\_DUST\_DEAD  
 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x  
 01 Mar 2012 - R. Yantosca - Now reference the new grid\_mod.F90  
 01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire\_mod.F90  
 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block  
 14 Nov 2012 - R. Yantosca - Add modifications for GIGC  
 04 Mar 2013 - R. Yantosca - Now call INIT\_DUST from the init stage  
       which facilitates connection to GEOS-5 GCM  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust  
       aerosols (T.D. Fairlie)  
 20 Jun 2014 - R. Yantosca - Remove obsolete emissions code; we now use HEMCO  
 13 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 01 Apr 2015 - R. Yantosca - Remove obsolete DUSTMIX, DRY\_DEPOSITION routines  
 16 Jun 2016 - J. Kaiser - Move tracerIDs into variable names  
 20 Jun 2016 - R. Yantosca - Rename IDT\* species ID's to id\_\*  
 20 Jun 2016 - R. Yantosca - Now only define species ID's in the INIT phase  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90  
 17 Mar 2017 - R. Yantosca - Remove reference to Input\_Opt fields  
 02 Nov 2017 - R. Yantosca - Add modifications for netCDF diagnostics  
 15 Nov 2017 - E. Lundgren - Implement dust AOD and surf area nc diagnostics  
 03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input\_Opt%LUCX

---

### 8.3.1 chemdust

Subroutine CHEMDUST is the interface between the GEOS-Chem main program and the dust chemistry routines that mostly calculates dust dry deposition.

#### INTERFACE:



```

SUBROUTINE CHEMDUST(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

30 Mar 2004 - T. D. Fairlie - Initial version
(1) Now references STT from "tracer_mod.f" and LDUST from "logical_mod.f"
 (bmy, 7/20/04)
(5) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(6) Now only do dry deposition if LDRYD = T (bmy, 5/23/06)
25 Aug 2010 - R. Yantosca - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
15 Nov 2012 - M. Payer - Now pass State_Met as an argument
05 Mar 2013 - R. Yantosca - Add ND70 debug print output
25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust
 aerosols (T.D. Fairlie)
01 Apr 2015 - R. Yantosca - Remove call to DRY_DEPOSITION, this is now
 done in mixing_mod.F90.
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
16 Mar 2017 - R. Yantosca - Clean up calls to DRY_SETTLING
02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

---

### 8.3.2 settledust

Subroutine SETTLEDUST is the interface between the size-resolved dry deposition subroutine AERO\_DRYDEP and the dust module. This is to call only gravitational settling and deals with removal of aerosol number with the dust mass. (win, 7/17/09)

#### INTERFACE:

```

SUBROUTINE SETTLEDUST(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD ! Size parameters
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD ! ND44
USE DIAG_MOD, ONLY : AD44
#endif
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD, ONLY : GET_AREA_CM2
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
USE TOMAS_MOD, ONLY : IBINS, Xk, SRTDUST
USE Species_Mod, ONLY : Species

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REVISION HISTORY:

```

17 Jul 2009 - W. Trivitanurak - Initial version
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
13 Dec 2012 - M. Payer - Add am_I_Root, Input_Opt, RC as arguments

```

31 May 2013 - R. Yantosca - Now pass State\_Chm via the arg list  
 16 Jul 2015 - R. Yantosca - Now pass INDEX and IDISP to DRY\_SETTLING  
 31 May 2016 - E. Lundgren - Replace Input\_Opt%XNUMOL with AVO/(emMW\_g\*1e-3)  
                   where emMW\_g is from species database  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
                   species ID from State\_Chm%Map\_Advect.  
 10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code  
 17 Mar 2017 - R. Yantosca - Update subroutine call to DRY\_SETTLING  
 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

### 8.3.3 emissdust

Subroutine EMISSDUST is the driver routine for the dust emission module. You may call either the GINOX or the DEAD dust source function.

#### INTERFACE:

```

SUBROUTINE EMISSDUST(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD ! Size parameters
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD ! ND59
USE DIAG_MOD, ONLY : AD59_DUST, AD59_NUMB !(win, 7/17/09)
#endif
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE State_Met_Mod, ONLY : MetState
USE TOMAS_MOD, ONLY : IBINS, XK !(win, 7/17/09)

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

## REVISION HISTORY:

30 Mar 2004 - T. D. Fairlie - Initial version  
 (1 ) Now reference LDEAD, LDUST, LPRT from "logical\_mod.f". Now reference!  
       STT from "tracer\_mod.f" (bmy, 7/20/04)  
 (2 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 (3 ) Add if condition for selecting between emitting 4-bin or 30-bin  
       dust. Add emission diagnostic calculation for 30bin dust(win, 7/17/09)  
 25 Aug 2010 - R. Yantosca - Added ProTeX headers  
 26 Nov 2012 - R. Yantosca - Now pass am\_I\_Root, Input\_Opt, State\_Met as args  
 26 Feb 2013 - R. Yantosca - Now pass Input\_Opt to SRC\_DUST\_GINOX  
 26 Feb 2013 - R. Yantosca - Changed INPUT\_OPT to INTENT(IN), since we are  
       now no longer calling INIT\_DUST from here,  
       it is now called in the init stage  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
       species ID from State\_Chm%Map\_Advect.  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
       species ID from State\_Chm%Map\_Advect.  
 11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code  
 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

### 8.3.4 dry\_settling

Subroutine DRY\_SETTLING computes the dry settling of dust species.

#### INTERFACE:

```
 SUBROUTINE DRY_SETTLING(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, Ind0,
& Ind1, RC)
```

#### USES:

```
 USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
 USE DIAG_MOD, ONLY : AD44
#endif
 USE ErrCode_Mod
 USE GC_GRID_MOD, ONLY : GET_AREA_CM2
 USE Input_Opt_Mod, ONLY : OptInput
 USE PhysConstants
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Diag_Mod, ONLY : DgnState
```

```

USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
USE Species_Mod, ONLY : Species

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
INTEGER, INTENT(IN) :: Ind0 ! Starting dust species #
INTEGER, INTENT(IN) :: Ind1 ! Ending dust species #

```

! INPUT/OUTPUT PARAMETERS

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

30 Mar 2004 - T. D. Fairlie - Initial version
(1) Updated comments, cosmetic changes (bmy, 3/30/04)
(2) Remove reference to CMN, it's not needed (bmy, 7/20/04)
(3) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
25 Aug 2010 - R. Yantosca - Added ProTeX headers
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
19 Mar 2013 - R. Yantosca - Now copy Input_Opt%XNUMOL(1:N_TRACERS)
12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust
 aerosols (T.D. Fairlie)
26 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID and
 remove dependency on pressure_mod
25 Jan 2016 - E. Lundgren - Update netcdf drydep flux diagnostic
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
31 May 2016 - E. Lundgren - Use TCVV instead of XNUMOL for molecular weights
22 Jun 2016 - M. Yannetti - Replace TCVV with species db MW and phys constant
16 Mar 2017 - R. Yantosca - Remove TC, IDISP; Loop over slots Ind0:Ind1
 of the State_Chm%Species array.
02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

**8.3.5 rdust\_online**

Subroutine RDUST\_ONLINE reads global mineral dust concentrations as determined by P. Ginoux. Calculates dust optical depth at each level for the FAST-J routine "set\_prof.f".

**INTERFACE:**

```

SUBROUTINE RDUST_ONLINE(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, DUST,
& ODSWITCH, RC)

```

**USES:**

```

USE CMN_FJX_MOD
USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : AD21
#endif
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TRANSFER_MOD, ONLY : TRANSFER_3D

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
REAL(fp), INTENT(IN) :: DUST(IIPAR,JJP,LLPAR,NDUST) !Dust [kg/m3]
INTEGER, INTENT(IN) :: ODSWITCH

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

- 01 Apr 2004 - R. Martin, R. Park - Initial version
- (1 ) Bundled into "dust\_mod.f" (bmy, 4/1/04)
- (2 ) Now references DATA\_DIR from "directory\_mod.f". Now parallelize over the L-dimension for ND21 diagnostics. (bmy, 7/20/04)
- (3 ) Archive only hydrophilic aerosol/aqueous dust surface area (excluding BCPO and OCPO), WTAREA and WERADIUS. (tmf, 3/6/09)
- 25 Aug 2010 - R. Yantosca - Added ProTeX headers
- 03 Feb 2011 - S. Kim. - Include wavelength argument to determine the wavelength at which the AOD should be computed.

This will set the optical properties that are used for the calculation of the AOD. The ND21 diagnostic should only be updated when WAVELENGTH = 1. (skim, 02/03/11)

04 Sep 2012 - D. Ridley - WAVELENGTH now ODSWITCH for clarity now that multiple wavelengths can be calculated at once.

09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object

23 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC

12 May 2016 - M. Sulprizio- Remove 1D arrays that depend on JLOOP. ERADIUS, TAREA, WERADIUS, WTAREA are now pointers that point to 3D fields in State\_Chm.

16 May 2016 - M. Sulprizio- Remove JLOOP entirely and loop over LLPAR, JJPAR, IIPAR instead.

27 Mar 2017 - R. Yantosca - Fixed bug in formula for NOUT diagnostic index

03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

15 Nov 2017 - E. Lundgren - Implement dust AOD and surf area nc diagnostics

23 Mar 2018 - E. Lundgren - Get number of wavelengths from Input\_Opt

### 8.3.6 rdust\_offline

Subroutine RDUST\_OFFLINE reads global mineral dust concentrations as determined by P. Ginoux. Calculates dust optical depth at each level for the FAST-J routine "set\_prof.F".

#### INTERFACE:

```

SUBROUTINE RDUST_OFFLINE(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, THISMONTH,
& THISYEAR, ODSWITCH, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
USE CMN_FJX_MOD
#if defined(BPCH_DIAG)
USE BPCH2_MOD, ONLY : GET_NAME_EXT, GET_RES_EXT
USE BPCH2_MOD, ONLY : GET_TAU0, READ_BPCH2
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : AD21
#endif
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TRANSFER_MOD, ONLY : TRANSFER_3D

```

IMPLICIT NONE

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
INTEGER, INTENT(IN) :: THISMONTH ! Current month (1-12)
INTEGER, INTENT(IN) :: THISYEAR ! Current year (YYYY format)
INTEGER, INTENT(IN) :: ODSWITCH ! Determine which wavelength
 ! to use for optical
 ! properties

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REMARKS:

```

#####
NOTE: BINARY PUNCH INPUT IS BEING PHASED OUT. THIS DATA
WILL EVENTUALLY BE READ IN FROM netCDF FILES VIA HEMCO!
-- Bob Yantosca (05 Mar 2015)
#####

```

#### REVISION HISTORY:

- (1 ) RDUST was patterned after rdaerosol.f (rvm, 9/30/00)
- (2 ) Don't worry about rewinding the binary file...reading from binary files is pretty fast. And it's only done once a month.
- (3 ) Now references punch file utility routines from F90 module "bpch2\_mod.f". Also reference variable DATA\_DIR from the header file "CMN\_SETUP". (bmy, 9/30/00)
- (4 ) Now selects proper GEOS-STRAT dust field for 1996 or 1997. Also need to pass THISYEAR thru the arg list. (rvm, bmy, 11/21/00)
- (5 ) CONC is now declared as REAL(fp) (rvm, bmy, 12/15/00)
- (6 ) Removed obsolete code from 12/15/00 (bmy, 12/21/00)
- (7 ) CONC(IIPAR,JJPARG,LLPAR,NDUST) is now CONC(IIPAR,JJPARG,LLPAR,NDUST). Now use routine TRANSFER\_3D from "transfer\_mod.f" to cast from REAL\*4 to REAL(fp) and also to convert from {IJL}GLOB to IIPAR,JJPARG,LLPAR space. Use 3 arguments in call to GET\_TAU0. Updated comments. (bmy, 9/26/01)
- (8 ) Removed obsolete code from 9/01 (bmy, 10/24/01)
- (9 ) Now reference ERADIUS, IXSAVE, IYSAVE, IZSAVE, TAREA from "comode\_mod.f". Compute ERADIUS and TAREA for the NDUST dust size bins from FAST-J. Renamed CONC to DUST to avoid conflicts.



- Also reference NTTLOOP from "comode.h". Also added parallel DO-loops. Also renamed MONTH and YEAR to THISMONTH and THISYEAR to avoid conflicts w/ other variables. (bmy, 11/15/01)
- (10) Bug fix: Make sure to use 1996 dust data for Dec 1995 for the GEOS-STRAT met field dataset. Set off CASE statement with an #if defined( GEOS\_STRAT ) block. (rvm, bmy, 1/2/02)
  - (11) Eliminate obsolete code from 1/02 (bmy, 2/27/02)
  - (12) Now report dust optical depths in ND21 diagnostic at 400 nm. Now report dust optical depths as one combined diagnostic field instead of 7 separate fields. Now reference JLOP from "comode\_mod.f". Now save aerosol surface areas as tracer #5 of the ND21 diagnostic. (rvm, bmy, 2/28/02)
  - (13) Remove declaration for TIME, since that is also defined in the header file "comode.h" (bmy, 3/20/02)
  - (14) Now read mineral dust files directly from the DATA\_DIR/dust\_200203/ subdirectory (bmy, 4/2/02)
  - (15) Now reference BXHEIGHT from "dao\_mod.f". Also reference ERROR\_STOP from "error\_mod.f". (bmy, 10/15/02)
  - (16) Now call READ\_BPCH2 with QUIET=TRUE to suppress extra informational output from being printed. Added cosmetic changes. (bmy, 3/14/03)
  - (17) Since December 1997 dust data does not exist, use November 1997 dust data as a proxy. (bnd, bmy, 6/30/03)
  - (18) Bundled into "dust\_mod.f" and renamed to RDUST\_OFFLINE. (bmy, 4/1/04)
  - (19) Now references DATA\_DIR from "directory\_mod.f". Now parallelize over the L-dimension for ND21 diagnostic. (bmy, 7/20/04)
  - (20) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
  - (21) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
  - (22) Archive only hydrophilic aerosol/aqueous dust surface area (excluding BCPO and OCPO), WTAREA and WERADIUS. (tmf, 3/6/09)
- 25 Aug 2010 - R. Yantosca - Added ProTeX headers
- 03 Feb 2011 - S. Kim - Include third input argument to determine the wavelength at which the AOD should be computed. This will set the optical properties that are used for the calculation of the AOD. The ND21 diagnostic should only be updated when WAVELENGTH = 1.
- 30 Jul 2012 - R. Yantosca - Now accept am\_I\_Root as an argument when running with the traditional driver main.F
- 04 Sep 2012 - D. Ridley - WAVELENGTH now ODSWITCH for clarity now that multiple wavelengths can be calculated at once.
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte
- 05 Mar 2015 - R. Yantosca - Add Input\_Opt%RES\_DIR to data path
- 23 Oct 2015 - E. Lundgren - Fix index for dust in ODAER/LUT arrays if UCX on
- 12 May 2016 - M. Sulprizio - Remove 1D arrays that depend on JLOOP. ERADIUS,



**REVISION HISTORY:**

08 Apr 2008 - T.D. Fairlie- Initial version  
 16 Sep 2013 - M. Sulprizio- Added ProTeX headers  
 17 Sep 2013 - M. Sulprizio- Now pass Input\_Opt, State\_Met, and State\_Chm  
                   as arguments  
 14 Nov 2013 - M. Sulprizio- Bug fix: Avoid div-by-zero in calculation of  
                   gas-to-particle rate constants for SO2 and HNO3  
 05 Jan 2016 - E. Lundgren - Use global PI  
 12 May 2016 - M. Sulprizio- Remove 1D arrays that depend on JLOOP. ERADIUS,  
                   TAREA, are now pointers that point to 3D fields  
                   fields in State\_Chm.  
 22 Jun 2016 - M. Yannetti - Replace TCVV with species db MW and phys constant  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
                   species ID from State\_Chm%Map\_Advect.

**8.3.8 init\_dust**

Subroutine INIT\_DUST allocates all module arrays.

**INTERFACE:**

```

SUBROUTINE INIT_DUST(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : PI, AIRMW
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
#if defined(TOMAS)
USE TOMAS_MOD, ONLY : IBINS, Xk
#endif

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

### REVISION HISTORY:

30 Mar 2004 - R. Yantosca - Initial version  
 (1 ) Now references LDEAD from "logical\_mod.f" (bmy, 7/20/04)  
 (2 ) Modify to work with 30bin dust. Reference to IBINS from tomas\_mod  
       for number of total bins = 30 bins. (win, 7/17/09)  
 25 Aug 2010 - R. Yantosca - Added ProTeX headers  
 14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments  
 26 Feb 2013 - M. Long      - Now use fields from Input\_Opt  
 12 Sep 2013 - M. Sulprizio- Add modifications for acid uptake on dust  
                           aerosols (T.D. Fairlie)  
 23 Sep 2015 - R. Yantosca - Now accept State\_Chm as an argument so that  
                           we can use the species database  
 30 Sep 2015 - R. Yantosca - DD\_A\_Density is now renamed to Density  
 30 Sep 2015 - R. Yantosca - DD\_A\_Radius is now renamed to Radius  
 05 Jan 2016 - E. Lundgren - Use global parameter PI  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 16 Jun 2016 - J. Kaiser   - Now define species ID's with Ind\_() function  
 20 Jun 2016 - R. Yantosca - Only define species ID's in the INIT phase  
 24 Jun 2016 - R. Yantosca - Add error checks for dust uptake species  
 05 Oct 2017 - R. Yantosca - Now accept State\_Diag as an argument

### 8.3.9 cleanup\_dust

Subroutine CLEANUP\_DUST deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_DUST
```

### REVISION HISTORY:

30 Mar 2004 - R. Yantosca - Initial version  
 25 Aug 2010 - R. Yantosca - Added ProTeX headers  
 26 Feb 2013 - R. Yantosca - Now use Input\_Opt instead of local arrays

## 8.4 Fortran: Module Interface seasalt\_mod.F

Module SEASALT\_MOD contains arrays and routines for performing either a coupled chemistry/aerosol run or an offline seasalt aerosol simulation. Original code taken from Mian Chin's GOCART model and modified accordingly. (bec, rjp, bmy, 6/22/00, 11/23/09)

#### INTERFACE:

MODULE SEASALT\_MOD

**USES:**

USE HCO\_ERROR\_MOD  
USE PRECISION\_MOD ! For GEOS-Chem Precision (fp)  
USE PHYSCONSTANTS

IMPLICIT NONE  
PRIVATE

**PUBLIC MEMBER FUNCTIONS:**

PUBLIC :: CHEMSEASALT  
PUBLIC :: CLEANUP\_SEASALT  
PUBLIC :: INIT\_SEASALT

**PRIVATE MEMBER FUNCTIONS:**

PRIVATE :: WET\_SETTLING  
PRIVATE :: CHEM\_MOPO  
PRIVATE :: CHEM\_MOPI

**PUBLIC DATA MEMBERS:**

PUBLIC :: SALT\_V  
PUBLIC :: DMID

**REMARKS:**

Seasalt aerosol species: (1) Accumulation mode (usually 0.1 - 0.5 um)  
(2) Coarse mode (usually 0.5 - 10.0 um)

NOTE: You can change the bin sizes for accumulation mode and coarse mode seasalt in the "input.geos" file in v7-yy-zz and higher.

References:

- =====
- (1) Chin, M., P. Ginoux, S. Kinne, B. Holben, B. Duncan, R. Martin, J. Logan, A. Higurashi, and T. Nakajima, "Tropospheric aerosol optical thickness from the GOCART model and comparisons with satellite and sunphotometers measurements", J. Atmos Sci., 2001.
  - (2) Gong, S., L. Barrie, and J.-P. Blanchet, "Modeling sea-salt aerosols in the atmosphere. 1. Model development", J. Geophys. Res., v. 102, 3805-3818, 1997.

**REVISION HISTORY:**

- (1) Now references "logical\_mod.f" and "tracer\_mod.f". Comment out SS\_SIZE, this has been replaced by SALA\_REDGE\_um and SALC\_REDGE\_um from "tracer\_mod.f". Increased NR\_MAX to 200. (bmy, 7/20/04)
- (2) Added error check in EMISSEASALT (bmy, 1/20/05)
- (3) Now references "pbl\_mix\_mod.f" (bmy, 2/22/05)

- (4 ) Added routine GET\_ALK to account for alkalinity. (bec, bmy, 4/13/05)
- (5 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)
- (6 ) Now only call dry deposition routine if LDRYD=T (bec, bmy, 5/23/06)
- (7 ) Remove unused variables from GET\_ALK. Also fixed variable declaration bug in WET\_SETTLING. (bec, bmy, 9/5/06)
- (8 ) Extra error check for low RH in WET\_SETTLING (phs, 6/11/08)
- (9 ) Bug fix to remove a double-substitution in GET\_ALK (bec, bmy, 7/18/08)
- (10) Save surface emissions separately (emis\_save) for non-local scheme. (ccc, 5/14/09)
- (11) Bug fixes in GET\_ALK and SRCSALT (bec, lyj, bmy, 11/23/09)
- (12) Add size-resolved emission subroutine SRCSALT30 and reference to tomas\_mod.f. (win, 7/17/09)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 16 Feb 2012 - R. Yantosca - Moved SRCSALT30 to end of module
- 01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90
- 04 Mar 2013 - R. Yantosca - Now call INIT\_SULFATE from the init stage which facilitates connection to GEOS-5 GCM
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 23 Jun 2014 - R. Yantosca - Remove code now made obsolete by HEMCO
- 03 Nov 2014 - C. Keller - Moved GET\_ALK to sulfate\_mod.F.
- 20 Nov 2014 - M. Yannetti - Added PRECISION\_MOD
- 12 Jun 2015 - R. Yantosca - Removed DRY\_DEPOSITION routine, because we now apply drydep in mixing\_mod.F90
- 16 Jun 2016 - C. Miller - Now define species ID's with Ind\_function
- 17 Jun 2016 - R. Yantosca - Add species ID's as module variables
- 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

#### 8.4.1 chemseasalt

Subroutine CHEMSEASALT is the interface between the GEOS-CHEM main program and the seasalt chemistry routines that mostly calculates seasalt dry deposition (rjp, bmy, 1/24/02, 5/23/06)

#### INTERFACE:

```

SUBROUTINE CHEMSEASALT(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

Dry deposition is now handled in mixing\_mod.F90. We have removed the calls to the DRY\_DEPOSITION routine here. (bmy, 6/12/15)

**REVISION HISTORY:**

```

(1) Now reference STT from "tracer_mod.f". Now references LPRT from
 "logical_mod.f" (bmy, 7/20/04)
(2) Now only call DRY_DEPOSITION if LDRYD=T (bec, bmy, 5/23/06)
22 Dec 2011 - M. Payer - Added ProTeX headers
30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
13 Nov 2012 - R. Yantosca - Now add Input_Opt, RC arguments for GIGC
15 Nov 2012 - M. Payer - Now pass met fields via State_Met object
25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing_mod.F90,
 so we can greatly collapse this code
17 Jun 2016 - R. Yantosca - Add error checks before calling routines
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
20 Sep 2016 - R. Yantosca - Bug fix: rewrite invalid (LMPOA > 0) test
02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

**8.4.2 wet\_settling**

Subroutine WET\_SETTLING performs wet settling of sea salt. (bec, rjp, bmy, 4/20/04, 6/11/08)

**INTERFACE:**

```

SUBROUTINE WET_SETTLING(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, TC,
& N, RC
)

```

**USES:**

```

 USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
 USE DIAG_MOD, ONLY : AD44
#endif
 USE ErrCode_Mod
 USE ERROR_MOD, ONLY : DEBUG_MSG
 USE ERROR_MOD, ONLY : ERROR_STOP
 USE GC_GRID_MOD, ONLY : GET_AREA_CM2
 USE Input_Opt_Mod, ONLY : OptInput
 USE PhysConstants
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Diag_Mod, ONLY : DgnState
 USE State_Met_Mod, ONLY : MetState
 USE TIME_MOD, ONLY : GET_TS_CHEM

```

**INPUT PARAMETERS:**

```

 INTEGER, INTENT(IN) :: N ! 1=accum mode;
 ! 2=coarse mode
 LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
 TYPE(MetState), INTENT(IN) :: State_Met ! MeteorologyState
 TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object

```

**INPUT/OUTPUT PARAMETERS:**

```

 TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
 REAL(fp), INTENT(INOUT) :: TC(IIPAR,JJP,LLPAR) ! Sea salt [kg]

```

**OUTPUT PARAMETERS:**

```

 INTEGER, INTENT(OUT) :: RC ! Success/failure

```

**REMARKS:**

TODO: Declare State\_Chm as INTENT(INOUT) and make TC a local pointer to State\_Chm%SPECIES. Pass in the ID field to index State\_Chm.

**REVISION HISTORY:**

- (1 ) Now references SALA\_REEDGE\_um and SALC\_REEDGE\_um from "tracer\_mod.f" (bmy, 7/20/04)
- (2 ) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)
- (3 ) Bug fix: DTCHEM has to be REAL(fp), not integer. (bmy, 9/7/06)
- (4 ) Now limit relative humidity to [tiny(real(fp)),0.99] range for DLOG argument (phs, 5/1/08)
- (5 ) Update sea salt density calculation using Tang et al. (1997) (bec, jaegle 5/11/11)
- (6 ) Update hygroscopic growth for sea salt using Lewis and Schwartz (2006)



and and density calculation based on Tang et al. (1997) (bec, jaegle 5/11/11)

(7 ) Itegrate settling velocity over entire size distribution (jaegle 5/11/11)

22 Dec 2011 - M. Payer - Added ProTeX headers

01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90

14 Nov 2012 - R. Yantosca - Add am\_I\_Root, Input\_Opt, RC as arguments

15 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object

19 Mar 2013 - R. Yantosca - Now copy Input\_Opt%XNUMOL(1:N\_TRACERS)

12 Jun 2013 - R. Yantosca - Bug fix: SALT\_MASS needs to be !OMP PRIVATE

12 Jun 2013 - R. Yantosca - Reformatted some comments for clarity

26 Feb 2015 - E. Lundgren - Replace GET\_PCENTER with State\_Met%PMID.  
Remove dependency on pressure\_mod.

25 Jan 2016 - E. Lundgren - Update netcdf drydep flux diagnostic

29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

31 May 2016 - E. Lundgren - Remove usage of Input\_Opt%XNUMOL

22 Jun 2016 - M. Yannetti - Replaced TCVV with spec db and phys constant

20 Oct 2016 - R. Yantosca - Only print out debug output on the root CPU

02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

### 8.4.3 chem\_mopo

Subroutine CHEM\_MOPO modifies hydrophobic marine organic aerosol concentrations based on the conversion to hydrophilic marine organic aerosols.

#### INTERFACE:

```
SUBROUTINE CHEM_MOPO(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:****REVISION HISTORY:**

```
10 Jul 2015 - E. Lundgren - Initial version (based on routine Chem_OCPO)
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
```

---

**8.4.4 chem\_mopi**

Subroutine CHEM\_MOPI modifies hydrophilic marine organic aerosol concentrations based on the conversion from hydrophobic marine organic aerosols.

**INTERFACE:**

```
SUBROUTINE CHEM_MOPI(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success?
```

**REMARKS:****REVISION HISTORY:**

```
10 Jul 2015 - E. Lundgren - Initial version (based on routine Chem_OCPI)
02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
```

---

### 8.4.5 init\_seasalt

Subroutine INIT\_SEASALT initializes and zeroes all module arrays (bmy, 4/26/04, 4/13/05)

#### INTERFACE:

```

SUBROUTINE INIT_SEASALT(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### REVISION HISTORY:

- (1 ) Now exit if we have allocated arrays before. Now also allocate  
ALK\_EMIS & N\_DENS. Now reference CMN\_SIZE. (bec, bmy, 4/13/05)
  - (2 ) Added SALT\_V and DMID (jaegle 5/11/11)
  - 22 Dec 2011 - M. Payer - Added ProTeX headers
  - 04 Mar 2013 - R. Yantosca - Now pass am\_I\_Root, Input\_Opt, RC as arguments
  - 04 Mar 2013 - R. Yantosca - Now test for DEPSALA, DEPSALC here
  - 12 Jun 2015 - R. Yantosca - Remove obsolete, commented-out code
  - 23 Sep 2015 - R. Yantosca - Now accept State\_Chm as an argument so that  
we can use the species database
  - 30 Sep 2015 - R. Yantosca - DD\_A\_Density is renamed to Density
  - 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
  - 17 Jun 2016 - R. Yantosca - Now define species ID's on the first call
  - 13 Oct 2016 - R. Yantosca - Now allocate OCCONV for marine-POA simulations
  - 02 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument
-

### 8.4.6 cleanup\_seasalt

Subroutine CLEANUP\_SEASALT deallocates all module arrays (bmy, 4/26/04, 4/13/05)

#### INTERFACE:

```
SUBROUTINE CLEANUP_SEASALT
```

#### REVISION HISTORY:

- (1 ) Now deallocates ALK\_EMIS, N\_DENS, SRC\_N (bec, bmy, 4/13/05)
  - (2 ) Deallocated SALT\_V and DMID (jaegle 5/11/11)
  - 22 Dec 2011 - M. Payer - Added ProTeX headers
  - 13 Oct 2016 - R. Yantosca - Bug fix: now deallocate OCCONV array
- 

### 8.4.7 srcsalt30

Subroutine SRCSALT30 emits sea-salt into the 30-bin sea-salt mass and aerosol number arrays. Sea-salt emission parameterization of Clarke et al. [2006] (win, 7/17/09)

#### INTERFACE:

```
SUBROUTINE SRCSALT30(TC1, TC2, State_Met)
```

#### USES:

```

USE CMN_SIZE_MOD ! Size parameters
USE CMN_DIAG_MOD ! ND59
USE DAO_MOD, ONLY : IS_WATER
USE DIAG_MOD, ONLY : AD59_NUMB, AD59_SALT
USE ERROR_MOD, ONLY : ERROR_STOP
USE ERROR_MOD, ONLY : IT_IS_NAN
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_TOP_L
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS
USE TOMAS_MOD, ONLY : IBINS, Xk

```

#### INPUT PARAMETERS:

```
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### INPUT/OUTPUT PARAMETERS:

```

! TC1 : Aerosol number tracer array [no.]
! TC2 (REAL(fp)) : Sea salt tracer array [kg]
REAL(fp), INTENT(INOUT) :: TC1(IIPAR,JJP,LLPAR,IBINS)
REAL(fp), INTENT(INOUT) :: TC2(IIPAR,JJP,LLPAR,IBINS)

```

#### AUTHOR:

Contact: Win Trivitayanurak (win@cmu.edu)

Arguments as Input/Output:

=====

## REMARKS:

References:

=====

- (1 ) Clarke, A.D., Owens, S., Zhou, J. " An ultrafine sea-salt flux from breaking waves: Implications for CCN in the remote marine atmosphere" JGR, 2006

## REVISION HISTORY:

- (1 ) Originally from emisnaN3clarke.f in GISS GCM-II' (win, 7/18/07)  
 (2 ) Now partition emission throughout the PBL (win, 7/18/07)  
 (3 ) Add COEF to adjust emission in a 1x1 nested-grid (win, 4/27/08)  
 16 Feb 2012 - R. Yantosca - Added ProTeX headers  
 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
 01 Mar 2012 - R. Yantosca - A\_M2 is now a scalar  
 10 Mar 2014 - J. Pierce - Avoid sea salt emissions over ice  
 25 Jul 2014 - R. Yantosca - Remove reference to function SFCWINDSQR

## 8.5 Fortran: Module Interface sulfate\_mod.F

Module SULFATE\_MOD contains arrays and routines for performing either a coupled chemistry/aerosol run or an offline sulfate aerosol simulation. Original code taken from Mian Chin's GOCART model and modified accordingly. (rjp, bdf, bmy, 6/22/00, 8/26/10)

### INTERFACE:

```
MODULE SULFATE_MOD
```

### USES:

```
USE HCO_ERROR_MOD ! For HEMCO error handling
USE PhysConstants ! Physical constants
USE PRECISION_MOD ! For GEOS-Chem Precision (fp, f4, f8)
```

```
IMPLICIT NONE
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: CHEMSULFATE
PUBLIC :: CLEANUP_SULFATE
PUBLIC :: INIT_SULFATE
#if defined(TOMAS)
```

```
PUBLIC :: EMISSSULFATETOMAS ! JKodros (6/2/15 - this is to connect TOMAS to HEMCO)
#endif
```

## REMARKS:

### References:

- ```
=====
```
- (1) Andreae, M.O. & P. Merlet, "Emission of trace gases and aerosols from biomass burning", Global Biogeochem. Cycles, 15, 955-966, 2001.
 - (2) Nightingale et al [2000a], J. Geophys. Res, 14, 373-387
 - (3) Nightingale et al [2000b], Geophys. Res. Lett, 27, 2117-2120
 - (4) Wanninkhof, R., "Relation between wind speed and gas exchange over the ocean", J. Geophys. Res, 97, 7373-7382, 1992.

REVISION HISTORY:

- (1) All module variables are declared PRIVATE (i.e., they can only be seen from within this module (bmy, 6/2/00)
- (2) The routines in "sulfate_mod.f" assume that we are doing chemistry over the global region (e.g. IIPAR=IIPAR, JJP=JJP). (bmy, 6/8/00)
- (3) Removed obsolete code from DRYDEP_SULFATE (bmy, 12/21/00)
- (4) Removed obsolete commented-out code from module routines (bmy, 4/23/01)
- (5) Now read data files from DATA_DIR/sulfate_sim_200106/ (bmy, 6/19/01)
- (6) Updated comments (bmy, 9/4/01)
- (7) XTRA2(IREF,JREF,5) is now XTRA2(I,J). Now reference COSSZA from "dao_mod.f". (bmy, 9/27/01)
- (8) Removed obsolete commented out code from 9/01 (bmy, 10/24/01)
- (9) Minor fixes to facilitate compilation on ALPHA (bmy, 11/15/01)
- (11) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
- (12) Replaced all instances of IM with IIPAR and JM with JJP, in order to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (13) Now reference "file_mod.f" (bmy, 6/27/02)
- (14) Now references GET_PEDGE from "pressure_mod.f", which computes P at the bottom edge of grid box (I,J,L). Also deleted obsolete, commented-out code. (dsa, bdf, bmy, 8/21/02)
- (15) Added updated code from Rokjin Park and Brendan Field, in order to perform coupled chemistry-aerosol simulations. Also added parallel DO-loops in several subroutines. Updated comments, cosmetic changes. Now reference "error_mod.f" and "wetscav_mod.f". Now only do chemistry below the tropopause. (rjp, bdf, bmy, 12/6/02)
- (16) Added ENH3_na array to hold natural source NH3 emissions. Also now facilitate passing DMS, SO2, SO4, NH3 to SMVGEAR for fullchem simulations. Added subroutine READ_NATURAL_NH3. (rjp, bmy, 3/23/03)
- (17) Now references "grid_mod.f" and "time_mod.f". Also made other minor cosmetic changes. (bmy, 3/27/03)
- (18) Updated chemistry routines to apply drydep losses throughout the

- entire PBL. (rjp, bmy, 8/1/03)
- (19) Now accounts for GEOS-4 PBL being in meters (bmy, 1/15/04)
 - (20) Fix ND44 diag so that we get same results for sp or mp (bmy, 3/24/04)
 - (21) Added COSZM array. Now use diurnal varying JH202 in CHEM_H2O2.
(rjp, bmy, 3/39/04)
 - (22) Added more parallel DO-loops (bmy, 4/14/04)
 - (23) Now add SO2 from ships (bec, bmy, 5/20/04)
 - (24) Now references "directory_mod.f", "logical_mod.f" and "tracer_mod.f".
Now removed IJSURF. (bmy, 7/20/04)
 - (25) Can overwrite USA with EPA/NEI99 emissions (rjp, rch, bmy, 11/16/04)
 - (26) Modified for AS, AHS, LET, SO4aq, NH4aq (cas, bmy, 1/11/05)
 - (27) Now also references "pbl_mix_mod.f". NOTE: Comment out phase
transition code for now since it is still under development and
will take a while to be rewritten. (bmy, 3/15/05)
 - (28) Modified for SO4s, NITs chemistry (bec, 4/13/05)
 - (29) Now reads updated files for SST and offline chemistry. Now read data
for both GCAP and GEOS grids. Now references "tropopause_mod.f".
(bmy, 8/22/05)
 - (30) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 - (31) Now references XNUMOL & XNUMOLAIR from "tracer_mod.f" (bmy, 10/25/05)
 - (32) Now read int'annual SST data on GEOS 1x1 grid (bmy, 11/17/05)
 - (33) Bug fix for offline aerosol sim in SEASALT_CHEM (bec, bmy, 3/29/06)
 - (34) Bug fix in INIT_DRYDEP (bmy, 5/23/06)
 - (35) Now references "bravo_mod.f" (rjp, kfb, bmy, 6/26/06)
 - (36) Now references "streets_anthro_mod.f" (yxw, bmy, 8/17/06)
 - (37) Now references "biomass_mod.f" (bmy, 9/27/06)
 - (38) Now prevent seg fault error in READ_BIOFUEL_SO2 (bmy, 11/3/06)
 - (39) Bug fix in SEASALT_CHEM (havalala, bec, bmy, 12/8/06)
 - (40) Extra error check for low RH in GRAV_SETTLING (phs, 6/11/08)
 - (41) Now references "cac_anthro_mod.f". And apply SO2 yearly scale factor
to SO2 from GEIA (amv, phs, 3/11/08)
 - (41) Bug fixes in reading EDGAR data w/ the right tracer number,
when we are doing offline or nonstd simulations (dkh, 10/31/08)
 - (42) Bug fix for AD13_SO2_sh in SRCSO2 (phs, 2/27/09)
 - (43) Bug fix: need to add CAC_AN to PRIVATE statements (bmy, 5/27/09)
 - (44) Constrain surface emissions to the first level and save them into
emis_save (lin, 5/29/09)
 - (45) Last year of SST data is now 2008 (see READ_SST) (bmy, 7/13/09)
 - (46) Updated rxns in CHEM_DMS and CHEM_SO2 to JPL 2006 (jaf, bmy, 10/15/09)
 - (47) Added new volcanic emissions of SO2 (jaf, bmy, 10/15/09)
 - (48) Now accounts for NEI 2005 emissions, and multilevels SOxan emissions
(amv, phs, 10/15/2009)
 - (49) Fixes in SRCSO2 for SunStudio compiler (bmy, 12/3/09)
 - (50) Add new subroutine SRCSF30 for emission to 30bin sulfate (win, 1/25/10)
 - (51) Add new array PSO4_SO2AQ for SO4 produced via aqueous chemistry of SO2
excluding that from heterogeneous reaction on sea-salt. (win, 1/25/10)
 - (52) Standardized patch in READ_ANTHRO_NH3 (dkh, bmy, 3/5/10)
 - (53) Use LWC from GEOS-5 met fields (jaf, bmy, 6/30/10)

- (54) Add module parameters MNYEAR_VOLC and MXYEAR_VOLC to define the 1st and last year with data for volcanic emissions. (ccc, 9/30/10)
- (55) Use updated volcanic emissions from 1979 to 2009
- 26 Aug 2010 - R. Yantosca - Add modifications for MERRA
- 12 Nov 2010 - R. Yantosca - Avoid div-by-zero when computing L2S, L3S
- 07 Sep 2011 - P. Kasibathla - Modified to include GFED3
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.2 met
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 13 Mar 2012 - M. Cooper - Changed regrid algorithm to map_a2a
- 28 Nov 2012 - R. Yantosca - Use SUNCOS fields from the State_Met object
- 04 Mar 2013 - R. Yantosca - Now call INIT_SULFATE from the init stage which facilitates connection to GEOS-5 GCM
- 05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNL PBL instead of LNL PBL from logical_mod.F
- 13 Mar 2013 - R. Yantosca - Bug fix: make sure we pass values to the SOIL_DRYDEP routine even when ND44 is off
- 30 May 2013 - S. Farina - Merged TOMAS code into sulfate_mod.F
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 12 Sep 2013 - M. Sulprizio - Add modifications for acid uptake on dust aerosol (T.D. Fairlie)
- 18 Sep 2014 - M. Sulprizio - Get oxidant fields for offline aerosol simulation from HEMCO
- 03 Nov 2014 - C. Keller - Incorporated GET_ALK from seasalt_mod.F
- 20 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 04 Mar 2015 - R. Yantosca - Remove obsolete, commented-out code
- 04 Mar 2015 - R. Yantosca - Use REAL(f4) for pointer args to HCO_GetPtr
- 22 May 2015 - R. Yantosca - Remove variables made obsolete by HEMCO
- 12 Jun 2015 - R. Yantosca - Now remove orphaned ND44 variables
- 12 Jun 2015 - R. Yantosca - Remove CHEM_NH3 and CHEM_NH4 routines because drydep is now done in mixing_mod.F90
- 23 Sep 2015 - R. Yantosca - Remove DRY* flags for most species except for those used in GRAV_SETTLING
- 05 Jan 2016 - E. Lundgren - Use global physical parameters
- 04 Aug 2016 - M. Yannetti - Replace TCVV with spc db MW and phys constant
- 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
- 24 Aug 2017 - M. Sulprizio - Remove support for GCAP, GEOS-4, GEOS-5 and MERRA; Also remove functions GET_VCLDF and GET_LWC, they're no longer needed

8.5.1 chemsulfate

Subroutine CHEMSULFATE is the interface between the GEOS-CHEM main program and the sulfate chemistry routines. The user has the option of running a coupled chemistry-aerosols simulation or an offline aerosol simulation. (rjp, bdf, bmy, 5/31/00, 3/16/06)

INTERFACE:


```

SUBROUTINE CHEMSULFATE( am_I_Root, Input_Opt, State_Met,
&                      State_Chm, State_Diag, FullRun, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH
USE TIME_MOD, ONLY : GET_TS_CHEM
USE TIME_MOD, ONLY : GET_ELAPSED_SEC
USE TIME_MOD, ONLY : ITS_A_NEW_MONTH
USE UCX_MOD, ONLY : SETTLE_STRAT_AER
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
LOGICAL, INTENT(IN) :: FullRun ! Modify species conc?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

REVISION HISTORY:

- (1) Now reference all arguments except FIRSTCHEM and RH from either F90 modules or from common block header files. Updated comments, cosmetic changes. Added NH3, NH4, NITRATE chemistry routines. Also call MAKE_RH and CONVERT_UNITS from "dao_mod.f". Now references IDTDMS, IDTSO2 etc. from "tracerid_mod.f". Now make FIRSTCHEM a local SAVED variable. Now reference DEPSAV from "drydep_mod.f". Also get rid of extraneous dimensions of DEPSAV. Added NTIME, NHMSb arrays for OHNO3TIME. (rjp, bdf, bmy, 12/16/02)
- (2) CHEM_DMS is now only called for offline sulfate simulations. (rjp, bmy, 3/23/03)
- (3) Now remove NTIME, NHMSb from the arg list and call to OHNO3TIME. Now references functions GET_MONTH, GET_TS_CHEM, and GET_ELAPSED_SEC

- from the new "time_mod.f". (bmy, 3/27/03)
- (4) Now reference STT, TCVV, N_TRACERS, ITS_AN_AEROSOL_SIM from "tracer_mod.f". Now reference ITS_A_NEW_MONTH from "time_mod.f". Now references LPRT from "logical_mod.f". (bmy, 7/20/04)
- (5) Updated for AS, AHS, LET, SO4aq, NH4aq. Now references LCRYST from logical_mod.f. Now locate species in the DEPSAV array w/in INIT_SULFATE. (bmy, 12/21/04)
- (6) Now handle gravitational settling of SO4s, NITs (bec, bmy, 4/13/05)
- (7) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (8) Remove reference to MAKE_RH, it's not needed here (bmy, 3/16/06)
- (9) Reference to LTOMAS and add call CHEM_SO4_AQ using aqueous oxidation which is one of the TOMAS microphysics subroutine (win, 1/25/10)
- 05 Oct 2011 - R. Yantosca - SUNCOS is no longer needed here
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when running with the traditional driver main.F
- 14 Nov 2012 - R. Yantosca - Add Input_Opt, RC as arguments
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 04 Mar 2013 - R. Yantosca - Remove call to INIT_SULFATE
- 19 Mar 2013 - R. Yantosca - Now copy Input_Opt%TCVV(1:N_TRACERS)
- 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
- 23 Apr 2013 - R. Yantosca - Remove LTOMAS logical, since we now invoke TOMAS with either TOMAS=yes or TOMAS40=yes
- 31 May 2013 - R. Yantosca - Now pass am_I_root, Input_Opt, State_Chm and RC to TOMAS routine CHEM_SO4_AQ
- 12 Sep 2013 - M. Sulprizio- Include gravitational settling of dust_sulfate and dust_nitrate. Changes made to CHEM_SO2 and CHEM_SO4 (tdf, 04/07/08)
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET_GLOBAL_OH routine
- 18 Sep 2014 - M. Sulprizio- Get oxidant fields for offline aerosol simulation from HEMCO
- 30 Sep 2015 - E. Lundgren - Now use UNITCONV_MOD conversion routines
- 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
- 16 Jun 2016 - M. Long - Remove references to TRACERID_MOD and replace with STATE_CHM_MOD::Ind_()
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
- 11 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
- 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine
- 02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
- 03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input_Opt%LUCX

8.5.2 emissssulfatetomas

Subroutine EMISSSULFATETOMAS connects HEMCO bulk emissions to the TOMAS tracers. Only use this for TOMAS sims. This should be quite similar to the TOMAS rele-

vant parts of 'emissulfate' in v9 (Jkodros 6/2/15)

INTERFACE:

```

SUBROUTINE EMISSSULFATETOMAS( am_I_Root, Input_Opt,
&                               State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TOMAS_MOD,         ONLY : IBINS,      ICOMP,   IDIAG
USE TOMAS_MOD,         ONLY : NH4BULKTOBIN
USE TOMAS_MOD,         ONLY : SRTNH4
USE UnitConv_Mod,     ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN)   :: Input_Opt    ! Input Options object
TYPE(MetState), INTENT(IN)   :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry State objectt

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)    :: RC            ! Success or failure?

```

Local variables

```

! Fields for TOMAS simulation
REAL*8      :: BINMASS(IIPAR,JJPAR,LLPAR,IBINS*ICOMP)
INTEGER     :: TID, I, J, L
INTEGER     :: ii=53, jj=29, ll=1
REAL(fp)    :: NH4_CONC
CHARACTER(LEN=63):: OrigUnit

```

! SAVED scalars

```

LOGICAL, SAVE  :: FIRST = .TRUE.

```

! Pointers

```

REAL*8, POINTER :: Spc(:, :, :, :, :)

```

! Arrays

```

REAL(fp)      :: tempnh4(ibins)

```

```

REAL(fp)          :: MK_TEMP2(IBINS)

!=====
! EMISSSULFATETOMAS begins here!
!=====

! First-time setup
IF ( FIRST ) THEN

    ! Reset first-time flag
    FIRST = .FALSE.
ENDIF

! Convert species to [kg] for TOMAS. This will be removed once
! TOMAS uses mixing ratio instead of mass as tracer units (ewl, 9/11/15)
CALL Convert_Spc_Units( am_I_Root, Input_Opt, State_Met,
&                      State_Chm, 'kg', RC, OrigUnit=OrigUnit )
IF ( RC /= GC_SUCCESS ) THEN
    CALL GC_Error('Unit conversion error', RC,
&               'Start of EMISSSULFATETOMAS in sulfate_mod.F')
    RETURN
ENDIF

! Point to chemical species array [kg]
Spc => State_Chm%Species

IF (id_SF1 > 0 .and. id_NK1 > 0 ) THEN

    !!! Get NH4 and aerosol water into the same array
    BINMASS(:, :, :, 1:IBINS*(ICOMP-IDIAG)) =
&        Spc(:, :, :, id_SF1:id_SF1+IBINS*(ICOMP-IDIAG) - 1)

    IF ( SRTNH4 > 0 ) THEN
        TID = IBINS*(ICOMP-IDIAG) + 1
    ENDIF
ENDIF

$OMP PARALLEL DO
$OMP+DEFAULT( SHARED )
$OMP+PRIVATE( I, J, L, TEMPNH4, MK_TEMP2, NH4_CONC )
$OMP+SCHEDULE( DYNAMIC )
    DO L=1,LLPAR
    DO J=1,JJPARG
    DO I=1,IIPARG

        ! Change pointer to a variable to avoid array temporary
        ! (bmy, 7/7/17)
        MK_TEMP2 = Spc(I,J,L,id_SF1:id_SF1-1+IBINS)
        NH4_CONC = Spc(I,J,L,id_NH4)
    END DO
    END DO
    END DO

```

```

        CALL NH4BULKTOBIN( MK_TEMP2, NH4_CONC, TEMPNH4 )

        BINMASS(I,J,L,TID:TID+IBINS-1) = TEMPNH4(1:IBINS)
        ENDDO
        ENDDO
        ENDDO
$OMP END PARALLEL DO

        ENDIF

        TID = IBINS*(ICOMP-1) +1
        BINMASS(:,:,:,TID:TID+IBINS-1) =
&          Spc(:,:,:,id_AW1:id_AW1+IBINS-1)

        !IF ( id_SF1 > 0 ) THEN
        CALL SRCSF30( Spc(:,:,:,id_NK1:id_NK1+IBINS-1),
&          BINMASS(:,:,:,), am_I_Root, Input_Opt, State_Met, RC )

        ! Return the aerosol mass after emission subroutine to Spc
        ! excluding the NH4 aerosol and aerosol water (win, 9/27/08)
        Spc(:,:,:,id_SF1:id_SF1+IBINS*(ICOMP-IDIAG)-1) =
&          BINMASS(:,:,:,1:IBINS*(ICOMP-IDIAG))
        ENDIF

        ! Free pointer
        NULLIFY( Spc )

        ! Convert species back to original units (ewl, 9/11/15)
        CALL Convert_Spc_Units( am_I_Root, Input_Opt, State_Met,
&          State_Chm, OrigUnit, RC )
        IF ( RC /= GC_SUCCESS ) THEN
        CALL GC_Error('Unit conversion error', RC,
&          'End of EMISSSULFATETOMAS in sulfate_mod.F')
        RETURN
        ENDIF

        END SUBROUTINE EMISSSULFATETOMAS
EOC
#endif
-----
                        Jack Kodros re-writing this
-----
#if defined( TOMAS )
        SUBROUTINE SRCSF30( TC1, TC2,
&          am_I_Root, Input_Opt, State_Met, RC )
USES:
        USE CMN_SIZE_MOD          ! Size parameters

```

```

#if defined( BPCH_DIAG )
    USE CMN_DIAG_MOD          ! ND13 (for now)
    USE DIAG_MOD,            ONLY : AD59_SULF,      AD59_NUMB
#endif
    USE ERROR_MOD,          ONLY : ERROR_STOP,    IT_IS_NAN
    USE Input_Opt_Mod,     ONLY : OptInput
    USE PBL_MIX_MOD,       ONLY : GET_FRAC_OF_PBL, GET_PBL_TOP_L
    USE State_Met_Mod,     ONLY : MetState
    USE TOMAS_MOD,        ONLY : IBINS, AVGMASS, ICOMP
    USE TOMAS_MOD,        ONLY : Xk
    USE TOMAS_MOD,        ONLY : SUBGRIDCOAG, MNFIX
    USE TOMAS_MOD,        ONLY : SRTSO4, SRTNH4,  DEBUGPRINT
    USE HCO_INTERFACE_MOD, ONLY : HcoState, GetHcoDiagn

```

INPUT PARAMETERS:

```

    LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
    TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
    TYPE(MetState), INTENT(IN)  :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

    REAL(fp), INTENT(INOUT) :: TC1(IIPAR,JJP,LLPAR,IBINS)
    REAL(fp), INTENT(INOUT) :: TC2(IIPAR,JJP,LLPAR,IBINS*ICOMP)

```

OUTPUT PARAMETERS:

```

    INTEGER,          INTENT(OUT)  :: RC          ! Success or failure?

```

Local variables

```

    INTEGER          :: I, J, K, L, DOW_LT, NTOP, C, Bi
    REAL*8          :: SO4(LLPAR), DTSRCE , EFRAC(LLPAR)
    REAL*8          :: TS04,          FEMIS
    REAL*8          :: AREA_CM2
    REAL*8          :: SO4an(IIPAR,JJP,LLPAR)
    REAL*8          :: SO4bf(IIPAR,JJP)
    REAL*8          :: SO4anbf(IIPAR,JJP,2)
    REAL*8 BFRAC(IBINS) ! Mass fraction emitted to each bin

```

```

#if defined( TOMAS12 ) || defined( TOMAS15 )
    DATA BFRAC/
# if defined( TOMAS15 )
    &          0.0d0      , 0.0d0      , 0.0d0,
# endif
    &          4.3760E-02, 6.2140E-02, 3.6990E-02, 1.8270E-02,
    &          4.2720E-02, 1.1251E-01, 1.9552E-01, 2.2060E-01,
    &          1.6158E-01, 7.6810E-02, 2.8884E-02, 2.0027E-04/
#else

```

```

      DATA BFRAC/
# if defined( TOMAS40)
      &          0.000d00 , 0.000d00 , 0.000d00 , 0.000d00 , 0.000d00 ,
      &          0.000d00 , 0.000d00 , 0.000d00 , 0.000d00 , 0.000d00 ,
# endif
      &          1.728d-02, 2.648d-02, 3.190d-02, 3.024d-02, 2.277d-02,
      &          1.422d-02, 9.029d-03, 9.241d-03, 1.531d-02, 2.741d-02,
      &          4.529d-02, 6.722d-02, 8.932d-02, 1.062d-01, 1.130d-01,
      &          1.076d-01, 9.168d-02, 6.990d-02, 4.769d-02, 2.912d-02,
      &          1.591d-02, 7.776d-03, 3.401d-03, 1.331d-03, 4.664d-04,
      &          1.462d-04, 4.100d-05, 1.029d-05, 2.311d-06, 4.645d-07/
#endif

      REAL(fp)          :: NDISTINIT(IBINS)
      REAL(fp)          :: NDISTFINAL(IBINS)
      REAL(fp)          :: MADDFINAL(IBINS)
      REAL(fp)          :: NDIST(IBINS), MDIST(IBINS,ICOMP)
      REAL(fp)          :: NDIST2(IBINS), MDIST2(IBINS,ICOMP)
      REAL*4            :: TSCALE, BOXVOL, TEMP, PRES

      REAL(fp)          :: NO(LLPAR,IBINS), MO(LLPAR,IBINS)
      REAL(fp)          :: Ndiag(IBINS), Mdiag(IBINS)
      REAL(fp)          :: MADDTOTAL !optimization variable for diag
      REAL(fp)          :: Avginit(IBINS), Avgfinal(IBINS)
      REAL(fp)          :: Avginner(IBINS)

      REAL(fp)          :: AREA(IIPAR, JJPARG)
      REAL(fp)          :: AREA3D(IIPARG, JJPARG,2)

      ! Pointers
      REAL(f4),          POINTER :: Ptr2D(:, : )
      REAL(f4),          POINTER :: Ptr3D(:, :, :)

      INTEGER            :: N_TRACERS

      LOGICAL            :: ERRORSWITCH, SGCOAG = .TRUE.
      INTEGER            :: FLAG, ERR
      logical            :: pdebug !(temporary) win, 10/24/07
      !integer           :: ii, jj, ll
      !data ii, jj, ll / 61, 1, 7 /
      INTEGER            :: ii=53, jj=29, ll=1

      ! Ratio of molecular weights: S/SO4
      REAL*8, PARAMETER :: S_SO4 = 32d0 / 96d0

      ! debugging
      real*8             dummy

```

```

! For fields from Input_Opt
LOGICAL :: LPRT, LNLPBL
LOGICAL          :: jkdbg=.true.

! Strings
CHARACTER(LEN= 63)      :: DgnName
CHARACTER(LEN=255)     :: MSG
CHARACTER(LEN=255)     :: LOC='srcsf30 (sulfate_mod.F)'

!=====
! SRCSF30 begins here!
!=====

! Free pointers
Ptr2D  => NULL()
Ptr3D  => NULL()

! Copy values from Input_Opt
LPRT   = Input_Opt%LPRT
LNLPBL = Input_Opt%LNLPBL

! Import emissions from HEMCO (through HEMCO state)
IF ( .NOT. ASSOCIATED(HcoState) ) THEN
  CALL ERROR_STOP ( 'HcoState not defined!', LOC )
ENDIF

! Emission timestep [seconds]
DTSRCE = HcoState%TS_EMIS

! Grid box aarea
AREA = HcoState%Grid%AREA_M2%Val(:, :)
AREA3D(:, :, 1) = AREA(:, :)
AREA3D(:, :, 2) = AREA(:, :)

! Define subgrid coagulation timescale (win, 10/28/08)
#if defined( GRID4x5 )
  TSCALE = 10.*3600. ! 10 hours
#elif defined( GRID2x25 )
  TSCALE = 5.*3600.
#elif defined( GRID05x0625 )
  !%% KLUDGE, just copied the 0.5 x 0.666 value
  !%% Someone needs to add the right value (bmy, 2/16/12)
  TSCALE = 1.*3600.
#elif defined( GRID025x03125 )
  !%% KLUDGE, just copied the 0.5 x 0.666 value
  !%% Someone needs to add the right value (bmy, 2/16/12)
  !%% 0.5 sounds about right. (sfarina, 11/14/16)
  TSCALE = 0.5*3600.

```



```

#endif
      !Prior to 10/28/08 (win)
      !TSCALE = 10.*3600.

      !=====
      ! READ IN HEMCO EMISSIONS
      !=====
      DgnName = 'SO4_ANTH'
      CALL GetHcoDiagn( am_I_Root, DgnName, .FALSE., ERR,
& Ptr3D=Ptr3D )
      IF ( .NOT. ASSOCIATED(Ptr3D) ) THEN
        CALL HCO_WARNING('Not found: '//TRIM(DgnName),ERR,THISLOC=LOC)
      ELSE
        SO4_ANTH = Ptr3D(:, :, :)
      ENDIF
      Ptr3D => NULL()

      DgnName = 'SO4_BIOF'
      CALL GetHcoDiagn( am_I_Root, DgnName, .FALSE., ERR,
& Ptr2D=Ptr2D )
      IF ( .NOT. ASSOCIATED(Ptr2D) ) THEN
        CALL HCO_WARNING('Not found: '//TRIM(DgnName),ERR,THISLOC=LOC)
      ELSE
        SO4_BIOF = Ptr2D(:, :)
      ENDIF
      Ptr2D => NULL()

      ! convert to kg/box/sec
      DO L = 1, LLPAR
        SO4an(:, :, L) = 0.0d0
        SO4an(:, :, L) = SO4_ANTH(:, :, L) * AREA(:, :)
      END DO
      SO4bf = SO4_BIOF(:, :) * AREA(:, :)

      !=====
      ! Compute SO4 emissions
      !=====
%%%%%%OMP+PRIVATE( Avginit, Avginner, Avgfinal, NO, MO)

$OMP PARALLEL DO
$OMP+DEFAULT( SHARED )
$OMP+PRIVATE( I, J, NTOP, SO4, TSO4, L, FEMIS, EFRAC, K )
$OMP+PRIVATE( NDISTINIT, NDIST, MDIST, NDISTFINAL, MADDFINAL )
$OMP+PRIVATE( Ndiag, Mdiag)
$OMP+PRIVATE( MADDTOTAL, NDIST2, MDIST2, C , ERRORSWITCH)
$OMP+PRIVATE( BOXVOL, TEMP, PRES, pdebug )
$OMP+SCHEDULE( DYNAMIC )

```

```

DO J = 1, JJPAR
DO I = 1, IIPAR
  !initialize diagnostics
  Ndiag(:) = 0.0D0
  Mdiag(:) = 0.0D0

  ! Top level of boundary layer at (I,J)
  NTOP = CEILING( GET_PBL_TOP_L( I, J ) )

  ! Zero SO4 array at all levels
  DO L = 1, LLPAR
    SO4(L) = 0.0
  ENDDO

  ! Compute total anthro SO4 (surface + 100m) plus biofuel SO4
  TSO4 = 0.d0
  TSO4 = SUM( SO4an(I,J,:) ) + SO4bf(I,J)
  IF ( TSO4 < 0d0 ) THEN
    WRITE(*,*) ' Negative Sulfate emis from hemco at IJ=', I, J
  ENDIF
  IF ( TSO4 == 0d0 ) CYCLE

debug      if(i==60.and.j==35) print *,'TSO4',TSO4
  !=====
  ! First calculate emission distribution vertically within PBL
  !=====
  ! EFRAC(30) = fraction of total emission splitted for each
  !           level until reaching PBL top.
  EFRAC = 0d0
  !=====
  ! Partition the total anthro SO4 emissions thru the entire
  ! boundary layer (if PBL top is higher than level 2)
  !=====
  ! Add option for non-local PBL (Lin, 03/31/09)
  IF (.NOT. LNL PBL) THEN
    IF ( NTOP > 2 ) THEN

      ! Loop thru boundary layer
      DO L = 1, LLPAR

        ! Fraction of PBL spanned by grid box (I,J,L) [unitless]
        EFRAC(L) = GET_FRAC_OF_PBL( I, J, L )

      ENDDO
    ELSE
      EFRAC(1) = ( SO4an(I,J,1) + SO4bf(I,J) ) / TSO4
    ENDIF
  ENDIF

```

```

      EFRAC(2) = S04an(I,J,2) / TSO4
    ENDIF

    IF ( ABS( SUM( EFRAC(:) ) - 1.d0 ) > 1.D-5 ) THEN
      PRINT*, '### ERROR in SRCSF30!'
      PRINT*, '### I, J : ', I, J
      print*, 'EFRAC',EFRAC(:)
      PRINT*, '### SUM(EFRAC) : ', SUM( EFRAC(:) )
      PRINT*, '### This should exactly 1.00'
      CALL ERROR_STOP( 'Check S04 redistribution',
&                      'SRCSF30 (sulfate_mod.f)' )
    ENDIF

    ELSE
      ! stop the program for now since I don't totally implement
      ! the subgrid coagulation option w/ Lin's new PBL scheme (win, 1/25/10)
      print *,'If the program stops here, that means you are ',
&          'running TOMAS simulation with the new PBL scheme ',
&          'implemented since GEOS-Chem v.8-02-01.',
&          '-----> Try not using the non-local PBL option'
      CALL ERROR_STOP( 'Code does not support new PBL scheme',
&                      'SRCSF30 (sulfate_mod.f)')

    ENDIF ! .not. LNL PBL
    !=====
    ! Add the size-resolved SO4 emission to tracer array
    ! Having the options to do sub-grid coagulation or simply
    ! emit.
    ! Sub-grid coagulation reduces the number being emitted
    ! and modifies the mass size distribution of existing particle
    ! as well as the size distribution being emitted.
    ! (win, 10/4/07)
    !=====
    IF ( SGCOAG ) THEN

      !save number and mass before emission
      !NO(:, :) = TC1(I,J, :, :)
      !MO(:, :) = TC2(I,J, :, 1:IBINS)

      DO L = 1, LLPAR
        !only really need to loop L=1,NTOP
        S04(L) = TSO4 * EFRAC(L) * DTSRCE
        IF ( S04(L) == 0.d0 ) CYCLE
        DO K = 1, IBINS
          !set number of sulfate particles emitted
          !as emitted mass * fraction in this bin / avg mass per particle for this bin
          NDISTINIT(K) = S04(L) * BFRAC(K) / AVGMASS(K)
          !sfarina - sqrt is expensive.

```

```

      NDISTINIT(K) = SO4(L) * BFRAC(K) /
&                ( SQRT( XK(K)*XK(K+1) ) )
      !set existing number of particles
      NDIST(K) = TC1(I,J,L,K)
      !sfarina - what are the chances aerosol water and ammonium are properly equi
DO C = 1, ICOMP
      !set existing mass of each component
      MDIST(K,C) = TC2(I,J,L,K+(C-1)*IBINS)
      IF( IT_IS_NAN( MDIST(K,C) ) ) THEN
          PRINT *, '+++++++ Found NaN in SRCSF30 +++++++'
          PRINT *, 'Location (I,J,L):', I,J,L, 'Bin', K, 'comp', C
          CALL ERROR_STOP('SRCSF30 SGCCOAG', 'sulfate_mod.f')
      ENDIF
ENDDO
      !initialize emitted sulfate number and mass returned from subgridcoag
      NDISTFINAL(K) = 0.0D0
      MADDFINAL(K) = 0.0D0
ENDDO
      !sfarina subgridcoag does its own mnfix. this call might be unnecessary?
      CALL MNFIX( NDIST, MDIST, ERRORSWITCH )
      IF( ERRORSWITCH ) PRINT *, 'SRCSF30: MNFIX found error ',
&          'before SUBGRIDCOAG at ', I,J,L
      ERRORSWITCH = .FALSE.

      !debug
      DO K = 1, IBINS
          ! Overwrite number and mass before emission for diagnostic
          ! just in case there was any change by MNFIX (win, 10/27/08)
          NO(L,K) = NDIST(K)
          MO(L,K) = MDIST(K,SRTS04)
          Avginit(K) = SUM(MDIST(K,:)) / NDIST(K)
      ENDDO

      BOXVOL = State_Met%AIRVOL(I,J,L) * 1.e6 !convert from m3 -> cm3
      TEMP = State_Met%T(I,J,L)
      PRES = State_Met%PMID(i,j,l)*100.0 ! in Pa

      pdebug=.false.

      CALL SUBGRIDCOAG( NDISTINIT, NDIST, MDIST, BOXVOL,TEMP,
&          PRES, TSCALE, NDISTFINAL, MADDFINAL,pdebug)
      DO K = 1, IBINS
          !add number from emissions
          NDIST2(K) = NDIST(K) + NDISTFINAL(K)
          !use this number for the diag
          Ndiag(K) = Ndiag(K) + NDISTFINAL(K)
          !sfarina - An example to illustrate what's happening here:
          !assuming mass doubling

```

```

!avgmass = .01, .02, .04 (AVGMASS(K) = sqrt(XK(K)*XK(K+1)))
!NO = 100, 50, 25
!MO = 1, 1, 1
!emitted SO4 particles: 10, 5, 1 (for a total mass of 0.24)
!but with subgrid coagulation, we lose 2 particles from bin 1
!onto particles in each of bins 2 and 3, and 1 from bin 2 to bin 3, giving
!Ndistribfinal = 6, 4, 1 (for a total mass of 6*.01 + 4*.02 + 1*.04 = 0.18)
!but that doesn't conserve mass... those particles are now a little heavier
!than they were before subgrid coag, so you have to add that additional mass
!(6*.01)+ (4*.02 + 2*.01) + (1*.04 + 1*.02 + 2*.01) = 0.24

MADDTOTAL = NDISTFINAL(K) * AVGMASS(K)
&
+ MADDFINAL(K)
!copy mass from all species
DO C = 1, ICOMP
  MDIST2(K,C) = MDIST(K,C)
ENDDO
!add mass from emissions as explained above
MDIST2(K,SRTSO4) = MDIST2(K,SRTSO4) + MADDTOTAL
!save this for the diag
Mdiag(K) = Mdiag(K) + MADDTOTAL

!sanity check
if(NDISTFINAL(K) < 0d0) then
  CALL ERROR_STOP('negative number emis','sulfate_mod.f')
endif
if(MADDTOTAL < 0d0) then
  CALL ERROR_STOP('negative mass emis','sulfate_mod.f')
endif
ENDDO

!debug - avg particle mass after emission but before mnfix
DO K = 1, IBINS
  Avginner(K) = SUM(MDIST2(K,:)) / NDIST2(K)
ENDDO

ERRORSWITCH = .FALSE.
CALL MNFIX( NDIST2, MDIST2, ERRORSWITCH )

IF( ERRORSWITCH ) PRINT *, 'SRCSF30: MNFIX found error ',
&
  'after SUBGRIDCOAG at ',I,J,L

DO K = 1, IBINS
  TC1(I,J,L,K) = NDIST2(K)
  DO C=1,ICOMP
    TC2(I,J,L,K+(C-1)*IBINS) = MDIST2(K,C)
  ENDDO
ENDDO

```

```

!debug - avg particle mass final
DO K = 1, IBINS
  Avgfinal(K) = SUM(MDIST2(K,:)) / NDIST2(K)
ENDDO
DO K = 1, IBINS
!sfarina debug
if(TC1(I,J,L,K)-NO(L,K) < Od0) then
  write(*,*) '"Negative NK emis" details:'
  write(*,*) 'NTOPI      ', NTOPI
  write(*,*) 'S_SO4:     ', S_SO4
  write(*,*) 'TSO4:      ', TSO4
  write(*,*) 'EFRAC(L):   ', EFRAC(L)
DO Bi=1,IBINS
  write(*,*) 'Bin          ',Bi
  write(*,*) 'n0, TC1      ', NO(l,bi), TC1(i,j,l,Bi)
  write(*,*) 'ndist1,2     ', NDIST(Bi), NDIST2(Bi)
  write(*,*) 'ndistfinal   ', NDISTFINAL(Bi)
  write(*,*) 'MADDFINAL    ', MADDFINAL(Bi)
  write(*,*) 'Avginit      ', Avginit(Bi)
  write(*,*) 'Avginner     ', Avginner(Bi)
  write(*,*) 'Avgfinal     ', Avgfinal(Bi)
  write(*,*) 'MO(so4)      ', MO(l,bi)
DO C=1,ICOMP
  write(*,*) 'Component    ', C
  write(*,*) 'mdist        ', MDIST(Bi, C)
  write(*,*) 'mdist2       ', MDIST2(Bi, C)
  write(*,*) 'TC2          ', TC2(i,j,l,(C-1)*IBINS+Bi)
END DO !c
END DO !bi
end if
ENDDO

ENDDO ! L loop
!=====
! ND59 Diagnostic: Size-resolved primary sulfate emission in
!                   [kg S/box/timestep] and the corresponding
!                   number emission [no./box/timestep]
!=====
IF ( ND59 > 0 ) THEN
!print*, 'JACK IN ND59 SULFATE'
DO K = 1, IBINS
  if(TC2(I,J,L,K)-MO(L,K) < Od0)
&      print *, 'Negative SF emis ', TC2(I,J,L,K)-MO(L,K),
&      'at', I, J, L, K
  if(TC1(I,J,L,K)-NO(L,K) < Od0) then
    print *, 'Negative NK emis ', TC1(I,J,L,K)-NO(L,K),
&      'at', I, J, L, K

```

```

                print *, 'tc1, NO ', TC1(I,J,L,K), NO(L,K)
            end if
sfarina - I have studied this extensively and determined that negative NK emis
as defined here is not an accurate statement.
The particle number in a given bin IS reduced by this subroutine, but
it is not reduced by emission. Particle number is reduced by mnfix.
if the bin is just about to boil over, that added sulfate mass will
trigger a big particle shift in mnfix and it will look like a
'negative number emission' event as defined by this inequality

#if defined( BPCH_DIAG )
sfarina - changing the definition of this diagnostic to ignore changes to the
distribution by mnfix
        AD59_SULF(I,J,1,K) = AD59_SULF(I,J,1,K) +
&                (TC2(I,J,L,K)-M0(L,K))*S_S04
        AD59_NUMB(I,J,1,K) = AD59_NUMB(I,J,1,K) +
&                TC1(I,J,L,K)-NO(L,K)
        AD59_SULF(I,J,1,K) = AD59_SULF(I,J,1,K) + Mdiag(K)
        AD59_NUMB(I,J,1,K) = AD59_NUMB(I,J,1,K) + Ndiag(K)
#endif

        ENDDO
    ENDIF

ELSE
! Distributing primary emission without sub-grid coagulation
!=====
! Add SO4 emissions to tracer array
! For SF: Convert from [kg SO4/box/s] -> [kg SO4/box/timestep]
! For NK: Convert from [kg SO4/box/s] -> [No. /box/timestep]
!=====
        DO L = 1, LLPAR
            SO4(L) = TS04 * EFRAC(L)
            DO K = 1, IBINS
                TC1(I,J,L,K) = TC1(I,J,L,K) +
&                ( SO4(L) * DTSRCE * BFRAC(K) / AVGMASS(K) )
                TC2(I,J,L,K) = TC2(I,J,L,K) +
&                ( SO4(L) * DTSRCE * BFRAC(K) )
            ENDDO
        ENDDO

#if defined( BPCH_DIAG )
!=====
! ND59 Diagnostic: Size-resolved primary sulfate emission in
!                [kg S/box/timestep] and the corresponding
!                number emission [no./box/timestep]
!=====
        IF ( ND59 > 0 ) THEN
            S04anbf(:, :, 1) = S04an(:, :, 1) + S04bf(:, :)

```

```

      S04anbf(:, :, 2) = S04an(:, :, 2)

      DO L = 1, 2
      DO K = 1, IBINS
        AD59_SULF(I, J, L, K) = AD59_SULF(I, J, L, K) +
&          ( S04anbf(I, J, L) * BFRAC(K)
&          * S_S04 * DTSRCE          )
        AD59_NUMB(I, J, L, K) = AD59_NUMB(I, J, L, K) +
&          ( S04anbf(I, J, L) * BFRAC(K)
&          / AVGMASS(K) * DTSRCE    )

      ENDDO
      ENDDO

      ENDIF
#endif

      ENDIF !SGCOAG

      ENDDO
      ENDDO
$OMP END PARALLEL DO
      IF ( LPRT ) print *, '   ### Finish SRCSF30'

      END SUBROUTINE SRCSF30
#endif
EOC
-----
                        GEOS-Chem Global Chemical Transport Model           !
-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\mbox{}\hrulefill\

\subsubsection [grav\_settling] {grav\_settling}

Subroutine GRAV\_SETTLING performs gravitational settling of
sulfate and nitrate in coarse sea salt (S04S and NITS).
(bec, rjp, bmy, 4/20/04, 7/20/04, 10/25/05)
\\
\\{\bf INTERFACE:}
\begin{verbatim}      SUBROUTINE GRAV_SETTLING( am_I_Root, Input_Opt, State_Met,
&          State_Chm, State_Diag, N,          RC )
USES:

      USE CMN_SIZE_MOD
      #if defined( BPCH_DIAG )

```



```

        USE CMN_DIAG_MOD
        USE DIAG_MOD,           ONLY : AD44
#endif
        USE ErrCode_Mod
        USE GC_GRID_MOD,       ONLY : GET_AREA_CM2
        USE Input_Opt_Mod,     ONLY : OptInput
        USE State_Chm_Mod,     ONLY : ChmState
        USE State_Diag_Mod,    ONLY : DgnState
        USE State_Met_Mod,     ONLY : MetState
        USE Species_Mod,       ONLY : Species
        USE TIME_MOD,          ONLY : GET_ELAPSED_SEC
        USE TIME_MOD,          ONLY : GET_TS_CHEM

```

INPUT PARAMETERS:

```

        LOGICAL,           INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
        TYPE(OptInput),   INTENT(IN)    :: Input_Opt    ! Input Options object
        TYPE(MetState),   INTENT(IN)    :: State_Met    ! Meteorology State object
        INTEGER,          INTENT(IN)    :: N             ! Species index

```

OUTPUT PARAMETERS:

```

        INTEGER,          INTENT(OUT)    :: RC           ! Success or failure?

```

INPUT/OUTPUT PARAMETERS:

```

        TYPE(ChmState),  INTENT(INOUT)  :: State_Chm    ! Chemistry State object
        TYPE(DgnState),  INTENT(INOUT)  :: State_Diag    ! Diagnostics State object

```

REMARKS:

N=1 is SO4S; N=2 is NITS

```

tdf Include Coarse Mode DUST size bins
N=3 is SO4d2; N=4 is NIT_d1
N=5 is SO4d3; N=6 is NIT_d2
N=7 is SO4d4; N=8 is NIT_d3
N=9 is SO4d4; N=10 is NIT_d4
tdf Treat these coated DUSTs as DRY for now

```

REVISION HISTORY:

- (1) Now references SALA_REEDGE_um and SALC_REEDGE_um from "tracer_mod.f" (bmy, 7/20/04)
- (2) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
- (3) Now limit relative humidity to [tiny(real(fp)),0.99] range for DLOG argument (phs, 5/1/08)
- (4) Bug fixes to the Gerber hygroscopic growth for sea salt aerosols (jaegle, 5/5/11)
- (5) Update hygroscopic growth to Lewis and Schwartz formulation (2006) and density calculation based on Tang et al. (1997) (bec, jaegle 5/5/11)

22 Dec 2011 - M. Payer - Added ProTeX headers
 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
 14 Nov 2012 - R. Yantosca - Now pass am_I_Root, Input_Opt, RC as arguments
 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
 12 Sep 2013 - M. Sulprizio- Include Coarse Mode DUST size bins (T.D. Fairlie)
 06 Jan 2015 - M. Yannetti - Changed some variables to f8 as needed
 26 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID and
 remove dependency on pressure_mod.
 22 Jan 2016 - E. Lundgren - Update netcdf drydep flux diagnostics
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 31 May 2016 - E. Lundgren - Remove usage of Input_Opt%XNUMOL
 24 Jun 2016 - R. Yantosca - Now return if id_S04d1<0 or id_NITd1<0 (not ==0)
 02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

8.5.3 chem_dms

Subroutine CHEM_DMS is the DMS chemistry subroutine from Mian Chin's GOCART model, modified for use with the GEOS-CHEM model. (rjp, bdf, bmy, 5/31/00, 10/15/09)

INTERFACE:

```

SUBROUTINE CHEM_DMS( am_I_Root, Input_Opt, State_Met,
&                   State_Chm, State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD
USE DIAG_MOD,          ONLY : AD05
#endif
USE ErrCode_Mod
USE Input_Opt_Mod,    ONLY : OptInput
USE State_Chm_Mod,    ONLY : ChmState
USE State_Diag_Mod,   ONLY : DgnState
USE State_Met_Mod,    ONLY : MetState
USE TIME_MOD,         ONLY : GET_TS_CHEM

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)  :: State_Diag    ! Diagnostics State object

```

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: RC ! Success or failure?

REMARKS:

Reaction List (by Mian Chin, chin@rondo.gsfc.nasa.gov)

R1: DMS + OH -> a*S02 + b*MSA OH addition channel
 k1 = { 1.7e-42*exp(7810/T)*[O2] / (1+5.5e-31*exp(7460/T)*[O2] }
 a = 0.75, b = 0.25

R2: DMS + OH -> S02 + ... OH abstraction channel
 k2 = 1.2e-11*exp(-260/T)

DMS_OH = DMS0 * exp(-(r1+r2)* NDT1)
 where DMS0 is the DMS concentration at the beginning,
 r1 = k1*[OH], r2 = k2*[OH].

R3: DMS + NO3 -> S02 + ...
 k3 = 1.9e-13*exp(500/T)

DMS = DMS_OH * exp(-r3*NDT1)
 where r3 = k3*[NO3].

R4: DMS + X -> S02 + ...
 assume to be at the rate of DMS+OH and DMS+NO3 combined.

The production of S02 and MSA here, PSO2_DMS and PMSA_DMS, are saved for use in CHEM_S02 and CHEM_MSA subroutines as a source term. They are in unit of [v/v/timestep].

REVISION HISTORY:

- (1) Now reference AD, AIRDEN, and SUNCOS from "dao_mod.f". Added parallel DO-loops. Also now extract OH and NO3 from SMVGEAR for coupled chemistry-aerosol runs. (rjp, bdf, bmy, 9/16/02)
- (2) Bug fix: remove duplicate definition of RK3 (bmy, 3/23/03)
- (3) Now use function GET_TS_CHEM from "time_mod.f". (bmy, 3/27/03)
- (4) Now reference STT and ITS_A_FULLCHEM_SIM from "tracer_mod.f" Now replace IJSURF w/ an analytic function. (bmy, 7/20/04)
- (5) Shift rows 8,9 in AD05 to 9,10 in to make room for P(S04) from O3 oxidation in sea-salt aerosols (bec, bmy, 4/13/05)
- (6) Now remove reference to CMN, it's obsolete. Now reference ITS_IN_THE_STRAT from "tropopause_mod.f". (bmy, 8/22/05)
- (7) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
- (8) Now correctly records P(S02) from OH in AD05 (pjh)
- (9) Update reaction rate to match JPL06 and full chem (jaf, bmy, 10/15/09)

22 Dec 2011 - M. Payer - Added ProTeX headers
 31 Jul 2012 - R. Yantosca - Now loop from 1..LLPAR for GIGC compatibility
 14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
 06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)
 24 Jun 2016 - R. Yantosca - Now Return if id_DMS < 0 (not == 0)
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 08 Dec 2017 - R. Yantosca - Now accept State_Diag as an argument

8.5.4 chem_h2o2

Subroutine CHEM_H2O2 is the H2O2 chemistry subroutine for offline sulfate simulations. For coupled runs, H2O2 chemistry is already computed by the SMVGEAR module. (rjp, bmy, 11/26/02, 10/25/05)

INTERFACE:

```

SUBROUTINE CHEM_H2O2( am_I_Root, Input_Opt, State_Met,
&                   State_Chm, State_Diag, RC           )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD,   ONLY : GET_FRAC_UNDER_PBLTOP
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD,     ONLY : GET_MONTH
USE TIME_MOD,     ONLY : GET_TS_CHEM
USE TIME_MOD,     ONLY : ITS_A_NEW_MONTH

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)   :: am_I_Root   ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)   :: Input_Opt   ! Input Options object
TYPE(MetState), INTENT(IN)    :: State_Met   ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm   ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag  ! Diagnostics State object

```

OUTPUT PARAMETERS:

30 Jun 2016 - R. Yantosca - Remove instances of Spc. Now get the advected species ID from State_Chm%Map_Advect.

EOC

BOC

DEFINED PARAMETERS:

REAL(fp), PARAMETER :: A = 2.9e-12_fp

LOCAL VARIABLES:

! SAVED Scalars

LOGICAL :: FIRST = .TRUE.

INTEGER, SAVE :: LASTMONTH = -99

! Scalars

INTEGER :: I, J, L

REAL(fp) :: DT, Koh, DH2O2, M, F, XTAU

REAL(fp) :: H2O2O, H2O2, ALPHA, FREQ, PHOTJ

! Strings

CHARACTER(LEN=255) :: FILENAME

! Arrays

REAL*4 :: ARRAY(IIPAR, JJPARG, LLCHEM)

! Pointers

REAL(fp), POINTER :: Spc(:, :, :, :)

!=====

! CHEM_H2O2 begins here!

!=====

IF (id_H2O2 < 0) RETURN

! Assume success

RC = GC_SUCCESS

! Point to chemical species array [v/v dry]

Spc => State_Chm%Species

! Chemistry timestep [s]

DT = GET_TS_CHEM()

! Factor to convert AIRDEN from kgair/m3 to molecules/cm3:

F = 1000.e+0_fp / AIRMW * AVO * 1.e-6_fp

!=====

```

! Loop over tropospheric grid boxes and do chemistry
!=====
$OMP PARALLEL DO
$OMP+DEFAULT( SHARED )
$OMP+PRIVATE( I, J, L, M, H2O20, KOH, FREQ, ALPHA, DH2O2, H2O2 )
$OMP+PRIVATE( PHOTJ )
$OMP+SCHEDULE( DYNAMIC )
  DO L = 1, LLPAR
  DO J = 1, JJPAR
  DO I = 1, IIPAR

    ! Initialize for safety's sake
    FREQ = 0e+0_fp

    ! Skip non-chemistry boxes
    IF ( .not. State_Met%InChemGrid(I,J,L) ) CYCLE

    ! Density of air [molec/cm3]
    M      = State_Met%AIRDEN(I,J,L) * f

    ! Initial H2O2 [v/v]
    H2O20 = Spc(I,J,L,id_H2O2)

    ! Loss frequently due to OH oxidation [s-1]
    KOH    = A * EXP( -160.e+0_fp / State_Met%T(I,J,L) ) *
&          GET_OH( I, J, L, Input_Opt, State_Chm, State_Met )

    ! Now do all dry deposition in mixing_mod.F90 (ckeller, 3/5/15)
    FREQ = 0.e+0_fp

    ! Impose a diurnal variation of jH2O2 by multiplying COS of
    ! solar zenith angle normalized by maximum solar zenith angle
    ! because the archived JH2O2 is for local noon time
    IF ( COSZM(I,J) > 0.e+0_fp ) THEN
      PHOTJ = JH2O2(I,J,L) * State_Met%SUNCOS(I,J) / COSZM(I,J)
      PHOTJ = MAX( PHOTJ, 0e+0_fp )
    ELSE
      PHOTJ = 0e+0_fp
    ENDIF

    ! Compute loss fraction from OH, photolysis, drydep [unitless].
    ALPHA = 1.e+0_fp + ( KOH + PHOTJ + FREQ ) * DT

    ! Delta H2O2 [v/v]
    ! PH2O2m is in kg/m3 (from HEMCO), convert to molec/cm3/s (mps,9/18/14)
    DH2O2 = ( PH2O2m(I,J,L) / TS_EMIS * XNUMOL_H2O2 / CM3PERM3 )
&          * DT / ( ALPHA * M )

```

```

      ! Final H2O2 [v/v]
      H2O2 = ( H2O20 / ALPHA + DH2O2 )
      IF ( H2O2 < SMALLNUM ) H2O2 = 0e+0_fp

      ! Store final H2O2 in Spc
      Spc(I,J,L,id_H2O2) = H2O2

      ENDDO
      ENDDO
      ENDDO
$OMP END PARALLEL DO

      ! Free pointer
      Spc => NULL()

      END SUBROUTINE CHEM_H2O2
EOC
-----
                        GEOS-Chem Global Chemical Transport Model                !
-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\mbox{\hrulefill\

\subsection [chem\_so2] {chem\_so2}

Subroutine CHEM\_SO2 is the SO2 chemistry subroutine.
  (rjp, bmy, 11/26/02, 8/26/10)
\\
\\{\bf INTERFACE:}
\begin{verbatim}
      SUBROUTINE CHEM_SO2( am_I_Root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, FullRun,  RC )

```

USES:

```

      USE CMN_SIZE_MOD
      USE DAO_MOD,           ONLY : IS_WATER
      USE DUST_MOD,         ONLY : GET_DUST_ALK      ! tdf 04/08/08
      USE ErrCode_Mod
      USE ERROR_MOD,       ONLY : IS_SAFE_EXP
      USE ERROR_MOD,       ONLY : SAFE_DIV
      USE ERROR_MOD,       ONLY : ERROR_STOP
      USE Input_Opt_Mod,   ONLY : OptInput
      USE PRESSURE_MOD,    ONLY : GET_PCENTER
      USE State_Chm_Mod,   ONLY : ChmState
      USE State_Met_Mod,   ONLY : MetState
      USE State_Diag_Mod,  ONLY : DgnState

```



```

        USE TIME_MOD,          ONLY : GET_TS_CHEM, GET_MONTH
        USE TIME_MOD,          ONLY : ITS_A_NEW_MONTH
        USE WETSCAV_MOD,       ONLY : H2O2s
        USE WETSCAV_MOD,       ONLY : SO2s
        USE HCO_INTERFACE_MOD, ONLY : GetHcoDiagn
#if defined( BPCH_DIAG )
        USE CMN_DIAG_MOD
        USE DIAG_MOD,          ONLY : AD05
#endif

```

INPUT PARAMETERS:

```

        LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
        LOGICAL,          INTENT(IN)    :: FullRun     ! Modify species conc?
        TYPE(OptInput),  INTENT(IN)    :: Input_Opt   ! Input Options object
        TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

        TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
        TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

        INTEGER,          INTENT(INOUT) :: RC          ! Success or failure?

```

REMARKS:

Reaction List (by Rokjin Park, rjp@io.harvard.edu)

=====

(1) S02 production:

DMS + OH, DMS + NO3 (saved in CHEM_DMS)

(2) S02 loss:

(a) S02 + OH -> S04

(b) S02 -> drydep

(c) S02 + H2O2 or O3 (aq) -> S04

(3) $S02 = S02_0 * \exp(-bt) + PS02_DMS/bt * [1-\exp(-bt)]$

where b is the sum of the reaction rate of S02 + OH and the dry deposition rate of S02, PS02_DMS is S02 production from DMS in MixingRatio/timestep.

If there is cloud in the gridbox (fraction = fc), then the aqueous phase chemistry also takes place in cloud. The amount of S02 oxidized by H2O2 in cloud is limited by the available H2O2; the rest may be oxidized due to additional chemistry, e.g, reaction with O3 or O2 (catalyzed by trace metal).

REVISION HISTORY:

- (1) Removed duplicate definition of Ki (bmy, 11/15/01)
 - (2) Eliminate duplicate HPLUS definition. Make adjustments to facilitate SMVGEAR chemistry for fullchem runs (rjp, bmy, 3/23/03)
 - (3) Now replace DXYP(J+J0)*1d4 with routine GET_AREA_CM2 of "grid_mod.f"
Now use function GET_TS_CHEM from "time_mod.f".
 - (4) Now apply dry deposition to entire PBL. Now references PBLFRAC array from "drydep_mod.f". (bmy, 8/1/03)
 - (5) Now use ND44_TMP array to store vertical levels of drydep flux, then sum into AD44 array. This prevents numerical differences when using multiple processors. (bmy, 3/24/04)
 - (6) Now use parallel DO-loop to zero ND44_TMP (bmy, 4/14/04)
 - (7) Now reference STT, TCVV, & ITS_AN_AEROSOL_SIM from "tracer_mod.f".
Now reference DATA_DIR from "directory_mod.f" (bmy, 7/20/04)
 - (8) Replace PBLFRAC from "drydep_mod.f" with GET_FRAC_UNDER_PBLTOP from "pbl_mix_mod.f" (bmy, 2/22/05)
 - (9) Modified for SO4s, NITs. Also modified for alkalinity w/in the seasalt chemistry. (bec, bmy, 4/13/05)
 - (10) Now remove reference to CMN, it's obsolete. Now reference ITS_IN_THE_STRAT from "tropopause_mod.f" (bmy, 8/22/05)
 - (11) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
 - (12) Updated to match JPL 2006 + full chem (jaf, bmy, 10/15/09)
 - (13) Now prevent floating-point exceptions when taking the exponential terms. (win, bmy, 1/4/10)
 - (14) Save aqueous production rate to PSO4_S02AQ for TOMAS microphysics (win, 1/25/10)
 - (15) Added extra error checks to prevent negative L2S, L3S (bmy, 4/28/10)
 - (16) Use liq. water content from met fields in GEOS-5 (jaf, bmy, 6/30/10)
- 26 Aug 2010 - R. Yantosca - Use liquid water content from MERRA
- 12 Nov 2010 - R. Yantosca - Prevent div-by-zero when computing L2S and L3S
- 27 May 2011 - L. Zhang - Divide LWC by cloud fraction for GEOS/MERRA
and adjust the L2S and L3S rates accordingly
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.2 in the same way as MERRA
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
- 31 Jul 2012 - R. Yantosca - Now loop over 1..LLPAR for GIGC compatibility
- 31 Jul 2012 - R. Yantosca - Declare temp drydep arrays w/ LLPAR (not LLTROP)
- 14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
- 05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNL PBL
- 19 Mar 2013 - R. Yantosca - Now copy Input_Opt%TCVV(1:N_TRACERS) and
Input_Opt%XNUMOL(1:N_TRACERS) -- avoid OOB errs
- 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
- 05 Sep 2013 - M. Sulprizio - Add modifications for cloud pH (B. Alexander)
- 06 Sep 2013 - M. Sulprizio - Bug fix: Prevent divide-by-zero if LWC=0. Only
do aqueous SO2 chemistry when LWC>0.
- 12 Sep 2013 - M. Sulprizio - Modified for SO4d, NITd. Also modified for
alkalinity w/in the dust chemistry. (tdf 4/07/08)


```

USE State_Chm_Mod,      ONLY : ChmState
USE State_Diag_Mod,    ONLY : DgnState
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : GET_TS_CHEM
USE TIME_MOD,          ONLY : GET_ELAPSED_SEC
USE TIME_MOD,          ONLY : GET_MONTH
USE TIME_MOD,          ONLY : ITS_A_NEW_MONTH

```

INPUT PARAMETERS:

```

INTEGER,      INTENT(IN)  :: I, J, L      ! Grid box indices
REAL(fp),    INTENT(IN)  :: SO2_cd       ! SO2 mixing ratio [v/v] after
                                           ! gas phase chemistry and
                                           ! dry deposition
REAL(fp),    INTENT(IN)  :: Kt1, Kt2     ! Rate constant [s-1] for
                                           ! sulfate formation on sea
                                           ! salt aerosols from GET_ALK
                                           ! (1=fine; 2=coarse)
REAL(fp),    INTENT(IN)  :: Kt1N, Kt2N
REAL(fp),    INTENT(IN)  :: ALK1, ALK2   ! Alkalinity [kg] from
                                           ! seasalt_mod
TYPE(MetState), INTENT(IN) :: State_Met  ! Meteorology State object
TYPE(OptInput), INTENT(IN) :: Input_Opt  ! Input Options object
LOGICAL,     INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
LOGICAL,     INTENT(IN)  :: FullRun     ! Modify species conc?

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

REAL(fp),    INTENT(OUT)  :: SO2_ss     ! SO2 mixing ratio [v/v]
                                           ! after sea salt chemistry
REAL(fp),    INTENT(OUT)  :: PS04E     ! SO4E (sulfate produced by
                                           ! S(IV)+O3 on fine seasalt)
                                           ! mixing ratio [v/v]
REAL(fp),    INTENT(OUT)  :: PS04F     ! SO4F (sulfate produced by
                                           ! S(IV)+O3 on coarse seasalt)
REAL(fp),    INTENT(OUT)  :: AlkA      ! Modified SSA alkalinity
REAL(fp),    INTENT(OUT)  :: AlkC      ! Modified SSA alkalinity
INTEGER,     INTENT(OUT)  :: RC        ! Success or failure?

```

REMARKS:

Chemical reactions:

```

=====
(R1) SO2 + O3 + ALK => SO4 + O2

```

Modeled after Chamedies and Stelson, 1992?

REVISION HISTORY:

- (1) Now references XNUMOLAIR from "tracer_mod.f" (bmy, 10/25/05)
 - (2) Bug fix: now avoid seg fault error if IDTHNO3 is zero, as it would be for an offline aerosol simulation. (bmy, 3/29/06)
 - (3) Fixed typo in FALK_A_SO2 equation: C_FLUX_C should be C_FLUX_A. (havala, bec, bmy, 12/8/06)
 - (4) Bug fix for mass balance, replace TITR_HNO3 w/ HNO3_SSC in the expression for HNO3_ss. Bug fix: now do equivalent computation for GET_GNO3, which is now no longer called because it's in "isoropia_mod.f". (bec, bmy, 7/30/08)
 - 22 Dec 2011 - M. Payer - Added ProTeX headers
 - 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
 - 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
 - 12 Sep 2013 - M. Sulprizio- Added loss of HNO3 on sea salt diagnostic (T.D. Fairlie)
 - 18 Sep 2014 - M. Sulprizio- Now get HNO3 for offline aerosol sim from HEMCO
 - 24 Mar 2015 - E. Lundgren - Remove dependency on tracer_mod since XNUMOLAIR now defined in CMN_GTCM_MOD
 - 06 Jan 2016 - E. Lundgren - Use global physical parameters
 - 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
 - 11 Oct 2016 - R. Yantosca - Bug fix: make sure that MW_HNO3 is defined for aerosol-only sims (where id_HNO3 < 0)
 - 10 Feb 2017 - R. Yantosca - Remove species molecular weights from conversions from v/v to equivalents & back
 - 08 Dec 2017 - R. Yantosca - Now accept State_Diag as an argument
-

8.5.6 dust_chem

Subroutine DUST_CHEM computes SO4 formed from S(IV) + O3 on dust aerosols as a function of dust alkalinity (tdf 3/28/2K8) Based on routine SEASALT_CHEM (bec, bmy, 4/13/05, 10/25/05)

INTERFACE:

```

SUBROUTINE DUST_CHEM ( I,          J,          L,
&                     ALK,          SO2_cd,    H2SO4_cd,
&                     KTS,          KTN,       KTH,
&                     SO2_gas,    H2SO4_gas, PSO4d,
&                     PH2SO4d,    PNITd,     ALKA,
&                     Input_Opt, State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD          ! Size parameters
USE ErrCode_Mod
USE ERROR_MOD,           ONLY : GEOS_CHEM_STOP
USE ERROR_MOD,           ONLY : IT_IS_NAN

```

```

USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : GET_TS_CHEM,      GET_ELAPSED_SEC
USE TIME_MOD,          ONLY : GET_ELAPSED_SEC,   GET_MONTH
USE TIME_MOD,          ONLY : ITS_A_NEW_MONTH

```

INPUT PARAMETERS:

```

INTEGER,      INTENT(IN)  :: I, J, L      ! Grid box indices
REAL(fp),    INTENT(IN)  :: SO2_cd       ! SO2 mixing ratio after
                                                ! gas phase chemistry and
                                                ! dry deposition [v/v]
REAL(fp),    INTENT(IN)  :: H2SO4_cd     ! H2SO4 mixing ratio after
                                                ! gas phase chemistry and
                                                ! dry deposition [v/v]
REAL(fp),    INTENT(IN)  :: ALK(NDSTBIN) ! Dust Alkalinity [v/v]
REAL(fp),    INTENT(IN)  :: KTS(NDSTBIN) ! Rate constant for uptake
                                                ! of SO2 on dust [s-1]
REAL(fp),    INTENT(IN)  :: KTN(NDSTBIN) ! Rate constant for uptake
                                                ! of HNO3 on dust [s-1]
REAL(fp),    INTENT(IN)  :: KTH(NDSTBIN) ! Size- and area-weighted
                                                ! FRACTION for uptake of
                                                ! H2SO4 on dust
TYPE(MetState), INTENT(IN) :: State_Met  ! Meteorology State object
TYPE(OptInput), INTENT(IN) :: Input_Opt  ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

OUTPUT PARAMETERS:

```

REAL(fp),    INTENT(OUT) :: SO2_gas      ! SO2 mixing ratio after
                                                ! dust chem [v/v]
REAL(fp),    INTENT(OUT) :: PS04d(NDSTBIN) ! Sulfate produced by
                                                ! S(IV)+O3 on dust in
                                                ! each size bin
REAL(fp),    INTENT(OUT) :: H2SO4_gas    ! H2SO4 mixing ratio
                                                ! after dust chem [v/v]
REAL(fp),    INTENT(OUT) :: PNITd (NDSTBIN) ! Nitrate produced by
                                                ! HNO3 uptake on dust
                                                ! in each size bin
REAL(fp),    INTENT(OUT) :: PH2SO4d(NDSTBIN) ! Sulfate produced by
                                                ! uptake of H2SO4 on
                                                ! dust in each size bin
REAL(fp),    INTENT(OUT) :: ALKA(NDSTBIN) ! Dust Alkalinity after
                                                ! dust chemistry [v/v]
INTEGER,     INTENT(OUT) :: RC           ! Success or failure?

```

REMARKS:

Chemical reactions:

```
=====
(R1) SO2 + O3 + CaCO3 => CaSO4 + O2 + CO2
```

```
(R2) 2(HNO3) + CaCO3 => Ca(NO3)2 + CO2 + H2O
```

Added sulfate production due to H2SO4 adsorption tdf 2/13/2K9

```
(R3) H2SO4 + CaCO3 => CaSO4 + H2O + CO2
```

REVISION HISTORY:

28 Mar 2008 - T.D. Fairlie- Initial version

16 Sep 2013 - M. Sulprizio- Added ProTeX headers

17 Sep 2013 - M. Sulprizio- Now pass Input_Opt, State_Met, State_Chm, and RC
as arguments

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
species ID from State_Chm%Map_Advect.

8.5.7 get_hplus

Subroutine GET_HPLUS computes H⁺ concentrations in cloud liquid water for pH dependent cloud chemistry. (bec, 4/11/11)

INTERFACE:

```
SUBROUTINE GET_HPLUS( S04nss, TNH3, TNO3, S02,
&                    T,      PRES, LWC, ihPLUS, HPLUS )
```

USES:

```
USE ERROR_MOD,      ONLY : IT_IS_NAN, GEOS_CHEM_STOP
```

INPUT PARAMETERS:

```
REAL(fp),  INTENT(IN)  :: S04nss ! Total nss sulfate mixing ratio [M]
REAL(fp),  INTENT(IN)  :: TNO3   ! Total nitrate (gas+particulate) mixing
                                ! ratio [v/v]
REAL(fp),  INTENT(IN)  :: TNH3   ! NH3 mixing ratio [v/v]
REAL(fp),  INTENT(IN)  :: S02    ! SO2 mixing ratio [v/v]
REAL(fp),  INTENT(IN)  :: T      ! Temperature [K]
REAL(fp),  INTENT(IN)  :: PRES   ! Dry air partial ressure [atm]
REAL(fp),  INTENT(IN)  :: LWC    ! Cloud liquid water content [m3/m3]
REAL(fp),  INTENT(IN)  :: ihPLUS ! Initial [H+] [M]
```

OUTPUT PARAMETERS:

```
REAL(fp),  INTENT(OUT) :: HPLUS  ! Calculated [H+] [M]
```

REMARKS:

Calculation:

=====

Solve the following electroneutrality equation:

$$[H+] = 2[S04]nss + [Cl] + [OH] + [HCO3] + 2[CO3] + [HSO3] + 2[S03] + [NO3] \\ - [Na] - 2[Ca] - [K] - 2[Mg] - [NH4]$$

Aqueous concentrations of [Cl], [Na], [Ca], [K], and [Mg] come from ISORROPIA II

Let concentrations of [HCO3], [CO3], [HSO3], [S03], [NO3] and [NH4] evolve according to Henry's law equilibrium.

Assume [S(VI)] = [S04]nss (this applies for pH > 3)

REVISION HISTORY:

- 25 Jan 2012 - M. Payer - Added ProTeX headers
- 06 Jan 2015 - M. Yannetti - Set some variables to f8 that required it
- 28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure

8.5.8 kCO21

Function kCO21

INTERFACE:

FUNCTION kCO21 (P, T, LWC, HPLUS) RESULT (KCO2p)

INPUT PARAMETERS:

REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS

OUTPUT PARAMETERS:

REAL(fp) :: KCO2p, KCO2p2

REVISION HISTORY:

- 25 Jan 2012 - M. Payer - Added ProTeX headers
- 28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
- 22 Mar 2017 - M. Sulprizio - Dhco2 value is from Table 7.3 of Seinfeld and Pandis (2006, pp 289) and should be positive for consistency with the way it is used here. Also, the value of R, in units of kcal mol⁻¹ K⁻¹, is 1.986x10⁻³, not 0.04. (V. Shah)

8.5.9 kCO22

Function kCO22

INTERFACE:


```
FUNCTION kCO22 ( P, T, LWC, HPLUS ) RESULT ( KC02p2 )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KC02p, KC02p2
```

REVISION HISTORY:

```
25 Jan 2012 - M. Payer      - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
22 Mar 2017 - M. Sulprizio- Dhco2 value is from Table 7.3 of Seinfeld and
                        Pandis (2006, pp 289) and should be positive for
                        consistency with the way it is used here. Also,
                        the value of R, in units of kcal mol-1 K-1, is
                        1.986x10-3, not 0.04. (V. Shah)
```

8.5.10 kSO21

Function kSO21

INTERFACE:

```
FUNCTION kSO21 ( P, T, LWC, HPLUS, SO2 ) RESULT ( KS02p )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS, SO2
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KS02p, KS02p2
```

REVISION HISTORY:

```
25 Jan 2012 - M. Payer      - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
22 Mar 2017 - M. Sulprizio- Dhso2 value is from Table 7.3 of Seinfeld and
                        Pandis (2006, pp 289) and should be positive for
                        consistency with the way it is used here. Also,
                        the value of R, in units of kcal mol-1 K-1, is
                        1.986x10-3, not 0.04. (V. Shah)
```

8.5.11 kSO22

Function kSO22

INTERFACE:

```
FUNCTION kSO22 ( P, T, LWC, HPLUS, SO2 ) RESULT ( KS02p2 )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS, SO2
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KS02p, KS02p2
```

REVISION HISTORY:

25 Jan 2012 - M. Payer - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
22 Mar 2017 - M. Sulprizio- Dhso2 value is from Table 7.3 of Seinfeld and Pandis (2006, pp 289) and should be positive for consistency with the way it is used here. Also, the value of R, in units of kcal mol⁻¹ K⁻¹, is 1.986x10⁻³, not 0.04. (V. Shah)

8.5.12 kHNO3

Function kNO3

INTERFACE:

```
FUNCTION kHNO3 ( P, T, LWC, HPLUS, HNO3 ) RESULT ( KHNO3p )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS, HNO3
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KHNO3p
```

REVISION HISTORY:

25 Jan 2012 - M. Payer - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
22 Mar 2017 - M. Sulprizio- Add fix for Hhno3eff from V. Shah

8.5.13 kHCl

Function kHCl

INTERFACE:

```
FUNCTION kHCl ( P, T, LWC, HPLUS, Cl ) RESULT ( KHClp )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS, Cl
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KHClp
```

REVISION HISTORY:

25 Jan 2012 - M. Payer - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
17 Oct 2017 - M. Sulprizio- Dhck value is from Table 7.3 of Seinfeld and Pandis (2006, pp 289) and should be positive for consistency with the way it is used here. Also, the value of R, in units of kcal mol⁻¹ K⁻¹, is 1.986x10⁻³, not 0.04. (Qianjie Chen)

8.5.14 kNH3

Function kNH3

INTERFACE:

```
FUNCTION kNH3 ( P, T, LWC, HPLUS, NH3, Kw ) RESULT ( KNH3p )
```

INPUT PARAMETERS:

```
REAL(fp), INTENT(IN) :: T, P, LWC, HPLUS, NH3, Kw
```

OUTPUT PARAMETERS:

```
REAL(fp)                :: KNH3p
```

REVISION HISTORY:

25 Jan 2012 - M. Payer - Added ProTeX headers
28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure
22 Mar 2017 - M. Sulprizio- Dnh3 value is from Table 7.3 of Seinfeld and Pandis (2006, pp 289) and should be positive for consistency with the way it is used here. Also, the value of R, in units of kcal mol⁻¹ K⁻¹, is 1.986x10⁻³, not 0.04. (V. Shah)

8.5.15 cubic

Subroutine CUBIC finds the roots of a cubic equation / 3rd order polynomial

INTERFACE:

```
SUBROUTINE CUBIC( A2, A1, A0, NR, CRUTES )
```

USES:

```
USE ERROR_MOD, ONLY : GEOS_CHEM_STOP !ERROR_STOP
```

INPUT PARAMETERS:

```
INTEGER          :: NR  
REAL(f8)         :: A2, A1, A0  
REAL(fp)         :: CRUTES(3)
```

REMARKS:

Formulae can be found in numer. recip. on page 145
kiran developed this version on 25/4/1990
Dr. Francis S. Binkowski modified the routine on 6/24/91, 8/7/97

*** modified 2/23/98 by fsb to incorporate Dr. Ingmar Ackermann's
recommendations for setting a0, a1,a2 as real(fp) variables.
Modified by Bob Yantosca (10/15/02)
- Now use upper case / white space
- force double precision with "D" exponents
- updated comments / cosmetic changes
- now call ERROR_STOP from "error_mod.f" to stop the run safely

REVISION HISTORY:

```
25 Jan 2012 - M. Payer    - Added ProTeX headers  
06 Jan 2015 - M. Yannetti - Manual changes of some variables to f8
```

8.5.16 aqchem_so2

Subroutine AQCHEM_SO2 computes the reaction rates for aqueous SO2 chemistry. (rjp, bmy, 10/31/02, 12/12/02)

INTERFACE:

```
SUBROUTINE AQCHEM_SO2( LWC,      T,      P,      S02,  H2O2,  
&                      O3,      Hplus, MnII,   FeIII,  
&                      KaqH2O2, KaqO3, KaqO3_1, KaqO2,  
&                      HS03aq,  S03aq )
```

INPUT PARAMETERS:

```

REAL(fp), INTENT(IN)  :: LWC      ! Liq water content [m3/m3]=1.E-6*L [g/m3]
REAL(fp), INTENT(IN)  :: T        ! Temperature [K]
REAL(fp), INTENT(IN)  :: P        ! Dry air partial pressure [atm]
REAL(fp), INTENT(IN)  :: SO2      ! SO2 mixing ratio [v/v]
REAL(fp), INTENT(IN)  :: H2O2     ! H2O2 mixing ratio [v/v]
REAL(fp), INTENT(IN)  :: O3       ! O3 mixing ratio [v/v]
REAL(fp), INTENT(IN)  :: HPLUS    ! Concentration of H+ ion (i.e. pH) [v/v]
REAL(fp), INTENT(IN)  :: MnII     ! Concentration of MnII [mole/l]
REAL(fp), INTENT(IN)  :: FeIII    ! Concentration of FeIII [mole/l]

```

OUTPUT PARAMETERS:

```

REAL(fp), INTENT(OUT) :: KaqH2O2 ! Reaction rate for H2O2
REAL(fp), INTENT(OUT) :: KaqO3   ! Reaction rate for O3
REAL(fp), INTENT(OUT) :: KaqO3_1 ! only the SO3-- oxidation, (qjc, 04/10/16)
REAL(fp), INTENT(OUT) :: KaqO2   ! Reaction rate for O2 (metal cat)
REAL(fp), INTENT(OUT) :: HS03aq  ! Cloud bisulfite [mol/l] (qjc, 06/10/16)
REAL(fp), INTENT(OUT) :: SO3aq   ! Cloud sulfite [mol/l] (qjc, 06/10/16)

```

REMARKS:

Chemical Reactions:

```

=====
(R1) HS03- + H2O2(aq) + H+ => SO4-- + 2H+ + H2O [Jacob, 1986]

```

$$d[S(VI)]/dt = k[H^+][H_2O_2(aq)][HSO_3^-]/(1 + K[H^+])$$

[Seinfeld and Pandis, 1998, page 366]

```

(R2) SO2(aq) + O3(aq) =>
     HS03-   + O3(aq) =>
     SO3--   + O3(aq) =>
     [Jacob, 1986; Jacobson, 1999]

```

$$d[S(VI)]/dt = (k_0[SO_2(aq)] + k_1[HSO_3^-] + K_2[SO_3--])[O_3(aq)]$$

[Seinfeld and Pandis, 1998, page 363]

Reaction rates can be given as

$$R_a = k [H_2O_2(aq)] [S(IV)] \quad [\text{mole/liter*s}] \quad \text{OR}$$

$$K_{rate} = R_a LWC R T / P \quad [1/s]$$

Where:

LWC = Liquid water content(g/m3)*10⁻⁶ [m3(water)/m3(gas)]
R = 0.08205 (atm L / mol-K), Universal gas const.
T = Temperature (K)
P = Pressure (atm)

Procedure:

```

=====
(a ) Given [SO2] which is assumed to be total SO2 (gas+liquid) in

```

equilibrium between gas and liquid phase.

- (b) We can compute $S_{O_2}(g)$ using Henry's law
 $P(so_2(g)) = X_g * [S_{O_2}]$
 $X_g = 1/(1 + F_{aq})$, Fraction of S_{O_2} in gas
 where:
 $F_{aq} = K_{heff} * R * T * LWC$,
 K_{heff} = Effective Henry's constant

- (c) Then Calculate Aquous phase, $S[IV]$ concentrations
 $S[IV] = K_{heff} * P(so_2(g) \text{ in atm}) [M]$

- (d) The exact same procedure is applied to calculate $H_2O_2(aq)$

REVISION HISTORY:

- (1) Updated by Rokjin Park (rjp, bmy, 12/12/02)
 22 Dec 2011 - M. Payer - Added ProTeX headers
 28 Apr 2015 - E. Lundgren - Input pressure is now dry air partial pressure

8.5.17 het_drop_chem

Subroutine HET_DROP_CHEM estimates the in-cloud sulfate production rate in heterogeneous cloud droplets based on the Yuen et al., 1996 parameterization. (bec, 6/16/11)

INTERFACE:

```

SUBROUTINE HET_DROP_CHEM( I,      J,      L,      LSTOT, SSCvv,
&                          aSO4, GNH3, SO2_sr, H2O2O, GN03,
&                          SR,   Input_Opt, State_Met,
&                          State_Chm
)
```

USES:

```

USE ERROR_MOD,          ONLY : IT_IS_FINITE, GEOS_CHEM_STOP
USE Input_Opt_Mod,     ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : GET_TS_CHEM
```

INPUT PARAMETERS:

```

INTEGER,                INTENT(IN)  :: I, J, L
REAL(fp),               INTENT(IN)  :: LSTOT
REAL(fp),               INTENT(IN)  :: SSCvv
REAL(fp),               INTENT(IN)  :: aSO4
REAL(fp),               INTENT(IN)  :: GNH3
REAL(fp),               INTENT(IN)  :: SO2_sr
REAL(fp),               INTENT(IN)  :: H2O2O
```

```

REAL(fp),          INTENT(IN)  :: GNO3
TYPE(OptInput),   INTENT(IN)  :: Input_Opt   ! Input Options object
TYPE(MetState),   INTENT(IN)  :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),   INTENT(INOUT) :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

REAL(fp),          INTENT(OUT) :: SR ! Sulfate production rate

```

REVISION HISTORY:

```

25 Jan 2012 - M. Payer    - Added ProTeX headers
05 Sep 2013 - M. Sulprizio- Now pass met fields using the State_Met object
06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)
22 Jun 2016 - M. Yannetti - Pass State_Chm as arg for spc db access
03 Feb 2017 - M. Sulprizio- Add fixes for sulfate production provided by
                        Qianjie Chen:
                        - Bug fix: NDss should be #/cm3, not #/m3
                        - Compute updraft velocity W from met fields
                        - Only compute SR when W>0

```

8.5.18 chem_so4

Subroutine CHEM_SO4 is the SO4 chemistry subroutine from Mian Chin's GOCART model, modified for the GEOS-CHEM model. Now also modified to account for production of crystalline and aqueous sulfur tracers. (rjp, bdf, cas, bmy, 5/31/00, 5/23/06)

INTERFACE:

```

SUBROUTINE CHEM_SO4( am_I_Root, Input_Opt, State_Met,
&                   State_Chm, State_Diag, RC           )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),   INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(MetState),   INTENT(IN)  :: State_Met    ! Meteorology State object
TYPE(DgnState),   INTENT(INOUT) :: State_Diag  ! Diagnostics State object

```

INPUT/OUTPUT PARAMETERS:

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: RC ! Success or failure?

REMARKS:

The only production is from SO2 oxidation (save in CHEM_SO2). Dry deposition is now handled in mixing_mod.F90, so we can must add the production from SO2 into the SO4 tracers.

REVISION HISTORY:

- (1) Now reference AD from "dao_mod.f". Added parallel DO-loops.
Updated comments, cosmetic changes. (rjp, bdf, bmy, 9/16/02)
 - (2) Now replace DXYP(JREF)*1d4 with routine GET_AREA_CM2 of "grid_mod.f"
Now use function GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
 - (3) Now reference PBLFRAC from "drydep_mod.f". Now apply dry deposition
to the entire PBL. (rjp, bmy, 8/1/03)
 - (4) Now use ND44_TMP array to store vertical levels of drydep flux, then
sum into AD44 array. This preents numerical differences when using
multiple processors. (bmy, 3/24/04)
 - (5) Now use parallel DO-loop to zero ND44_TMP (bmy, 4/14/04)
 - (6) Now reference STT & TCVV from "tracer_mod.f" (bmy, 7/20/04)
 - (7) Now references LCRYST from "logical_mod.f". Modified for crystalline
and aqueous sulfate2 tracers: AS, AHS, LET, SO4aq. Also changed name
of ND44_TMP to T44 to save space. (cas, bmy, 12/21/04)
 - (8) Replace PBLFRAC from "drydep_mod.f" with GET_FRAC_UNDER_PBLTOP from
"pbl_mix_mod.f" (bmy, 2/22/05)
 - (9) Now remove reference to CMN, it's obsolete. Now reference
ITS_IN_THE_STRAT from "tropopause_mod.f" (bmy, 8/22/05)
 - (10) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
 - (11) Rearrange error check to avoid SEG FAULTS (bmy, 5/23/06)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
- 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
running with the traditional driver main.F
- 31 Jul 2012 - R. Yantosca - Now loop from 1..LLPAR for GIGC compatibility
- 31 Jul 2012 - R. Yantosca - Declare temp drydep arrays w/ LLPAR (not LLTROP)
- 14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
- 05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNL PBL
- 19 Mar 2013 - R. Yantosca - Now copy Input_Opt%TCVV(1:N_TRACERS) and
Input_Opt%XNUMOL(1:N_TRACERS) -- avoid OOB errs
- 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list

12 Sep 2013 - M. Sulprizio- Now include references to dust-sulfate chemistry
(tdf 04/07/08)

12 Jun 2015 - R. Yantosca - Now remove orphaned ND44 variables

12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing_mod.F90,
so we can greatly collapse this code

23 Sep 2015 - R. Yantosca - Remove references to DRYSO4 and DRYSO4s

24 Jun 2016 - R. Yantosca - Now return if id_SO4<0 or id_SO4s<0 (not == 0)

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
species ID from State_Chm%Map_Advect.

07 Dec 2017 - R. Yantosca - Now accept State_DIag as an argument

8.5.19 chem_so4_aq

Subroutine CHEM_SO4_AQ takes the SO4 produced via aqueous chemistry of SO2 and distribute onto the size-resolved aerosol number and sulfate mass as a part of the TOMAS aerosol microphysics module (win, 1/25/10)

INTERFACE:

```

SUBROUTINE CHEM_SO4_AQ( am_I_Root, Input_Opt,
&                      State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD
USE Input_Opt_Mod,      ONLY : OptINput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TOMAS_MOD,         ONLY : AQOXID, GETACTBIN
USE UnitConv_Mod,      ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(MetState), INTENT(IN)  :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

NOTE: This subroutine is ignored unless we compile for TOMAS microphysics.

REVISION HISTORY:

- (1) As of now the SO4 produced via heterogeneous reaction on the 2-mode seasalt is not include in this treatment (win, 7/23/07)
- (2) Change a fixed kmin = 8 (corresponding to the assumed activation dia. of 55nm to be varying with current chemical composition. Take average of the activating bin for LS and CONV rains. (win, 9/25/07)
- 16 Feb 2012 - R. Yantosca - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 31 May 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm and RC arguments
- 31 May 2013 - R. Yantosca - Now pass State_Chm to TOMAS routines
- 30 Sep 2015 - E. Lundgren - Now use UNITCONV_MOD conversion routine
- 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

8.5.20 chem_msa

Subroutine CHEM_MSA is the SO4 chemistry subroutine from Mian Chin's GOCART model, modified for the GEOS-CHEM model. (rjp, bdf, bmy, 5/31/00, 10/25/05)

INTERFACE:

```

SUBROUTINE CHEM_MSA( am_I_Root, Input_Opt,
&                   State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState), INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)    :: RC            ! Success or failure?

```

REMARKS:

The only production is from DMS oxidation (saved in CHEM_DMS).
 Dry deposition is now treaded in mixing_mod.F90.

REVISION HISTORY:

- (1) Now reference AD from "dao_mod.f". Added parallel DO-loops.
Updated comments, cosmetic changes. (rjp, bmy, bdf, 9/16/02)
- (2) Now replace DXYP(JREF)*1d4 with routine GET_AREA_CM2 of "grid_mod.f"
Now use function GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
- (3) Now reference PBLFRAC from "drydep_mod.f". Now apply dry deposition
to the entire PBL. (rjp, bmy, 8/1/03)
- (4) Now use ND44_TMP array to store vertical levels of drydep flux, then
sum into AD44 array. This prevents numerical differences when using
multiple processors. (bmy, 3/24/04)
- (5) Now use parallel DO-loop to zero ND44_TMP (bmy, 4/14/04)
- (6) Now references STT & TCVV from "tracer_mod.f" (bmy, 7/20/04)
- (7) Replace PBLFRAC from "drydep_mod.f" with GET_FRAC_UNDER_PBLTOP from
"pbl_mix_mod.f". Also reference GET_PBL_MAX_L from "pbl_mix_mod.f"
Vertical DO-loops can run up to PBL_MAX and not LLTROP. Also
remove reference to header file CMN. (bmy, 2/22/05)
- (8) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
- (9) Change loop back to over entire troposphere to correctly add production
of MSA (PMSA_dms) to the MSA tracer array.
Added reference USE_TROPOPAUSE_MOD, ONLY : ITS_IN_THE_STRAT
as a precaution. (pjh, 8/19/2009)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
- 31 Jul 2012 - R. Yantosca - Now loop from 1..LLPAR for GIGC compatibility
- 31 Jul 2012 - R. Yantosca - Declare temp drydep arrays w/ LLPAR (not LLTROP)
- 14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
- 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
- 05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNL PBL
- 19 Mar 2013 - R. Yantosca - Now copy Input_Opt%TCVV(1:N_TRACERS) and
Input_Opt%XNUMOL(1:N_TRACERS) -- avoid OOB errs
- 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
- 12 Jun 2015 - R. Yantosca - Now remove orphaned ND44 variables
- 12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing_mod.F90,
so we can greatly collapse this code
- 23 Sep 2015 - R. Yantosca - Remove references to obsolete DRYMSA flag
- 24 Jun 2016 - R. Yantosca - Now return if id_MSA < 0 (not == 0)
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
species ID from State_Chm%Map_Advect.

8.5.21 chem_nit

Subroutine CHEM_NIT removes SULFUR NITRATES (NIT) from the surface via dry deposition. (rjp, bdf, bmy, 1/2/02, 5/23/06)

INTERFACE:

```

SUBROUTINE CHEM_NIT( am_I_Root, Input_Opt,
&                   State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)      :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)      :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)    :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC          ! Success or failure?

```

REMARKS:

Dry deposition is now applied in mixing_mod.F90. Therefore we can

REMARKS:

Reaction List:

```

=====
(1 ) NIT = NIT_0 * EXP( -dt )  where d = dry deposition rate [s-1]

```

REVISION HISTORY:

- (1) Now reference AD from "dao_mod.f". Added parallel DO-loops.
Updated comments, cosmetic changes. (rjp, bmy, bdf, 9/20/02)
- (2) Now replace DXYP(J+J0)*1d4 with routine GET_AREA_CM2 from "grid_mod.f".
Now use function GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
- (3) Now reference PBLFRAC from "drydep_mod.f". Now apply dry deposition
to the entire PBL. Added L and FREQ variables. Recode to avoid
underflow from EXP(). (rjp, bmy, 8/1/03)
- (4) Now use ND44_TMP array to store vertical levels of drydep flux, then
sum into AD44 array. This prevents numerical differences when using
multiple processors. (bmy, 3/24/04)
- (5) Now use parallel DO-loop to zero ND44_TMP (bmy, 4/14/04)
- (6) Now reference STT & TCVV from "tracer_mod.f". Also remove reference
to CMN, it's not needed anymore. (bmy, 7/20/04)

(7) Replace PBLFRAC from "drydep_mod.f" with GET_FRAC_UNDER_PBLTOP from "pbl_mix_mod.f". Also reference GET_PBL_MAX_L from "pbl_mix_mod.f" Vertical DO-loops can run up to PBL_MAX and not LLTROP. (bmy, 2/22/05)

(8) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)

(9) Rearrange error check to avoid SEG FAULTS (bmy, 5/23/06)

22 Dec 2011 - M. Payer - Added ProTeX headers

01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90

31 Jul 2012 - R. Yantosca - Now loop from 1..LLPAR for GIGC compatibility

31 Jul 2012 - R. Yantosca - Declare temp drydep arrays w/ LLPAR (not LLTROP)

09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object

05 Mar 2013 - R. Yantosca - Now use Input_Opt%LNLPLBL

13 Mar 2013 - R. Yantosca - Bug fix: make sure we pass values to the SOIL_DRYDEP routine even when ND44 is off (this happens when LNLPLBL = F)

19 Mar 2013 - R. Yantosca - Now copy Input_Opt%TCVV(1:N_TRACERS) and Input_Opt%XNUMOL(1:N_TRACERS) -- avoid OOB errs

25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list

12 Sep 2013 - M. Sulprizio- Now include references to nitrate on dust (tdf, 04/07/08)

23 Sep 2013 - M. Sulprizio- Bug fix: Skip stratospheric boxes to prevent out-of-bounds error in PNIT_DUST

12 Jun 2015 - R. Yantosca - Now remove orphaned ND44 variables

12 Jun 2015 - R. Yantosca - Drydep is now handled in mixing_mod.F90, so we can greatly collapse this code

23 Sep 2015 - R. Yantosca - Remove reference to DRYNIT, DRYNITS

24 Jun 2016 - R. Yantosca - Now return if id_NIT<0 or id_NITs<0 (not == 0)

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.

8.5.22 sulfate_pbl_mix

Subroutine SULFATE_PBL_MIX partitions the total anthro sulfate emissions thru the entire boundary layer. Emissions above the PBL are not used, and left in their level, regardless of the mixing scheme. For non-local mixing scheme, all emissions within the PBL are put in the first level.

INTERFACE:

```

SUBROUTINE SULFATE_PBL_MIX ( EMISS, SULFATE, FRAC_OF_PBL,
$                               PBL_TOP, IS_LOCAL )

```

USES:

```

USE ERROR_MOD, ONLY : ERROR_STOP
IMPLICIT NONE

```

INPUT PARAMETERS:

```

INTEGER, INTENT(IN)      :: PBL_TOP ! Top level of boundary layer
LOGICAL, INTENT(IN)      :: IS_LOCAL ! mixing scheme
REAL(fp), INTENT(IN)     :: FRAC_OF_PBL(:) !
REAL(fp), INTENT(IN)     :: EMISS(:)

```

OUTPUT PARAMETERS:

```

REAL(fp), INTENT(INOUT) :: SULFATE(:) ! partitioned emissions

```

REVISION HISTORY:

```

27 Oct 2009 - P. Le Sager - initial
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

```

8.5.23 get_oh

Function GET_OH returns OH from State_Chm (coupled runs) or monthly mean OH (for offline runs). Imposes a diurnal variation on OH for offline simulations. (bmy, 12/16/02, 7/20/04)

INTERFACE:

```

FUNCTION GET_OH( I, J, L, Input_Opt, State_Chm, State_Met )
&      RESULT( OH_MOLEC_CM3 )

```

USES:

```

USE CMN_SIZE_MOD
USE ERROR_MOD,          ONLY : ERROR_STOP
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
USE State_Met_Mod,      ONLY : MetState
USE TIME_MOD,           ONLY : GET_TS_CHEM

```

INPUT PARAMETERS:

```

INTEGER, INTENT(IN) :: I, J, L      ! Lon, lat, level indices
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object

```

REVISION HISTORY:

```

(1) We assume SETTRACE has been called to define IDOH (bmy, 11/1/02)
(2) Now use function GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
(3) Now reference ITS_A_FULLCHEM_SIM, ITS_AN_AEROSOL_SIM from
    "tracer_mod.f". Also replace IJSURF w/ an analytic function.
    (bmy, 7/20/04)
22 Dec 2011 - M. Payer - Added ProTeX headers
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS

```

28 Nov 2012 - R. Yantosca - Add State_Met to argument list
 04 Mar 2013 - R. Yantosca - Now pass Input_Opt%ITS_A_FULLLCHEM_SIM and
 Input_Opt%ITS_AN_AEROSOL_SIM
 18 Sep 2014 - M. Sulprizio- Now get OH for offline aerosol sim from HEMCO
 22 Dec 2015 - M. Sulprizio- Remove references to CSPEC and JLOOP. We now get
 species concentrations from State_Chm%Species.

8.5.24 get_no3

Function GET_NO3 returns NO3 from State_Chm(coupled runs) or monthly mean OH (for offline runs). For offline runs, the concentration of NO3 is set to zero during the day. (rjp, bmy, 12/16/02)

INTERFACE:

```
FUNCTION GET_NO3( I, J, L, Input_Opt, State_Chm, State_Met )
&          RESULT( NO3_MOLEC_CM3 )
```

USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD,          ONLY : ERROR_STOP
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
USE State_Met_Mod,      ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)  :: I, J, L      ! Lon, lat, vertical level
TYPE(OptInput),  INTENT(IN)  :: Input_Opt     ! Input Options object
TYPE(MetState),  INTENT(IN)  :: State_Met     ! Meteorology State object
TYPE(ChmState),  INTENT(IN)  :: State_Chm     ! Chemistry State object
```

REVISION HISTORY:

(1) Now references ERROR_STOP from "error_mod.f". We also assume that SETTRACE has been called to define IDNO3. Now also set NO3 to zero during the day. (rjp, bmy, 12/16/02)
 (2) Now reference ITS_A_FULLLCHEM_SIM and ITS_AN_AEROSOL_SIM from "tracer_mod.f". Also remove reference to CMN. Also replace IJSURF with an analytic function. (bmy, 7/20/04)
 22 Dec 2011 - M. Payer - Added ProTeX headers
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
 04 Mar 2013 - R. Yantosca - Now pass Input_Opt%ITS_A_FULLLCHEM_SIM and
 Input_Opt%ITS_AN_AEROSOL_SIM
 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
 18 Sep 2014 - M. Sulprizio- Now get NO3 for offline aerosol sims from HEMCO
 22 Dec 2015 - M. Sulprizio- Remove references to CSPEC and JLOOP. We now get
 species concentrations from State_Chm%Species.

8.5.25 get_o3

Function GET_O3 returns monthly mean O3 for offline sulfate aerosol simulations. (bmy, 12/16/02, 10/25/05)

INTERFACE:

```
FUNCTION GET_O3( I, J, L, Input_Opt, State_Chm, State_Met )
&                RESULT( O3_VV )
```

USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD,          ONLY : ERROR_STOP
USE Input_Opt_Mod,     ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)  :: I, J, L      ! Lon, lat, vertical level
TYPE(OptInput),  INTENT(IN)  :: Input_Opt
TYPE(ChmState),  INTENT(IN)  :: State_Chm     ! Chemistry State object
TYPE(MetState),  INTENT(IN)  :: State_Met     ! Meteorology State object
```

REVISION HISTORY:

```
(1 ) We assume SETTRACE has been called to define IDO3. (bmy, 12/16/02)
(2 ) Now reference inquiry functions from "tracer_mod.f" (bmy, 7/20/04)
(3 ) Now remove reference to CMN, it's obsolete. (bmy, 8/22/05)
(4 ) Now references XNUMOLAIR from "tracer_mod.f" (bmy, 10/25/05)
22 Dec 2011 - M. Payer      - Added ProTeX headers
09 Nov 2012 - M. Payer      - Replaced all met field arrays with State_Met
                              derived type object
04 Mar 2013 - R. Yantosca - Now pass Input_Opt%ITS_A_FULLCHEM_SIM and
                              Input_Opt%ITS_AN_AEROSOL_SIM
06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)
24 Mar 2015 - E. Lundgren - Replace dependency on tracer_mod with
                              CMN_GTCM_MOD to get XNUMOLAIR.
22 Dec 2015 - M. Sulprizio - Remove references to CSPEC and JLOOP. We now get
                              species concentrations from State_Chm%Species.
```

8.5.26 ohno3time

Subroutine OHNO3TIME computes the sum of cosine of the solar zenith angle over a 24 hour day, as well as the total length of daylight. This is needed to scale the offline OH and NO3 concentrations. (rjp, bmy, 12/16/02, 3/30/04)

INTERFACE:

SUBROUTINE OHNO3TIME

USES:

```
USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_XMID,    GET_YMID_R
USE TIME_MOD,    ONLY : GET_NHMSb,   GET_ELAPSED_SEC
USE TIME_MOD,    ONLY : GET_TS_CHEM,  GET_DAY_OF_YEAR, GET_GMT
```

REVISION HISTORY:

- (1) Copy code from COSSZA directly for now, so that we don't get NaN values. Figure this out later (rjp, bmy, 1/10/03)
- (2) Now replace XMID(I) with routine GET_XMID from "grid_mod.f".
Now replace RLAT(J) with routine GET_YMID_R from "grid_mod.f".
Removed NTIME, NHMSb from the arg list. Now use GET_NHMSb,
GET_ELAPSED_SEC, GET_TS_CHEM, GET_DAY_OF_YEAR, GET_GMT from
"time_mod.f". (bmy, 3/27/03)
- (3) Now store the peak SUNCOS value for each surface grid box (I,J) in
the COSZM array. (rjp, bmy, 3/30/04)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 16 May 2016 - M. Sulprizio- Remove IJLOOP and change SUNTMP array dimensions
from (MAXIJ) to (IIPAR,JJPAR)

8.5.27 get_alk

Subroutine GET_ALK returns the seasalt alkalinity emitted at each timestep to sulfate_mod.f for chemistry on seasalt aerosols. (bec, 12/7/04, 11/23/09)

INTERFACE:

```
SUBROUTINE GET_ALK( am_I_Root, I,J,L, ALK1, ALK2, Kt1, Kt2,
&                  Kt1N, Kt2N, Input_Opt, State_Met, State_Chm,
&                  RC)
```

USES:

```
USE CMN_SIZE_MOD,    ONLY : NDUST
USE ErrCode_Mod
USE ERROR_MOD,      ONLY : IT_IS_NAN, ERROR_STOP
USE HCO_INTERFACE_MOD, ONLY : GetHcoVal
(jas, 8/4/15) calculate kt using sea salt surface area concentration
USE CMN_SIZE_MOD,    ONLY : NDUST
USE Input_Opt_Mod,   ONLY : OptInput
USE PBL_MIX_MOD,     ONLY : GET_FRAC_OF_PBL, GET_PBL_TOP_L
USE State_Chm_Mod,   ONLY : ChmState
USE State_Met_Mod,   ONLY : MetState
```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Root CPU?
INTEGER,          INTENT(IN)    :: I, J, L      ! Lon-lat-alt indices
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

OUTPUT PARAMETERS:

```

REAL(fp),        INTENT(OUT)    :: ALK1, ALK2  ! [kg]
REAL(fp),        INTENT(OUT)    :: Kt1, Kt2, Kt1N, Kt2N ! [s-1]

```

INPUT/OUTPUT PARAMETERS:

```

INTEGER,          INTENT(INOUT) :: RC
TYPE(ChmState),  INTENT(INOUT) :: State_Chm    ! Chemistry State object

```

REVISION HISTORY:

- (1) Becky Alexander says we can remove AREA1, AREA2 (bec, bmy, 9/5/06)
- (2) Bug fix to remove a double-substitution. Replace code lines for
TERM{123}A, TERM{123}B, TERM{123}AN, TERM{123}BN. (bec, bmy, 7/18/08)
- (3) Updated hygroscopic growth parameters (bec, bmy, 11/23/09)
- (4) Fixed bug when calculating rate constants for uptake of SO2 and HNO3
(kt). The bug caused kt to be too small and prevented acidification
of SSA. (jas, 8/4/15)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
- 25 Jun 2014 - R. Yantosca - Now accept Input_Opt via the arg list
- 25 Jun 2014 - R. Yantosca - Removed references to tracer_mod.F
- 02 Nov 2014 - C. Keller - Now get alkalinity and number density from HEMCO
- 04 Nov 2014 - C. Keller - Moved from seasalt_mod.F
- 12 Nov 2014 - C. Keller - Now weight # dens. and alk. by fraction of PBL
- 12 May 2016 - M. Sulprizio- Remove 1D arrays that depend on JLOOP. ERADIUS,
TAREA, are now pointers that point to 3D fields
fields in State_Chm.

8.5.28 init_sulfate

Subroutine INIT_SULFATE initializes and zeros all allocatable arrays declared in "sulfate_mod.f" (bmy, 6/2/00, 10/15/09)

INTERFACE:

```

SUBROUTINE INIT_SULFATE( am_I_Root, Input_Opt,
&                        State_Chm, State_Diag, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod

```

```

USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
TYPE(ChmState),  INTENT(IN)  :: State_Chm     ! Chemistry State object
TYPE(DgnState),  INTENT(IN)  :: State_Diag    ! Diagnostics State object

```

INPUT/OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT) :: RC           ! Success or failure?

```

REVISION HISTORY:

- (1) Only allocate some arrays for the standalone simulation (NSRCX==10).
Also reference NSRCX from "CMN". Now eferences routine ALLOC_ERR
from "error_mod.f" ((rjp, bdf, bmy, 10/15/02)
 - (2) Now also allocate the IJSURF array to keep the 1-D grid box indices
for SUNCOS (for both coupled & offline runs). Now allocate PH2O2m
and O3m for offline runs. Also allocate ES02_bf (bmy, 1/16/03)
 - (3) Now allocate ENH3_na array (rjp, bmy, 3/23/03)
 - (4) Now allocate COSZM for offline runs (bmy, 3/30/04)
 - (5) Now allocate ES02_sh array (bec, bmy, 5/20/04)
 - (6) Now allocates ITS_AN_AEROSOL_SIM from "tracer_mod.f". Now remove
IJSURF (bmy, 7/20/04)
 - (7) Now locate species in the DEPSAV array here instead of in CHEMSULFATE.
Now reference LDRYD from "logical_mod.f". Updated for AS, AHS, LET,
SO4aq, NH4aq. (bmy, 1/6/06)
 - (8) Now allocates PSO4_ss, PNITs (bec, bmy, 4/13/05)
 - (9) Initialize drydep flags outside of IF block (bmy, 5/23/06)
 - (10) Now redimension EEV & NEV arrays for new SO2 volcanic emissions
inventory. Deleted obsolete arrays from older SO2 volcanic
emissions inventory. (jaf, bmy, 10/15/09)
 - (11) Now allocate PSO4_SO2AQ (win, 1/25/10)
 - 22 Dec 2011 - M. Payer - Added ProTeX headers
 - 04 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC arguments
 - 05 Mar 2013 - R. Yantosca - Now use Input_Opt%ITS_AN_AEROSOL_SIM
 - 30 May 2013 - S. Farina - Allocate PSO4_SO2AQ for TOMAS
 - 12 Sep 2013 - M. Sulprizio- Add PSO4_dust and PNIT_dust for acid update on
dust aerosols (T.D. Fairlie)
 - 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
 - 22 May 2015 - R. Yantosca - Remove variables made obsolete by HEMCO
 - 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 - 21 Jun 2016 - R. Yantosca - Reorder the Ind_ declaration statements a bit
 - 24 Jun 2016 - R. Yantosca - Add error checks for dust uptake species
 - 02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
-

8.5.29 cleanup_sulfate

Subroutine CLEANUP_SULFATE deallocates all previously allocated arrays for sulfate emissions – call at the end of the run (bmy, 6/1/00, 10/15/09)

INTERFACE:

```
SUBROUTINE CLEANUP_SULFATE()
```

REVISION HISTORY:

- (1) Now also deallocates IJSURF. (bmy, 11/12/02)
- (2) Now also deallocates ENH3_na (rjp, bmy, 3/23/03)
- (3) Now also deallocates COSZM (rjp, bmy, 3/30/04)
- (4) Now also deallocates ES04_sh (bec, bmy, 5/20/04)
- (5) Now remove IJSURF (bmy, 7/20/04)
- (6) Bug fix: now deallocate PS04_ss, PNITs (bmy, 5/3/06)
- (7) Deleted obsolete arrays from older SO2 volcanic emissions inventory (jaf, bmy, 10/15/09)
- (8) Deallocate PS04_SO2AQ (win, 1/25/10)
- 22 Dec 2011 - M. Payer - Added ProTeX headers
- 30 May 2013 - S. Farina - Deallocate PS04_SO2AQ for TOMAS
- 04 Mar 2015 - R. Yantosca - Remove EEV, NEV. Volcano eruptions are now handled via HEMCO.
- 22 May 2015 - R. Yantosca - Remove variables made obsolete by HEMCO

8.6 Fortran: Module Interface isoropiaii_mod.F

Module ISOROPIAII_MOD contains the routines that provide the interface between ISORROPIA II and GEOS-Chem.

The actual ISORROPIA II code which performs Na-SO4-NH3-NO3-Cl-(Ca-K-Mg) aerosol thermodynamic equilibrium is in *isoropiaIIcode.f*.

INTERFACE:

```
MODULE ISOROPIAII_MOD
```

USES:

```
USE HCO_ERROR_MOD      ! For real precisions (hp)
USE PRECISION_MOD      ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: CLEANUP_ISOROPIAII
PUBLIC  :: DO_ISOROPIAII
PUBLIC  :: GET_GNO3
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: GET_HNO3
PRIVATE :: INIT_ISOROPIAII
PRIVATE :: SAFELOG10
PRIVATE :: SET_HNO3
```

REMARKS:

Original Author:

```
*** COPYRIGHT 1996-2006, UNIVERSITY OF MIAMI, CARNEGIE MELLON UNIVERSITY,
*** GEORGIA INSTITUTE OF TECHNOLOGY
*** WRITTEN BY ATHANASIOS NENES
*** UPDATED BY CHRISTOS FOUNTOUKIS
```

Original v1.3 isoropia implementation into GEOS-Chem by
Becky Alexander and Bob Yantosca (bec, bmy, 4/12/05, 11/2/05)

For Ca,K,Mg = 0, ISOROPIA II performs exactly like ISOROPIAv1.7
Ca, K, Mg, Na from dust is not currently considered

To implement ISOROPIA II into GEOS-Chem:

- * cleanup_isoropiaII needs to be called from cleanup.f
- * DO_ISOROPIA needs to be replaced with DO_ISOROPIAII in chemistry_mod.f
- * Change ISOROPIA to ISOROPIAII in sulfate_mod.f
- * add isoropiaII_mod.f, isoropiaIIcode.f, and irspia.inc to Makefile

ISOROPIA II implementation notes by Havala O.T. Pye:

- (1) The original isoropia code from T.Nenes is left as unmodified as possible. Original isoropia code can be found in isoropiaIIcode.f and common blocks can be found in isrpia.inc. For future upgrades to isoropia, replace isrpia.inc and isoropiaIIcode.f with the new version of isoropia and modify the call to ISOROPIA in this module. Please let the original author know of any changes made to ISOROPIA.
- (2) As of Nov 2007, routines using non-zero Ca, K, and Mg do not always conserve mass. Ca, K, and Mg are set to zero.

NOTE: ISORROPIA is Greek for "equilibrium", in case you were wondering.

REVISION HISTORY:

```
06 Jul 2007 - H. O. T. Pye - Initial version
29 Jan 2010 - R. Yantosca - Added ProTeX headers
21 Apr 2010 - R. Yantosca - Bug fix in DO_ISOROPIAII for offline aerosol
16 Feb 2011 - R. Yantosca - Add modifications for APM from G. Luo
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
23 Sep 2014 - M. Sulprizio - Get global HNO3 for offline aerosol simulation
                        from HEMCO
14 Nov 2014 - M. Yannetti - Added PRECISION_MOD
23 Jun 2016 - R. Yantosca - Remove references to APM code; it is no longer
                        compatible with the FlexChem implementation
```

8.6.1 do.isoropiaii

Subroutine DO_ISOROPIAII is the interface between the GEOS-Chem model and the aerosol thermodynamical equilibrium routine ISORROPIA II.

INTERFACE:

```

SUBROUTINE DO_ISOROPIAII( am_I_Root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, RC          )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,          ONLY : DEBUG_MSG
USE ERROR_MOD,          ONLY : ERROR_STOP
USE ERROR_MOD,          ONLY : SAFE_DIV
USE HCO_INTERFACE_MOD,  ONLY : HcoState
USE HCO_EMISLIST_MOD,   ONLY : HCO_GetPtr
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
USE State_Diag_Mod,     ONLY : DgnState
USE State_Chm_Mod,      ONLY : Ind_
USE State_Met_Mod,      ONLY : MetState
USE TIME_MOD,           ONLY : GET_MONTH
USE TIME_MOD,           ONLY : ITS_A_NEW_MONTH
USE IsorropiaII_Main_Mod, ONLY : Isoropia
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD,       ONLY : LD73,   ND73
USE DIAG_MOD,          ONLY : AD73
#endif

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Is this the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

Original isoropia v1.3 implmentation: (rjp, bec, bmy, 12/17/01, 8/22/05)

REVISION HISTORY:

24 Aug 2007 - H. O. T. Pye - Initial version, in ISORROPIA II
 18 Dec 2009 - H. O. T. Pye - Added division checks
 29 Jan 2010 - R. Yantosca - Added ProTeX headers
 21 Apr 2010 - E. Sofen - Prevent out-of-bounds errors for offline
 aerosol simulations where HNO3 is undefined
 23 Jul 2010 - R. Yantosca - Bug fix: corrected typo in ND42 diag section
 30 Jul 2012 - R. Yantosca - Now accept am_I_Root as an argument when
 running with the traditional driver main.F
 31 Jul 2012 - R. Yantosca - Now loop from 1..LLPAR for GIGC compatibilit
 14 Nov 2012 - R. Yantosca - Add am_I_Root, Input_Opt, RC as arguments
 15 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
 25 Mar 2013 - M. Payer - Now pass State_Chm object via the arg list
 23 Sep 2014 - M. Sulprizio - Now get HNO3 for offline aerosol sim from HEMCO
 16 Jun 2016 - K. Travis - Now define species ID's with the Ind_ function
 17 Jun 2016 - R. Yantosca - Now only define species ID's on the first call;
 we don't want to do it on each loop iteration
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
 10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
 25 Jan 2017 - S. Eastham - Can skip calling ISORROPIA if P and T are not
 in the proper range to yield a valid solution.
 (Activate this with the #ifdef block above.)
 10 Mar 2017 - C. Keller - Add print out statement after initialization.
 02 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

8.6.2 safelog10

Calculates the LOG (base 10) of a number X. Returns a minimum value if X is too small,
 in order to avoid NaN or Infinity problems.

INTERFACE:

FUNCTION SAFELOG10(X) RESULT (SAFLOG)

INPUT PARAMETERS:

REAL(fp), INTENT(IN) :: X ! Argument for LOG10 function

RETURN VALUE:

REAL(fp) :: SAFLOG ! LOG10 output --

REVISION HISTORY:

11 Aug 2009 - H. O. T. Pye - Initial version, in ISORROPIA II
 29 Jan 2010 - R. Yantosca - Added ProTeX headers

8.6.3 get_hno3

Subroutine GET_HNO3 allows the HNO3 concentrations to evolve with time, but relaxes back to the monthly mean concentrations every 3 hours.

INTERFACE:

```
FUNCTION GET_HNO3( I, J, L, State_Met ) RESULT ( HNO3_UGM3 )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE PhysConstants,     ONLY : AIRMW
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : GET_ELAPSED_SEC
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)  :: I           ! GEOS-Chem longitude index
INTEGER,      INTENT(IN)  :: J           ! GEOS-Chem latitude index
INTEGER,      INTENT(IN)  :: L           ! GEOS-Chem level index
TYPE(MetState), INTENT(IN) :: State_Met  ! Meteorology State object
```

REVISION HISTORY:

```
16 Dec 2002 - R. Yantosca - Initial version, in ISORROPIA I
24 Mar 2003 - R. Yantosca - Now use function GET_ELAPSED_MIN() from the
                           new "time_mod.f" to get the elapsed minutes
                           since the start of run.
06 Jul 2007 - H. O. T. Pye - Initial version, in ISORROPIA II
29 Jan 2010 - R. Yantosca - Added ProTeX headers
23 Sep 2014 - M. Sulprizio - Now get HNO3 for offline aerosol sims from HEMCO
06 Jan 2016 - E. Lundgren - Use global physical parameters
06 Feb 2018 - E. Lundgren - Change GET_ELAPSED_MIN to GET_ELAPSED_SEC to
                           match new timestep unit of seconds
```

8.6.4 set_hno3

Subroutine SET_HNO3 stores the modified HNO3 value back into the HNO3_sav array for the next timestep.

INTERFACE:

```
SUBROUTINE SET_HNO3( I, J, L, HNO3_UGM3 )
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I           ! GEOS-Chem longitude index
INTEGER, INTENT(IN) :: J           ! GEOS-Chem longitude index
INTEGER, INTENT(IN) :: L           ! GEOS-Chem longitude index
REAL(f8), INTENT(IN) :: HNO3_UGM3 ! HNO3 concentration [ug/m3]
```


REVISION HISTORY:

16 Dec 2002 - R. Yantosca - Initial version, in ISORROPIA I
 06 Jul 2007 - H. O. T. Pye - Initial version, in ISORROPIA II
 29 Jan 2010 - R. Yantosca - Added ProTeX headers
 11 Jan 2015 - M. Yannetti - Some precision edits, for safety.
 19 Sep 2016 - R. Yantosca - Bug fix: #ifdef's should be lower case
 19 Sep 2016 - R. Yantosca - Bug fix: elif should be else

8.6.5 get_gno3

Function GET_GNO3 returns the gas-phase HNO₃ [v/v] for calculation of sea-salt chemistry in sulfate_mod (SEASALT_CHEM).

INTERFACE:

```
SUBROUTINE GET_GNO3( I, J, L, HNO3_kg, State_Met )
```

USES:

```
USE State_Met_Mod, ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)   :: I           ! GEOS-Chem longitude index
INTEGER,          INTENT(IN)   :: J           ! GEOS-Chem latitude index
INTEGER,          INTENT(IN)   :: L           ! GEOS-Chem level index
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met  ! Meteorology State object
```

OUTPUT PARAMETERS:

```
REAL(fp),          INTENT(OUT)  :: HNO3_kg    ! Gas-phase HNO3 [kg]
```

REVISION HISTORY:

15 Apr 2005 - B. Alexander - Initial version, in ISORROPIA I
 06 Jul 2007 - H. O. T. Pye - Initial version, in ISORROPIA II
 29 Jan 2010 - R. Yantosca - Added ProTeX headers
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object

8.6.6 init_isoropiaII

Subroutine INIT_ISORROPIAII initializes all module arrays.

INTERFACE:

```
SUBROUTINE INIT_ISOROPIAII
```

USES:

```
USE CMN_SIZE_MOD      ! Size parameters
USE ERROR_MOD,       ONLY : ALLOC_ERR
```

REVISION HISTORY:

```
06 Jul 2007 - H. O. T. Pye - Initial version
29 Jan 2010 - R. Yantosca - Added ProTeX headers
31 Jul 2012 - R. Yantosca - Declare diag arrays w/ LLPAR (not LLTROP)
```

8.6.7 cleanup_isoropiaII

Subroutine CLEANUP_ISOROPIAII deallocates all module arrays.

INTERFACE:

```
SUBROUTINE CLEANUP_ISOROPIAII
```

REVISION HISTORY:

```
06 Jul 2007 - H. O. T. Pye - Initial version
29 Jan 2010 - R. Yantosca - Added ProTeX headers
```

9 Meteorological field modules

These modules contain routines to read archived meteorology into GEOS-Chem.

9.1 Fortran: Module Interface dao_mod.F

Module DAO_MOD contains both arrays that hold DAO met fields, as well as subroutines that compute, interpolate, or otherwise process DAO met field data.

INTERFACE:

```
MODULE DAO_MOD
```

USES:

```
USE CMN_SIZE_MOD      ! Size parameters
USE PhysConstants     ! Physical constants
USE PRECISION_MOD     ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```

PUBLIC  :: AVGPOLE
PUBLIC  :: AIRQNT
PUBLIC  :: COPY_I3_I6_FIELDS
PUBLIC  :: GET_COSINE_SZA
PUBLIC  :: GET_OBK
PUBLIC  :: INTERP
PUBLIC  :: IS_LAND
PUBLIC  :: IS_WATER
PUBLIC  :: IS_ICE
PUBLIC  :: IS_NEAR
PUBLIC  :: SET_DRY_SURFACE_PRESSURE
#if defined( ESMF_ ) || defined( EXTERNAL_GRID )
PUBLIC  :: GIGC_Cap_Tropopause_Pr
#endif

```

REVISION HISTORY:

- 26 Jun 2010 - R. Yantosca - Initial version
- (1) Added sea level pressure (SLP) met field for GEOS-3 (bmy, 10/10/00)
- (2) Moved MAKE_QQ to "wetscav_mod.f" (bmy, 10/12/00)
- (3) Now get LWI from ALBEDO for GEOS-3 in routines IS_LAND and IS_WATER (bmy, 4/4/01)
- (4) Define OPTDEP allocatable array for GEOS-3 -- this is the grid box optical depth and is now stored as a met field (bmy, 8/15/01)
- (5) Updated comments (bmy, 9/4/01)
- (6) Now make AVGW an allocatable module array. Also replace obsolete parameters {IJL}GCMPAR with IIPAR,JJPAR,LLPAR. (bmy, 9/27/01)
- (7) Remove arguments LMAKEPW, PW, and LM from AIRQNT (bmy, 10/3/01)
- (8) Remove obsolete code from 9/01 (bmy, 10/23/01)
- (9) Bug fixes in IS_LAND and IS_WATER. Also cosmetic changes and updated some comments. (mje, bmy, 1/9/02)
- (10) Now add additional array PSC2 in order to pass to TPCORE, which will fix the mixing ratio bug. Compute PSC2 in subroutine INTERP. Now bundle "convert_units.f" into "dao_mod.f". Updated comments. (bmy, 3/27/02)
- (11) Updated comments (bmy, 5/28/02)
- (12) Replaced all instances of IM with IIPAR and JM with JJPAR, in order to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (13) Eliminated PS, PSC arrays. Now reference "pressure_mod.f". Also updated AIRQNT for hybrid grid. Added routine MAKE_RH to this module. (dsa, bdf, bmy, 8/27/02)
- (14) Added arrays AD, BXHEIGHT, and T to "dao_mod.f". Also removed obsolete code from 8/02 from several module routines. Now references "error_mod.f". Remove all references to QQ, it is now declared in "wetscav_mod.f". (bmy, 11/8/02)
- (15) Now references "grid_mod.f". Also added PHIS field, which was formerly stored as PALTD in "CMN". Added bug fix in routine

- AVGPOLE for 1x1 nested grid. (bmy, 3/11/03)
- (16) Added SUNCOSB array for SMVGEAR II. Also removed KZZ array, since that is now obsolete. (bmy, 4/28/03)
 - (17) Now moved MAKE_CLDFRC into "a6_read_mod.f". Added HKBETA, HKETA, TSKIN, GWETTOP, ZMEU, ZMMD, ZMMU, PARDF, PARDR fields for GEOS-4/fvDAS. (bmy, 6/25/03)
 - (18) Added CLDFRC, RADSWG, RADLWG, SNOW arrays (bmy, 12/9/03)
 - (19) Added routine COPY_I6_FIELDS w/ parallel DO-loops (bmy, 4/13/04)
 - (20) Now also allocate AVGW for offline aerosol simulation (bmy, 9/28/04)
 - (21) AVGPOLE now uses NESTED_CH and NESTED_NA cpp switches (bmy, 12/1/04)
 - (22) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
 - (23) Now allocate SNOW and GWET for GCAP (bmy, 8/17/05)
 - (24) Now also add TSKIN for GEOS-3 (tmf, bmy, 10/20/05)
 - (25) Modifications for near-land formulation (ltm, bmy, 5/16/06)
 - (26) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
 - (27) Modified for variable tropopause (phs, bdf, 9/14/06)
 - (28) Add in extra fields for GEOS-5. Updated COSSZA. Now cap var trop at 200hPa near poles in INTERP (bmy, phs, 9/18/07)
 - (29) Bug fix in INIT_DAO for CMFMC array (bmy, jaf, 6/11/08)
 - (30) Add heat flux EFLUX for GEOS5. (lin, ccc, 5/29/09)
 - (31) Add fractions of land and water, FRLAND, FROCEAN, FRLANDIC, FRLAKE for methane (kjl, 8/18/09)
 - (32) Bug fix in AVGPOLE (bmy, 12/18/09)
 - (33) Remove obsolete SUNCOSB array (bmy, 4/28/10)
- 16 Aug 2010 - R. Yantosca - Added ProTeX headers
- 18 Aug 2010 - R. Yantosca - Added modifications for MERRA data
- 18 Aug 2010 - R. Yantosca - Move CMN_SIZE, CMN_DIAG to top of module
- 25 Aug 2010 - R. Yantosca - Now read LWI (land/water/ice) for MERRA met
- 05 Oct 2011 - R. Yantosca - Add SUNCOS_30 array to hold the cos(SZA) computed @ 30 mins after each GMT hour.
- 07 Oct 2011 - R. Yantosca - Rename SUNCOS30 to SUNCOS_MID, which is the cos(SZA) at the midpt of the chemistry timestep
- 06 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x met fields
- 06 Feb 2012 - R. Yantosca - Split up INIT_DAO into several routines
- 07 Feb 2012 - M. Payer - Add subroutine GET_COSINE_SZA to compute sun angles at the current time and 5 hours prior to the current time (for the PARANOX ship emissions plume model) (R. Yantosca)
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 01 Mar 2012 - R. Yantosca - Now references the new grid_mod.F90
- 06 Mar 2012 - R. Yantosca - Now allocate T03 for all met fields
- 21 Nov 2012 - R. Yantosca - Removed met fields now contained in State_met
- 21 Nov 2012 - R. Yantosca - Remove functions INIT_DAO_GCAP, INIT_DAO_GEOS4, INIT_DAO_GEOS5, INIT_DAO_GEOS57, INIT_DAO_MERRA
- 27 Nov 2012 - R. Yantosca - Removed obsolete AIRQNT_FULLGRID routine and obsolete arrays AIRDEN_FULLGRID, T_FULLGRID
- 28 Nov 2012 - R. Yantosca - Removed SUNCOS, SUNCOS_MID, SUNCOS_MID_5hr
- 28 Nov 2012 - R. Yantosca - Removed routines INIT_DAO, INIT_DAO_DERIVED, and

CLEANUP_DAO; we have no more allocatable arrays

14 Mar 2013 - M. Payer - Restored routines AIRQNT_FULLLGRID, INIT_DAO, CLEANUP_DAO and arrays AIRDEN_FULLLGRID and T_FULLLGRID. They are required to correct vertical regridding of OH for offline simulations.

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

07 Nov 2014 - M. Yannetti - Added PRECISION_MOD

23 Feb 2015 - E. Lundgren - Change AIRQNT and formulas to take into account the presence of water vapor

26 Feb 2015 - E. Lundgren - Move calculations of RH and AVGW from MAKE_RH and MAKE_AVGW to AIRQNT. Update algorithms to account for the presence of water vapor and to use [g water / kg moist air] for the definition of specific humidity (SPHU).

16 Apr 2015 - R. Yantosca - Removed AIRQNT_FULLLGRID, this was used to regrid OH fields, but that is now done by HEMCO

16 Apr 2015 - R. Yantosca - Also remove INIT_DAO and CLEANUP_DAO, since these allocated/deallocated the FULLGRID arrays

29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

24 Aug 2017 - M. Sulprizio - Remove support for GCAP, GEOS-4, GEOS-5 and MERRA

08 Mar 2018 - E. Lundgren - Add GIGC_Cap_Tropopause_Prns from GCHP gchp_util

9.1.1 avgpole

Subroutine AVGPOLE computes average quantity near polar caps, defined by ($J = 1, 2$) and ($J = JJP\text{AR}-1, JJP\text{AR}$).

INTERFACE:

```
SUBROUTINE AVGPOLE( Z )
```

USES:

```
USE GC_GRID_MOD, ONLY : GET_AREA_M2
```

INPUT/OUTPUT PARAMETERS:

```
REAL(fp), INTENT(INOUT) :: Z(IIPAR,JJP\text{AR})    ! Quantity to be averaged
                                                ! over the pole (usually PS)
```

REVISION HISTORY:

30 Jan 1998 - R. Yantosca - Initial version

(1) AVGPOLE is written in Fixed-Form Fortran 90. Use F90 syntax for declarations, etc (bmy, 4/14/99)

(2) MAIN now passes the Harvard CTM variable for surface area of a gridbox, DXYP(JJP\text{AR}), to AVGPOLE. Use window offset J+J0 when accessing DXYP. Add JJP\text{AR} to the parameter list.

- (3) Added this routine to "dao_mod.f" (bmy, 6/27/00)
 - (4) Updated comments (bmy, 4/4/01)
 - (5) Now replaced DXYP(J) with routine GET_AREA_M2 of "grid_mod.f"
Now also return immediately if GRID1x1 is selected. (bmy, 3/11/03)
 - (6) Now use cpp switches NESTED_CH and NESTED_NA to denote nested
grids...GRID1x1 can now also denote a global grid (bmy, 12/1/04)
 - (7) Also need to RETURN for 0.5 x 0.666 nested grid simulations
(mpb, bmy, 12/18/09)
 - 16 Aug 2010 - R. Yantosca - Added ProTeX headers
 - 01 Mar 2012 - R. Yantosca - Now use GET_AREA_M2(I,J,L) from grid_mod.F90
 - 26 Sep 2013 - R. Yantosca - Remove SEAC4RS C-preprocessor switch
-

9.1.2 airqnt

Subroutine AIRQNT sets several members of State_Met, the meteorology object of derived type MetState, and optionally updates the tracer concentrations to conserve tracer mass when air quantities change.

INTERFACE:

```

SUBROUTINE AIRQNT( am_I_Root, Input_Opt, State_Met,
&                 State_Chm, RC,           Update_Mixing_Ratio )

```

USES:

```

USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : Get_Area_M2
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE PhysConstants, ONLY : AIRMW, AVO
USE Pressure_Mod
USE Time_Mod,      ONLY : Get_LocalTime

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)           :: am_I_Root ! On the root CPU?
TYPE(OptInput),  INTENT(IN)           :: Input_Opt ! Input Options obj
LOGICAL,          INTENT(IN), OPTIONAL :: update_mixing_ratio
                                                         ! Update mixing ratio?
                                                         ! Default is yes

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)         :: State_Met ! Met State obj
TYPE(ChmState), INTENT(INOUT)         :: State_Chm ! Chemistry State obj

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)         :: RC           ! Success or failure?

```

REMARKS:

DAO met fields updated by AIRQNT:

```

=====
(1) PEDGE      (REAL(fp)) : Moist air pressure at grid box bottom      [hPa]
(2) PEDGE_DRY (REAL(fp)) : Dry air partial pressure at box bottom      [hPa]
(3) PMID       (REAL(fp)) : Moist air pressure at grid box centroid    [hPa]
(4) PMID_DRY  (REAL(fp)) : Dry air partial pressure at box centroid    [hPa]
(5) PMEAN     (REAL(fp)) : Altitude-weighted mean moist air pressure  [hPa]
(6) PMEAN_DRY (REAL(fp)) : Alt-weighted mean dry air partial pressure [hPa]
(7) DELP      (REAL(fp)) : Delta-P extent of grid box                  [hPa]
                               (Same for both moist and dry air since we
                               assume constant water vapor pressure
                               across box)
(8) AIRDEN    (REAL(fp)) : Mean grid box dry air density                [kg/m^3]
                               (defined as total dry air mass/box vol)
(9) MAIRDEN   (REAL(fp)) : Mean grid box moist air density              [kg/m^3]
                               (defined as total moist air mass/box vol)
(10) AD       (REAL(fp)) : Total dry air mass in grid box                [kg]
(11) ADMIIST  (REAL(fp)) : Total moist air mass in grid box            [kg]
(12) BXHEIGHT (REAL(fp)) : Vertical height of grid box                  [m]
(13) AIRVOL   (REAL(fp)) : Volume of grid box                          [m^3]
(14) MOISTMW  (REAL(fp)) : Molecular weight of moist air in box        [g/mol]

```

REVISION HISTORY:

30 Jan 1998 - R. Yantosca - Initial version

- (1) AIRQNT is written in Fixed-Form Fortran 90. Use F90 syntax for declarations etc. (bmy, 4/14/99)
- (2) AIRQNT can now compute PW from PS (if LMAKEPW=T) or PS from PW.
- (3) AIRQNT should also be called after TPCORE, since TPCORE changes the PW values. AIRQNT must then be called to compute the post-TPCORE values of AD, BXHEIGHT, AIRVOL, and AIRDEN.
- (4) The AIRDEN and DELP arrays are now dimensioned as (LLPAR,IIPAR,JJPARG) for better efficiency when processing a whole (I,J) column layer by layer. In FORTRAN, the best efficiency is obtained when the leftmost array index corresponds to the innermost loop.
- (5) Remove PTOP from the arg list. PTOP is now a parameter in "CMN_SIZE". Also updated comments. (bmy, 2/22/00)
- (6) Replace IM, JM, LM with IIPAR, JJPARG, LLPARG as loop boundaries. This ensures that all quantities get defined up to the top of the atmosphere. (bmy, 6/15/00)
- (7) Added to "dao_mod.f" (bmy, 6/26/00)
- (8) Updated comments (bmy, 4/4/01)
- (9) P(IREF,JREF) is now P(I,J). T(IREF,JREF,L) is now T(I,J,L). Also removed LM from the arg list, it is obsolete. Also updated comments. (bmy, 9/26/01)
- (10) Remove PW -- it is now obsolete. Also make PW a local variable, we need to preserve the way it computes P so as to avoid numerical

- drift. (bmy, 10/4/01)
- (11) Removed obsolete code from 9/01 and 10/01 (bmy, 10/23/01)
 - (12) Removed LMAKEPW from arg list. Added parallel DO loops (bmy, 11/15/01)
 - (13) Removed obsolete code from 11/01 (bmy, 1/9/02)
 - (14) Now rename G_SIGE to SIGE, and dimension it (1:LLPAR+1). Updated comments, cosmetic changes. (bmy, 4/4/02)
 - (15) Removed obsolete, commented-out code (bmy, 6/25/02)
 - (16) Removed PS, P, SIGE from the arg list for hybrid grid. Now reference routines GET_PEDGE and GET_BP from "pressure_mod.f". Removed obsolete, commented-out code. (dsa, bdf, bmy, 8/27/02)
 - (17) Now only pass DXYP via the arg list -- the other arguments are actually already contained within "dao_mod.f" (bmy, 11/15/02)
 - (18) Now replace DXYP(JREF) with routine GET_AREA_M2 of "grid_mod.f". (bmy, 3/11/03)
 - (19) Now move computation of DELP into main loop. Also remove P, LOGP, JREF, DSIG variables -- these are obsolete for fvDAS. (bmy, 6/19/03)
- 16 Aug 2010 - R. Yantosca - Added ProTeX headers
 - 01 Mar 2012 - R. Yantosca - Now use GET_AREA_M2(I,J,L) from grid_mod.F90
 - 22 Oct 2012 - R. Yantosca - Now reference gipc_state_met_mod.F90
 - 22 Oct 2012 - R. Yantosca - Renamed LOCAL_MET argument to State_Met
 - 09 Nov 2012 - M. Payer - Copy met field arrays to the State_Met derived type object
 - 06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)
 - 06 Nov 2014 - R. Yantosca - Now use State_Met%DELP(I,J,L)
 - 07 Jan 2015 - E. Lundgren - Add new Met field MAIRDEN (moist air density)
 - 13 Feb 2015 - E. Lundgren - Incorporate water vapor in calculations of densities. Calculate dry air pressures.
 - 23 Feb 2015 - E. Lundgren - Change box height formula to include water vapor
 - 24 Feb 2015 - E. Lundgren - Add calculation of RH to subroutine. Use new formulate for H2O saturation pressure.
 - 03 Mar 2015 - E. Lundgren - Set State_Met%TV (virtual temperature)
 - 16 Apr 2015 - E. Lundgren - Add new State_Met variables PMEAN, PMEAN_DRY, MOISTMW, and ADMOIST
 - 28 Oct 2015 - E. Lundgren - Pass update_mixing_ratio flag, Input_Opt, and State_Chm to optionally update tracer units
 - 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
 - 15 Aug 2016 - E. Lundgren - Update conc w/ RH for all species
 - 08 Jan 2018 - R. Yantosca - Now define query fields from State_Met here
 - 17 Jan 2018 - R. Yantosca - Now compute tropopause quantities here, which allows us to retire GeosCore/chemgrid_mod.F

9.1.3 interp

Subroutine INTERP linearly interpolates GEOS-Chem I6 and I3 fields (winds, surface pressure, temperature, surface albedo, specific humidity etc.) to the current dynamic timestep.

INTERFACE:

```
SUBROUTINE INTERP( NTIME0, NTIME1, NTD, Input_Opt, State_Met )
```

USES:

```
USE GC_GRID_MOD,      ONLY : GET_YEDGE
USE Input_Opt_Mod,   ONLY : OptInput
USE State_Met_Mod,   ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: NTIME0    ! Elapsed time [s] at
                                           ! start of outer time step
INTEGER,      INTENT(IN)    :: NTIME1    ! Elapsed time [s] at
                                           ! current time
INTEGER,      INTENT(IN)    :: NTD       ! Dynamic timestep [s]
TYPE(OptInput), INTENT(IN)  :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Different met fields are archived at I6 (instantaneous 6-hr) and I3 (instantaneous 3-hr) time resolution depending on the specific product. For example, relative humidity is an instantaneous 6hr field in MERRA and a 6-hr time averaged field in GEOS-5.

REVISION HISTORY:

- 30 Jan 1998 - R. Yantosca - Initial version
- (1) INTERP is written in Fixed-Form Fortran 90.
 - (2) Subtract PINT from PSC since the only subroutine that uses PSC is TPCORE. This prevents having to subtract and add PINT to PSC before and after each call of TPCORE.
 - (3) Pass the Harvard CTM temperature variable T(IGCM, JGCM, LGCM) to INTERP via the argument list (instead of including file CMN). It is computationally inefficient to keep two large arrays for the same quantity. Use the proper window offsets with T.
 - (4) Added to "dao_mod.f" (bmy, 6/26/00)
 - (5) Updated comments (bmy, 4/4/01)
 - (6) Replaced {IJL}GCMPAR w/ IIPAR, JJP, LLP. Also now use parallel DO-loop for interpolation. Updated comments. (bmy, 9/26/01)
 - (7) Removed obsolete code from 9/01 (bmy, 10/23/01)
 - (8) Add PSC2 as the surface pressure at the end of the dynamic timestep. This needs to be passed to TPCORE and AIRQNT so that the mixing ratio can be converted to mass properly. Removed PINT from the arg list, since we don't need it anymore. Also updated comments and made

- some cosmetic changes. (bmy, 3/27/02)
- (9) Removed obsolete, commented-out code (bmy, 6/25/02)
 - (10) Eliminated PS, PSC from the arg list, for floating-pressure fix.
(dsa, bdf, bmy, 8/27/02)
 - (11) Met field arrays are module variables, so we don't need to pass them
as arguments. (bmy, 11/20/02)
 - (12) Removed NDT from the arg list since that is always 21600. For GEOS-4
met fields, only interpolate PSC2; the other fields are 6-h averages.
Eliminate TC variable, it's obsolete. Now use double precision to
compute TM and TC2 values. Renamed NTIME to NTIME1 and NTIME1 to
NTIME0. Updated comments. (bmy, 6/19/03)
 - (13) Now modified for GEOS-5 and GCAP met fields. (swu, bmy, 5/25/05)
 - (14) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
 - (15) Now interpolate TROPP, only if variable tropopause is used
(phs, 9/12/06)
 - (16) Don't interpolate TROPP for GEOS-5 (bmy, 1/17/07)
 - (17) Now limit tropopause pressure to 200 mbar at latitudes above 60deg
(phs, 9/18/07)
- 16 Aug 2010 - R. Yantosca - Added ProTeX headers
 - 18 Aug 2010 - R. Yantosca - Rewrite #if block logic for clarity
 - 06 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x met fields
 - 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
 - 01 Mar 2012 - R. Yantosca - Now use GET_YEDGE(I,J,L) from new grid_mod.F90
 - 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
 - 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
 - 29 Oct 2013 - R. Yantosca - Now interpolate T_FULLGRID field for GEOS-FP met
 - 12 May 2015 - E. Lundgren - Fix I3 interpolation bug where time fraction
spans from 1 to 2, not 0 to 1, for hrs 3 to 6
 - 11 Aug 2015 - R. Yantosca - MERRA2 behaves the same way as GEOS-FP
 - 28 Oct 2015 - E. Lundgren - Set previous SPHU to current SPHU before update
 - 04 May 2016 - E. Lundgren - Now also interpolate dry pressure calculated
in SET_DRY_SURFACE_PRESSURE using GMAO wet P
 - 07 Nov 2017 - R. Yantosca - Dynamic tropopause is now always assumed

9.1.4 gigc_cap_tropopause_prs

Subroutine GIGC_CAP_TROPOPAUSE_PRS caps the tropopause pressure in polar latitudes to 200 hPa, so that we don't end up doing tropospheric chemistry too high over the poles. This is done in the standalone GEOS-Chem, and we also need to apply this when running GEOS-Chem within the GEOS-5 GCM.

INTERFACE:

```

SUBROUTINE GIGC_Cap_Tropopause_Prns( am_I_Root, IM,          JM,
&                                     Input_Opt, State_Met, RC )

```

USES:

```

USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE GC_Grid_Mod, ONLY : Get_XEdge
USE GC_Grid_Mod, ONLY : Get_YEdge

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root      ! Are we on the root CPU?
INTEGER,          INTENT(IN)    :: IM             ! # of lons on this CPU
INTEGER,          INTENT(IN)    :: JM             ! # of lats on this CPU
TYPE(OptInput),  INTENT(IN)    :: Input_Opt      ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met      ! Meteorology State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC             ! Success or failure

```

REMARKS:

Jennifer Logan (see correspondence below) suggested that we should cap the variable tropopause at 200hPa in near-polar regions (90-60S and 60-90N), to avoid the problem with anomalously high tropopause heights at high latitudes. This fix was standardized in GEOS-Chem v7-04-13.

Jennifer Logan wrote:

I think we should restrict the tropopause at latitudes > 60 deg. to pressures greater than 200 mb (about 11 km). From Fig. 3 in Seidel and Randel, there are tropopause (TP) heights as high as 13.5 km in the Antarctic (median height is ~9.8 km, 250 mb), but I don't think we want to be doing trop. chem there. The median TP pressure at ~80 N is ~300 mb, compared to ~250 mb at 70-85 S. The extratropical TP heights are higher (lower pressure) in the SH than in the NH according to Fig. 3. This approach is also very easy to explain in a paper.

REVISION HISTORY:

```

14 Mar 2013 - R. Yantosca - Initial version
08 Mar 2018 - E. Lundgren - Move this rtn from obsolete GCHP/gchp_utils.F90

```

9.1.5 set_dry_surface_pressure

Subroutine SET_DRY_SURFACE_PRESSURE sets the dry surface pressures PS1_DRY or PS2_DRY by removing the water vapor from the column constructed with MET pressure fields PS1_WET or PS2_WET.

INTERFACE:

```
SUBROUTINE SET_DRY_SURFACE_PRESSURE( State_Met, PS_ID )
```

USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD,           ONLY : CHECK_VALUE
USE State_Met_Mod,       ONLY : MetState
USE PRESSURE_MOD,        ONLY : GET_AP, GET_BP
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)    :: PS_ID          ! 1 = PS1, 2 = PS2
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT)  :: State_Met     ! Object for met fields
```

REVISION HISTORY:

```
03 May 2002 - E. Lundgren - Initial version (M. Lee, JPL)
18 Feb 2017 - S. D. Eastham - Updated to include TOA pressure
```

REMARKS:

This subroutine is an adaptation of the GEOS-Chem moisture fix implemented by Meemong Lee (JPL) in the adjoint model. Like PS1_WET and PS2_WET, from which PS1_DRY and PS2_DRY are derived, these values are interpolated within routine INTERP to derive instantaneous PSC2_DRY. Note that PSC2_WET and PSC2_DRY are not used to fetch pressures and calculate air quantities until after advection. Also note that the dry surface pressure calculated in this routine may be used to calculate delta dry pressures across a level by utilizing GMAO parameters Ap and Bp but should not be used to extract dry pressure edge values as a height proxy.

9.1.6 is_land

Function IS_LAND returns TRUE if surface grid box (I,J) is a land box.

INTERFACE:

```
FUNCTION IS_LAND( I, J, State_Met ) RESULT ( LAND )
```

USES:

```
USE State_Met_Mod,       ONLY : MetState
USE TIME_MOD,           ONLY : GET_YEAR
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)    :: I            ! Longitude index of grid box
INTEGER,          INTENT(IN)    :: J            ! Latitude index of grid box
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object
```



```

INTEGER,          INTENT(IN) :: I          ! Longitude index of grid box
INTEGER,          INTENT(IN) :: J          ! Latitude index of grid box
TYPE(MetState),  INTENT(IN) :: State_Met  ! Meteorology State object

```

RETURN VALUE:

```

LOGICAL           :: WATER                ! =T if this is a water box

```

REVISION HISTORY:

```

30 Jan 1998 - R. Yantosca - Initial version
(1 ) Now use ALBEDO field to determine water or water ice boxes for GEOS-3.
      (bmy, 4/4/01)
(2 ) For 4x5 data, regrided albedo field can cause small inaccuracies
      near the poles (bmy, 4/4/01)
(3 ) Add references to CMN_SIZE and CMN, so that we can use the JYEAR
      variable to get the current year. Also, for 1998, we need to compute
      if is an ocean box or not from the surface albedo, since for this
      year the LWI/SURFTYPE field is not given. For other years than 1998,
      we use LWI(I,J) >= 50 as our ocean box criterion. Deleted obsolete
      code and updated comments. (mje, bmy, 1/9/02)
(4 ) Deleted GEOS-2 #ifdef statement. GEOS-2 met fields never really
      materialized, we use GEOS-3 instead. (bmy, 9/18/02)
(5 ) Now uses function GET_YEAR from "time_mod.f". Removed reference
      to CMN header file. (bmy, 3/11/03)
(6 ) Added code to determine water boxes for GEOS-4 (bmy, 6/18/03)
(7 ) Now modified for GEOS-5 and GCAP met fields (swu, bmy, 5/25/05)
(8 ) Now remove test for sea ice (bmy, 8/10/05)
(9 ) Now use NINT to round LWI for GEOS-4/GEOS-5 (ltm, bmy, 5/9/06)
(10) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
16 Aug 2010 - R. Yantosca - Added ProTeX headers
25 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
06 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA/GEOS-5
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
                          derived type object

```

9.1.8 is_ice

Function IS_ICE returns TRUE if surface grid box (I,J) contains either land-ice or sea-ice.

INTERFACE:

```

FUNCTION IS_ICE( I, J, State_Met ) RESULT ( ICE )

```

USES:

```

USE State_Met_Mod,      ONLY : MetState
USE TIME_MOD,          ONLY : GET_YEAR

```

INPUT PARAMETERS:

```

    INTEGER,          INTENT(IN) :: I           ! Longitude index of grid box
    INTEGER,          INTENT(IN) :: J           ! Latitude  index of grid box
    TYPE(MetState),  INTENT(IN) :: State_Met    ! Meteorology State object

```

RETURN VALUE:

```

    LOGICAL              :: ICE                ! =T if this is an ice box

```

REVISION HISTORY:

```

09 Aug 2005 - R. Yantosca - Initial version
(1 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
16 Aug 2010 - R. Yantosca - Added ProTeX headers
25 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
06 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA/GEOS-5
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3

```

9.1.9 is_near

Function IS_NEAR returns TRUE if surface grid box (I,J) contains any land above a certain threshold (THRESH) or any of the adjacent boxes up to NEIGHBOR boxes away contain land.

INTERFACE:

```

    FUNCTION IS_NEAR( I, J, THRESH, NEIGHBOR, State_Met )
    &   RESULT ( NEAR )

```

USES:

```

    USE State_Met_Mod,      ONLY : MetState

```

INPUT PARAMETERS:

```

    ! Arguments
    INTEGER,          INTENT(IN) :: I, J       ! Lon & lat grid box indices
    INTEGER,          INTENT(IN) :: NEIGHBOR   ! # of neighbor boxes to consider
    REAL(fp),         INTENT(IN) :: THRESH     ! LWI threshold for near-land
    TYPE(MetState),  INTENT(IN) :: State_Met  ! Meteorology State object

```

RETURN VALUE:

```

    LOGICAL              :: NEAR              ! # of near land boxes

```

REMARKS:

```

Typical values for:
GCAP   : THRESH = 0.2, NEIGHBOR = 1
GEOS-3 : THRESH = 80.0, NEIGHBOR = 1

```

```

GEOS-4 : THRESH = 0.2, NEIGHBOR = 1
GEOS-5 : THRESH = 0.2, NEIGHBOR = 1

```

NOTE: This routine is mostly obsolete now.

REVISION HISTORY:

```

09 May 2006 - R. Yantosca - Initial version
(1 ) Modified for GCAP and GEOS-3 met fields (bmy, 5/16/06)
(2 ) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
16 Aug 2010 - R. Yantosca - Added ProTeX headers
19 Aug 2010 - R. Yantosca - Rewrote logic of #if block for clarity
25 Aug 2010 - R. Yantosca - Treat MERRA in same way as GEOS-5
06 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA/GEOS-5
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
                        derived type object

```

9.1.10 get_obk

Function GET_OBK returns the Monin-Obhukov length at a grid box (I,J).

INTERFACE:

```

FUNCTION GET_OBK( I, J, State_Met ) RESULT( OBK )

```

USES:

```

USE State_Met_Mod, ONLY : MetState

```

INPUT PARAMETERS:

```

INTEGER,          INTENT(IN) :: I           ! Longitude index
INTEGER,          INTENT(IN) :: J           ! Latitude index
TYPE(MetState),  INTENT(IN) :: State_Met    ! Meteorology State object

```

RETURN VALUE:

```

REAL(fp)          :: OBK                    ! Monin-Obhukhov length

```

REMARKS:

REVISION HISTORY:

```

25 May 2005 - R. Yantosca - Initial version
16 Aug 2010 - R. Yantosca - Added ProTeX headers
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
                        derived type object

```


9.1.11 get_cosine_sza

Routine GET_COSINE_SZA is a driver for calling the COSSZA routine from dao_mod.F. This routine calls COSSZA twice. The first call computes the sun angles at the current time and midpoint of the current chemistry time step. The second call computes the sun angles 5 hours prior to the current time (for the PARANOX ship emissions plume model).

INTERFACE:

```
      SUBROUTINE GET_COSINE_SZA( am_I_Root, Input_Opt, State_Met, RC )
```

USES:

```
      USE ErrCode_Mod
      USE Input_Opt_Mod,      ONLY : OptInput
      USE JULDAY_MOD,        ONLY : JULDAY
      USE State_Met_Mod,     ONLY : MetState
      USE TIME_MOD,          ONLY : GET_DAY_OF_YEAR
      USE TIME_MOD,          ONLY : GET_DAY
      USE TIME_MOD,          ONLY : GET_GMT
      USE TIME_MOD,          ONLY : GET_HOUR
      USE TIME_MOD,          ONLY : GET_MINUTE
      USE TIME_MOD,          ONLY : GET_SECOND
      USE TIME_MOD,          ONLY : GET_MONTH
      USE TIME_MOD,          ONLY : GET_YEAR
```

INPUT PARAMETERS:

```
      LOGICAL,      INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
      TYPE(OptInput), INTENT(IN)  :: Input_Opt    ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
      TYPE(MetState), INTENT(INOUT) :: State_Met  ! Meteorology State object
```

OUTPUT PARAMETERS:

```
      INTEGER,      INTENT(OUT)   :: RC           ! Success or failure?
```

REVISION HISTORY:

```
      07 Feb 2012 - R. Yantosca - Initial version
      27 Nov 2012 - R. Yantosca - Add am_I_root, Input_Opt, State_Met, RC args
      27 Nov 2012 - R. Yantosca - Now pass State_Met to COSSZA so that the
                                SUNCOS fields may be updated
      28 Nov 2012 - R. Yantosca - Removed references to 1-D SUNCOS arrays
      04 Dec 2014 - M. Yannetti - REAL*8 needs to stay *8 due to JULDAY
```

9.1.12 cossza

COSSZA computes the cosine of the solar zenith angle, given the day of the year and GMT hour. The cosine of the solar zenith angle is returned at both the current time and at the midpoint of the chemistry timestep (i.e. for the centralized chemistry timestep option).

INTERFACE:

```
SUBROUTINE COSSZA( DOY, GMT_HOUR, DO_5hr_AGO, State_Met )
```

USES:

```
USE GC_GRID_MOD,          ONLY : GET_YMID_R
USE State_Met_Mod,       ONLY : MetState
USE TIME_MOD,           ONLY : GET_MINUTE, GET_SECOND
USE TIME_MOD,           ONLY : GET_LOCALTIME
USE TIME_MOD,           ONLY : GET_TS_CHEM
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)   :: DOY           ! Day of the year
INTEGER,      INTENT(IN)   :: GMT_HOUR      ! Hour of day
LOGICAL,      INTENT(IN)   :: DO_5hr_AGO   ! Compute 5h ago?
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State
```

REMARKS:

Hour angle (AHR) is a function of longitude. AHR is zero at solar noon, and increases by 15 deg for every hour before or after solar noon. Hour angle can be thought of as the time in hours since the sun last passed the meridian (i.e. the time since the last local noon).

The cosine of the solar zenith angle (SZA) is given by:

$$\cos(\text{SZA}) = \sin(\text{LAT}) * \sin(\text{DEC}) + \cos(\text{LAT}) * \cos(\text{DEC}) * \cos(\text{AHR})$$

where LAT = the latitude angle,
 DEC = the solar declination angle,
 AHR = the hour angle, all in radians.

If SUNCOS < 0, then the sun is below the horizon, and therefore does not contribute to any solar heating.

REVISION HISTORY:

- 21 Jan 1998 - R. Yantosca - Initial version
- (1) COSSZA is written in Fixed-Form Fortran 90.
- (2) Use IMPLICIT NONE
- (3) Use C-preprocessor #include statement to include CMN_SIZE, which has IIPAR, JJPARG, LLPARG, IIPARG, JJPARG, LGLOB.
- (4) Use IM and JM (in CMN_SIZE) as loop limits.
- (5) Include Harvard CTM common blocks and rename variables where needed.
- (6) Use SUNCOS(MAXIJ) instead of a 2D array, in order for compatibility with the Harvard CTM subroutines. SUNCOS loops over J, then I.
- (7) Added DO WHILE loops to reduce TIMLOC into the range 0h - 24h.
- (8) Cosmetic changes. Also use F90 declaration statements (bmy, 6/5/00)
- (9) Added to "dao_mod.f". Also updated comments. (bmy, 9/27/01)

- (10) Replaced all instances of IM with IIPAR and JM with JJPARG, in order to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (11) Deleted obsolete code from 6/02 (bmy, 8/21/02)
- (12) Removed RLAT and XLON from the arg list. Now compute these using functions from "grid_mod.f" (bmy, 2/3/03)
- (13) Now uses GET_LOCALTIME from "time_mod.f" to get the local time. Added parallel DO loop. Removed NHMSb, NSEC arguments. (bmy, 2/13/07)
- (14) Now compute SUNCOS at the midpoint of the relevant time interval (i.e. the chemistry timestep). Also make the A and B coefficients parameters instead of variables. (bmy, 4/27/10)
- 16 Aug 2010 - R. Yantosca - Added ProTeX headers
- 05 Oct 2011 - R. Yantosca - Now also return the cosine of the solar zenith angle at 30m after the GMT hour.
- 07 Oct 2011 - R. Yantosca - Now return SUNCOS_MID, the cos(SZA) at the midpt of the chem step (not always at 00:30).
- 07 Feb 2012 - R. Yantosca - Now add GMT_HOUR as a new argument, which ! will facilitate computing sun angles 5h ago
- 01 Mar 2012 - R. Yantosca - Now use GET_YMID_R(I,J,L) from grid_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET_LOCALTIME(I,J,L) from time_mod.F90
- 27 Nov 2012 - R. Yantosca - Update SUNCOS fields of the State_Met object
- 10 Mar 2017 - E. Lundgren - Bug fix: include partial hour in current time

9.1.13 copy_i3_i6_fields

Subroutine COPY_I3_I6_FIELDS copies the I-6 fields at the end of a 6-hr timestep. The I-6 fields at the end of a given 6-hr timestep become the fields at the beginning of the next 6-hr timestep.

INTERFACE:

```
SUBROUTINE COPY_I3_I6_FIELDS( State_Met )
```

USES:

```
USE State_Met_Mod, ONLY : MetState
```

INPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REVISION HISTORY:

- 13 Apr 2004 - R. Yantosca - Initial version
- (1) Added parallel DO-loops (bmy, 4/13/04)
- (2) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
- (3) Added TROPP (phs 11/10/06)
- (4) Don't copy TROPP2 to TROPP1 for GEOS-5 (bmy, 1/17/07)
- 16 Aug 2010 - R. Yantosca - Added ProTeX headers


```
PRIVATE :: Check_Dimensions
PRIVATE :: Merra2_Read_A3cld
PRIVATE :: Merra2_Read_A3dyn
PRIVATE :: Merra2_Read_A3mstC
PRIVATE :: Merra2_Read_A3mstE
PRIVATE :: Get_Resolution_String
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC   :: Merra2_Read_CN
PUBLIC   :: Merra2_Read_A1
PUBLIC   :: Merra2_Read_A3
PUBLIC   :: Merra2_Read_I3_1
PUBLIC   :: Merra2_Read_I3_2
PUBLIC   :: Cleanup_Merra2_Read
```

REMARKS:

Assumes that you have a netCDF library (either v3 or v4) installed on your system.

REVISION HISTORY:

```
12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
03 Dec 2015 - R. Yantosca - Add file ID's as module variables
03 Dec 2015 - R. Yantosca - Add CLEANUP_MERRA2_READ to close any open
                           netCDF files left at the end of a simulation
02 Feb 2016 - E. Lundgren - Block of diagnostics with if defined BPCH
```

9.2.1 Get_Resolution_String

Function Get_Resolution_String returns the proper filename extension for the GEOS-Chem horizontal grid resolution. This is used to construct the various file names.

INTERFACE:

```
FUNCTION Get_Resolution_String() RESULT( resString )
```

RETURN VALUE:

```
CHARACTER(LEN=255) :: resString
```

REVISION HISTORY:

```
12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
13 Aug 2015 - R. Yantosca - MERRA2 data is now storead as netCDF4 (.nc4)
```

9.2.2 Check_Dimensions

Subroutine CHECK_DIMENSIONS checks to see if dimensions read from the netCDF file match the defined GEOS-Chem dimensions. If not, then it will stop the GEOS-Chem simulation with an error message.

INTERFACE:

```
SUBROUTINE Check_Dimensions( lon, lat, lev, time, time_expected, caller )
```

INPUT PARAMETERS:

```

INTEGER,          OPTIONAL, INTENT(IN)  :: lon           ! Lon dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lat           ! Lat dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lev           ! Alt dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time          ! Time dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time_expected ! Expected # of
                                                    ! time slots
CHARACTER(LEN=*), OPTIONAL, INTENT(IN)  :: caller        ! Name of caller
                                                    ! routine

```

REMARKS:

Call this routine with keyword arguments, e.g

```
CALL CHECK_DIMENSION( lon=X, lat=Y, lev=Z, &
                    time=T, time_expected=8, caller=caller )
```

REVISION HISTORY:

02 Feb 2012 - R. Yantosca - Initial version

03 Feb 2012 - R. Yantosca - Now pass the caller routine name as an argument

9.2.3 Merra2_Read_cn

Routine to read variables and attributes from a MERRA2 met fields file containing constant (CN) data.

INTERFACE:

```
SUBROUTINE Merra2_Read_CN( Input_Opt, State_Met )
```

USES:

```

USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
 13 Aug 2015 - R. Yantosca - Bug fix: change CN date to 2015/01/01

9.2.4 Merra2_Read_a1

Routine to read variables and attributes from a MERRA2 met fields file containing 1-hr time-averaged (A1) data.

INTERFACE:

```
SUBROUTINE Merra2_Read_A1( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS   ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

Special handling for surface precipitation fields:

In MERRA2 (and in MERRA), the PRECTOT etc. surface precipitation met fields fields have units of [kg/m²/s]. In all other GEOS versions, PREACC and PRECON have units of [mm/day].

Therefore, for backwards compatibility with existing code, apply

the following unit conversion to the GEOS-5 PRECTOT and PRECCON fields:

$$\begin{array}{cccc}
 \text{kg} & | & \text{m}^3 & | & 86400 \text{ s} & | & 1000 \text{ mm} \\
 \hline
 \text{m}^2 \text{ s} & | & 1000 \text{ kg} & | & \text{day} & | & \text{m} \\
 & & \wedge & & & & \\
 & & | & & & & \\
 & & 1 / \text{density of water} & & & &
 \end{array} = 86400$$

REMARKS:

MERRA2 pressure quantities are stored on disk with units of Pa. For now we will convert to hPa for compatibility with GEOS-Chem. But in the near future we will probably recode GEOS-Chem to use the native units, which facilitate GEOS-Chem HP development.

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
 23 Sep 2015 - E. Lundgren - Now assign SWGDN to State_Met SWGDN not RADSWG
 03 Dec 2015 - R. Yantosca - Now open file only once per day

9.2.5 Merra2_Read_a3

Convenience wrapper for the following routines which read 3-hour time averaged data from disk:

- Merra2_Read_A3cld
- Merra2_Read_A3dyn
- Merra2_Read_A3mstC
- Merra2_Read_A3mstE

INTERFACE:

```
SUBROUTINE Merra2_Read_A3( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

INPUT/OUTPUT PARAMETERS:


```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90

9.2.6 Merra2_Read_a3cld

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr time-averaged (A3) data (cloud fields).

INTERFACE:

```
SUBROUTINE Merra2_Read_A3cld( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)  :: YYYYMMDD ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)  :: HHMMSS   ! GMT time in hh:mm:ss  format
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
 03 Dec 2015 - R. Yantosca - Now open file only once per day
 17 Mar 2016 - M. Sulprizio- Read optical depth into State_Met%OPTD instead of State_Met%OPTDEP (obsolete).

9.2.7 Merra2_Read_a3dyn

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr time-averaged (A3) data (dynamics fields).

INTERFACE:

```
SUBROUTINE Merra2_Read_A3dyn( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)  :: YYYYMMDD  ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)  :: HHMMSS    ! GMT time in hh:mm:ss  format
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

```
12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
03 Dec 2015 - R. Yantosca - Now open file only once per day
03 Feb 2017 - M. Sulprizio- Activate reading OMEGA fields from file
```

9.2.8 Merra2_Read_a3mstc

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr time-averaged (A3) data (moist fields, saved on level centers).

INTERFACE:

```
SUBROUTINE Merra2_Read_A3mstC( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)  :: YYYYMMDD  ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)  :: HHMMSS    ! GMT time in hh:mm:ss  format
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
 03 Dec 2015 - R. Yantosca - Now open file only once per day

9.2.9 Merra2_Read_a3mste

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr time-averaged (A3) data (moist fields, saved on level edges).

INTERFACE:

```
SUBROUTINE Merra2_read_A3mstE( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss  format
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
 03 Dec 2015 - R. Yantosca - Now open file only once per day

9.2.10 Merra2_Read_I3_1

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr instantaneous (I3) data.

INTERFACE:

```
SUBROUTINE Merra2_Read_I3_1( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are. MERRA2 pressure quantities are stored on disk with units of Pa. For now we will convert to hPa for compatibility with GEOS-Chem. But in the near future we will probably recode GEOS-Chem to use the native units, which facilitate GEOS-Chem HP development.

REVISION HISTORY:

```
12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
03 Dec 2015 - R. Yantosca - Now open file only once per day
```

9.2.11 Merra2_Read_I3_2

Routine to read variables and attributes from a MERRA2 met fields file containing 3-hr instantaneous (I3) data.

INTERFACE:

```
SUBROUTINE Merra2_Read_I3_2( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```

    INTEGER,          INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
    INTEGER,          INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss  format
    TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

    TYPE(MetState), INTENT(INOUT)  :: State_Met    ! Meteorology State object

```

REMARKS:

Even though the netCDF file is self-describing, the MERRA2 data, dimensions, and units are pre-specified according to the GMAO MERRA2 file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are. MERRA2 pressure quantities are stored on disk with units of Pa. For now we will convert to hPa for compatibility with GEOS-Chem. But in the near future we will probably recode GEOS-Chem to use the native units, which facilitate GEOS-Chem HP development.

REVISION HISTORY:

```

12 Aug 2015 - R. Yantosca - Initial version, based on geosfp_read_mod.F90
03 Dec 2015 - R. Yantosca - Now open file only once per day
20 Sep 2016 - R. Yantosca - Bug fix: FIRST must be declared as LOGICAL

```

9.2.12 Cleanup_Merra2_Read

Closes any open netCDF files at the end of a simulation. This can occur if the simulation ends at a time other than 00:00 GMT.

INTERFACE:

```

SUBROUTINE Cleanup_Merra2_Read()

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Initial version

```

9.3 Fortran: Module Interface geosfp_read_mod.F90

Module GEOSFP_READ_MOD contains subroutines for reading the GEOS-FP data from disk (in netCDF format).

INTERFACE:

```

MODULE GeosFp_Read_Mod

```

USES:

```

! NcdfUtil modules for netCDF I/O
USE m_netcdf_io_open           ! netCDF open
USE m_netcdf_io_get_dimlen     ! netCDF dimension queries
USE m_netcdf_io_read          ! netCDF data reads
USE m_netcdf_io_close         ! netCDF close

! GEOS-Chem modules
USE CMN_SIZE_MOD              ! Size parameters
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD             ! Diagnostic arrays & counters
USE DIAG_MOD,                ONLY : AD21    ! Array for ND21 diagnostic
USE DIAG_MOD,                ONLY : AD66    ! Array for ND66 diagnostic
USE DIAG_MOD,                ONLY : AD67    ! Array for ND67 diagnostic
#endif
USE ERROR_MOD,               ONLY : ERROR_STOP ! Stop w/ error message
USE PhysConstants            ! Physical constants
USE TIME_MOD                 ! Date & time routines
USE TRANSFER_MOD             ! Routines for casting

IMPLICIT NONE
PRIVATE

# include "netcdf.inc"       ! Include file for netCDF library

```

PRIVATE MEMBER FUNCTIONS:

```

PRIVATE :: Check_Dimensions
PRIVATE :: GeosFp_Read_A3cld
PRIVATE :: GeosFp_Read_A3dyn
PRIVATE :: GeosFp_Read_A3mstC
PRIVATE :: GeosFp_Read_A3mstE
PRIVATE :: Get_Resolution_String

```

PUBLIC MEMBER FUNCTIONS:

```

PUBLIC  :: GeosFp_Read_CN
PUBLIC  :: GeosFp_Read_A1
PUBLIC  :: GeosFp_Read_A3
PUBLIC  :: GeosFp_Read_I3_1
PUBLIC  :: GeosFp_Read_I3_2
PUBLIC  :: Cleanup_GeosFp_Read

```

REMARKS:

Assumes that you have a netCDF library (either v3 or v4) installed on your system.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
03 Feb 2012 - R. Yantosca - Add Geos57_Read_A3 wrapper function
07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
10 Feb 2012 - R. Yantosca - Add function Get_Resolution_String
05 Apr 2012 - R. Yantosca - Convert units for specific humidity properly
15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
11 Apr 2013 - R. Yantosca - Now pass directory fields via Input_Opt
26 Sep 2013 - R. Yantosca - Renamed to geosfp_read_mod.F90
14 Jan 2014 - R. Yantosca - Remove "define GEOS572_FILES #ifdef blocks
14 Aug 2014 - R. Yantosca - Compute CLDTOPS field in GeosFp_Read_A3mstE
03 Dec 2015 - R. Yantosca - Add file ID's as module variables
03 Dec 2015 - R. Yantosca - Add CLEANUP_GEOSFP_READ to close any open
netCDF files left at the end of a simulation
02 Feb 2016 - E. Lundgren - Block of diagnostics with if defined BPCH

9.3.1 Get_Resolution_String

Function `Get_Resolution_String` returns the proper filename extension for the GEOS-Chem horizontal grid resolution. This is used to construct the various file names.

INTERFACE:

```
FUNCTION Get_Resolution_String() RESULT( resString )
```

RETURN VALUE:

```
CHARACTER(LEN=255) :: resString
```

REVISION HISTORY:

10 Feb 2012 - R. Yantosca - Initial version
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
26 Sep 2013 - R. Yantosca - Remove SEAC4RS C-preprocessor switch

9.3.2 Check_Dimensions

Subroutine `CHECK_DIMENSIONS` checks to see if dimensions read from the netCDF file match the defined GEOS-Chem dimensions. If not, then it will stop the GEOS-Chem simulation with an error message.

INTERFACE:

```
SUBROUTINE Check_Dimensions( lon, lat, lev, time, time_expected, caller )
```

INPUT PARAMETERS:

```

INTEGER,          OPTIONAL, INTENT(IN)  :: lon           ! Lon dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lat           ! Lat dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: lev           ! Alt dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time          ! Time dimension
INTEGER,          OPTIONAL, INTENT(IN)  :: time_expected ! Expected # of
                                                ! time slots
CHARACTER(LEN=*), OPTIONAL, INTENT(IN)  :: caller        ! Name of caller
                                                ! routine

```

REMARKS:

Call this routine with keyword arguments, e.g

```

CALL CHECK_DIMENSION( lon=X, lat=Y, lev=Z, &
                    time=T, time_expected=8, caller=caller )

```

REVISION HISTORY:

02 Feb 2012 - R. Yantosca - Initial version

03 Feb 2012 - R. Yantosca - Now pass the caller routine name as an argument

9.3.3 GeosFp_Read_cn

Routine to read variables and attributes from a GEOS-FP met fields file containing constant (CN) data.

INTERFACE:

```

SUBROUTINE GeosFp_Read_CN( Input_Opt, State_Met )

```

USES:

```

USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState

```

INPUT PARAMETERS:

```

TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object

```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
 07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
 10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
 09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
 15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
 11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
 26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_CN
 06 Nov 2014 - R. Yantosca - Replace TRANSFER_2D with direct casts

9.3.4 GeosFp_Read_a1

Routine to read variables and attributes from a GEOS-FP met fields file containing 1-hr time-averaged (A1) data.

INTERFACE:

```
SUBROUTINE GeosFp_Read_A1( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

Special handling for surface precipitation fields:

In GEOS-FP (and in MERRA), the PRECTOT etc. surface precipitation met fields fields have units of [kg/m2/s]. In all other GEOS

versions, PREACC and PRECON have units of [mm/day].

Therefore, for backwards compatibility with existing code, apply the following unit conversion to the GEOS-5 PRECTOT and PRECCON fields:

$$\frac{\text{kg} \mid \text{m}^3 \mid 86400 \text{ s} \mid 1000 \text{ mm}}{\text{m}^2 \text{ s} \mid 1000 \text{ kg} \mid \text{day} \mid \text{m}} = 86400$$

1 / density of water

REVISION HISTORY:

- 30 Jan 2012 - R. Yantosca - Initial version
- 07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
- 10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
- 09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
- 15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
- 04 Jan 2013 - M. Payer - Bug fix: Use State_Met%TSKIN for ND67 surface skin temperature diagnostic, not State_MET%TS
- 11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
- 02 Dec 2013 - S. Philip - Correction for GEOS-FP boundary layer height
- 04 Dec 2013 - R. Yantosca - Now comment out GEOS-FP BL height correction
- 06 Nov 2014 - R. Yantosca - Replace TRANSFER_2D with direct casts
- 23 Sep 2015 - E. Lundgren - Now assign SWGDN to State_Met SWGDN not RADSWG
- 03 Dec 2015 - R. Yantosca - Now open file only once per day

9.3.5 GeosFp_Read_a3

Convenience wrapper for the following routines which read 3-hour time averaged data from disk:

- GeosFp_Read_A3cld
- GeosFp_Read_A3dyn
- GeosFp_Read_A3mstC
- GeosFp_Read_A3mstE

INTERFACE:

SUBROUTINE GeosFp_Read_A3(YYYYMMDD, HHMMSS, Input_Opt, State_Met)

USES:

USE Input_Opt_Mod, ONLY : OptInput
 USE State_Met_Mod, ONLY : MetState

INPUT PARAMETERS:

```

INTEGER,          INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,          INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss  format
TYPE(OptInput),  INTENT(IN)    :: Input_Opt   ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)  :: State_Met   ! Meteorology State object

```

REVISION HISTORY:

```

30 Jan 2012 - R. Yantosca - Initial version
11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_A3

```

9.3.6 GeosFp_Read_a3cld

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr time-averaged (A3) data (cloud fields).

INTERFACE:

```

SUBROUTINE GeosFp_Read_A3cld( YYYYMMDD, HHMMSS, Input_Opt, State_Met )

```

USES:

```

USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```

INTEGER,          INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,          INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss  format
TYPE(OptInput),  INTENT(IN)    :: Input_Opt   ! Input Options object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(MetState), INTENT(INOUT)  :: State_Met   ! Meteorology State object

```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
 07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
 10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
 05 Apr 2012 - R. Yantosca - Fixed bug: TAUCLI was overwritten w/ TAUCLW
 09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
 15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
 11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
 26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_A3Cld
 06 Nov 2014 - R. Yantosca - Replace TRANSFER_A6 with TRANSFER_3D
 03 Dec 2015 - R. Yantosca - Now open file only once per day
 17 Mar 2016 - M. Sulprizio- Read optical depth into State_Met%OPTD instead of State_Met%OPTDEP (obsolete).

9.3.7 GeosFp_Read_a3dyn

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr time-averaged (A3) data (dynamics fields).

INTERFACE:

```
SUBROUTINE GeosFp_Read_A3dyn( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD   ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS     ! GMT time in hh:mm:ss  format
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
 07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
 10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
 09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
 15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
 11 Apr 2013 - R. Yantosca - Now pass directories with Input_Opt
 26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_A3dyn
 15 Nov 2013 - R. Yantosca - Now convert RH from [1] to [%], in order to be consistent with GEOS-Chem convention
 14 Aug 2014 - R. Yantosca - Now compute CLDTOPS in GeosFP_Read_A3mstE
 03 Dec 2015 - R. Yantosca - Now open file only once per day
 03 Feb 2017 - M. Sulprizio- Activate reading OMEGA fields from file

9.3.8 GeosFp_Read_a3mstc

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr time-averaged (A3) data (moist fields, saved on level centers).

INTERFACE:

```
SUBROUTINE GeosFp_Read_A3mstC( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD   ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS     ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script `ncCodeRead`, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_A3mstC
03 Dec 2015 - R. Yantosca - Now open file only once per day

9.3.9 GeosFp_Read_a3mste

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr time-averaged (A3) data (moist fields, saved on level edges).

INTERFACE:

```
SUBROUTINE GeosFp_Read_A3mstE( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput  
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD   ! GMT date in YYYY/MM/DD format  
INTEGER,      INTENT(IN)    :: HHMMSS     ! GMT time in hh:mm:ss format  
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script *ncCodeRead*, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object

15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_A3mstE
26 Sep 2013 - R. Yantosca - Now read CMFMC from GEOSFP*.nc files
14 Aug 2014 - R. Yantosca - Now compute CLDTOPS here; it depends on CMFMC
03 Dec 2015 - R. Yantosca - Now open file only once per day

9.3.10 GeosFp_Read_I3.1

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr instantaneous (I3) data.

INTERFACE:

```
SUBROUTINE GeosFp_Read_I3_1( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput  
USE State_Met_Mod,     ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD   ! GMT date in YYYY/MM/DD format  
INTEGER,      INTENT(IN)    :: HHMMSS     ! GMT time in hh:mm:ss format  
TYPE(OptInput), INTENT(IN)  :: Input_Opt  ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script ncCodeRead, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
05 Apr 2012 - R. Yantosca - Now convert QV1 from [kg/kg] to [g/kg]
09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type object
15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt

06 Sep 2013 - R. Yantosca - Bug fix: we need to initialize State_Met%T with State_Met%TMPU1 to avoid errors. The State_Met%T field will be set again in INTERP.
26 Sep 2013 - R. Yantosca - Renamed to GeosFp_Read_I3_1
29 Oct 2013 - R. Yantosca - Now read T_FULLLGRID_1 for offline simulations
06 Nov 2014 - R. Yantosca - Replace TRANSFER_2D with direct casts
16 Apr 2015 - R. Yantosca - Remove reference to T_FULLLGRID; it's obsolete
12 Jun 2015 - E. Lundgren - Initialize State_MET%SPHU with State_Met%SPHU1
03 Dec 2015 - R. Yantosca - Now open file only once per day

9.3.11 GeosFp_Read_I3_2

Routine to read variables and attributes from a GEOS-FP met fields file containing 3-hr instantaneous (I3) data.

INTERFACE:

```
SUBROUTINE GeosFp_Read_I3_2( YYYYMMDD, HHMMSS, Input_Opt, State_Met )
```

USES:

```
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Met_Mod,      ONLY : MetState
```

INPUT PARAMETERS:

```
INTEGER,      INTENT(IN)    :: YYYYMMDD    ! GMT date in YYYY/MM/DD format
INTEGER,      INTENT(IN)    :: HHMMSS      ! GMT time in hh:mm:ss format
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(MetState), INTENT(INOUT) :: State_Met ! Meteorology State object
```

REMARKS:

This routine was automatically generated by the Perl script `ncCodeRead`, and was subsequently hand-edited for compatibility with GEOS-Chem.

Even though the netCDF file is self-describing, the GEOS-FP data, dimensions, and units are pre-specified according to the GMAO GEOS-FP file specification. Therefore we can "cheat" a little bit and not have to read netCDF attributes to find what these values are.

REVISION HISTORY:

30 Jan 2012 - R. Yantosca - Initial version
07 Feb 2012 - R. Yantosca - Now echo info after reading fields from disk
10 Feb 2012 - R. Yantosca - Now get a string for the model resolution
05 Apr 2012 - R. Yantosca - Now convert QV2 from [kg/kg] to [g/kg]
09 Nov 2012 - M. Payer - Copy all met fields to the State_Met derived type

object

- 15 Nov 2012 - R. Yantosca - Now replace dao_mod.F arrays with State_Met
- 11 Apr 2013 - R. Yantosca - Now pass directory fields with Input_Opt
- 26 Sep 2013 - R. Yantosca - Rename to GeosFp_Read_I3_2
- 29 Oct 2013 - R. Yantosca - Now read T_FULLLGRID_2 for offline simulations
- 06 Nov 2014 - R. Yantosca - Replace TRANSFER_2D with direct casts
- 03 Dec 2015 - R. Yantosca - Now open file only once per day
- 20 Sep 2016 - R. Yantosca - Bug fix: FIRST must be declared as LOGICAL

9.3.12 Cleanup_GeosFp_Read

Closes any open netCDF files at the end of a simulation. This can occur if the simulation ends at a time other than 00:00 GMT.

INTERFACE:

```
SUBROUTINE Cleanup_GeosFp_Read()
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Initial version

10 Specialty simulation modules

These modules contain routines used by the GEOS-Chem "specialty" simulations.

10.1 Fortran: Module Interface *c2h6_mod.F*

Module C2H6_MOD contains variables and routines used for the tagged C2H6 (ethane) simulation. (xyp, qli, bmy, 7/28/01, 4/5/06)

INTERFACE:

```
MODULE C2H6_MOD
```

USES:

```
USE PRECISION_MOD  
USE PhysConstants, ONLY : AVO
```

```
IMPLICIT NONE  
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: GET_C2H6_ANTHRO  
PUBLIC  :: CHEMC2H6  
PUBLIC  :: CLEANUP_C2H6
```

```

PUBLIC   :: INIT_C2H6
PRIVATE DATA MEMBERS:
PRIVATE  :: NGASC2H6
PRIVATE  :: FMOL_C2H6
PRIVATE  :: XNUMOL_C2H6

```

REMARKS:

Setting LSPLIT = T in "input.geos" will run with the following tracers:

- (1) Total C2H6
- (2) C2H6 from biomass burning
- (3) C2H6 from biofuel burning
- (4) C2H6 from natural gas leaking/venting (e.g. "anthro" C2H6)

Setting LSPLIT = F in "input.geos" will run w/ the following tracers:

- (1) Total C2H6

REVISION HISTORY:

- (1) Eliminated obsolete code from 1/02 (bmy, 2/27/02)
- (2) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
- (3) Now reference BXHEIGHT and T from "dao_mod.f". Also references "error_mod.f". Removed obsolete code. Now references F90 module tracerid_mod.f". (bmy, 11/15/02)
- (4) Now references "grid_mod.f" and the new "time_mod.f" (bmy, 2/11/03)
- (5) Now references "directory_mod.f", "logical_mod.f", and "tracer_mod.f". (bmy, 7/20/04)
- (6) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (7) Now modified
- 22 Mar 2012 - M. Payer - Add function GET_C2H6_ANTHRO
- 22 Mar 2012 - M. Payer - Added ProTeX headers
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 11 Apr 2014 - R. Yantosca - Now declare INIT_C2H6 public, so that it can be called from input_mod.F
- 20 Jun 2014 - R. Yantosca - Remove obsolete code now replaced by HEMCO
- 06 Nov 2014 - M. Yannetti - Added PRECISION_MOD
- 06 Jan 2016 - E. Lundgren - Use global physical params

10.1.1 chemc2h6

Subroutine CHEMC2H6 performs C2H6 chemistry. Loss of C2H6 is via reaction with OH. (xyp, qli, bmy, 10/19/99, 7/20/04)

INTERFACE:

```

SUBROUTINE CHEMC2H6( am_I_Root, Input_Opt,
&                   State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Error_Mod,          ONLY : Error_Stop
USE HCO_Interface_Mod,  ONLY : HcoState
USE HCO_EmisList_Mod,   ONLY : HCO_GetPtr
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
USE State_Met_Mod,      ONLY : MetState
USE TIME_MOD,           ONLY : GET_MONTH, GET_TS_CHEM

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?

```

REVISION HISTORY:

- (1) Now do chemistry all the way to the model top.
 - (2) Use monthly mean OH fields for oxidation -- reference the monthly mean OH array and the routine which reads it from disk in "global_oh_mod.f" (bmy, 1/25/02)
 - (3) Now reference T from "dao_mod.f". Also make FIRSTCHEM a local SAVED variable. (bmy, 11/15/02)
 - (4) Now use functions GET_MONTH and GET_TS_CHEM from "time_mod.f".
 - (5) Now reference STT & N_TRACERS from "tracer_mod.f". Now reference LSPLIT from "logical_mod.f" (bmy, 7/20/04)
- 22 Mar 2012 - M. Payer - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET_GLOBAL_OH routine
- 06 Nov 2014 - M. Yannetti - Converted REAL*8 to REAL(fp)
- 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
- 10 Aug 2016 - R. Yantosca - Removed temporary tracer-removal code
-

10.1.2 get_c2h6_anthro

Function GET_C2H6_ANTHRO returns the monthly average anthropogenic C2H6 emissions at GEOS-Chem grid box (I,J). Data will be returned in units of [atoms C/cm2/s].

INTERFACE:

```
FUNCTION GET_C2H6_ANTHRO( I, J, N ) RESULT( C2H6_ANTHRO )
```

USES:

```
USE CMN_SIZE_MOD
USE State_Chm_Mod,      ONLY : Ind_
```

INPUT PARAMETERS:

```
INTEGER, INTENT(IN)    :: I   ! GEOS-Chem longitude index
INTEGER, INTENT(IN)    :: J   ! GEOS-Chem latitude index
INTEGER, INTENT(IN)    :: N   ! GEOS-Chem tracer index
```

RETURN VALUE:

```
REAL(fp)               :: C2H6_ANTHRO
```

REVISION HISTORY:

```
22 Mar 2012 - M. Payer      - Initial version adapted from GET_RETRO_ANTHRO
```

10.1.3 init_c2h6

Subroutine INIT_C2H6 allocates and zeroes the NGASC2H6 array, which holds global monthly mean natural gas C2H6 emissions.

INTERFACE:

```
SUBROUTINE INIT_C2H6( am_I_Root, Input_Opt, RC )
```

USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,        ONLY : ALLOC_ERR
USE Input_Opt_Mod,    ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,      INTENT(IN)    :: am_I_Root   ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN)  :: Input_Opt   ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,      INTENT(OUT)   :: RC          ! Success or failure?!
```

REMARKS:

This routine is now called from GC_INIT_EXTRA (in GeosCore/input_mod.F).

REVISION HISTORY:

(1) Now references ALLOC_ERR from "error_mod.f" (bmy, 10/15/02)
22 Mar 2012 - M. Payer - Added ProTeX headers
11 Apr 2014 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC arguments

10.1.4 cleanup_c2h6

Subroutine CLEANUP_C2H6 deallocates the natural gas C2H6 emission array.

INTERFACE:

SUBROUTINE CLEANUP_C2H6

REVISION HISTORY:

22 Mar 2012 - M. Payer - Added ProTeX headers

10.2 Fortran: Module Interface co2_mod.F

Module CO2_MOD contains variables and routines used for the CO2 simulation. A tagged CO2 simulation capability has now been added.

References:

- Andres, R.J, G. Marland, I. Fung, and E. Matthews, *A 1x1 distribution of carbon dioxide emissions from fossil fuel consumption and cement manufacture*, Glob. Biogeochem. Cycles, **10**, 419-429, 1996.
- Corbett and Koehler (2003) *Updated emissions from ocean shipping*, J. Geophys. Res., **108**, D20, 4650.
- Corbett and Koehler (2004) *Considering alternative input parameters in an activity-based ship fuel consumption and emissions model: Reply ...* J. Geophys. Res., D23303.
- Endresen et al. (2007) *A historical reconstruction of ships fuel consumption and emissions*, J. Geophys. Res. **112**, D12301.
- Kim et al. (2005) *System for assessing Aviation's Global Emissions (SAGE) Version 1.5 global Aviation Emissions Inventories for 2000-2004*
- Kim et al. (2007) *System for assessing Aviation's Global Emissions (SAGE) Part 1: Model description and inventory results*

- LeQuere et al. (2009) *Trends in the sources and sinks of carbon dioxide*, Nature Geoscience, doi:10.1038/ngeo689.
- Olsen and Randerson (2004), *Differences between surface and column atmospheric CO2 and implications for carbon cycle research*, J. Geophys. Res., **109**, D02301,
- Potter et al. (1993), *Terrestrial Ecosystem Production: A process model based on global satellite and surface data*, Glob. Biogeochem. Cycles, **7**(4), 811-841.
- Randerson, J.T, M.V. Thompson, T.J.Conway, I.Y. Fung, and C.B. Field, *The contribution of terrestrial sources and sinks to trends in the seasonal cycle of atmospheric carbon dioxide*, Glob. Biogeochem. Cycles,**11**, 535-560, 1997.
- Suntharalingam et al. (2005) *Infulence of reduced carbon emissions and oxidation on the distribution of atmospheric CO2: Implications for inversion analysis*, BGC, 19, GB4003.
- Takahashi, T, R. Feely, R. Weiss, R. Wanninkof, D. Chipman, S. Sutherland, and T. Takahashi (1997), *Global air-sea flux of CO2: An estimate based on measurements of sea-air pCO2 difference*, Proceedings of the National Academy of Sciences, **94**, 8292-8299.
- Takahashi, T, et al. (2009), *Climatological mean and decadal change in surface ocean pCO2, and net sea-air CO2 flux over the global oceans*, **Deep-Sea Research II**, doi:10.1016/j.jdsr2/2008.12.009.
- Yevich, R. and J. A. Logan, *An assesment of biofuel use and burning of agricultural waste in the developing world*, Glob. Biogeochem. Cycles, **17**, 1095, doi:10.1029/2002GB001952, 2003.
- Sausen, R. and Schumann, U. "Estimates of the Climate Response to Aircraft CO2 and NOx Emissions Scenarios", Climate Change, 44: 27-58, 2000
- Wilkerson, J.T. et al. *Analysis of emission data from global commercial Aviation: 2004 and 2006*, Atmos. chem. Phys. Disc., **10**, 2945-2983, 2010.

INTERFACE:

MODULE CO2_MOD

USES:

```
USE PhysConstants      ! Physical constants
USE inquireMod,       ONLY : findFreeLUN
USE PRECISION_MOD     ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

PUBLIC :: CLEANUP_CO2
PUBLIC :: INIT_CO2
PUBLIC :: EMISSCO2

PRIVATE MEMBER FUNCTIONS:

PRIVATE :: DEF_BIOSPH_CO2_REGIONS_F
PRIVATE :: DEF_OCEAN_CO2_REGIONS_F
PRIVATE :: DEF_FOSSIL_CO2_REGIONS_F

REMARKS:

%%
%% WARNING! Tagged CO2 simulation only work for 2 x 2.5 grid! %%
%% Someone will have to make this more general later on... %%
%%

REVISION HISTORY:

- 16 Aug 2005 - P. Suntharalingam - Initial version
- (1) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (2) Now references biomass_mod.f (bmy, 9/27/06)
- (3) Tagged CO2 capability developed (dbj)
- (4) Implemented monthly and annual fossil fuel inventories
(R.Nassar 2009-03-10)
- (5) Implemented CO2 emissions from shipping and aviation (R.Nassar 2010)
- (6) Implemented monthly CO2 chemical production and surface correction
(R.Nassar 2010)
- 25 Feb 2011 - R. Nassar - Now read updated CDIAC CO2 emissions data
- 07 Sep 2011 - P. Kasibhatla - Modified to include GFED3
- 01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire_mod.F90
- 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 20 Jun 2014 - R. Yantosca - Remove obsolete emissions code; we now use HEMCO
- 07 Nov 2014 - M. Yannetti - Added PRECISION_MOD

10.2.1 emissco2

Subroutine EMISSCO2 is the driver routine for CO2 emissions.

INTERFACE:

SUBROUTINE EMISSCO2(am_I_Root, Input_Opt,
& State_Met, State_Chm, RC)

USES:

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,          ONLY : ERROR_STOP
USE DIAGO4_MOD,         ONLY : ADO4, NDO4
USE DIAGO4_MOD,         ONLY : ADO4_plane,  ADO4_chem
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EmisList_Mod,   ONLY : HCO_GetPtr
USE HCO_STATE_MOD,      ONLY : HCO_GetHcoID
USE Input_Opt_Mod,     ONLY : OptInput
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

The initial condition for CO2 has to be at least 50 ppm or higher or else the balanced biosphere fluxes will make STT negative. (pns, bmy, 8/16/05)

REVISION HISTORY:

```

16 Aug 2005 - P. Suntharalingam - Initial version
(1 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
(2 ) We now get CO2 biomass emissions from biomass_mod.f. This allows us
    to use either GFED2 or default Duncan et al biomass emissions.
    (bmy, 9/27/06)
(3 ) Tagged tracer capability added. This requires the editable region
    files Regions_land.dat and Regions_ocean.dat in the run directory
    (rnassar,dbj, 2009)
(4 ) New tracers for emissions from international and domestic shipping,
    international and domestic aviation, and the chemical CO2 source
    from the oxidation of CO, CH4, and other organics (rnassar,dbj, 2009)
01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
09 Nov 2012 - M. Payer    - Replaced all met field arrays with State_Met
    derived type object
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
27 Aug 2014 - C. Keller   - Now interface with HEMCO.
13 Apr 2015 - R. Nassar   - Some bug fixes for HEMCO implementation,

```


e.g. unit fixes for aviation and chem source
 11 Sep 2015 - E. Lundgren - Tracer units are now kg/kg dry air (prev kg)
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.

10.2.2 def_biosph_co2_regions_f

Subroutine DEF_BIOSPH_CO2_REGIONS defines the land biospheric and ocean CO2 exchange regions.

INTERFACE:

```
SUBROUTINE DEF_BIOSPH_CO2_REGIONS_F( REGION )
```

USES:

```
USE CMN_SIZE_MOD      ! Size parameters
USE FILE_MOD,        ONLY : IOERROR
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT)  :: REGION(IIPAR,JJPAR)
```

REMARKS:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% WARNING! Tagged CO2 simulation only work for 2 x 2.5 grid! %%
%% Someone will have to make this more general later on... %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

REVISION HISTORY:

```
18 May 2010 - R. Nassar, D. Jones - Initial version
03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
13 Apr 2013 - R. Nassar - minor fixes to region boundary indices
```

10.2.3 def_ocean_co2_regions_f

Subroutine DEF_OCEAN_CO2_REGIONS defines CO2 regions for ocean exchange.

INTERFACE:

```
SUBROUTINE DEF_OCEAN_CO2_REGIONS_F( REGION )
```

USES:

```
USE CMN_SIZE_MOD      ! Size parameters
USE FILE_MOD,        ONLY : IOERROR
```

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: REGION(IIPAR,JJPARG)

REMARKS:

%%
%% WARNING! Tagged CO2 simulation only work for 2 x 2.5 grid! %%
%% Someone will have to make this more general later on... %%
%%

REVISION HISTORY:

18 May 2010 - R. Nassar, D. Jones - Initial version
03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block

10.2.4 def_fossil_co2_regions.f

Subroutine DEF_FOSSIL_CO2_REGIONS defines CO2 regions for anthropogenic emissions

INTERFACE:

SUBROUTINE DEF_FOSSIL_CO2_REGIONS_F(REGION)

USES:

USE CMN_SIZE_MOD ! Size parameters
USE FILE_MOD, ONLY : IOERROR

OUTPUT PARAMETERS:

INTEGER, INTENT(OUT) :: REGION(IIPAR,JJPARG)

REMARKS:

%%
%% WARNING! Tagged CO2 simulation only work for 2 x 2.5 grid! %%
%% Someone will have to make this more general later on... %%
%%

REVISION HISTORY:

18 May 2010 - R. Nassar, D. Jones - Initial version
03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block

10.2.5 init_co2

Subroutine INIT_CO2 allocates memory to module arrays and reads in annual mean emissions.

INTERFACE:

```
SUBROUTINE INIT_CO2( am_I_Root, Input_Opt, RC )
```

USES:

```
! References to F90 modules
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD,          ONLY : ALLOC_ERR
USE Input_Opt_Mod,      ONLY : OptInput
```

INPUT PARAMETERS:

```
LOGICAL,          INTENT(IN)  :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput),  INTENT(IN)  :: Input_Opt    ! Input Options object
```

OUTPUT PARAMETERS:

```
INTEGER,          INTENT(OUT) :: RC           ! Success or failure?
```

REVISION HISTORY:

```
16 Aug 2005 - P. Suntharalingam - Initial version
18 May 2010 - R. Nassar, D. Jones - Updated
25 Mar 2013 - R. Yantosca - Now use logical fields from Input_Opt
```

10.2.6 cleanup_co2

Subroutine CLEANUP_CO2 deallocates all module arrays.

INTERFACE:

```
SUBROUTINE CLEANUP_CO2
```

REVISION HISTORY:

```
16 Aug 2005 - P. Suntharalingam - Initial version
18 May 2010 - R. Nassar, D. Jones - Updated
```

10.3 Fortran: Module Interface exchange_mod.F90

Module EXCHANGE_MOD contains variables and routines which are used to exchange data between two or more runs (Global Domain and Nested Domains), thus to combine them into TWO-WAY nested run. (yanyy, 6/18/14)

INTERFACE:

```
MODULE EXCHANGE_MOD
```

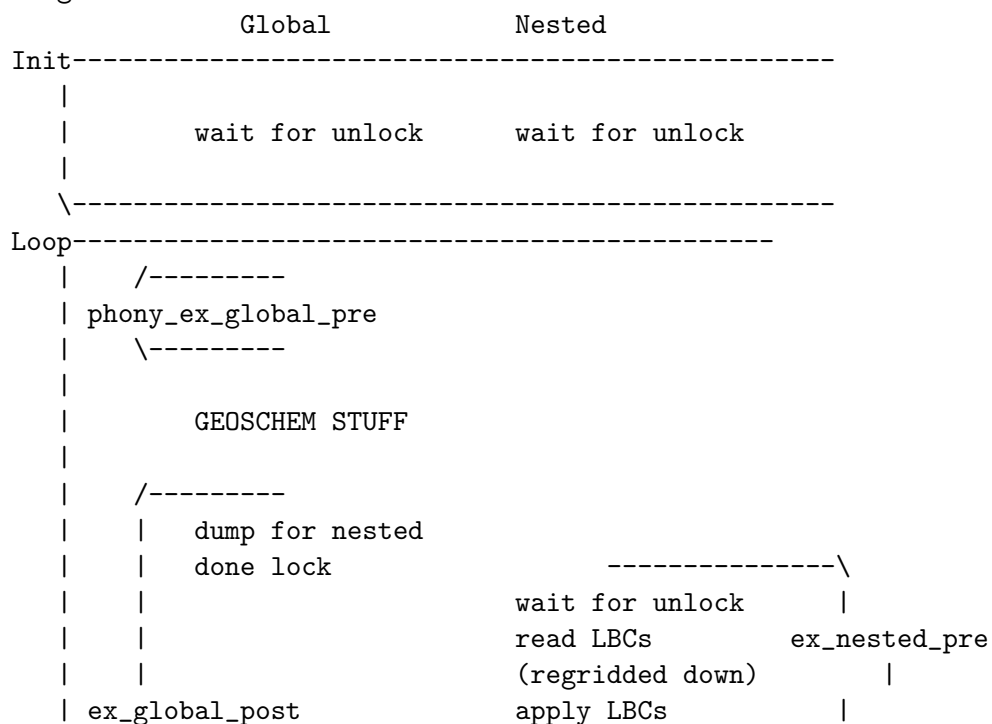
USES:

```
IMPLICIT NONE
PRIVATE
!PUBLIC MEMBER FUNCTIONS
PUBLIC    :: EXCHANGE_GLOBAL_INIT
PUBLIC    :: EXCHANGE_NESTED_INIT
PUBLIC    :: EXCHANGE_GLOBAL_POST
PUBLIC    :: EXCHANGE_NESTED_PRE
PUBLIC    :: EXCHANGE_NESTED_POST
PUBLIC    :: ITS_TIME_FOR_EXCHANGE
PUBLIC    :: INIT_EXCHANGE
PUBLIC    :: CLEANUP_EXCHANGE
```

PRIVATE MEMBER FUNCTIONS:

REMARKS:

Diagram:



```

|   |                                     -----/
|   |                                     GEOSCHEM STUFF
|   |                                     -----\
|   |                                     dump for global(regridded up)
|   |                                     ex_nested_post
|   | done lock                           |
|   | wait for unlock                       -----/
|   | read & apply feedback
|   | \-----
| \-----

```

REVISION HISTORY:

30 Mar 2014 - Y.Y. Yan - Initial Version
 (1) Establish the frame of the data exchange between models
 (kuangye,yanyy, 6/10/12)
 (2) Now update for version 9-02 (yanyy, 6/21/14)
 06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.1 exchange_global_init

Global initialization for exchange

INTERFACE:

```
SUBROUTINE EXCHANGE_GLOBAL_INIT()
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.2 exchange_nested_init

Nested initialization for exchange

INTERFACE:

```
SUBROUTINE EXCHANGE_NESTED_INIT()
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.3 exchange_global_post

Carry out the process to communicate the nested simulated results back to global model.

INTERFACE:

```

      SUBROUTINE EXCHANGE_GLOBAL_POST( am_I_Root, Input_Opt,
&                                     State_Met, State_Chm, RC )

```

USES:

```

      USE CMN_SIZE_MOD           ! Size parameters
      USE DAO_MOD
      USE ErrCode_Mod
      USE Input_Opt_Mod,        ONLY : OptInput
      USE State_Chm_Mod,       ONLY : ChmState
      USE State_Met_Mod,       ONLY : MetState
      USE TPCORE_BC_MOD
      USE UnitConv_Mod,        ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)    :: am_I_Root
      TYPE(OptInput),  INTENT(IN)    :: Input_Opt
      TYPE(MetState), INTENT(IN)    :: State_Met

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(ChmState), INTENT(INOUT)  :: State_Chm

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)   :: RC

```

REMARKS:

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers
13 Aut 2015 - E. Lundgren - Now input tracer conc in kg/kg not v/v.
26 Jun 2017 - R. Yantosca - GC_ERROR is now contained in errcode_mod.F90
28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

```

10.3.4 exchange_nested_pre

Before nested grid exchange

INTERFACE:

```

      SUBROUTINE EXCHANGE_NESTED_PRE()

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers

```

10.3.5 exchange_nested_post

After nested-grid exchange

INTERFACE:

```

      SUBROUTINE EXCHANGE_NESTED_POST( am_I_Root, Input_Opt,
&                                     State_Met, State_Chm, RC )

```

USES:

```

      USE CMN_SIZE_MOD           ! Size parameters
      USE DAO_MOD
      USE ErrCode_Mod
      USE ERROR_MOD,            ONLY : GC_Error
      USE Input_Opt_Mod,        ONLY : OptInput
      USE State_Chm_Mod,        ONLY : ChmState
      USE State_Met_Mod,        ONLY : MetState
      USE UnitConv_Mod          ONLY : Convert_Spc_Units

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)      :: am_I_Root
      TYPE(OptInput),  INTENT(IN)      :: Input_Opt
      TYPE(MetState),  INTENT(IN)      :: State_Met

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(ChmState), INTENT(INOUT)    :: State_Chm

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)    :: RC

```

RETURN VALUE:

REMARKS:

REVISION HISTORY:

```

      06 Jan 2015 - R. Yantosca - Added ProTeX headers
      13 Aug 2015 - E. Lundgren - Input tracer conc in kg/kg not v/v
      28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

```

10.3.6 its_time_for_exchange

Returns TRUE if it's time to do the 2-way exchange of data.

INTERFACE:

```
FUNCTION ITS_TIME_FOR_EXCHANGE()
```

USES:

```
USE TIME_MOD, ONLY : ITS_TIME_FOR_EXCH
```

RETURN VALUE:

```
LOGICAL :: ITS_TIME_FOR_EXCHANGE
```

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Added ProTeX headers
```

10.3.7 init_exchange

Initialization routine.

INTERFACE:

```
SUBROUTINE INIT_EXCHANGE( am_I_Root )
```

USES:

```
USE CMN_SIZE_MOD          ! Size parameters
USE ERROR_MOD
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)      :: am_I_Root
```

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Now only print output on the root CPU
06 Jan 2015 - R. Yantosca - Added ProTeX headers
24 Mar 2015 - E. Lundgren - Remove tracer_mod dependency since unused
```

10.3.8 cleanup_exchange

Cleanup routine.

INTERFACE:

```
SUBROUTINE CLEANUP_EXCHANGE( am_I_Root )
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN)      :: am_I_Root
```

REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Now only write output on the root CPU
06 Jan 2015 - R. Yantosca - Added ProTeX headers
```

10.3.9 ex_dump_for_global

?

INTERFACE:

```

SUBROUTINE EX_DUMP_FOR_GLOBAL( am_I_Root, Input_Opt,
&                               State_Met, State_Chm, RC )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE ERROR_MOD
USE Input_Opt_Mod,        ONLY : OptInput
USE State_Chm_Mod,        ONLY : ChmState
USE State_Met_Mod,        ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root
TYPE(OptInput),  INTENT(IN)    :: Input_Opt
TYPE(MetState),  INTENT(IN)    :: State_Met

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT)  :: State_Chm

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)    :: RC

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers
06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies
15 Aug 2016 - E. Lundgren - Replace tracers with species

```

10.3.10 ex_read_and_apply_feedback

?

INTERFACE:

```

SUBROUTINE EX_READ_AND_APPLY_FEEDBACK( am_I_Root, Input_Opt,
&                                       State_Met, State_Chm, RC, DOMAIN )

```

USES:

```

USE CMN_SIZE_MOD           ! Size parameters
USE ERROR_MOD
USE Input_Opt_Mod,        ONLY : OptInput
USE State_Chm_Mod,        ONLY : ChmState
USE State_Met_Mod,        ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)      :: am_I_Root
TYPE(OptInput),  INTENT(IN)      :: Input_Opt
TYPE(MetState), INTENT(IN)      :: State_Met
INTEGER,         INTENT(IN)      :: DOMAIN

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT)    :: State_Chm

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)     :: RC

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Now get LPRT from Input_Opt
06 Jan 2015 - R. Yantosca - Added ProTeX headers
24 Mar 2015 - E. Lundgren - Remove tracer_mod dependency since not used
06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies
15 Aug 2016 - E. Lundgren - Replace tracers with species

```

10.3.11 ex_regrid_up

This subroutine will be called within EX_DUMP_FOR_GLOBAL, Using module's global vars: REGRID_MAT_L, REGRID_MAT_R

INTERFACE:

```

SUBROUTINE EX_REGRID_UP( NX_ORIG, NY_ORIG, NX_NEW, NY_NEW,
&                        NZ, ORIG_ARRAY, NEW_ARRAY )

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers

```

10.3.12 ex_convert_units_cspect

Converts molec/cm3/box to fakemass (e.g. convert cspectfull to a masslike unit)

INTERFACE:

```

SUBROUTINE EX_CONVERT_UNITS_CSPEC( am_I_Root, Input_Opt,
&                                  State_Met, State_Chm, RC, FLAG, NX, NY, NZ )

```

USES:

```

USE CMN_SIZE_MOD          ! Size parameters
USE DAO_MOD
USE Input_Opt_Mod,       ONLY : OptInput
USE State_Chm_Mod,       ONLY : ChmState
USE State_Met_Mod,       ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root
TYPE(OptInput),  INTENT(IN)    :: Input_Opt
TYPE(MetState),  INTENT(IN)    :: State_Met

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState),  INTENT(INOUT) :: State_Chm

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC

```

RETURN VALUE:**REMARKS:****REVISION HISTORY:**

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers
06 Jun 2016 - M. Sulprizio- Replace NTSPEC with State_Chm%nSpecies

```

10.3.13 wait_for_unlock

Waits for MPI unlock.

INTERFACE:

```

SUBROUTINE WAIT_FOR_UNLOCK()

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers

```

10.3.14 done_lock_me_up

?

INTERFACE:

```

SUBROUTINE DONE_LOCK_ME_UP()

```

REVISION HISTORY:

```

06 Jan 2015 - R. Yantosca - Added ProTeX headers

```

10.3.15 no_more_lock

?

INTERFACE:

```
SUBROUTINE NO_MORE_LOCK()
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.16 alert_for_kill_command

Stops if there is a kill command issued.

INTERFACE:

```
SUBROUTINE ALERT_FOR_KILL_COMMAND()
```

USES:

```
USE ERROR_MOD
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.17 touch_file

Touches a file (i.e. updates the timestamp on disk).

INTERFACE:

```
SUBROUTINE TOUCH_FILE( FNAME )
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)      :: FNAME
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.18 open_bin_file_for_read

Opens a binary file for reading.

INTERFACE:

```
SUBROUTINE OPEN_BIN_FILE_FOR_READ( FNAME )
```

USES:

```
USE FILE_MOD
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.19 open_bin_file_for_write

Opens a binary file for writing.

INTERFACE:

```
SUBROUTINE OPEN_BIN_FILE_FOR_WRITE( FNAME )
```

USES:

```
USE FILE_MOD
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)      :: FNAME
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.20 open_bin_file_for_append

Opens a binary file for appending.

INTERFACE:

```
SUBROUTINE OPEN_BIN_FILE_FOR_APPEND( FNAME )
```

USES:

```
USE FILE_MOD
```

INPUT PARAMETERS:

```
CHARACTER(LEN=*), INTENT(IN)      :: FNAME
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.21 close_bin_file

Closes a binary file.

INTERFACE:

```
SUBROUTINE CLOSE_BIN_FILE()
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.22 write_3d_real8_array

Writes data to a 3-D REAL*8 array.

INTERFACE:

```
SUBROUTINE WRITE_3D_REAL8_ARRAY( NI, NJ, NK, ARRAY )
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)      :: NI, NJ, NK  
REAL*8,          INTENT(IN)      :: ARRAY(NI,NJ,NK)
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.23 read_3d_real8_array

Reads data from 3-D REAL*8 array.

INTERFACE:

```
SUBROUTINE READ_3D_REAL8_ARRAY( NI, NJ, NK, ARRAY )
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)      :: NI, NJ, NK
```

OUTPUT PARAMETERS:

```
REAL*8,          INTENT(OUT)     :: ARRAY(NI,NJ,NK)
```

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.3.24 gen_regrid_mat

Regridding utility.

INTERFACE:

```
SUBROUTINE GEN_REGRID_MAT( N1, N2, RATIO, OFFSET, RES )
```

INPUT PARAMETERS:

```
INTEGER,          INTENT(IN)      :: N1, N2  
REAL*8,          INTENT(IN)      :: RATIO, OFFSET
```

OUTPUT PARAMETERS:

```
REAL*8,          INTENT(OUT)     :: RES(N1, N2)
```

REMARKS:

REVISION HISTORY:

06 Jan 2015 - R. Yantosca - Added ProTeX headers

10.4 Fortran: Module Interface *global_ch4_mod.F*

Module GLOBAL_CH4_MOD contains variables and routines for simulating CH₄ chemistry in the troposphere.

INTERFACE:

```
MODULE GLOBAL_CH4_MOD
```

USES:

```
USE HCO_ERROR_MOD      ! For HEMCO error reporting  
USE PhysConstants, ONLY : AVO, AIRMW  
USE PRECISION_MOD     ! For GEOS-Chem Precision (fp, f4, f8)
```

```
IMPLICIT NONE  
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: EMISSCH4  
PUBLIC :: CHEMCH4  
PUBLIC :: INIT_GLOBAL_CH4  
PUBLIC :: CLEANUP_GLOBAL_CH4
```

PUBLIC DATA MEMBERS:

```

REAL(fp), PARAMETER, PUBLIC :: XNUMOL_CH4 = AVO / 16d-3 ! hard-coded MW

! Make CH4_EMIS now public so that it can be used by vdiff_mod.F90
! Methane emissions units are [kg/m2/s]
REAL(fp), ALLOCATABLE, PUBLIC :: CH4_EMIS(:,:,:)

```

REVISION HISTORY:

- 17 Jan 2001- J. Wang, B. Duncan, R. Yantosca -- Initial version
- (1) Merged routines from jsw's CH4 code into "global_ch4_mod.f"
(bmy, 1/16/01)
- (2) XNUMOL_CH4 and TCH4 have to be public - all other variables can
be made private, so as not to conflict with other common-block
definitions (bmy, 1/17/01)
- (3) Minor fixes from jsw added (jsw, bmy, 2/17/01)
- (4) Removed some F90 module references from EMISSCH4 (bmy, 3/20/01)
- (5) Eliminate obsolete commented-out code (bmy, 4/20/01)
- (6) Updated comments (bmy, 9/4/01)
- (7) Fixes for binary punch file in READ_COPROD (bmy, 9/26/01)
- (8) Removed obsolete code from READ_COPROD (bmy, 10/24/01)
- (9) Minor bug fixes for compilation on ALPHA (bmy, 11/15/01)
- (10) Eliminate obsolete code from 11/01 (bmy, 2/27/02)
- (11) Now eliminate PS from the arg list to CH4_AVGTP (4/11/02)
- (12) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and
MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
- (13) Replaced all instances of IM with IIPAR and JM with JJPARG, in order
to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (14) Now reference "file_mod.f". Also removed obsolete code. (bmy, 6/27/02)
- (15) Now references "pressure_mod.f" (bmy, 8/21/02)
- (16) Now reference AD and T from "dao_mod.f". Now reference "error_mod.f".
Remove obsolete code from various routines. Remove reference to
header file "comtrid.h" -- it's not used. (bmy, 11/6/02)
- (17) Minor bug fix in FORMAT statements (bmy, 3/23/03)
- (18) Now references "grid_mod.f" and "time_mod.f" (bmy, 3/27/03)
- (19) Updates to GET_GLOBAL_CH4 (bmy, 7/1/03)
- (20) Now references "directory_mod.f", "tracer_mod.f", and "diag_oh_mod.f"
(bmy, 7/20/04)
- (21) Now can read data for both GEOS and GCAP grids (bmy, 8/16/05)
- (22) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (23) Updated CH4 simulation (kju, cph, ccarouge, 10/1/09)
- (24) Added modifications for MERRA (bmy, 8/13/10)
- 08 Feb 2012 - R. Yantosca - Added modifications for GEOS-5.7.x
- 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
- 07 Mar 2012 - M. Payer - Added ProTeX headers
- 01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire_mod.F90
- 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 12 Feb 2014 - K. Wecht - Add modifications for 0.25 x 0.3125 NA grid

12 Feb 2014 - K. Wecht - Disable CH4 budget diagnostic (bracket the code out with #ifdef blocks so it can be used)

20 Jun 2014 - R. Yantosca - Remove obsolete emissions code; we now use HEMCO

09 Sep 2014 - C. Keller - Implemented HEMCO for emissions.

16 Sep 2014 - C. Keller - Now also get chemical fields (OH and CH4 prod) through HEMCO. Removed READ_COPROD and READ_CH4LOSS (now obsolete)

16 Sep 2014 - C. Keller - Removed all references to Clarisa's OH climatology (not used anywhere)

14 Nov 2014 - M. Yannetti - Added PRECISION_MOD

04 Mar 2015 - R. Yantosca - Now use REAL(f4) for HCO_GetPtr pointer args

06 Jan 2016 - E. Lundgren - Use global physical parameters

16 Jun 2016 - M. Sulprizio- Now define IDTCH4 locally

20 Jun 2016 - R. Yantosca - Rename IDTCH4 to id_CH4 for consistency

29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90

16 Jun 2017 - M. Sulprizio- Remove routines CH4_AVGTP, GET_CH4_ANTHRO, and GET_CH4_BIOG - they are not used anymore

18 Oct 2017 - M. Sulprizio- Remove CH4_BUDGET code. It did not work anymore and was in need of rewriting. This information can be calculated using GEOS-Chem output.

BOC

DEFINED PARAMETERS:

```

!=====
! Module Variables:
! BAIRDENS   : Array for air density           [molec/cm3]
! BOH        : Array for OH values             [kg/m3]
! BC1        : Array for Cl values             [ppbv]
! COPROD     : Array for zonal mean P(CO)      [v/v/s]
! CH4LOSS    : Array for monthly average CH4 loss freq [1/s]
! FMOL_CH4   : Molecular weight of CH4        [kg/mole]
! XNUMOL_CH4 : Molecules CH4 / kg CH4         [molec/kg]
! CH4_EMIS   : Array for CH4 Emissions        [kg/m2/s]
!=====

REAL(fp), PARAMETER  :: XNUMOL_OH = AVO / 17e-3_fp ! molec OH / kg OH
                                                           ! hard-coded MW

REAL(fp), PARAMETER  :: CM3PERM3 = 1.e+6_fp

```

LOCAL VARIABLES:

```

! Diagnostic flags
LOGICAL          :: Do_ND43
LOGICAL          :: Archive_OHconcAfterChem

! Species ID flag

```

```

      INTEGER                :: id_CH4

      ! Various arrays
      REAL(fp), ALLOCATABLE :: BAIRDENS(:,:,:)

      ! Pointers to fields in the HEMCO data structure.
      ! These need to be declared as REAL(f4), aka REAL*4.
      ! NOTE: These are globally SAVED variables so we can
      ! nullify these in the declaration statement (bmy, 4/29/16)
      REAL(f4), POINTER      :: BOH      (:,,:,:) => NULL()
      REAL(f4), POINTER      :: BCl     (:,,:,:) => NULL()
      REAL(f4), POINTER      :: CH4LOSS(:,,:,:) => NULL()
      REAL(f4), POINTER      :: CLUSTERS(:,:) => NULL()

      REAL(fp)                :: TROPOCH4

      CONTAINS

EOC
-----
                        GEOS-Chem Global Chemical Transport Model                !
-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\mbox{}\hrulefill\

\subsubsection [emissch4] {emissch4}

Subroutine EMISSCH4 places emissions of CH4 [kg] into the
chemical species array.
\\
\\{\bf INTERFACE:}
\begin{verbatim}
      SUBROUTINE EMISSCH4( am_I_Root, Input_Opt,
&                          State_Met, RC )

```

USES:

```

      USE CMN_SIZE_MOD
      #if defined( BPCH_DIAG )
      USE CMN_DIAG_MOD
      USE DIAG_MOD,          ONLY : AD58
      #endif
      USE HCO_INTERFACE_MOD, ONLY : HcoState, GetHcoDiagn
      USE ErrCode_Mod
      USE ERROR_MOD,        ONLY : ERROR_STOP
      USE GC_GRID_MOD,     ONLY : GET_XOFFSET, GET_YOFFSET
      USE Input_Opt_Mod,   ONLY : OptInput
      USE State_Met_Mod,   ONLY : MetState

```

INPUT PARAMETERS:

```

LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(Optional),  INTENT(IN)    :: Input_Opt    ! Input Options object
TYPE(MetState),  INTENT(IN)    :: State_Met     ! Meteorology State object

```

OUTPUT PARAMETERS:

```

INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

WARNING: Soil absorption has to be the 15th field in CH4_EMIS

REVISION HISTORY:

- (1) Created by Bryan Duncan (1/99). Adapted for CH4 chemistry by James Wang (7/00). Inserted into module "global_ch4_mod.f" by Bob Yantosca. (bmy, 1/16/01)
- (2) EMISSCH4 is independent of "CMN_OH", "CMN_CO", and "CMN_CO_BUDGET". (bmy, 1/16/01)
- (3) GLOBSEAEMIS, GLOBSEAEMIS are diagnostics by jsw.
- (4) Do not multiply CO emissions by 1.28 anymore (jsw, bmy, 2/12/01)
- (5) Renamed input files to CH4_monthly.geos.{RES} and CH4_aseasonal.geos.{RES}. (bmy, 2/12/01)
- (6) Add reference to "CMN_SETUP" for the DATA_DIR variable (bmy, 2/13/01)
- (7) Removed references to "biofuel_mod.f" and "biomass_mod.f"; these weren't necessary (bmy, 3/20/01)
- (8) Now reference IU_FILE and IOERROR from "file_mod.f". Now use IU_FILE instead of IUNIT as the file unit #. (bmy, 6/27/02)
- (9) Now reference BXHEIGHT and SUNCOS from "dao_mod.f". Remove reference to header file "comtrid.h" -- it's not used. Make FIRSTEMISS a local SAVED variable. Also use MONTH from "CMN" instead of the variable LMN. (bmy, 11/15/02)
- (10) Now replace DXYP(JREF)*1d4 with routine GET_AREA_CM2 of "grid_mod.f". Now use function GET_MONTH and GET_TS_EMIS from "time_mod.f". Now use functions GET_XOFFSET and GET_YOFFSET from "grid_mod.f". IO and JO are now local variables. (bmy, 3/27/03)
- (11) Now reference STT from "tracer_mod.f". Now reference DATA_DIR from "directory_mod.f". (bmy, 7/20/04)
- (12) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (13) Add non-local PBL capability (ccc, 8/31/09)
- 01 Mar 2012 - R. Yantosca - Now use GET_AREA_CM2(I,J,L) from grid_mod.F90
- 07 Mar 2012 - M. Payer - Added ProTeX headers
- 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
- 11 Apr 2014 - R. Yantosca - Remove call to INIT_GLOBAL_CH4
- 09 Sep 2014 - C. Keller - Implemented HEMCO
- 11 Sep 2015 - E. Lundgren - Remove area-dependency (except for global mass sums) by outputting diagnostics as kg/m2 not kg
- 11 Sep 2015 - E. Lundgren - Remove State_Chm from arg list since not used

10.4.1 chemch4

Subroutine CHEMCH4 computes the chemical loss of CH4 (sources - sinks). (jsw, bnd, bmy, 6/8/00, 10/3/05)

INTERFACE:

```

      SUBROUTINE CHEMCH4( am_I_Root, Input_Opt, State_Met,
&                        State_Chm, State_Diag, RC           )

```

USES:

```

      USE CMN_SIZE_MOD
      #if defined( BPCH_DIAG )
      USE CMN_DIAG_MOD
      USE DIAG_MOD,          ONLY : AD43
      #endif
      USE ErrCode_Mod
      USE ERROR_MOD,        ONLY : GEOS_CHEM_STOP, ERROR_STOP
      USE HCO_INTERFACE_MOD, ONLY : HcoState
      USE HCO_EMITLIST_MOD, ONLY : HCO_GetPtr
      USE Input_Opt_Mod,    ONLY : OptInput
      USE State_Chm_Mod,    ONLY : ChmState
      USE State_Diag_Mod,   ONLY : DgnState
      USE State_Met_Mod,    ONLY : MetState

```

INPUT PARAMETERS:

```

      LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
      TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
      TYPE(MetState),  INTENT(IN)    :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

      TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
      TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

      INTEGER,          INTENT(OUT)   :: RC          ! Success or failure?

```

REMARKS:

CH4 SOURCES

- ```

=====
(1) Oxidation of methane, isoprene and monoterpenes (SRCO_fromHCs).
(2) Direct emissions of CO from fossil fuel combustion, biomass
 burning and wood (for fuel) burning (SR SETEMIS).
(3) Emissions.

```

## CH4 SINKS:

- ```
=====
```
- (1) Removal of CO by OH (SR OHparam & CO_decay).
 - (2) CO uptake by soils (neglected).
 - (3) Transport of CO to stratosphere from troposphere
(in dynamical subroutines).
 - (4) Removal by OH (Clarissa's OH--climatol_OH.f and CO_decay.f)
 - (5) Transport of CH4 between troposphere and stratosphere, and
destruction in strat (CH4_strat.f).
 - (6) Removal by Cl

REVISION HISTORY:

- (1) Created by Bryan Duncan (1/99). Adapted for CH4 chemistry by
James Wang (6/8/00). Inserted into module "global_ch4_mod.f"
by Bob Yantosca. (bmy, 1/16/01)
- (2) CHEMCH4 is independent of "CMN_OH", "CMN_CO", and "CMN_CO_BUDGET".
(bmy, 1/16/01)
- (3) Updated comments (jsw, bmy, 2/12/01)
- (4) LD43 is already declared in CMN_DIAG; don't redefine it (bmy, 11/15/01)
- (5) Replaced all instances of IM with IIPAR and JM with JJPARG, in order
to prevent namespace confusion for the new TPCORE (bmy, 6/25/02)
- (6) Now reference AD from "dao_mod.f". Now reference GEOS_CHEM_STOP from
"error_mod.f" Now make FIRSTCHEM a local SAVED variable. Now
reference ALBD from "dao_mod.f". Now use MONTH and JDATE from "CMN"
instead of LMN and LDY. (bmy, 11/15/02)
- (7) Remove NYMdb, NYMde from the arg list. Now use functions GET_MONTH,
GET_NYMdb, GET_NYMde, GET_MONTH, GET_DAY from the new "time_mod.f"
(bmy, 3/27/03)
- (8) Now reference DATA_DIR from "directory_mod.f" (bmy, 7/20/04)
- (9) Remove reference to BPCH2_MOD, it's not needed (bmy, 10/3/05)
- 07 Mar 2012 - M. Payer - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
derived type object
- 25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET_GLOBAL_OH routine
- 12 Feb 2014 - K. Wecht - Disable CH4 budget diagnostic (bracket the
code out with #ifdef blocks so it can be used)
- 23 Jul 2014 - R. Yantosca - Remove reference to obsolete CMN_mod.F
- 23 Jul 2014 - R. Yantosca - Reference ITS_IN_THE_CHEMGRID (chemgrid_mod.F)
- 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
- 17 Sep 2014 - C. Keller - Now use HEMCO to get CH4 loss and OH conc. field.
- 06 Jan 2016 - E. Lundgren - Use global physical parameters
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
species ID from State_Chm%Map_Advect.
- 10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code
- 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

10.4.2 ch4_decay

Subroutine CH4_DECAY calculates the decay rate of CH4 by OH. OH is the only sink for CH4 considered here. (jsw, bnd, bmy, 1/16/01, 7/20/04)

INTERFACE:

```
SUBROUTINE CH4_DECAY( State_Met, State_Chm )
```

USES:

```
USE CMN_SIZE_MOD
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD
USE DIAG_MOD,          ONLY : AD19
#endif
USE State_Chm_Mod,     ONLY : ChmState
USE State_Met_Mod,     ONLY : MetState
USE TIME_MOD,          ONLY : GET_TS_CHEM, ITS_A_NEW_YEAR
USE TIME_MOD,          ONLY : GET_MONTH
```

INPUT PARAMETERS:

```
TYPE(MetState), INTENT(IN)  :: State_Met  ! Meteorology State object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object
```

REMARKS:

We now use function ITS_IN_THE_CHEMGRID from chemgrid_mod.F to diagnose if box (I,J,L) is in the troposphere or stratosphere.

Monthly loss of CH4 is summed in TCH4(3)
TCH4(3) = CH4 sink by OH

REVISION HISTORY:

- (1) Created by Bryan Duncan (1/99). Adapted for CH4 chemistry by James Wang (7/00). Inserted into module "global_ch4_mod.f" by Bob Yantosca. (bmy, 1/16/01)
 - (2) CH4_DECAY is independent of "CMN_OH", "CMN_CO", and "CMN_CO_BUDGET". (bmy, 1/16/01)
 - (3) Now use function GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
 - (4) Now references STT from "tracer_mod.f" (bmy, 7/20/04)
- 07 Mar 2012 - M. Payer - Added ProTeX headers
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met

derived type object

12 Feb 2014 - K. Wecht - Disable CH4 budget diagnostic (bracket the code out with #ifdef blocks so it can be used)

24 Jul 2014 - R. Yantosca - Now compute BOXVL internally

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.

16 Jun 2017 - M. Sulprizio- Add loss of CH4 by Cl

10.4.3 ch4_ohsave

Subroutine CH4_OHSAVE archives the CH3CCl3 lifetime from the OH used in the CH4 simulation. (bnd, jsw, bmy, 1/16/01, 7/20/04)

INTERFACE:

```
SUBROUTINE CH4_OHSAVE( State_Met, State_Chm )
```

USES:

```
USE CMN_SIZE_MOD
USE DIAG_OH_MOD,      ONLY : DO_DIAG_OH_CH4
USE GC_GRID_MOD,      ONLY : GET_AREA_CM2
USE State_Chm_Mod,    ONLY : ChmState
USE State_Met_Mod,    ONLY : MetState
```

INPUT PARAMETERS:

```
TYPE(MetState), INTENT(IN)  :: State_Met  ! Meteorology State object
TYPE(ChmState), INTENT(IN)  :: State_Chm   ! Chemistry State object
```

REMARKS:

We now use function ITS_IN_THE_CHEMGRID from chemgrid_mod.F to diagnose if box (I,J,L) is in the troposphere or stratosphere.

REVISION HISTORY:

- (1) Created by Bryan Duncan (1/99). Adapted for CH4 chemistry by James Wang (7/00). Inserted into module "global_ch4_mod.f" by Bob Yantosca. (bmy, 1/16/01)
 - (2) CH4_OHSAVE is independent of "CMN_OH", "CMN_CO", and "CMN_CO_BUDGET". (bmy, 1/16/01)
 - (3) Now call DO_DIAG_OH_CH4 to pass OH diagnostic info to the "diag_oh_mod.f" (bmy, 7/20/04)
- 07 Mar 2012 - M. Payer - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally

30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
 14 Dec 2017 - M. Sulprizio- Parallelize DO loop

10.4.4 ch4_strat

Subroutine CH4_STRAT calculates uses production rates for CH4 to calculate loss of CH4 in above the tropopause. (jsw, bnd, bmy, 1/16/01, 7/20/04)

INTERFACE:

```

SUBROUTINE CH4_STRAT( am_I_Root, Input_Opt,
&                   State_Met, State_Chm, RC )

```

USES:

```

USE CMN_DIAG_MOD
USE CMN_SIZE_MOD
#if defined( BPCH_DIAG )
USE DIAG_MOD,          ONLY : AD19
#endif
USE ErrCode_Mod

```

Prior to 12/11/17:

Comment this out to speed up GEOS-Chem (bmy, 12/11/17)

```

USE ERROR_MOD,          ONLY : CHECK_SPC

```

```

USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
USE State_Met_Mod,      ONLY : MetState
USE TIME_MOD,           ONLY : GET_TS_CHEM

```

INPUT PARAMETERS:

```

LOGICAL,      INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN)  :: Input_Opt    ! Input options
TYPE(MetState), INTENT(IN)  :: State_Met    ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm  ! Chemistry State object

```

OUTPUT PARAMETERS:

```

INTEGER,      INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

Production (mixing ratio/sec) rate provided by Dylan Jones.
Only production by CH₄ + OH is considered.

We now use function ITS_IN_THE_CHEMGRID from chemgrid_mod.F to diagnose if box (I,J,L) is in the troposphere or stratosphere.

REVISION HISTORY:

- (1) Created by Bryan Duncan (1/99). Adapted for CH₄ chemistry by James Wang (7/00). Inserted into module "global_ch4_mod.f" by Bob Yantosca. (bmy, 1/16/01)
- (2) CH₄_STRAT is independent of "CMN_OH", "CMN_CO", and "CMN_CO_BUDGET". (bmy, 1/16/01)
- (3) Removed LMN from the arg list and made it a local variable. Now use functions GET_MONTH and GET_TS_CHEM from "time_mod.f" (bmy, 3/27/03)
- (4) Now references STT from "tracer_mod.f" (bmy, 7/20/04)
- 07 Mar 2012 - M. Payer - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met derived type object
- 23 Jun 2014 - R. Yantosca - Now accept am_I_Root and RC
- 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
- 24 Mar 2015 - E. Lundgren - Now pass Input_Opt to Check_STT
- 06 Jan 2016 - E. Lundgren - Use global physical parameters
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
- 14 Dec 2017 - M. Sulprizio- Parallelize DO loop

10.4.5 ch4_distrib

Subroutine CH₄_DISTRIB allocates the chemistry sink to different emission species. (ccc, 10/2/09)

INTERFACE:

```
SUBROUTINE CH4_DISTRIB( PREVCH4, Input_Opt, State_Chm )
```

USES:

```
USE CMN_SIZE_MOD
USE ERROR_MOD,          ONLY : SAFE_DIV
USE Input_Opt_Mod,      ONLY : OptInput
USE State_Chm_Mod,      ONLY : ChmState
```

```
IMPLICIT NONE
```

INPUT PARAMETERS:

```
REAL(fp)                :: PREVCH4(IIPAR,JJPAR,LLPAR)! CH4 bef chem
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

REVISION HISTORY:

```
07 Mar 2012 - M. Payer - Added ProTeX headers
25 Mar 2013 - R. Yantosca - Now accept Input_Opt, State_Chm args
20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
species ID from State_Chm%Map_Advect.
```

10.4.6 init_global_ch4

Subroutine INIT_GLOBAL_CH4 allocates and zeroes module arrays.

INTERFACE:

```
SUBROUTINE INIT_GLOBAL_CH4( am_I_Root, Input_Opt, State_Diag, RC )
```

USES:

```
USE CMN_SIZE_MOD
#if defined( BPCH_DIAG )
USE CMN_DIAG_MOD
#endif
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
```

INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
```

OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

REMARKS:

This routine is called from GC_INIT_EXTRA (in GeosCore/input_mod.f)

REVISION HISTORY:


```
USE OCEAN_MERCURY_MOD, ONLY : LRED_CLOUDONLY
USE OCEAN_MERCURY_MOD, ONLY : LBRYCHEM
USE OCEAN_MERCURY_MOD, ONLY : LCLCHEM
USE OCEAN_MERCURY_MOD, ONLY : LHGAQCHEM
USE OCEAN_MERCURY_MOD, ONLY : LHg2HalfAerosol
USE OCEAN_MERCURY_MOD, ONLY : STRAT_BR_FACTOR
USE OCEAN_MERCURY_MOD, ONLY : LAnthroHgOnly
USE OCEAN_MERCURY_MOD, ONLY : LOHO3CHEM
USE OCEAN_MERCURY_MOD, ONLY : LGCBROMINE
USE OCEAN_MERCURY_MOD, ONLY : LnoUSAemis
USE OCEAN_MERCURY_MOD, ONLY : LBROCHEM
USE OCEAN_MERCURY_MOD, ONLY : LNEI2005
USE OCEAN_MERCURY_MOD, ONLY : LInPlume
USE OCEAN_MERCURY_MOD, ONLY : LOCEANCOEF
USE PhysConstants           ! Physical constants
USE PRECISION_MOD           ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

PUBLIC MEMBER FUNCTIONS:

```
PUBLIC  :: CHEMMERCURY
PUBLIC  :: CLEANUP_MERCURY
PUBLIC  :: INIT_MERCURY
PUBLIC  :: EMISSMERCURY
PUBLIC  :: PARTITIONHG2
```

PRIVATE MEMBER FUNCTIONS:

```
PRIVATE :: EMITHG
PRIVATE :: SRCHg0
PRIVATE :: SRCHg2
PRIVATE :: SRCHgP
PRIVATE :: MERCURY_READYR
PRIVATE :: OHNO3TIME
PRIVATE :: CALC_HG2_SEASALT_LOSSRATE
PRIVATE :: RED_INPLUME_GRID
PRIVATE :: DO_RED_INPLUME
PRIVATE :: GET_HGBR_RATE
PRIVATE :: GET_O3
PRIVATE :: GET_OH
PRIVATE :: GET_BR
PRIVATE :: GET_JNO2
PRIVATE :: GET_HGCL_RATE
PRIVATE :: GET_HGBRY_RATE
PRIVATE :: GET_HGAQ_RATE
PRIVATE :: GET_NO2
PRIVATE :: GET_HO2
```

PRIVATE :: GET_C1
PRIVATE :: GET_C10
PRIVATE :: GET_HOCl

REMARKS:

Nomenclature:

```
=====
(1 ) Hg(0)  a.k.a. Hg0   : Elemental  mercury
(2 ) Hg(II) a.k.a. Hg2   : Divalent   mercury
(3 ) HgP    : Particulate mercury
```

Mercury Species (1-3 are always defined; 4-87 are defined for tagged runs)

```
=====
(1 ) Hg(0)           : Hg(0) - total species
(2 ) Hg(II)          : Hg(II) - total species
(3 ) HgP             : HgP   - total species
-----+-----
(4 ) Hg0_can         : Hg(0) - Canadian Anthropogenic
(5 ) Hg0_usa         : Hg(0) - USA Anthropogenic
(6 ) Hg0_cam         : Hg(0) - Central American Anthropogenic
(7 ) Hg0_sam         : Hg(0) - South American Anthropogenic
(8 ) Hg0_waf         : Hg(0) - West African Anthropogenic
(9 ) Hg0_eaf         : Hg(0) - East African Anthropogenic
(10) Hg0_saf         : Hg(0) - South African Anthropogenic
(11) Hg0_naf         : Hg(0) - North African Anthropogenic
(12) Hg0_eur         : Hg(0) - OECD European Anthropogenic
(13) Hg0_eeu        : Hg(0) - Eastern European Anthropogenic
(14) Hg0_sov        : Hg(0) - Former USSR Anthropogenic
(15) Hg0_mde        : Hg(0) - Middle Eastern Anthropogenic
(16) Hg0_sas        : Hg(0) - South Asian Anthropogenic
(17) Hg0_eas        : Hg(0) - East Asian Anthropogenic
(18) Hg0_sea        : Hg(0) - Southeast Asian Anthropogenic
(19) Hg0_jpn        : Hg(0) - Japanese Anthropogenic
(20) Hg0_ocn        : Hg(0) - Oceanian Anthropogenic
(21) Hg0_so         : Hg(0) - Organic Soils
(22) Hg0_bb         : Hg(0) - Biomass Burning
(23) Hg0_geo        : Hg(0) - Geogenic
(24) Hg0_atl        : Hg(0) - Middle Atlantic Subsurface Waters
(25) Hg0_nat        : Hg(0) - North Atlantic Subsurface Waters
(26) Hg0_sat        : Hg(0) - South Atlantic Subsurface Waters
(27) Hg0_npa        : Hg(0) - North Pacific Subsurface Waters
(28) Hg0_arc        : Hg(0) - Arctic Subsurface Waters
(29) Hg0_ant        : Hg(0) - Antarctic Subsurface Waters
(30) Hg0_oce        : Hg(0) - Indo-Pacific Subsurface Waters
(31) Hg0_str        : Hg(0) - Stratospheric Hg from Intial Conditions
(32-59) Same as (4-31) but for Hg(II)
```

(60-87) Same as (4-31) but for Hg(P)

References:

- =====
- (1) Hall, B. (1995). "The gas phase oxidation of elemental mercury by ozone.", *Water, Air, and Soil Pollution* 80: 301-315.
 - (2) Sommar, J., et al. (2001). "A kinetic study of the gas-phase reaction between the hydroxyl radical and atomic mercury." *Atmospheric Environment* 35: 3049-3054.
 - (3) Selin, N., et al. (2007). "Chemical cycling and deposition of atmospheric mercury: Global constraints from observations." *J. Geophys. Res.* 112.
 - (4) Selin, N., et al. (2008). "Global 3-D land-ocean-atmosphere model for mercury: present-day versus preindustrial cycles and anthropogenic enrichment factors for deposition." *Global Biogeochemical Cycles* 22: GB2011.
 - (5) Allison, J.D. and T.L. Allison (2005) "Partition coefficients for metals in surface water, soil and waste." Rep. EPA/600/R-05/074, US EPA, Office of Research and Development, Washington, D.C.
 - (6) Mintz, Y and G.K. Walker (1993). "Global fields of soil moisture and land surface evapotranspiration derived from observed precipitation and surface air temperature." *J. Appl. Meteorol.* 32 (8), 1305-1334.
 - (7) Soerensen, A. et al. (2010), An improved global model for air-sea exchange of mercury: High concentrations over the North Atlantic, *Environ. Sci. Technol.*, 44, 8574-8580.
 - (8) Corbitt, E.S. et al. (2011), Global source-receptor relationships for mercury deposition under present-day and 2050 emissions scenarios, *Environ. Sci. Technol.*, 45, 10477-10484.
 - (9) Street, D.G. et al. (2009), Projections of global mercury emissions in 2050, *Environ. Sci. Technol.*, 43, 2983-2988.
 - (10) Holmes, C.D., et al. (2010) Global atmospheric model for mercury including oxidation by bromine atoms, *AC&P*, 10, 12,037-12,057.
 - (11) Parrella, J. et al. (2012), Tropospheric bromine chemistry: implications for present and pre-industrial ozone and mercury, *ACP*.
 - (12) Pohler, D. et al. (2010), Observation of halogen species in the Amundsen Gulf, Arctic, by active long-path differential optical absorption spectroscopy, *Proc. Natl. Acad. Sci*, 107(15): 6528-6587.
 - (13) Prados-Roman, C. et al. (2011), Airborne DOAS limb measurements of tropospheric trace gas profiles: case studies on the profile retrieval of O₄ and BrO, *Atmos. Meas. Tech.*, 4: 1241-1260.

REVISION HISTORY:

- (1) Updated for reduction rxn and online Hg⁰ ocean flux. Now use diagnostic arrays from "diag03_mod.f". (eck, sas, bmy, 1/21/05)
- (2) Now references "pbl_mix_mod.f". Remove FEMIS array and routine COMPUTE_FEMIS. (bmy, 2/15/05)
- (3) Now can read data for both GEOS and GCAP grids (bmy, 8/16/05)

- (4) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
 (5) Now references XNUMOL from "tracer_mod.f" (bmy, 10/25/05)
 (6) Various updates added for tagged Hg sim. (eck, sas, cdh, bmy, 4/6/06)
 (7) Now includes LPREINDHG logical switch for preindustrial simulation
 (eds 7/30/08)
- 13 Aug 2010 - R. Yantosca - Add modifications for MERRA (treat like GEOS-5)
 01 Apr 2011 - H. Amos - added LVEGEMIS, distinct from LGCAPEMIS
 27 Sep 2011 - H. Amos - Remove LHg_WETDasHNO3 logical, it's obsolete
 01 Nov 2011 - Y. Zhang - Add subroutines to do in plume reduction of Hg
 07 Feb 2012 - E. Corbitt - Imported tagged tracers; renamed nt to geo.
 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.x met
 01 Mar 2012 - R. Yantosca - Now reference new grid_mod.F90
 01 Mar 2012 - R. Yantosca - Use updated GET_LOCALTIME from time_mod.F
 19 Apr 2012 - E. Corbitt - Added LGCBROMINE to use GEOS-Chem bromine.
 03 Jun 2013 - C. Holmes - Extensively rewritten for simplicity
 03 Jun 2013 - R. Yantosca - Added ProTeX headers
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
 17 Nov 2014 - M. Yannetti - Added PRECISION_MOD
 12 Mar 2015 - R. Yantosca - Now get JNO2 from the HEMCO data structure
 12 Mar 2015 - R. Yantosca - Remove GET_GLOBAL_JNO2 routine
 10 Aug 2015 - J. Fisher - Updated treatment of polar BrO chemistry
 (orig. 29/11/11)
 05 Jan 2016 - E. Lundgren - Now use global dry air gas and Henry's Law consts
 31 May 2016 - E. Lundgren - Replace Input_Opt%XNUMOL with AVO / (emMW_g*1e-3)
 where emMW_g is emitted MW from species database
 29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
 24 Aug 2017 - M. Sulprizio - Remove support for GCAP, GEOS-4, GEOS-5 and
 MERRA; Also remove functions GET_VCLDF and
 GET_LWC, they're no longer needed

10.5.1 chemmercury

Subroutine CHEMMERCURY is the driver routine for mercury chemistry in the GEOS-CHEM module.

INTERFACE:

```

SUBROUTINE CHEMMERCURY( am_I_Root, Input_Opt, State_Met,
&                      State_Chm, State_Diag, RC          )

```

USES:

```

LL      USE CHEMGRID_MOD,          ONLY : GET_TPAUSE_LEVEL
        USE CMN_SIZE_MOD
        USE CMN_DIAG_MOD
        USE DAO_MOD,              ONLY : IS_ICE
        USE DAO_MOD,              ONLY : IS_LAND
        USE DAO_MOD,              ONLY : IS_WATER

```

```

        USE DEPO_MERCURY_MOD,    ONLY : ADD_HG2_DD
        USE DEPO_MERCURY_MOD,    ONLY : ADD_HgP_DD
#ifdef defined( BPCH_DIAG )
        USE DIAG_MOD,            ONLY : AD44
#endif
        USE DIAG03_MOD,          ONLY : AD03_Hg2_Hg0, AD03_Hg2_03
        USE DIAG03_MOD,          ONLY : AD03_Hg2_Br,  AD03_Hg2_OH
        USE DIAG03_MOD,          ONLY : AD03_Hg2_BRY,  AD03_Hg2_CLY
        USE DIAG03_MOD,          ONLY : AD03_Hg2_Br_Y
        USE DIAG03_MOD,          ONLY : AD03_Hg2_SS,  LD03
        USE DIAG03_MOD,          ONLY : AD03_Hg2_SSR, ND03
        USE DIAG03_MOD,          ONLY : AD03_Br
        USE ErrCode_Mod
        USE ERROR_MOD,           ONLY : DEBUG_MSG
        USE ERROR_MOD,           ONLY : ERROR_STOP
        USE ERROR_MOD,           ONLY : SAFE_DIV
        USE HCO_INTERFACE_MOD,   ONLY : HcoState
        USE HCO_EmisList_Mod,    ONLY : HCO_GetPtr
        USE GLOBAL_BR_MOD,       ONLY : GET_GLOBAL_BR
        USE GC_GRID_MOD,         ONLY : GET_AREA_CM2
        USE Input_Opt_Mod,       ONLY : OptInput
        USE PBL_MIX_MOD,         ONLY : GET_FRAC_UNDER_PBLTOP
        USE Species_Mod,         ONLY : Species
        USE State_Chm_Mod,       ONLY : ChmState
        USE State_Diag_Mod,      ONLY : DgnState
        USE State_Met_Mod,       ONLY : MetState
        USE TIME_MOD,           ONLY : GET_MONTH
        USE TIME_MOD,           ONLY : GET_TS_CHEM
        USE TIME_MOD,           ONLY : ITS_A_NEW_MONTH
        USE TIME_MOD,           ONLY : ITS_TIME_FOR_A3

```

INPUT PARAMETERS:

```

        LOGICAL,          INTENT(IN)    :: am_I_Root    ! Are we on the root CPU?
        TYPE(OptInput),  INTENT(IN)    :: Input_Opt    ! Input Options object
        TYPE(MetState),  INTENT(IN)    :: State_Met     ! Meteorology State object

```

INPUT/OUTPUT PARAMETERS:

```

        TYPE(ChmState), INTENT(INOUT)  :: State_Chm    ! Chemistry State object
        TYPE(DgnState), INTENT(INOUT)  :: State_Diag   ! Diagnostics State object

```

OUTPUT PARAMETERS:

```

        INTEGER,          INTENT(OUT)   :: RC           ! Success or failure?

```

REMARKS:

Description of the chemistry mechanism:

=====

The comments and code below refer to 3 mercury species:

1. Hg0 or Hg(0) : gaseous elemental mercury
2. Hg2g or Hg(II)g : gaseous Hg(II) (comparable to RGM observations)
3. Hg2p or Hg(II)p : particle-bound Hg(II) (comparable to PBM observations)

Following Amos et al. (2012), the Hg(II) partitioning between Hg2g and Hg2p is based on temperature and aerosol surface area. There is no refractory particulate Hg.

In some legacy code below Hg2p is called HgP and Hg2g is called Hg2.

The differential equations for oxidation, reduction and deposition are:

$$d[\text{Hg0}]/dt = -(k_{0x} + k_{\text{Dep0}}) [\text{Hg0}] + k_{\text{Red_Hg2g}} [\text{Hg2g}] + k_{\text{Red_Hg2p}} [\text{Hg2p}]$$

$$d[\text{Hg2g}]/dt = k_{0x} [\text{Hg0}] - (k_{\text{Red_Hg2g}} + k_{\text{Dep_Hg2g}}) [\text{Hg2g}]$$

$$d[\text{Hg2p}]/dt = -(k_{\text{Red_Hg2p}} + k_{\text{Dep_Hg2p}}) [\text{Hg2p}]$$

The chemical mechanism currently assumes that the product of Hg(0) oxidation is in the gas phase. This can be easily changed.

Equilibrium partitioning between Hg2g and Hg2p is established at the beginning and end of CHEMMERCURY.

Notes on chemistry and deposition:

(1) Oxidation: Hg(0) --> Hg(II):

Oxidation by Br, BrO, OH, and O3 can be selectively enabled or disabled with switches (LBRCHEM, LBROCHEM, LOHO3CHEM) in INIT_MERCURY.

Hg(0)(g) + Br(g) --> + Br/OH --> Hg(II)g, rates are selected with METHOD keyword below. Recommended kinetics are 'DonohoueYBBalabanov' which use rates from Donohoue et al. (2006), Goodsite et al. (2004) and Balabanov et al. (2005)

Hg(0)(g)+ O3(g) --> Hg(II) , k = 3.0e-20 cm3 molec-1 s-1
Source: Hall, 1995

Hg(0)(g)+ OH(g) --> Hg(II) , k = 8.7e-14 cm3 s-1
Source: Sommar et al. 2001

(2) Reduction:

Reduction rates can be scaled to NO2 photolysis rates or OH concentrations with the switch LRED_JNO2. In either case, aqueous-phase photochemical reduction of Hg(II) is included based on estimate of rate constant and scaled to NO2 photolysis or [OH].

The rate is tuned to match the global Hg(0) concentration and seasonal cycle.

(3) Hg(0) dry deposition:

The dry deposition frequency is calculated by drydep_mod. If the non-local PBL mixing scheme is used, however, all dry deposition is done elsewhere. The ocean module separately calculates Hg(0) dry deposition over ocean, so CHEMMERCURY only includes Hg(0) dry deposition over land.

(4) Hg(II)g and Hg(II)p dry deposition:

The dry deposition frequencies are calculated by drydep_mod. If the non-local PBL mixing scheme is used, however, all dry deposition is done elsewhere.

(5) Sea-salt uptake of Hg(II)g

Hg(II)g is taken up into sea-salt aerosol and deposited to the ocean surface at a rate based on wind speed and relative humidity. Hg(II)p uptake into sea-salt aerosol may also occur at a slower rate, but is not yet treated here.

REVISION HISTORY:

- 01 Oct 1995 - R. Yantosca - Initial version
- 08 Dec 2009 - R. Yantosca - Added ProTeX headers
- 06 Dec 2004 - N. (Eckley) Selin - Initial version
- (1) Now references routine GET_PBL_MAX_L from "pbl_mix_mod.f". Now references AD44 from "diag_mod.f". Now sum the levels from T44 into the AD44 array. Now references N_TRACERS from "tracer_mod.f". (bmy, 2/24/05)
- (2) Bug fix: Set T44 to 0e0 for single precision. Now allow for zero dry deposition velocity. Now call INIT_MERCURY from "input_mod.f" (bmy, 4/6/06)
- 02 Apr 2013 - C. Holmes - CHEMMERCURY rewritten for clarity and easier maintainability. The chemistry and deposition for particle-bound mercury is now solved simultaneously with Hg(0) and gaseous Hg(II). The subroutine now uses a 4th-order Runge-Kutta method to solve the coupled chemistry. The subroutines CHEM_HGO_HG2 and CHEM_HGP are now obsolete.
- 04 Apr 2013 - C. Holmes - Use State_Met for met fields.
- 03 Jun 2013 - R. Yantosca - Now use fields from Input_OpT instead of from logical_mod.F and tracer_mod.F
- 26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET_GLOBAL_OH routine
- 06 Nov 2014 - R. Yantosca - Now use State_Met%AIRDEN(I,J,L)
- 06 Nov 2014 - R. Yantosca - Now use State_Met%CLDF(I,J,L)
- 26 Feb 2015 - E. Lundgren - Replace GET_PCENTER with State_Met%PMID_DRY. Remove dependency on pressure_mod.
- 12 Aug 2015 - R. Yantosca - Added support for MERRA2 meteorology

05 Jan 2016 - E. Lundgren - Use Henry's constant from species database and calculate dry air gas constant from global params
 25 Apr 2016 - R. Yantosca - Now pass Hg category # to ADD_Hg2_* routines
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State_Chm%Map_Advect.
 01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
 11 Oct 2016 - R. Yantosca - Pass additional args to SET_OPTIONS_FROM_HEMCO
 30 Sep 2017 - C. Thackray - Brought in Horowitz 2017 chemistry
 03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument
 30 Nov 2017 - C. Thackray - Option to use offline ocean Hg concentrations
 02 Feb 2018 - C. Thackray - Made Hg2(aq) for red. temperature dependent
 10 Feb 2018 - C. Thackray - Switches for chem options

10.5.2 get_hgaq_rate

Function GET_HGAQ_RATE computes the effective 1st order conversion rate of Hg(0) to Hg(II) via aqueous-phase oxidation in cloud droplets for three reactions: HOCl, O3, and OH (2-step). hmh 11/19/14

INTERFACE:

```
FUNCTION GET_HGAQ_RATE(I, J, L, BR,OH,O3,HOCL,State_Met)
&      RESULT( K_AQ_OX )
```

USES:

```
USE State_Met_Mod,      ONLY : MetState
USE CMN_SIZE_MOD ! Size parameters
USE ERROR_MOD,        ONLY : SAFE_DIV
```

INPUT PARAMETERS:

```
! BR      : Concentration of atomic Br [molec/cm3]
! OH      : Concentration of OH [molec/cm3]
! O3      : Concentration of O3 [molec/cm3]
! HOCL    : Concentration of HOCL[molec/cm3]
! State_Met : Meteorology state object
INTEGER,   INTENT(IN)  :: I, J, L
REAL(fp),  INTENT(IN)  :: BR, OH, O3, HOCL
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State obj
```

RETURN VALUE:

```
! Effective 1st order loss rate of Hg(0) [1/s] for:
! reaction with aqueous (1) HOCl, (2) OH, and (3) O3
```

```
REAL(fp)                                :: K_AQ_OX(3)
```

References:

- ```
=====
```
1. Wang and Pehkonen, 2004
  2. Pehkonen and Lin, 1997
  3. Buxton et al., 1988
  4. Munthe, 1992

#### REVISION HISTORY:

```
19 November 2014 - H. Horowitz (HMH) - initial version finalized
30 September 2017 - C. Thackray (cpt) - merged into v11
```

---

#### 10.5.3 get\_hgbr\_rate

Function GET\_HGBR\_RATE computes the effective 1st order conversion rate of Hg(0) to Hg(II) via two-step recombination with Br and OH.

#### INTERFACE:

```
FUNCTION GET_HGBR_RATE(I, J, L, BR, State_Met, OH, METHOD)
& RESULT(K_HGBR)
USES:
USE State_Met_Mod, ONLY : MetState
```

#### INPUT PARAMETERS:

```
! BR : Concentration of atomic Br [molec/cm3]
! State_Met : Meteorology state object
! OH : Concentration of OH [molec/cm3]
! METHOD : Set of rate constants to use in calculation
INTEGER, INTENT(IN) :: I, J, L
REAL(fp), INTENT(IN) :: BR, OH
CHARACTER(LEN=*), INTENT(IN) :: METHOD
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State obj
```

#### RETURN VALUE:

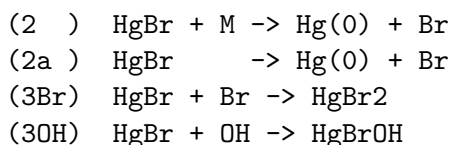
```
! K_HgBr : Effective 1st order loss rate of Hg(0) [1/s]
REAL(fp) :: K_HGBR
```

#### REMARKS:

```
=====
```

This subroutine calculates the net rate of Hg(0) oxidation to Hg(II) through the following reactions. All are gas phase.

```
(1) Hg(0) + Br -> HgBr
```



## References:

- 
1. Ariya, P. A., A. Khalizov, and A. Gidas (2002), Reaction of gaseous mercury with atomic and molecular halogens: kinetics, product studies, and atmospheric implications, *Journal of Physical Chemistry A*, 106, 7310-7320.
  2. Balabanov, N. B., B. C. Shepler, and K. A. Peterson (2005), Accurate global potential energy surface and reaction dynamics for the ground state of HgBr<sub>2</sub>, *Journal of Physical Chemistry A*, 109(39), 8765-8773.
  3. Donohoue, D. L., D. Bauer, B. Cossairt, and A. J. Hynes (2006), Temperature and Pressure Dependent Rate Coefficients for the Reaction of Hg with Br and the Reaction of Br with Br: A Pulsed Laser Photolysis-Pulsed Laser Induced Fluorescence Study, *Journal of Physical Chemistry A*, 110, 6623-6632.
  4. Goodsite, M. E., J. M. C. Plane, and H. Skov (2004), A theoretical study of the oxidation of Hg-0 to HgBr<sub>2</sub> in the troposphere, *Environmental Science & Technology*, 38(6), 1772-1776.
  5. Holmes, C. D., et al. (2006), Global lifetime of elemental mercury against oxidation by atomic bromine in the free troposphere, *Geophys. Res. Lett.*, 33(20).
  6. Khalizov, A. F., B. Viswanathan, P. Larregaray, and P. A. Ariya (2003), A theoretical study on the reactions of Hg with halogens: Atmospheric implications, *Journal of Physical Chemistry A*, 107(33), 6360-6365.

**REVISION HISTORY:**

06 Jul 2006 - C. Holmes - Initial version  
 05 Jan 2016 - E. Lundgren - Use global physical parameters

---

**10.5.4 get\_hgcl\_rate**

Function GET\_HGCL\_RATE computes the effective 1st order conversion rate of Hg(0) to Hg(II) initiated by Cl and 2nd step with Cl, OH, NO<sub>2</sub>, HO<sub>2</sub>, ClO, and BrO. hmh 11/21/14 copied from get\_hgbry\_rate.

**INTERFACE:**



2. Holmes et al., 2009
3. Hynes et al., 2009
4. Wilcox, 2009
5. Balabanov et al., 2005

**REVISION HISTORY:**

21 Nov 2014 - H. Horowitz (HMH) - first version, copied from get\_hgbry\_rat

---

**10.5.5 get\_hgbry\_rate**

Function GET\_HGBRY\_RATE computes the effective 1st order conversion rate of Hg(0) to Hg(II) via two-step recombination with Br OH, NO<sub>2</sub>, HO<sub>2</sub>, ClO, and BrO. hmh 6/4/14 based on eds 5/2/13, modified for consistency with GET\_HGBR\_RATE.

**INTERFACE:**

```

 FUNCTION GET_HGBRY_RATE(I, J, L, BR, OH, NO2, HO2, CLO, BRO, CL,
& State_Met)
& RESULT(K_HGBRY)

```

**USES:**

```

 USE State_Met_Mod, ONLY : MetState

```

**INPUT PARAMETERS:**

```

 ! BR : Concentration of atomic Br [molec/cm3]
 ! HO2 : Concentration of HO2 [molec/cm3]
 ! OH : Concentration of OH [molec/cm3]
 ! NO2 : Concentration of NO2 [molec/cm3]
 ! CLO : Concentration of ClO [molec/cm3]
 ! BRO : Concentration of BrO [molec/cm3]
 ! CL : Concentration of Cl [molec/cm3]
 ! State_Met : Meteorology state object
 INTEGER, INTENT(IN) :: I, J, L
 REAL(fp), INTENT(IN) :: BR, OH, NO2, HO2
 REAL(fp), INTENT(IN) :: CLO, BRO, CL!
 TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State obj

```

**RETURN VALUE:**

```

 ! K_HgBrY : Effective 1st order loss rate of Hg(0) [1/s]
 ! each pathway - Br, OH, HO2, NO2, ClO, BrO, Cl in that order.
 ! #s 8 - 10 are for R1, R2, and R2a!
 REAL(fp) :: K_HGBRY(11)

```

**REMARKS:**

=====

This subroutine calculates the net rate of Hg(0) oxidation to Hg(II) through the following reactions. All are gas phase.

```
(1) Hg(0) + Br -> HgBr
(2) HgBr + M -> Hg(0) + Br
(2a) HgBr + Br-> Hg(0) + Br2 ! HMH there was a typo in this rxn!
(3BR) HgBr + Br -> HgBr2
(3OH) HgBr + OH -> HgBrOH
(3H02) HgBr + H02 -> HgBrH02
(3N02) HgBr + N02 -> HgBrN02
(3CLO) HgBr + Cl0 -> HgBrCl0
(3BR0) HgBr + Br0 -> HgBrBr0
(3CL) HgBr + Cl -> HgBrCl
```

References:

- =====
1. Dibble et al. 2012
  2. Horowitz et al. in prep

## REVISION HISTORY:

03 May 2013 - E. Corbitt - initial version  
 05 June 2014 - H. Horowitz (HMH) - revising format to be consistent  
 20 Nov 2014 - H. Horowitz (HMH) - saving out separate rate constants for each pathway for comparison purposes! And fixing the reactions listed.  
 10 Aug 2015 - H. Horowitz (HMH) - adding Cl as 'Y', was missing before.  
 01 Mar 2016 - H. Horowitz (HMH) - saving out other reaction rates!  
 10 Dec 2016 - H. Horowitz (HMH) - changing to Ted Dibble revised rates  
 30 Sep 2017 - C. Thackray (cpt) - merged into v11

---

### 10.5.6 emissmercury

Subroutine EMISSMERCURY is the driver routine for mercury emissions.

NOTE/TODO: The mercury simulation is the only GEOS-Chem emission code that is not yet fully compatible with HEMCO. So far, only the anthropogenic emissions are included in HEMCO. For all other sources, the original mercury code is used.

For the non-local PBL mixing, all emissions are written into module array HG\_EMIS (in kg m<sup>-2</sup> s<sup>-1</sup>). These values are then used by vdiff\_mod.F90. This is just a workaround to ensure backwards compatibility of the mercury code. Once we have added all mercury emissions to HEMCO, HG\_EMIS is not used any more (ckeller, 10/21/2014).

### INTERFACE:

```
SUBROUTINE EMISSMERCURY(am_I_Root, Input_Opt,
```



& State\_Met, State\_Chm, RC )

**USES:**

```

USE CMN_SIZE_MOD
USE DEPO_MERCURY_MOD, ONLY : RESET_HG_DEP_ARRAYS
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE LAND_MERCURY_MOD, ONLY : LAND_MERCURY_FLUX, VEGEMIS
USE LAND_MERCURY_MOD, ONLY : SOILEMIS, BIOMASSHG
USE LAND_MERCURY_MOD, ONLY : SNOWPACK_MERCURY_FLUX
USE OCEAN_MERCURY_MOD, ONLY : OCEAN_MERCURY_FLUX
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH, ITS_A_NEW_MONTH
USE UnitConv_Mod, ONLY : Convert_Spc_Units

```

! Added for GTMM (ccc, 11/19/09)

```

USE LAND_MERCURY_MOD, ONLY : GTMM_DR

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:****REVISION HISTORY:**

- 03 Jun 2013 - N. (Eckley) Selin - Initial version
- (1 ) Now call OCEAN\_MERCURY\_FLUX from "ocean\_mercury\_mod.f" to compute the emissions of Hg0 from the ocean instead of reading it from disk. (sas, bmy, 1/20/05)
- (2 ) Now no longer call COMPUTE\_FEMIS, since we can get the same information from routine GET\_FRAC\_OF\_PBL in "pbl\_mix\_mod.f" (bmy, 2/22/05)
- (3 ) Now modified for new ocean mercury module. (cdh, sas, bmy, 4/6/06)
- 01 Mar 2012 - R. Yantosca - Now use GET\_XMID(I,J,L) from grid\_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID(I,J,L) from grid\_mod.F90
- 25 Mar 2013 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Chm, RC

08 Dec 2009 - R. Yantosca - Added ProTeX headers  
 20 Jun 2014 - R. Yantosca - Now pass Input\_Opt to VEGEMIS  
 23 Jun 2014 - R. Yantosca - Now pass am\_I\_Root to OCEAN\_MERCURY\_FLUX  
 05 Oct 2015 - E. Lundgren - Now convert tracer units to kg locally  
 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine

### 10.5.7 emithg

Subroutine EMITHG directs emission either to the chemical species array (State\_Chmnon-local PBL mixing. This is a programming convenience.

#### INTERFACE:

```
SUBROUTINE EMITHG(I, J, L, ID, E_HG, Input_Opt, State_Chm)
```

#### USES:

```
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE TIME_MOD, ONLY : GET_TS_EMIS
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I, J, L, ID ! Grid boxes + species #
REAL(fp), INTENT(IN) :: E_Hg ! Hg emissions
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

#### INPUT/OUTPUT PARAMETERS:

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

#### REVISION HISTORY:

27 Aug 2009 - C. Holmes - Initial version  
 25 Mar 2013 - R. Yantosca - Now accept Input\_Opt, State\_Chm, arguments

### 10.5.8 srcHg0

Subroutine SRCHg0 is the subroutine for Hg(0) emissions. Emissions of Hg(0) will be distributed throughout the boundary layer. (eck, cdh, bmy, 1/21/05, 4/6/06)

#### INTERFACE:

```
SUBROUTINE SRCHg0(am_I_Root, Input_Opt, State_Chm, RC)
```

#### USES:

```

USE CMN_SIZE_MOD
USE DIAG03_MOD, ONLY : AD03, ND03, AD03_nat
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_MAX_L
USE State_Chm_Mod, ONLY : ChmState
USE TIME_MOD, ONLY : GET_TS_EMIS

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

- 21 Jan 2005 - N. (Eckley) Selin, C. Holmes - Initial version
- (1 ) Now use diagnostic arrays from "diag03\_mod.f" (bmy, 1/21/05)
- (2 ) Now references GET\_FRAC\_OF\_PBL and GET\_PBL\_MAX\_L from "pbl\_mix\_mod.f".  
Remove reference to FEMIS. (bmy, 2/22/05)
- (3 ) EHg0\_an is now a 2-D array. Modified for new ocean mercury module.  
Now use ID\_Hg0 index array from "tracerid\_mod.f". Now make sure  
STT does not underflow. (cdh, bmy, 4/6/06)
- 25 Mar 2013 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Chm, RC
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected  
species ID from State\_Chm%Map\_Advect.

**10.5.9 srcHg2**

Subroutine SRCHg2 is the subroutine for Hg(II) emissions. Emissions of Hg(II) will be distributed throughout the boundary layer.

**INTERFACE:**

```

SUBROUTINE SRCHg2(am_I_Root, Input_Opt, State_Chm, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE DIAG03_MOD, ONLY : AD03, ND03
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput

```

```

USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_MAX_L
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

- 07 Dec 2004 - N. (Eckley) Selin - Initial version
- (1 ) Now use diagnostic arrays from "diag03\_mod.f" (bmy, 1/21/05)
- (2 ) Now references GET\_FRAC\_OF\_PBL and GET\_PBL\_MAX\_L from "pbl\_mix\_mod.f".  
Remove reference to FEMIS. (bmy, 2/22/05)
- (3 ) EHg2\_an is now a 2-D array. Now use ID\_Hg2 index array from  
"tracerid\_mod.f". Now make sure STT does not underflow.  
(eck, cdh, bmy, 4/6/06)
- 25 Mar 2013 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Chm, RC

**10.5.10 srcHgp**

Subroutine SRCHgP is the subroutine for HgP emissions. Emissions of HgP will be distributed throughout the boundary layer.

**INTERFACE:**

```

SUBROUTINE SRCHgP(am_I_Root, Input_Opt, State_Chm, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE DIAG03_MOD, ONLY : AD03, ND03
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE PBL_MIX_MOD, ONLY : GET_FRAC_OF_PBL, GET_PBL_MAX_L
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

- 07 Dec 2004 - N. (Eckley) Selin - Initial version
- (1 ) Now use diagnostic arrays from "diag03\_mod.f" (bmy, 1/21/05)
  - (2 ) Now references GET\_FRAC\_OF\_PBL and GET\_PBL\_MAX\_L from "pbl\_mix\_mod.f".  
Remove reference to FEMIS. (bmy, 2/22/05)
  - (3 ) EHgP\_an is now a 2-D array. Now use ID\_HgP index array from  
"tracerid\_mod.f". Now make sure STT does not underflow.  
(eck, cdh, bmy, 4/6/06)
- 

**10.5.11 mercury\_readyr**

Subroutine MERCURY\_READYR reads the year-invariant emissions for Mercury from anthropogenic, ocean, and land sources.

**INTERFACE:**

```

SUBROUTINE MERCURY_READYR(am_I_Root, Input_Opt, RC)

```

**USES:**

```

USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_ERROR_MOD
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE HCO_INTERFACE_MOD, ONLY : GetHcoDiagn
USE Input_Opt_Mod, ONLY : OptInput

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

06 Dec 2004 - N. (Eckley) Selin - Initial version  
 (1 ) Now read data from mercury\_200501 subdirectory. Now compute oceanic Hg(0) emissions w/ ocean flux module instead of reading them from disk. Now use 1985 TAU values. (sas, bmy, 1/20/05)  
 (2 ) Now can read data for both GEOS and GCAP grids (bmy, 8/16/05)  
 (3 ) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)  
 (4 ) Now read anthro emissions on GEOS 1x1 grid in DATA\_1x1\_DIR. Also keep 2x25 and 4x5 files together in DATA\_1x1\_DIR. Also now use new land re-emissions files from Noelle Selin. (eck, bmy, 4/6/06)  
 (5 ) Now reads anthro emissions for 17 world regions from David Streets for use in tagged simulations. (eds 2/7/12)  
 (6 ) Bug fix: emissions are read on GENERIC grid (jaf, 2/8/12)  
 01 Mar 2012 - R. Yantosca - Now use GET\_XMID(I,J,L) from grid\_mod.F90  
 01 Mar 2012 - R. Yantosca - Now use GET\_YMID(I,J,L) from grid\_mod.F90  
 13 Mar 2012 - M. Cooper - Changed regrid algorithm to map\_a2a  
 24 May 2012 - R. Yantosca - Fixed minor bugs in map\_a2a algorithm  
 24 May 2012 - H. Amos - updated file paths for soil Hg files  
 28 Jun 2012 - S. Kim - Bug fix: you need to provide a location in the call to ERROR\_STOP  
 24 Aug 2012 - R. Yantosca - DO\_REGRID\_A2A now reads netCDF input file  
 03 Jan 2013 - M. Payer - Renamed PERAREA to IS\_MASS in DO\_REGRID\_A2A  
 13 Jan 2013 - M. Payer - Update soil Hg files for v9-02c (Y. Zhang)  
 06 Jun 2013 - R. Yantosca - Get ANTHRO\_Hg\_YEAR, Hg\_SCENARIO from Input\_Opt  
 06 Sep 2013 - R. Yantosca - Kludge: use MERRA soil file for GEOS-FP testing and debugging. Worry about this later.  
 06 Sep 2013 - R. Yantosca - Kludge: use GEOS-5 soil file for GEOS-4 testing and debugging. Worry about this later.  
 24 Sep 2014 - C. Keller - HEMCO update  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 27 Feb 2018 - C. Thackray - Removed unused ANTHRO\_Hg\_YEAR and Hg\_SCENARIO

**10.5.12 get\_o3**

Function GET\_O3 returns monthly mean O3 for the mercury simulation.

**INTERFACE:**

```
FUNCTION GET_O3(I, J, L, State_Met) RESULT(O3_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I ! Longitude index
```

```

INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**RETURN VALUE:**

```

REAL(fp) :: O3_MOLEC_CM3 ! O3 conc [molec/cm3]

```

**REVISION HISTORY:**

```

06 Dec 2002 - R. Yantosca - Initial version
(1) We assume SETTRACE has been called to define IDO3. (bmy, 12/16/02)
(2) Now reference inquiry functions from "tracer_mod.f" (bmy, 7/20/04)
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
11 Mar 2015 - R. Yantosca - Remove reference to GLOBAL_O3_MOD

```

**10.5.13 get\_oh**

Function GET\_OH returns monthly mean OH and imposes a diurnal variation.

**INTERFACE:**

```

FUNCTION GET_OH(I, J, L, State_Met) RESULT(OH_MOLEC_CM3)

```

**USES:**

```

USE CMN_SIZE_MOD
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
LL USE CHEMGRID_MOD, ONLY : GET_TPAUSE_LEVEL

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**RETURN VALUE:**

```

REAL(fp) :: TPL
REAL(fp) :: HEM_OH
REAL(fp) :: OH_MOLEC_CM3 ! OH conc [molec/cm3]

```

**REVISION HISTORY:**

```

07 Dec 2004 - N. (Eckley) Selin - Initial version
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
28 Nov 2012 - R. Yantosca - Now pass State_Met via the argument list

```

11 Mar 2015 - R. Yantosca - Remove reference to GLOBAL\_OH\_MOD  
 06 Feb 2018 - E. Lundgren - Update time unit conversion factor for timestep  
 unit change from min to sec

#### 10.5.14 get\_ho2

Function GET\_HO2 returns monthly mean HO2 and imposes a diurnal variation. HMH  
 9/24/14 adding for dibble chemistry

#### INTERFACE:

```
FUNCTION GET_HO2(I, J, L, State_Met) RESULT(HO2_MOLEC_CM3)
```

#### USES:

```
USE CMN_SIZE_MOD
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
LL USE CHEMGRID_MOD, ONLY : GET_TPAUSE_LEVEL
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: HO2_VV
REAL(fp) :: HO2_MOLEC_CM3 ! HO2 conc [molec/cm3]
REAL(fp) :: HEM_HO2
REAL(fp) :: TPL
```

#### REVISION HISTORY:

07 Dec 2004 - N. (Eckley) Selin - Initial version  
 28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
 28 Nov 2012 - R. Yantosca - Now pass State\_Met via the argument list  
 ca. 2014 - H. Horowitz -  
 30 Sep 2017 - C. Thackray (cpt) - Merged into v11



**10.5.15 get\_clo**

Function GET\_CLO imposes a diurnal scaling for ClO, hmh 10/6/ adding for dibble chemistry. Now including scaling of Cl and HOCl for aqueous and gas-phase oxidation 11/19/14.

**INTERFACE:**

```
FUNCTION GET_CLO(I, J, L, State_Met) RESULT(CLO_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD
USE State_Met_Mod, ONLY : MetState
USE ERROR_MOD, ONLY : SAFE_DIV
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_LOCALTIME
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

**RETURN VALUE:**

```
REAL(fp) :: CLO_PPBV
REAL(fp) :: CLO_MOLEC_CM3 ! CLO conc [molec/cm3]

INTENT(OUT)
```

**REVISION HISTORY:**

```
06 Oct 2014 - H. Horowitz - initial version copied from get_br.
30 Sep 2017 - C. Thackray (cpt) - merged into v11
```

---

**10.5.16 get\_cl**

Function GET\_CL imposes a diurnal scaling for Cl, hmh 11/19/14 adding for Cl-initiated oxidation.

**INTERFACE:**

```
FUNCTION GET_CL(I, J, L, State_Met) RESULT(CL_MOLEC_CM3)
```

**USES:**

```

USE CMN_SIZE_MOD
USE State_Met_Mod, ONLY : MetState
USE ERROR_MOD, ONLY : SAFE_DIV
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_LOCALTIME

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**RETURN VALUE:**

```

REAL(fp) :: Cl_PPBV
REAL(fp) :: Cl_MOLEC_CM3 ! Cl conc [molec/cm3]

INTENT(OUT)

```

**REVISION HISTORY:**

19 Nov 2014 - H. Horowitz - initial version copied from get\_cl0.  
30 Sep 2017 - C. Thackray (cpt) - merged into v11

---

**10.5.17 get\_HOCl**

Function GET\_HOCl returns instantaneous HOCl concentration calculated from the monthly mean and an imposed diurnal variation.

**INTERFACE:**

```

FUNCTION GET_HOCl(I,J,L,State_Met)
& RESULT(HOCL_INST)

```

**USES:**

```

USE ERROR_MOD, ONLY : ERROR_STOP, SAFE_DIV
USE TIME_MOD, ONLY : GET_LOCALTIME
USE State_Met_Mod, ONLY : MetState
USE CMN_SIZE_MOD

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I,J,L
TYPE(MetState), INTENT(IN) :: State_Met

```

**REVISION HISTORY:**

ca. 2014 - H. Horowitz - Initial version  
30 Sep 2017 - C. Thackray (cpt) - Merged into v11

---

**10.5.18 get\_NO2**

Function GET\_NO2 returns instantaneous NO2 concentration calculated from the monthly mean and an imposed diurnal variation.

**INTERFACE:**

```
FUNCTION GET_NO2(I, J, L, State_Met, NO_MO, NO2_MO, O3_MO)
& RESULT(NO2_INST)
```

**USES:**

```
USE ERROR_MOD, ONLY : ERROR_STOP, SAFE_DIV
USE State_Met_Mod, ONLY : MetState
USE CMN_SIZE_MOD
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I,J,L
TYPE(MetState), INTENT(IN) :: State_Met
REAL(4), INTENT(IN) :: NO_MO
REAL(4), INTENT(IN) :: NO2_MO
REAL(fp), INTENT(IN) :: O3_MO
```

**REVISION HISTORY:**

```
ca. 2014 - H. Horowitz - Initial version
30 Sep 2017 - C. Thackray (cpt) - Merged into v11
```

---

**10.5.19 get\_Br**

Function GET\_BR returns instantaneous Br concentration calculated from the monthly mean and an imposed diurnal variation.

**INTERFACE:**

```
FUNCTION GET_BR(I, J, L, BRO_MOLEC_CM3, State_Met)
& RESULT(BR_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD
USE DAO_MOD, ONLY : GET_OBK
USE DAO_MOD, ONLY : IS_WATER
USE DIAG03_MOD, ONLY : AD03_Br, LD03, ND03
USE ERROR_MOD, ONLY : SAFE_DIV
USE GLOBAL_BR_MOD, ONLY : BR_MERGE, BR_TROP
USE GLOBAL_BR_MOD, ONLY : BR_STRAT, J_BRO
USE GLOBAL_BR_MOD, ONLY : BRO_MERGE, BRO_TROP, BRO_STRAT
USE GC_GRID_MOD, ONLY : GET_YMID
USE PBL_MIX_MOD, ONLY : GET_FRAC_UNDER_PBLTOP
```

```

USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_LOCALTIME
USE TIME_MOD, ONLY : ITS_A_NEW_DAY
USE TIME_MOD, ONLY : GET_MONTH

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
REAL(fp), INTENT(OUT) :: BrO_MOLEC_CM3 ! BrO conc [molec/cm3]
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**RETURN VALUE:**

```

REAL(fp) :: BR_MOLEC_CM3 ! Br conc [molec/cm3]

```

**REVISION HISTORY:**

```

06 Jul 2006 - C. Holmes - Initial version
01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_LOCALTIME(I,J,L) from time_mod.F90
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
26 Feb 2015 - E. Lundgren - Remove dependency on pressure_mod (not used)
10 Aug 2015 - J. Fisher - implement variable polar BrO concentrations based
 on temperature and incoming solar radiation
10 Aug 2015 - J. Fisher - only apply diurnal cycle to BrO from pTOMCAT
 daily means, NOT to polar BrO which represents
 instantaneous values
10 Aug 2015 - J. Fisher - add upper ice threshold to MERRA conditions for
 polar BrO release

```

**10.5.20 ohno3time**

Subroutine OHNO3TIME computes the sum of cosine of the solar zenith angle over a 24 hour day, as well as the total length of daylight. This is needed to scale the offline OH and NO3 concentrations.

**INTERFACE:**

```

SUBROUTINE OHNO3TIME

```

**USES:**

```

USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID_R

```

```

USE TIME_MOD, ONLY : GET_NHMSb, GET_ELAPSED_SEC
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_DAY_OF_YEAR, GET_GMT

```

## REVISION HISTORY:

- 16 Dec 2002 - R. Park & R. Yantosca - Initial version
- (1 ) Copy code from COSSZA directly for now, so that we don't get NaN values. Figure this out later (rjp, bmy, 1/10/03)
- (2 ) Now replace XMID(I) with routine GET\_XMID from "grid\_mod.f".  
Now replace RLAT(J) with routine GET\_YMID\_R from "grid\_mod.f".  
Removed NTIME, NHMSb from the arg list. Now use GET\_NHMSb, GET\_ELAPSED\_SEC, GET\_TS\_CHEM, GET\_DAY\_OF\_YEAR, GET\_GMT from "time\_mod.f". (bmy, 3/27/03)
- (3 ) Now store the peak SUNCOS value for each surface grid box (I,J) in the COSZM array. (rjp, bmy, 3/30/04)
- (4 ) Also added parallel loop over grid boxes (eck, bmy, 12/8/04)
- 01 Mar 2012 - R. Yantosca - Now use GET\_XMID(I,J,L) from grid\_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID\_R(I,J,L) from grid\_mod.F90
- 16 May 2016 - M. Sulprizio- Remove IJLOOP and change SUNTMP array dimensions from (MAXIJ) to (IIPAR,JJPAR)
- 06 Feb 2018 - E. Lundgren - Update unit conversion factor for timestep unit change from min to sec

### 10.5.21 calc\_hg2\_seasalt\_lossrate

Subroutine CALC\_HG2\_SEASALT\_LOSSRATE calculates the loss rate of RGM (/s) by uptake of RGM into sea salt aerosol for each model grid. Return value is a loss frequency (/s)

#### INTERFACE:

```

SUBROUTINE CALC_HG2_SEASALT_LOSSRATE(State_Met)

```

#### USES:

```

USE CMN_SIZE_MOD ! Size parameters
USE DAO_MOD, ONLY : IS_WATER
USE PBL_MIX_MOD, ONLY : GET_PBL_TOP_M
USE State_Met_Mod, ONLY : MetState

```

#### INPUT PARAMETERS:

```

TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### REMARKS:

The formula used here is a least-squares fit to the full-physics model of sea-salt aerosol emissions, hygroscopic growth, mass-transport limited uptake of Hg(II), and aerosol deposition presented by Holmes et al. (2009)

See Holmes et al. 2010 for evaluation of this parameterization.  
(cdh, 11/25/09)

## REVISION HISTORY:

25 Nov 2009 - C. Holmes - Initial version  
25 Jul 2014 - R. Yantosca - Remove reference to function SFCWINDSQR  
04 Dec 2014 - M. Yannetti - Remove reference to RNPBE\_MOD

---

### 10.5.22 get\_jno2

Function GET\_JNO2 returns monthly mean JNO2 and imposes a diurnal variation.

#### INTERFACE:

```
FUNCTION GET_JNO2(I, J, L, State_Met) RESULT(JNO2_NOW)
```

#### USES:

```
USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_YMID_R
USE PhysConstants
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_DAY_OF_YEAR
```

#### INPUT PARAMETERS:

```
INTEGER, INTENT(IN) :: I ! Longitude index
INTEGER, INTENT(IN) :: J ! Latitude index
INTEGER, INTENT(IN) :: L ! Level index
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### RETURN VALUE:

```
REAL(fp) :: JNO2_NOW ! J(NO2) value [s-1]
```

#### REMARKS:

Impose the diurnal variation of JNO2 found by Parrish et al. (1983) under clear skies.  $J\text{-NO}_2 \sim \exp(-0.360 * \sec(\text{SZA}))$

## REVISION HISTORY:

01 Mar 2012 - R. Yantosca - Now use GET\_YMID\_R(I,J,L) from grid\_mod.F90  
28 Nov 2012 - R. Yantosca - Now pass State\_Met via the argument list  
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS

---

**10.5.23 partitionhg2**

Subroutine PARTITIONHG2 splits Hg(II) into gas and aerosol portions according to the thermodynamic equilibrium determined by temperature and aerosol surface area.

**INTERFACE:**

```
SUBROUTINE PARTITIONHG2(am_I_Root, Input_Opt, State_Chm, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : SAFE_DIV , GEOS_CHEM_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE OCEAN_MERCURY_MOD, ONLY : Fg
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
12-Jul-2010 - H. Amos - Add option to partition Hg2 according to Fg/Fp
02-Jan-2011 - H. Amos - Add/clean up comments
04 Jan 2012 - H. Amos - modify algorithms to reflect the fact that
 anthropogenic Hg(p) is now emitted as Hg(II)
 (i.e. it's no longer considered refractory).
28-Mar-2013 - C. Holmes - Since we now assume that all HgP and Hg2
 actively partition between gas and aerosol,
 we no longer need to 'reverse' partition.
03 Jun 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
18 Nov 2014 - M. Yannetti - Took out OCEAN_MERCURY_MOD calling Fp
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
 species ID from State_Chm%Map_Advect.
```

**10.5.24 red\_inplume\_grid**

Subroutine RED\_INPLUME\_GRID conducts in plume reduction of Hg2 for selected grids

**INTERFACE:**

```
SUBROUTINE RED_INPLUME_GRID(I, J, E_plant)
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I, J
REAL(fp), INTENT(IN) :: E_plant
```

**REVISION HISTORY:**

```
11 Jan 2011 - Y. Zhang - Initial version
```

---

**10.5.25 do\_red\_inplume**

Subroutine DO\_RED\_INPLUME conducts in plume reduction of Hg2 for selected grids.

**INTERFACE:**

```
SUBROUTINE DO_RED_INPLUME(am_I_Root, Input_Opt, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE TIME_MOD, ONLY : EXPAND_DATE
```

```
IMPLICIT NONE
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
11 Jan 2011 - Y. Zhang - Initial version
23 Jun 2014 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC
06 Nov 2014 - R. Yantosca - Replace TRANSFER_2D with direct casts
17 Dec 2014 - R. Yantosca - Leave time/date variables as 8-byte
05 Mar 2015 - R. Yantosca - Add Input_Opt%RES_DIR to data path'
16 Mar 2015 - R. Yantosca - Remove bpch input
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
```

---



**10.5.26 offlineocean\_readmo**

Subroutine OFFLINEOCEAN\_READMO reads the monthly varying offline ocean evasion emissions if LDYNOCEAN is FALSE. Will not actually need mixed layer depth when i get stuff from Yanxu

**INTERFACE:**

```

SUBROUTINE OFFLINEOCEAN_READMO(FLUX, RC, am_I_Root,
& State_Met, State_Chm)

```

**USES:**

```

USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE Input_Opt_Mod, ONLY : OptInput
USE TIME_MOD, ONLY : EXPAND_DATE, GET_YEAR, GET_TS_EMIS !hmh 9/16/15
USE TIME_MOD, ONLY : ITS_A_NEW_MONTH, GET_MONTH
USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE DIAG03_MOD, ONLY : AD03, ND03 ! hmh 9/16/15 for Hg_up/down
! adding hmh 5/17/16
USE DAO_MOD, ONLY : IS_WATER
USE CMN_SIZE_MOD ! Size parameters
USE State_Chm_Mod, ONLY : ChmState
USE State_Met_Mod, ONLY : MetState
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr

```

**INPUT PARAMETERS:**

```

! hmh adding 5/17/16
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?

```

**OUTPUT PARAMETERS:**

```

REAL*8, INTENT(OUT) :: FLUX(IIPAR, JJPAR, 1) ! hmh 9/14/15
INTEGER, INTENT(OUT) :: RC ! Success or failure?
CHARACTER(LEN=255) :: LOC = 'OFFLINEOCEAN_READMO (mercury_mod.F)'

```

**REVISION HISTORY:**

```

12 Aug 2015 - H. Horowitz - Initial version
15 Sep 2015 - H. Horowitz - removing am_I_root & trying to fix netcdf issue.
also removing input_opt; don't need it here. Try making Array 2D instead

```

of 3-D.

16 Sep 2015 - H. Horowitz - adding diags for Hg\_up and Hg\_down (separates the positive and negative oceanic fluxes from the net EHg0\_oc.

17 May 2016 - H. Horowitz - setting up to read concentrations and do flux calculations like in ocean\_mercury\_mod subroutine ocean\_mercury\_flux

30 Nov 2017 - C. Thackray (cpt) - merging into v11

-----  
BOC

#### LOCAL VARIABLES:

```

INTEGER :: I, J, M, DN(12) ! M is mon., DN is days in mon
INTEGER :: NNN
INTEGER :: THISYEAR
INTEGER :: THISMONTH

LOGICAL, SAVE :: FIRST = .TRUE.
LOGICAL :: IS_OCEAN_BOX
INTEGER :: NN, N
REAL(fp) :: A_M2, DTSRCE
REAL(fp) :: CHg0aq, CHg0, vi, Hg0aqtemp
REAL(fp) :: TC, TK, Kw
REAL(fp) :: Sc, ScCO2, USQ
REAL(fp) :: FRAC_L, FRAC_O, H, D
REAL(fp) :: FUP(IIPAR,JJPAN,N_Hg_CATS)
REAL(fp) :: FDOWN(IIPAR,JJPAN,N_Hg_CATS)
REAL(fp) :: Hg0aq(IIPAR,JJPAN,N_Hg_CATS)
REAL(fp) :: MHg0_air

! Conversion factor from [cm/h * ng/L] --> [kg/m2/s]
REAL(fp), PARAMETER :: TO_KGM2S = 1.0e-11_fp / 3600.0e+0 _fp

! Small numbers to avoid dividing by zero
REAL(fp), PARAMETER :: SMALLNUM = 1.0e-32_fp

REAL(fp) :: SFCWINDSQR

! Pointers
! We need to define local arrays to hold corresponding values
! from the Chemistry State (State_Chm) object. (mpayer, 12/6/12)
REAL(fp), POINTER :: STT(:, :, :, :)

! For values from Input_Opt
LOGICAL :: LDYNOCEAN

!=====
! OFFLINEOCEAN_READMO begins here!

```

```

=====

! Assume success
RC = GC_SUCCESS

THISMONTH = GET_MONTH()

M = THISMONTH

! Days in each month (will use later) 9/16/15 hmh
DN = (/ 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /)

!%%%%%%%%%%
! read monthly ocean evasion !
!%%%%%%%%%%

IF (ITS_A_NEW_MONTH()) THEN

 CALL HCO_GetPtr(am_I_Root, HcoState, 'GLOBAL_OCEAN',
& OCEAN_CONC, RC)
 IF (RC /= GC_SUCCESS) THEN
 CALL ERROR_STOP ('Cannot get pointer to GLOBAL_OCEAN', LOC)
 ENDIF

ENDIF

! Only doing Hg0 overall, should add trap for LSPLIT (cpt - 2017)
DO NNN = 1, N_Hg_CATS
 IF (NNN .eq. 1) THEN
 Hg0aq(:, :, NNN) = OCEAN_CONC(:, :, 1)
 ELSE
 Hg0aq(:, :, NNN) = 0.0e+0_fp
 ENDIF
ENDDO

! Emission timestep [s]
DTSRCE = GET_TS_EMIS()

STT => State_Chm%Species

! Loop over surface boxes
$OMP PARALLEL DO
$OMP+DEFAULT(SHARED)
$OMP+PRIVATE(I, A_M2, vi, ScCO2)
$OMP+PRIVATE(J, NN, TC, TK)
$OMP+PRIVATE(N, CHg0, FRAC_L, FRAC_O)
$OMP+PRIVATE(H, Kw, CHg0aq, Hg0aqtemp, MHg0_air)

```

```

$OMP+PRIVATE(IS_OCEAN_BOX, Sc, Usq, D)
$OMP+SCHEDULE(DYNAMIC)
 DO J = 1, JJPAR
 DO I = 1, IIPAR

 ! Grid box surface area [m2]
 A_M2 = GET_AREA_M2(I, J, 1)
 ! Initialize values
 Kw = 0e0_fp
 TK = 0e0_fp
 TC = 0e0_fp

 ! Get fractions of land and ocean in the grid box [unitless]
 ! Use fractional land type information in MERRA. Also make sure
 ! we do not use boxes that are mostly sea ice for consistency
 ! FROCEAN is a constant, so to get correct ocean fraction we
 ! need to subtract the sea ice fraction. Don't let the fraction
 ! be less than zero (jaf, 4/26/11)
#if defined(MERRA) || defined(GEOS_FP) || defined(MERRA2)
 FRAC_L = State_Met%FRLAND(I,J)
 FRAC_O = MAX(State_Met%FROCEAN(I,J) -
& State_Met%FRSEAICE(I,J), 0e0_fp)
 IS_OCEAN_BOX = ((FRAC_O > 0e0_fp
& (State_Met%SEAICE00(I,J) > 0.5e0_fp)))
#else
 FRAC_L = State_Met%FRCLND(I,J)
 FRAC_O = 1e0_fp - FRAC_L
 IS_OCEAN_BOX = (IS_WATER(I, J, State_Met))
#endif

 IF ((IS_OCEAN_BOX)) THEN
 !-----
 ! Sea surface temperature in both [K] and [C]
 !-----
 ! where TSKIN is the temperature (K) at the ground/sea surface
 ! (Use as surrogate for SST, cap at freezing point)

 TK = MAX(State_Met%TSKIN(I,J), 273.15e0_fp)

 TC = TK - 273.15e0_fp

 !=====
 ! Volatilisation of Hg0
 !=====

 ! Henry's law constant (gas->liquid) [unitless] [L water/L air]
 ! (ref: Andersson et al. 2008)

```

```

H = EXP((-2404.3e0_fp / TK) + 6.92e0_fp)

! Viscosity as a function of changing temperatures
! (ref: Loux 2001)
! The paper says the viscosity is given in cP but us really P
! and we therefor multiply with 100 to get cP.
vi = (10** ((1301.0e0_fp / (998.333e0_fp + 8.1855e0_fp
& * (TC - 20.0e0_fp) + 0.00585e0_fp
& * (TC - 20.0e0_fp)**2))
& - 3.30233e0_fp)) * 100.0e0_fp

! Schmidt # for Hg [unitless]
! Sc = v/D = kinematic viscosity/diffusivity
! (ref: Poissant et al 2000; Wilke and Chang 1995)
! to correct for seawater D0 is decreased by 6% as suggested
! by Wanninkhof (1992)

D = 7.4e-8_fp * sqrt(2.26e0_fp * 18.0e0_fp) * TK /
& ((14.8e0_fp**0.6e0_fp) * vi)

Sc = (0.017e0_fp * EXP(-0.025e0_fp * TC)) / D

! Schmidt # of CO2 [unitless] for CO2 in seawater at 20 degrees C
! The value is set to a constant based on other ocean studies
! (Gardfeld et al. 2003, Rolffhus & Fitzgerald2004,Mason et al.2001)

! Correction of the Schmidt # with temperature based on Poissant
! et al. (2000) (for freshwatersystems).

ScCO2 = 644.7e0_fp + TC * (-6.16e0_fp + TC * (0.11e0_fp))

! Square of surface (actually 10m) wind speed [m2/s2]
Usq = State_Met%U10M(I,J)**2 + State_Met%V10M(I,J)**2

!-----
! Parameterizations for calculating water side mass trasfer
! coefficient
!-----
! Mass transfer coefficient [cm/h], from Nightingale et al. 2000
Kw = (0.25e0_fp * Usq) / SQRT(Sc / ScCO2)
! Loop over all Hg categories
DO NN = 1, N_Hg_CATS

 ! Hg0 tracer number (for STT)
 N = Hg0_Id_List(NN)

!-----
! Calculate oceanic and gas-phase concentration of Hg(0)

```

```

!-----
! Concentration of Hg(0) in the ocean [ng/L]
! now converting from Hg0aq in mol/m3 to ng/L
CHg0aq = Hg0aq(I,J,NN) *200.59e0_fp * 1.0e9_fp / 1.0e3_fp

! Gas phase Hg(0) concentration: convert [kg] -> [ng/L]
MHg0_air = STT(I,J,1,N)
CHg0 = MHg0_air
& * 1.0e9_fp /State_Met%AIRVOL(I,J,1)

!-----
! Compute flux of Hg(0) from the ocean to the air
!-----

! Compute ocean flux of Hg0 [cm/h*ng/L]
FLUX(I,J,NN) = Kw * (CHg0aq - (CHg0 / H))

!Extra diagnostic: compute flux up and flux down
FUP(I,J,NN) = (Kw * CHg0aq)
FDOWN(I,J,NN) = (Kw * CHg0 / H)
!-----
! Convert [cm/h*ng/L] --> [kg/m2/s] --> [kg/s]
! Also account for ocean fraction of grid box
FLUX(I,J,NN) = FLUX(I,J,NN) * TO_KGM2S * A_M2 * FRAC_0

! hmh 5/11/16 reverting to old version and uncommenting here
FUP(I,J,NN) = FUP(I,J,NN) * TO_KGM2S * A_M2 * FRAC_0
FDOWN(I,J,NN) = FDOWN(I,J,NN) * TO_KGM2S * A_M2 * FRAC_0
!-----

!-----
! Flux limited by ocean and atm Hg(0)
!-----

! Cap the flux w/ the available Hg(0) ocean mass
! THIS IS THE PROBLEM!
Hg0aqtemp = CHg0aq * A_M2 * FRAC_0 *1.0e-8_fp

IF (FLUX(I,J,NN) * DTSRCE > Hg0aqtemp) THEN
 FLUX(I,J,NN) = Hg0aqtemp / DTSRCE
 FUP(I,J,NN) = FLUX(I,J,NN)-FDOWN(I,J,NN)
ENDIF

! Cap the neg flux w/ the available Hg(0) atm mass

IF ((-FLUX(I,J,NN) * DTSRCE) > MHg0_air) THEN
 FLUX(I,J,NN) = -MHg0_air / DTSRCE

```

```

ENDIF

! make sure Fup and Fdown do not underflow either
! debug 2x2.5 diagnostic?
FUP(I,J,NN) = MAX(FUP(I,J,NN), SMALLNUM)
FDOWN(I,J,NN) = MAX(FDOWN(I,J,NN), SMALLNUM)

IF (ND03 > 0) THEN
 ADO3(I,J,16,NN) = ADO3(I,J,16,NN)
& + FUP(I,J,NN) * DTSRCE
 ADO3(I,J,17,NN) = ADO3(I,J,17,NN)
& + FDOWN(I,J,NN) *DTSRCE
ENDIF
ENDDO

ELSE

DO NN = 1, N_Hg_CATS
 FLUX(I,J,NN) = 0e0_fp
 FUP(I,J,NN)=0e0_fp
 FDOWN(I,J,NN)=0e0_fp
ENDDO

ENDIF
ENDDO
ENDDO
$OMP END PARALLEL DO

! Free pointer
NULLIFY(STT)

END SUBROUTINE OFFLINEOCEAN_READMO
EOC

```

---

GEOS-Chem Global Chemical Transport Model

---

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\mbox{}\hrulefill\

\subsubsection [init\\_mercury] {init\\_mercury}

Subroutine INIT\\_MERCURY allocates and zeroes all  
module arrays.

\\

\\{\bf INTERFACE:}

```
\begin{verbatim} SUBROUTINE INIT_MERCURY(am_I_Root, Input_Opt, State_Chm, RC)
```

**USES:**

```
 USE CMN_DIAG_MOD
 USE CMN_SIZE_MOD
 USE ErrCode_Mod
 USE ERROR_MOD, ONLY : ALLOC_ERR, ERROR_STOP
 USE Input_Opt_Mod, ONLY : OptInput
 USE Species_Mod, ONLY : Species
 USE State_Chm_Mod, ONLY : ChmState
```

**INPUT PARAMETERS:**

```
 LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
 TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
```

**OUTPUT PARAMETERS:**

```
 INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
02 Dec 2004 - N. (Eckley) Selin - Initial version
(1) Removed reference to FEMIS array. Now also allocates and zeroes
 the T44 array. Added reference to CMN_DIAG. Now references
 N_TRACERS from "tracer_mod.f". (bmy, 2/24/05)
(2) EHg0_an, EHg2_an, EHgP_an are now 2-D arrays. Now modified for
 updated ocean mercury module. (eck, cdh, sas, bmy, 4/6/06)
27 Sep 2011 - H. Amos - remove LHg_WETDasHNO3 logical, it's obsolete
11 Apr 2012 - R. Yantosca - Now retire lai_mod.F; remove call to INIT_LAI
11 Apr 2012 - R. Yantosca - Now move LAI initialization to main.F
03 Jun 2013 - R. Yantosca - Now use fields from Input_Opt
12 Mar 2015 - R. Yantosca - Now test if HEMCO_Collections are defined
25 Apr 2016 - R. Yantosca - Now use species database for tagHg indexing
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo
```

**10.5.27 set\_options\_from\_hemco**

Overrides some of the Hg simulation settings depending on the inputs that are specified in the HEMCO configuration file.

**INTERFACE:**

```
 SUBROUTINE SET_OPTIONS_FROM_HEMCO(am_I_Root, Input_Opt, RC)
```

**USES:**



```
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_ERROR_MOD
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_ExtList_Mod, ONLY : GetExtOpt
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root
TYPE(OptInput), INTENT(IN) :: Input_Opt
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REMARKS:

#### REVISION HISTORY:

```
06 Jan 2015 - R. Yantosca - Initial version
11 Oct 2016 - R. Yantosca - Move call to DO_RED_INPLUME here
13 Feb 2018 - C. Thackray - Put chem switches here
```

---

#### 10.5.28 cleanup\_mercury

Subroutine CLEANUP\_MERCURY deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_MERCURY
```

#### REVISION HISTORY:

```
06 Dec 2004 - N. (Eckley) Selin - Initial version
(1) Now deallocate MLD, NPP, RAD (sas, bmy, 1/18/05)
(2) Now deallocate T44 (bmy, 2/24/05)
```

---

#### 10.6 Fortran: Module Interface depo\_mercury\_mod.F

Module DEPO\_MERCURY\_MOD contains routines to handle deposition fluxes for mercury.

#### INTERFACE:

```
MODULE DEPO_MERCURY_MOD
```

#### USES:

```
USE PRECISION_MOD ! For Geos-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: ADD_Hg2_DD
PUBLIC :: ADD_Hg2_WD
PUBLIC :: ADD_HgP_DD
PUBLIC :: ADD_HgP_WD
PUBLIC :: ADD_HG2_SNOWPACK
PUBLIC :: RESET_HG_DEP_ARRAYS
PUBLIC :: CHECK_DIMENSIONS
PUBLIC :: READ_GTMM_RESTART
PUBLIC :: MAKE_GTMM_RESTART
PUBLIC :: UPDATE_DEP
PUBLIC :: INIT_DEPO_MERCURY
PUBLIC :: CLEANUP_DEPO_MERCURY
```

#### PUBLIC DATA MEMBERS:

```
PUBLIC :: DD_HG2, DD_HGP, WD_HG2, WD_HGP
PUBLIC :: HG2mth_wd, HG0mth_dd, HG2mth_dd
PUBLIC :: SNOW_HG_OC
PUBLIC :: SNOW_HG_LN
PUBLIC :: SNOW_HG_STORED_OC
PUBLIC :: SNOW_HG_STORED_LN
PUBLIC :: LHGSNOW
```

```
REAL(fp), ALLOCATABLE :: DD_Hg2(:,:,:)
REAL(fp), ALLOCATABLE :: DD_HgP(:,:,:)
REAL(fp), ALLOCATABLE :: WD_Hg2(:,:,:)
REAL(fp), ALLOCATABLE :: WD_HgP(:,:,:)
REAL(fp), ALLOCATABLE :: HG0mth_dd(:,:)
REAL(fp), ALLOCATABLE :: HG2mth_dd(:,:)
REAL(fp), ALLOCATABLE :: HG2mth_wd(:,:)
REAL(fp), ALLOCATABLE :: SNOW_HG_OC(:,:,:)
REAL(fp), ALLOCATABLE :: SNOW_HG_LN(:,:,:)
REAL(fp), ALLOCATABLE :: SNOW_HG_STORED_OC(:,:,:)
REAL(fp), ALLOCATABLE :: SNOW_HG_STORED_LN(:,:,:)
REAL(fp), ALLOCATABLE :: Hg0dryGEOS(:,:), HgIIdryGEOS(:,:),
& HgIIwetGEOS(:,:)
```

#### !PRIVATE DATA MEMBERS:

```
CHARACTER(LEN=255) :: GTMM_RST_FILE
LOGICAL :: LHGSNOW
```

#### REVISION HISTORY:

23 Apr 2010 - C. Carouge - Initial version



```

INTEGER, INTENT(IN) :: I, J ! Grid box lon & lat indices
INTEGER, INTENT(IN) :: Hg_Cat ! Hg category number
REAL(fp), INTENT(IN) :: WET_Hg2 ! Hg(II) scavenged out of the
 ! atmosphere [kg]

```

**REVISION HISTORY:**

```

19 Jan 2005 - S. Strode, C. Holmes - Initial version
(1) WD_Hg2 is now a 3-D array. Also pass N via the argument list. Now
 call GET_Hg2_CAT to return the Hg category #. (cdh, bmy, 3/28/06)
23 Apr 2010 - C. Carouge - Moved from ocean_mercury_mod.f to
 depo_mercury_mod.f
25 Apr 2016 - R. Yantosca - Remove reference to GET_HG2_CAT, we now pass
 this in as an argument. Rename NN to Hg2_Cat.

```

**10.6.3 add\_HgP\_dd**

Subroutine ADD\_HgP\_DD computes the amount of HgP dry deposited out of the atmosphere into the column array DD\_HgP.

**INTERFACE:**

```

SUBROUTINE ADD_HgP_DD(I, J, Hg_Cat, DRY_HgP)

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I, J ! Grid box lon & lat indices
INTEGER, INTENT(IN) :: Hg_Cat ! Hg category number
REAL(fp), INTENT(IN) :: DRY_HgP ! HgP dry deposited out of the
 ! atmosphere [kg]

```

**REVISION HISTORY:**

```

19 Jan 2005 - S. Strode, C. Holmes - Initial version
(1) DD_HgP is now a 3-D array. Also pass N via the argument list. Now
 call GET_HgP_CAT to return the Hg category #. (cdh, bmy, 3/28/06)
23 Apr 2010 - C. Carouge - Moved from ocean_mercury_mod.f to
 depo_mercury_mod.f
25 Apr 2016 - R. Yantosca - Remove reference to GET_HGP_CAT, we now pass
 this in as an argument. Rename NN to HgP_Cat.

```

**10.6.4 add\_HgP\_wd**

Subroutine ADD\_HgP\_WD computes the amount of HgP wet scavenged out of the atmosphere into the column array WD\_HgP.

**INTERFACE:**

```

SUBROUTINE ADD_HgP_WD(I, J, Hg_Cat, WET_HgP)

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: I, J ! Grid box lon & lat indices
INTEGER, INTENT(IN) :: Hg_Cat ! Hg category number
REAL(fp), INTENT(IN) :: WET_HgP ! HgP scavenged out of the
 ! atmosphere [kg]

```

**REVISION HISTORY:**

```

19 Jan 2005 - S. Strode, C. Holmes - Initial version
(1) WD_HgP is now a 3-D array. Also pass N via the argument list. Now
 call GET_HgP_CAT to return the Hg category #. (cdh, bmy, 3/28/06)
23 Apr 2010 - C. Carouge - Moved from ocean_mercury_mod.f to
 depo_mercury_mod.f
25 Apr 2016 - R. Yantosca - Remove reference to GET_HGP_CAT, we now pass
 this in as an argument. Rename NN to HgP_Cat.

```

**10.6.5 add\_hg2\_snowpack**

Subroutine ADD\_Hg2\_SNOWPACKS adds Hg2 deposition to snowpack.

**INTERFACE:**

```

SUBROUTINE ADD_HG2_SNOWPACK(I, J, Hg_Cat, DEP_Hg2, State_Met)

```

**USES:**

```

USE DAO_MOD, ONLY : IS_ICE, IS_LAND
USE State_Met_Mod, ONLY : MetState
#if defined(BPCH_DIAG)
USE DIAGO3_MOD, ONLY : ADO3, NDO3
#endif

```

**INPUT PARAMETERS:**

```

! Arguments as input
INTEGER, INTENT(IN) :: I, J ! Grid box lon & lat indices
INTEGER, INTENT(IN) :: Hg_Cat ! Hg category number
REAL(fp), INTENT(IN) :: Dep_Hg2 ! Hg2 (or HgP) deposited
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**REVISION HISTORY:**

```

02 Sep 2008 - C. Holmes - Initial version
23 Apr 2010 - C. Carouge - Moved from mercury_mod.f to depo_mercury_mod.f
25 Aug 2010 - R. Yantosca - Treat MERRA in the same way as GEOS-5
26 Apr 2011 - J. Fisher - Use MERRA land fraction information
12 Apr 2011 - J. Fisher - Add missing code from Holmes 2010
13 Apr 2011 - R. Yantosca - Bug fix: reference IS_LAND from dao_mod.f
8 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA

```

09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

26 Sep 2013 - R. Yantosca - Renamed GEOS\_57 Cpp switch to GEOS\_FP

30 May 2013 - J. Fisher - Add land snow Hg reservoirs

06 Aug 2015 - J. Fisher - Store reservoir of non-reducible Hg in snowpack (orig. 29/11/11)

11 Aug 2015 - R. Yantosca - MERRA2 behaves the same way as GEOS\_FP

25 Apr 2016 - R. Yantosca - Remove reference to routines GET\_HG2\_CAT and GET\_HGP\_CAT, these are now passed in as the Hg\_Cat argument.

### 10.6.6 reset\_hg\_dep\_arrays

Subroutine RESET\_Hg\_DEP\_ARRAYS resets the wet and dry deposition arrays for Hg(II) and Hg(p) to zero. This allows us to call OCEAN\_MERCURY\_FLUX and LAND\_MERCURY\_FLUX in any order in MERCURY\_MOD.

#### INTERFACE:

```
SUBROUTINE RESET_HG_DEP_ARRAYS
```

#### REVISION HISTORY:

02 Sep 2008 - C. Holmes - Initial version

23 Apr 2010 - C. Carouge - Moved from ocean\_mercury\_mod.f to depo\_mercury\_mod.f

### 10.6.7 make\_gtmm\_restart

MAKE\_GTMM\_RESTART writes a GTMM restart file with deposition fluxes and store deposition fluxes for continuous runs.

#### INTERFACE:

```
SUBROUTINE MAKE_GTMM_RESTART(am_I_Root, Input_Opt, NYMD,
& NHMS, TAU, RC)
```

#### USES:

```
USE BPCH2_MOD
USE CMN_SIZE_MOD ! Size parameters
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_XOFFSET, GET_YOFFSET
USE Input_Opt_Mod, ONLY : OptInput
USE inquireMod, ONLY : findFreeLUN
USE TIME_MOD, ONLY : EXPAND_DATE
USE TIME_MOD, ONLY : GET_CT_DYN, GET_CT_CHEM
```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: NYMD ! Year-Month-Date
INTEGER, INTENT(IN) :: NHMS ! and Hour-Min-Sec
REAL(f8), INTENT(IN) :: TAU ! TAU value: hrs since 1/1/85
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

15 Sep 2009 - C. Carouge - Initial version
03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block
07 Aug 2012 - R. Yantosca - Now print LUN used to open file
20 Jun 2014 - R. Yantosca - Remove reference to directory_mod.F

```

**10.6.8 read\_gtmm\_restart**

Subroutine READ\_GTMM\_RESTART reads dry and wet deposition for mercury from GTMM restart.

**INTERFACE:**

```

SUBROUTINE READ_GTMM_RESTART(am_I_Root,
& Input_Opt, State_Chm,
& YYYYMMDD, HHMMSS,
& HgOdryGEOS, HgIIDryGEOS,
& HgIIwetGEOS, RC)

```

**USES:**

```

USE BPCH2_MOD, ONLY : OPEN_BPCH2_FOR_READ
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
USE inquireMod, ONLY : findFreeLun
USE State_Chm_Mod, ONLY : ChmState
USE TIME_MOD, ONLY : EXPAND_DATE

```

**INPUT PARAMETERS:**

```

INTEGER, INTENT(IN) :: YYYYMMDD, HHMMSS
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?!
REAL(fp), DIMENSION(IIPAR, JJPAR) :: Hg0dryGEOS
REAL(fp), DIMENSION(IIPAR, JJPAR) :: HgIIdryGEOS
REAL(fp), DIMENSION(IIPAR, JJPAR) :: HgIIwetGEOS
!REMAKRS:
NOTE: THIS ROUTINE WILL HAVE TO BE UPDATED TO CONVERT TO NETCDF FORMAT!!!
```

**REVISION HISTORY:**

```
15 Sep 2009 - C. Carouge - Initial version
25 Jun 2014 - R. Yantosca - Remove references to tracer_mod.F
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
16 Mar 2017 - R. Yantosca - Replace NNPAR with State_Chm%nAdvect
```

---

**10.6.9 update\_dep**

Subroutine UPDATE\_DEP update the monthly average for wet and dry deposition of Hg0 and Hg2 for mercury from GTMM restart.

**INTERFACE:**

```
SUBROUTINE UPDATE_DEP(NN)
```

**USES:**

```
#if defined(BPCH_DIAG)
 USE DIAG_MOD, ONLY : AD38, AD39
 USE DIAG_MOD, ONLY : AD44
#endif
 USE TIME_MOD, ONLY : GET_CT_DYN, GET_CT_CHEM
```

**INPUT PARAMETERS:**

```
INTEGER :: NN ! Hg2 ID for wet deposition
```

**REVISION HISTORY:**

```
04 June 2010 - C. Carouge - Initial version
```

---



**10.6.10 check\_dimensions**

Subroutine CHECK\_DIMENSIONS makes sure that the dimensions of the Hg restart file extend to cover the entire grid.

**INTERFACE:**

```
SUBROUTINE CHECK_DIMENSIONS(NI, NJ, NL)
```

**USES:**

```
USE ERROR_MOD, ONLY : GEOS_CHEM_STOP
USE CMN_SIZE_MOD
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: NI, NJ, NL
```

**REVISION HISTORY:**

```
30 Aug 2010 - S. Strode, C. Holmes - Initial version
```

---

**10.6.11 init\_depo\_mercury**

Subroutine INIT\_DEPO\_MERCURY initialize deposition arrays for mercury.

**INTERFACE:**

```
SUBROUTINE INIT_DEPO_MERCURY(am_I_Root, Input_Opt,
& State_Chm, RC)
!USES
```

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE Input_Opt_Mod, ONLY : OptInput
USE Species_Mod, ONLY : Species
USE State_Chm_Mod, ONLY : ChmState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

23 Apr 2010 - C. Carouge - Moved arrays allocation from ocean\_mercury\_mod.f  
23 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC  
30 May 2013 - J. Fisher - Add land snow Hg reservoirs  
06 Aug 2015 - J. Fisher - Add stored snow Hg reservoirs (orig. 29/11/11)  
25 Apr 2016 - R. Yantosca - Now handle all Hg indexing locally  
29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
01 Jul 2016 - R. Yantosca - Now rename species DB object ThisSpc to SpcInfo

---

### 10.6.12 cleanup\_depo\_mercury

Subroutine CLEANUP\_DEPO\_MERCURY deallocate all arrays

#### INTERFACE:

```
SUBROUTINE CLEANUP_DEPO_MERCURY
```

#### REVISION HISTORY:

23 Apr 2010 - C. Carouge - Moved from ocean\_mercury\_mod.f  
30 May 2013 - J. Fisher - Add land snow Hg reservoirs  
06 Aug 2015 - J. Fisher - Add stored snow Hg reservoirs (orig. 29/11/11)

---

## 10.7 Fortran: Module Interface land\_mercury\_mod.F

Module LAND\_MERCURY\_MOD contains variables and routines for the land emissions for the GEOS-Chem mercury simulation.

#### INTERFACE:

```
MODULE LAND_MERCURY_MOD
```

#### USES:

```
USE HCO_ERROR_MOD ! For real precisions (hp)
USE PRECISION_MOD ! For GEOS-Chem Precision (fp, f4, f8)
```

```
IMPLICIT NONE
PRIVATE
```

#### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: BIOMASSHG
PUBLIC :: VEGEMIS
PUBLIC :: SOILEMIS
PUBLIC :: LAND_MERCURY_FLUX
PUBLIC :: GTMM_DR
PUBLIC :: SNOWPACK_MERCURY_FLUX
PUBLIC :: INIT_LAND_MERCURY
PUBLIC :: CLEANUP_LAND_MERCURY
```

**REVISION HISTORY:**

02 Jun 2010 - N. E. Selin, C. Carouge - Initial version  
 02 Jun 2010 - C. Carouge - Group all land emissions routine for mercury  
                   into this new module.  
 13 Aug 2010 - R. Yantosca - Added modifications for MERRA  
 25 Aug 2010 - R. Yantosca - Treat MERRA in same way as GEOS-5  
 30 Aug 2010 - R. Yantosca - Added more ProTeX headers  
 12 Apr 2011 - J. Fisher - Add missing code from Holmes 2010  
 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA  
 01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90  
 11 Apr 2012 - R. Yantosca - Now reference new modis\_lai\_mod.F90  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 17 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90  
 24 Aug 2017 - M. Sulprizio- Remove support for GCAP, GEOS-4, GEOS-5 and MERRA

**10.7.1 land\_mercury\_flux**

Subroutine LAND\_MERCURY\_FLUX calculates emissions of Hg(0) from prompt recycling of previously deposited mercury to land, in [kg/s].

**INTERFACE:**

```
SUBROUTINE LAND_MERCURY_FLUX(LFLUX, LHGSNOW, State_Met)
```

**USES:**

```
USE CMN_SIZE_MOD ! Size parameters
USE DAO_MOD, ONLY : IS_ICE, IS_LAND
USE DEPO_MERCURY_MOD, ONLY : WD_HGP, WD_HG2, DD_HGP, DD_HG2
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: LHGSNOW ! Use Hg0 from snow?
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

**OUTPUT PARAMETERS:**

```
REAL(fp), INTENT(OUT) :: LFLUX(IIPAR,JJPAN,N_Hg_CATS) ! Hg0 flux
 ! [kg/s]
```

**REVISION HISTORY:**

30 Aug 2010 - N. E. Selin, C. Holmes, B. Corbitt - Initial version  
 (1 ) Now uses SNOWMAS from DAO\_MOD for compatibility with GEOS-5.  
       (eds 7/30/08)  
 (2 ) Now includes REEMFRAC in parallelization; previous versions may have

overwritten variable. (cdh, eds 7/30/08)

(3 ) Now also reemit Hg(0) from ice surfaces, including sea ice  
(cdh, 8/19/08)

13 Aug 2010 - R. Yantosca - Add modifications for MERRA  
 25 Aug 2010 - R. Yantosca - Treat MERRA in same way as GEOS-5  
 26 Apr 2011 - J. Fisher - Use MERRA land fraction information  
 12 Apr 2011 - J. Fisher - Add missing code from Holmes 2010  
 08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA  
 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met  
 derived type object  
 26 Sep 2013 - R. Yantosca - Renamed GEOS\_57 Cpp switch to GEOS\_FP  
 12 Aug 2015 - R. Yantosca - Add support for MERRA2 meteorology

### 10.7.2 biomasshg

Subroutine BIOMASSHG is the subroutine for Hg(0) emissions from biomass burning. These emissions are active only for present day simulations and not for preindustrial simulations.

#### INTERFACE:

```
SUBROUTINE BIOMASSHG(am_I_Root, Input_Opt, EHg0_bb, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_ERROR_MOD
USE HCO_STATE_MOD, ONLY : HCO_STATE
USE HCO_INTERFACE_MOD, ONLY : GetHcoDiagn
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE Input_Opt_Mod, ONLY : OptInput
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Root CPU
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options
```

#### OUTPUT PARAMETERS:

```
REAL(fp), DIMENSION(:,,:),INTENT(OUT) :: EHg0_bb
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REMARKS:

Emissions are based on an inventory of CO emissions from biomass burning (Duncan et al. J Geophys Res 2003), multiplied by a Hg/CO ratio in BB plumes from Franz Slemr (Poster, EGU 2006).

Slemr surveyed emission factors from measurements worldwide. Although his best estimate was  $1.5e-7$  mol Hg/ mol CO, we chose the highest value ( $2.1e-7$  mol Hg/ mol CO) in the range because the simulations shown in Selin et al. (GBC 2008) required large Hg(0) emissions to sustain reasonable atmospheric Hg(0) concentrations. (eck, 11/13/2008)

## REVISION HISTORY:

30 Jul 2008 - N. E. Selin, C. Holmes, B. Corbitt - Initial version  
 12 Apr 2011 - J. Fisher - Add missing code from Holmes 2010  
 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
 15 Aug 2014 - R. Yantosca - Removed reference to IDBCO; now obsolete  
 23 Sep 2014 - C. Keller - Now use HEMCO diagnostics  
 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts

### 10.7.3 vegemis

Subroutine VEGEMIS is the subroutine for Hg(0) emissions from vegetation by evapotranspiration.

## INTERFACE:

```
! Bug fix: VEGEMIS shouldn't be tied to GCAP emissions
! (jaf, eds, 4/1/11)
SUBROUTINE VEGEMIS(am_I_Root, Input_Opt, State_Met,
& LVEGEMIS, EHg0_dist, EHg0_vg, RC)
```

## USES:

```
USE CMN_SIZE_MOD
USE DAO_MOD, ONLY : IS_LAND
USE ErrCode_Mod
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE Input_Opt_Mod, ONLY : OptInput
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH, ITS_A_NEW_MONTH
```

## INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Met State object
LOGICAL, INTENT(IN) :: LVEGEMIS !
REAL(fp), DIMENSION(:,:), INTENT(IN) :: EHg0_dist !
```

## OUTPUT PARAMETERS:

```
REAL(fp), DIMENSION(:,:), INTENT(OUT) :: EHg0_vg
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REMARKS:**

Vegetation emissions are proportional to the evapotranspiration rate and the soil water mercury content. We assume a constant concentration of mercury in soil matter, based on the preindustrial and present-day simulations described in Selin et al. (GBC 2008) and in SOILEMIS subroutine. From the soil matter Hg concentration, we calculate a soil water Hg concentration in equilibrium (Allison and Allison, 2005).

NASA provides a climatology of evapotranspiration based on a water budget model (Mintz and Walker, 1993).

Calculate vegetation emissions following Xu et al (1999)

$$F_c = E_c C_w$$

$F_c$  is Hg0 flux (ng m<sup>-2</sup> s<sup>-1</sup>)

$E_c$  is canopy transpiration (m s<sup>-1</sup>)

$C_w$  is conc of Hg0 in surface soil water (ng m<sup>-3</sup>)

Calculate  $C_w$  from the Allison and Allison (2005) equilibrium formula

$$C_w = C_s / K_d$$

$C_s$  is the concentration of Hg in surface soil solids, ng/g

$K_d$  is the equilibrium constant = [sorbed]/[dissolved]

$$\log K_d = 3.8 \text{ L/kg} \rightarrow K_d = 6310 \text{ L/kg} = 6.31 \text{D-3 m}^3/\text{g}$$

We assume a global mean  $C_s = 45$  ng/g for the preindustrial period. In iterative simulations we redistribute this according to the deposition pattern while maintaining the global mean. The scaling factor, EHg0\_dist, also accounts for the anthropogenic enhancement of soil Hg in the present day.

**REVISION HISTORY:**

30 Aug 2010 - N. Eckley, C. Holmes, B. Corbitt - Initial version

01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_M2(I,J,L) from grid\_mod.F90

20 Jun 2014 - R. Yantosca - Now accept Input\_Opt via the arg list

**10.7.4 soilemis**

Subroutine SOILEMIS is the subroutine for Hg(0) emissions from soils.

**INTERFACE:**

```
SUBROUTINE SOILEMIS(EHg0_dist, EHg0_so, State_Met)
```

**USES:**

```
USE CMN_SIZE_MOD ! Size parameters
USE DAO_MOD, ONLY : IS_LAND
USE GC_GRID_MOD, ONLY : GET_AREA_M2
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_MONTH, ITS_A_NEW_MONTH
```

**INPUT PARAMETERS:**

```
REAL(fp), DIMENSION(:,,:), INTENT(IN) :: EHg0_dist
TYPE(MetState), INTENT(IN) :: State_Met ! Met State object
```

**OUTPUT PARAMETERS:**

```
REAL(fp), DIMENSION(:,,:), INTENT(OUT) :: EHg0_so
```

**REMARKS:**

Soil emissions are a function of solar radiation at ground level (accounting for attenuation by leaf canopy) and surface temperature. The radiation dependence from Zhang et al. (2000) is multiplied by the temperature dependence from Poissant and Casimir (1998). Finally, this emission factor is multiplied by the soil mercury concentration and scaled to meet the global emission total. Comments on soil Hg concentration:

-----  
We chose the preindustrial value of 45 ng Hg /g dry soil as the mean of the range quoted in Selin et al. (GBC 2008): 20-70 ng/g (Andersson, 1967; Shacklette et al., 1971; Richardson et al., 2003; Frescholtz and Gustin, 2004). Present-day soil concentrations are thought to be 15% greater than preindustrial (Mason and Sheu 2002), but such a difference is much less than the range of concentrations found today, so not well constrained. We calculate the present-day soil Hg distribution by adding a global mean 6.75 ng/g (=0.15 \* 45 ng/g) according to present-day Hg deposition. (eck, 11/13/08)

**REVISION HISTORY:**

30 Aug 2010 - N. Eckley, B. Corbitt - Initial version  
(1 ) Added comments. (cdh, eds, 7/30/08)  
(2 ) Now include light attenuation by the canopy after sunset. Emissions change by < 1% in high-emission areas (cdh, 8/13/2008)  
(3 ) Removed FRCLND for consistency with other Hg emissions (cdh, 8/19/08)  
2 June 2010 - C. Carouge - Solve  
13 Aug 2010 - R. Yantosca - Added modifications for MERRA  
25 Aug 2010 - R. Yantosca - Treat MERRA in same way as GEOS-5  
26 Apr 2011 - J. Fisher - Use MERRA land fraction information  
12 Apr 2011 - J. Fisher - Bug fixes, add missing code from Holmes 2010  
08 Feb 2012 - R. Yantosca - Treat GEOS-5.7.x in the same way as MERRA  
10 Feb 2012 - R. Yantosca - Extend #if statement for SOIL\_EMIS\_FAC in order to get the code to compile w/o error.  
01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90  
11 Apr 2012 - R. Yantosca - Replace lai\_mod.F with modis\_lai\_mod.F90  
09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object  
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State\_Met%SUNCOS  
26 Sep 2013 - R. Yantosca - Renamed GEOS\_57 Cpp switch to GEOS\_FP  
12 Aug 2015 - R. Yantosca - Add support for MERRA2 meteorology

---

### 10.7.5 snowpack\_mercury\_flux

Subroutine SNOWPACK\_MERCURY\_FLUX calculates emission of Hg(0) from snow and ice.

#### INTERFACE:

```
SUBROUTINE SNOWPACK_MERCURY_FLUX(FLUX, LHGSNOW, State_Met)
```

#### USES:

```
USE CMN_SIZE_MOD ! Size parameters
USE DEPO_MERCURY_MOD, ONLY : SNOW_HG_OC, SNOW_HG_LN
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_EMIS
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: LHGSNOW ! Use Hg from snow?
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

#### OUTPUT PARAMETERS:

```
REAL(fp), INTENT(OUT) :: FLUX(IIPAR,JJPAR,N_Hg_CATS) ! Hg0 flux
 ! [kg/s]
```

#### REMARKS:

Emissions are a linear function of Hg mass stored in the snowpack. The Hg lifetime in snow is assumed to be 180 d when  $T < 270\text{K}$  and 7 d when  $T > 270\text{K}$

$$E = k * \text{SNOW\_HG} \quad : \quad k = 6\text{D-}8 \text{ if } T < 270\text{K}, 1.6\text{D-}6 \text{ otherwise}$$

These time constants reflect the time scales of emission observed in the Arctic and in field studies. Holmes et al 2010

Formulation from Holmes et al. 2010 is now obsolete. Instead, emissions from snow are tied to solar radiation, not temperature. Effective rate constant is in the mid-range of values estimated by Durnford and Dastoor (2011) and consistent with surface air Hg0 observations, as described in Fisher et al. (2011, in review).

#### REVISION HISTORY:

```
15 Sep 2009 - C. Holmes, S. Carouge - Initial version
30 Aug 2010 - R. Yantosca - Added ProTex headers
12 Apr 2011 - J. Fisher - Add missing code from Holmes 2010
18 Nov 2011 - J. Fisher - Update to radiation-dependent formulation
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
28 Nov 2012 - R. Yantosca - Replace SUNCOS with State_Met%SUNCOS
30 Mar 2015 - R. Yantosca - Remove obsolete variables
```

---



**10.7.6 gtmr\_dr**

GTMM\_DR is a driver to call GTMM from GEOS-Chem.

**INTERFACE:**

```

SUBROUTINE GTMM_DR(am_I_Root, Input_Opt, State_Met,
& State_Chm, HgOgtm, RC)

```

**USES:**

```

USE BPCH2_MOD
USE CMN_SIZE_MOD ! Size parameters
USE DAO_MOD, ONLY : IS_LAND
USE DEPO_MERCURY_MOD, ONLY : CHECK_DIMENSIONS
USE DEPO_MERCURY_MOD, ONLY : WD_Hg2, WD_HgP, DD_HgP, DD_Hg2
USE DEPO_MERCURY_MOD, ONLY : READ_GTMM_RESTART
USE ErrCode_Mod
USE FILE_MOD, ONLY : IOERROR
USE Input_Opt_Mod, ONLY : OptInput
USE inquireMod, ONLY : findFreeLun
USE State_Met_Mod, ONLY : MetState
USE State_Chm_Mod, ONLY : ChmState
USE TIME_MOD, ONLY : EXPAND_DATE, YMD_EXTRACT
USE TIME_MOD, ONLY : GET_NYMD, GET_NHMS

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

! Emission of Hg0 calculated by GTMM for the month [kg/s]
REAL(fp), INTENT(OUT) :: HgOgtm(IIPAR, JJPAR)

! Success or failure?
INTEGER, INTENT(OUT) :: RC

```

**REMARKS:**

```

#####
NOTE: BINARY PUNCH INPUT IS BEING PHASED OUT. THIS DATA
WILL EVENTUALLY BE READ IN FROM netCDF FILES VIA HEMCO!
-- Bob Yantosca (05 Mar 2015)
#####

```

**REVISION HISTORY:**

15 Sep 2009 - C. Carouge - Initial version  
 03 Aug 2012 - R. Yantosca - Move calls to findFreeLUN out of DEVEL block  
 05 Mar 2015 - R. Yantosca - Now add Input\_Opt%RES\_DIR to data path  
 16 Mar 2017 - R. Yantosca - Now pass State\_Chm to READ\_GTMM\_RESTART

---

**10.7.7 init\_land\_mercury**

Subroutine INIT\_LAND\_MERCURY allocates and zeroes all module arrays.

**INTERFACE:**

```

SUBROUTINE INIT_LAND_MERCURY(am_I_Root, Input_Opt,
& State_Chm, RC)

```

**USES:**

```

USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

14 Sep 2009 - C. Carouge - Initial version  
 23 Jun 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC

---

**10.7.8 cleanup\_land\_mercury**

Subroutine CLEANUP\_LAND\_MERCURY deallocates all module arrays.

**INTERFACE:**

```

SUBROUTINE CLEANUP_LAND_MERCURY

```

**REVISION HISTORY:**

14 Sep 2009 - C. Carouge - Initial version

---

## 10.8 Fortran: Module Interface pops\_mod.F

Module POPS\_MOD contains variables and routines for the GEOS-Chem persistent organic pollutants (POPs) simulation.

### INTERFACE:

```
MODULE POPS_MOD
```

### USES:

```
USE PhysConstants ! For physical constants
USE PRECISION_MOD ! For GEOS-Chem Precision (fp, f4, f8)
```

```
IMPLICIT NONE
PRIVATE
```

### PUBLIC MEMBER FUNCTIONS:

```
PUBLIC :: CHEMPOPS
PUBLIC :: INIT_POPS
```

### REMARKS:

POPs Tracers

=====

- (1 ) POPG : Gaseous POP - total tracer
- (2 ) POPPOC : OC-sorbed POP - total tracer
- (3 ) POPPBC : BC-sorbed POP - total tracer

### REVISION HISTORY:

20 Sep 2010 - N.E. Selin - Initial Version  
 04 Jan 2011 - C.L. Friedman - Expansion on initial version  
 21 Aug 2014 - M. Sulprizio - Removed emissions routines now handled by HEMCO  
 04 Mar 2015 - R. Yantosca - Use REAL(f4) for pointer args to HCO\_GetPtr  
 23 Dec 2015 - E. Lundgren - Use global physical constants  
 22 Jun 2016 - R. Yantosca - Rename species ID flags from IDT\* to id\_\*  
 22 Jun 2016 - R. Yantosca - Rename species drydep flags from DRY\* to dd\_\*  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

---

#### 10.8.1 chempops

Subroutine CHEMPOPS is the driver routine for POPs chemistry (eck, 9/20/10)

### INTERFACE:

```
SUBROUTINE CHEMPOPS(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)
```

### USES:

```

USE CMN_SIZE_MOD
USE DRYDEP_MOD, ONLY : DEPSAV
USE ErrCode_Mod
USE ERROR_MOD, ONLY : DEBUG_MSG
USE ERROR_MOD, ONLY : ERROR_STOP
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

```

20 September 2010 - N.E. Selin - Initial Version based on CHEMMERCURY
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
23 Oct 2013 - R. Yantosca - Now pass objects to GET_GLOBAL_OH routine
10 Sep 2014 - M. Sulprizio- Remove calls to GET_GLOBAL_OH and GET_GLOBAL_O3,
 these fields are now read in HEMCO
10 Sep 2014 - M. Sulprizio- Now get global concentrations of OH, O3, OC, and
 BC from HEMCO
03 Mar 2015 - C. Keller - Now do dry-deposition always outside of chempops
03 Nov 2017 - R. Yantosca - Now accept State_Diag as an argument

```

**10.8.2 chem\_pogpg**

Subroutine CHEM\_POPGP is the chemistry subroutine for the oxidation, gas-particle partitioning, and deposition of POPs. (eck, clf, 1/4/2011)

**INTERFACE:**

```

SUBROUTINE CHEM_POPGP (V_DEP_G,
& V_DEP_P_OCPO, V_DEP_P_OCPI,
& V_DEP_P_BCPO, V_DEP_P_BCPI,
& am_I_Root, Input_Opt,
& State_Met, State_Chm,
7 RC)

```

**USES:**

```

 USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
 USE DIAG_MOD, ONLY : AD44
 USE DIAG53_MOD
#endif
 USE ErrCode_Mod
 USE ERROR_MOD, ONLY : DEBUG_MSG
 USE ERROR_MOD, ONLY : SAFE_DIV
 USE GC_GRID_MOD, ONLY : GET_AREA_CM2
 USE Input_Opt_Mod, ONLY : OptInput
 USE PBL_MIX_MOD, ONLY : GET_FRAC_UNDER_PBLTOP
 USE State_Chm_Mod, ONLY : ChmState
 USE State_Met_Mod, ONLY : MetState
 USE TIME_MOD, ONLY : GET_TS_CHEM

```

**INPUT PARAMETERS:**

```

 ! Dry deposition frequencies [/s]
 REAL(fp), INTENT(IN) :: V_DEP_G(IIPAR,JJPAR)
 REAL(fp), INTENT(IN) :: V_DEP_P_OCPO(IIPAR,JJPAR)
 REAL(fp), INTENT(IN) :: V_DEP_P_BCPO(IIPAR,JJPAR)
 REAL(fp), INTENT(IN) :: V_DEP_P_OCPI(IIPAR,JJPAR)
 REAL(fp), INTENT(IN) :: V_DEP_P_BCPI(IIPAR,JJPAR)

 LOGICAL, INTENT(IN) :: am_I_Root ! Is this the root CPU?
 TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
 TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

 TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object

```

**OUTPUT PARAMETERS:**

```

 INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

## References:

```

=====
(1) For OH rate constant: Brubaker & Hites. 1998. OH reaction kinetics of
PAHs and PCDD/Fs. J. Phys. Chem. A. 102:915-921.

```

**REVISION HISTORY:**

```

20 Sep 2010 - N.E. Selin - Initial Version based on CHEM_HGO_HG2
29 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
14 Apr 2014 - R. Yantosca - Now prevent div-by-zero in computations below

```



```
REAL(fp), INTENT(IN) :: OLD_POPG
REAL(fp), INTENT(IN) :: DT
REAL(fp), INTENT(IN) :: K_OX
REAL(fp), INTENT(IN) :: K_DEPG
REAL(fp), INTENT(IN) :: E_KOX_T
```

**OUTPUT PARAMETERS:**

```
REAL(fp), INTENT(OUT) :: NEW_POPG
REAL(fp), INTENT(OUT) :: GROSS_OX
REAL(fp), INTENT(OUT) :: DEP_POPG
```

**REVISION HISTORY:**

27 January 2011 - CL Friedman - Initial Version

---

**10.8.5 no\_rxn\_withdep**

Subroutine NO\_RXN\_WITHDEP calculates new mass of POPP for given rate of deposition. No oxidation of POPP. This is for the boundary layer. (clf, 2/9/11)

**INTERFACE:**

```
SUBROUTINE NO_RXN_WITHDEP(OLD_POPP, K_DEPP, DT,
& NEW_POPP, DEP_POPP)
```

**USES:**

```
USE ERROR_MOD, ONLY : ERROR_STOP
```

**INPUT PARAMETERS:**

```
REAL(fp), INTENT(IN) :: OLD_POPP
REAL(fp), INTENT(IN) :: K_DEPP
REAL(fp), INTENT(IN) :: DT
```

**OUTPUT PARAMETERS:**

```
REAL(fp), INTENT(OUT) :: NEW_POPP
REAL(fp), INTENT(OUT) :: DEP_POPP
```

**REVISION HISTORY:**

9 February 2011 - CL Friedman - Initial Version

---

**10.8.6 get\_oh**

Function GET\_OH returns monthly mean OH and imposes a diurnal variation.

**INTERFACE:**

```
FUNCTION GET_OH(I, J, L, State_Met) RESULT(OH_MOLEC_CM3)
```

**USES:**

```
USE CMN_SIZE_MOD ! Size parameters
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: I, J, L

TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

**REMARKS:**

Copied GET\_OH function from mercury\_mod.f - CLF

**REVISION HISTORY:**

```
03 Feb 2011 - CL Friedman - Initial Version
29 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
23 Sep 2014 - M. Sulprizio- Now get OH for offline aerosol sim from HEMCO
```

**10.8.7 ohno3time**

Subroutine OHNO3TIME computes the sum of cosine of the solar zenith angle over a 24 hour day, as well as the total length of daylight. This is needed to scale the offline OH and NO3 concentrations. (rjp, bmy, 12/16/02, 12/8/04)

**INTERFACE:**

```
SUBROUTINE OHNO3TIME
```

**USES:**

```
USE CMN_SIZE_MOD ! Size parameters
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID_R
USE TIME_MOD, ONLY : GET_NHMSb, GET_ELAPSED_SEC
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_DAY_OF_YEAR, GET_GMT
```

**REVISION HISTORY:**

```
20 Sep 2010 - N.E. Selin - Initial Version for POPS_MOD
16 May 2016 - M. Sulprizio- Remove IJLOOP and change SUNTMP array dimensions
 from (MAXIJ) to (IIPAR,JJPAR)
```



### 10.8.8 init\_pops

Subroutine INIT\_POPS allocates and zeroes all module arrays.

#### INTERFACE:

```
SUBROUTINE INIT_POPS(am_I_Root, Input_Opt, State_Chm, RC)
```

#### USES:

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE ERROR_MOD, ONLY : ALLOC_ERR
USE ERROR_MOD, ONLY : DEBUG_MSG
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
```

#### INPUT PARAMETERS:

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
```

#### OUTPUT PARAMETERS:

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

#### REVISION HISTORY:

```
20 Sep 2010 - N.E. Selin - Initial Version based on INIT_MERCURY
25 Mar 2013 - R. Yantosca - Now accept Input_Opt argument
25 Aug 2014 - M. Sulprizio- Now accept am_I_Root and RC arguments
10 Sep 2014 - M. Sulprizio- Remove calls to GET_GLOBAL_OC and GET_GLOBAL_BC,
 these fields are now read in HEMCO
10 Mar 2015 - R. Yantosca - Now always allocate ZERO_DVEL
21 Sep 2015 - R. Yantosca - Now takes State_Chm as an argument, so that
 we can take advantage of the species database
02 May 2016 - R. Yantosca - Now make id_POP* flags as module variables
 instead of getting them from tracerid_mod.F
22 Jun 2016 - R. Yantosca - Now use Ind_() to define species & drydep ID's
22 Jun 2016 - R. Yantosca - Rename IDT* flags to id* and DRY* to dd_*
```

### 10.8.9 cleanup\_pops

Subroutine CLEANUP\_POPS deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_POPS
```

#### REVISION HISTORY:

```
20 September 2010 - N.E. Selin - Initial Version
```

**10.9 Fortran: Module Interface RnPbBe\_mod.F**

Module RnPbBe\_MOD contains variables and routines used for the 222Rn-210Pb-7Be simulation. (hyl, swu, bmy, 6/14/01, 8/4/06)

**INTERFACE:**

```
MODULE RnPbBe_MOD
```

**USES:**

```
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
```

```
PRIVATE
```

**PUBLIC MEMBER FUNCTIONS:**

```
PUBLIC :: CHEMRnPbBe
```

**REMARKS:**

References:

```
=====
```

- (1 ) Liu, H., D. Jacob, I. Bey, and R. M. Yantosca, Constraints from 210Pb and 7Be on wet deposition and transport in a global three-dimensional chemical tracer model driven by assimilated meteorological fields, JGR, 106, D11, 12,109-12,128, 2001.
- (2 ) Jacob et al., Evaluation and intercomparison of global atmospheric transport models using Rn-222 and other short-lived tracers, JGR, 1997 (102):5953-5970
- (3 ) Dorothy Koch, JGR 101, D13, 18651, 1996.
- (4 ) Lal, D., and B. Peters, Cosmic ray produced radioactivity on the Earth. Handbuch der Physik, 46/2, 551-612, edited by K. Sitte, Springer-Verlag, New York, 1967.

**REVISION HISTORY:**

- 14 Jun 2001 - H. Liu - Initial version
- (1 ) Added existing routines to this module (bmy, 6/14/01)
  - (2 ) Updated comments (bmy, 9/4/01)
  - (3 ) Eliminate AVGF; redimensioned XTRA2 (bmy, 9/25/01)
  - (4 ) Replace references to PW(I,J) with P(I,J) (bmy, 10/3/01)
  - (5 ) Remove obsolete code from 9/01 and 10/01 (bmy, 10/23/01)
  - (6 ) Removed duplicate variable declarations (bmy, 11/15/01)
  - (7 ) Now read files from DATA\_DIR/RnPbBe\_200203/ directory. Also updated comments. (bmy, 3/29/02)
  - (8 ) Incorporated latest changes from Hongyu Liu. Also split off the code to read in the 7Be emissions into a separate routine. Add parallel DO-loops in several places. Cleaned up DRYFLXRnPbBe, and now make sure ND44 accurately represents the drydep fluxes of 210Pb and 7Be. (hyl, bmy, 8/7/02)

(9 ) Now reference AD from "dao\_mod.f". Now references "error\_mod.f".  
 Moved routine DRYFLXRnPbBe into "drydep\_mod.f". (bmy, 1/27/03)

(10) Now references the new "time\_mod.f" (bmy, 2/11/03)

(11) Bug fix in EMISSRnPbBe -- take abs( lat) for 7Be emiss. (bmy, 6/10/03)

(12) Bug fix in EMISSRnPbBe -- shut off 222Rn emissions in polar regions  
 (swu, bmy, 10/28/03)

(13) Now references "directory\_mod.f", "logical\_mod.f", and "tracer\_mod.f"  
 (bmy, 7/20/04)

(14) Now modified for GCAP and GEOS-5 met fields (swu, bmy, 5/24/05)

(15) Now references "tropopause\_mod.f"

(16) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)

19 Nov 2010 - R. Yantosca - Added ProTeX headers

08 Nov 2011 - R. Yantosca - Prevent out-of-bounds errors in diagnostics

28 Feb 2012 - R. Yantosca - Removed support for GEOS-3

01 Mar 2012 - R. Yantosca - Now use routines from the new grid\_mod.F90

01 Aug 2012 - R. Yantosca - Add reference to findFreeLUN from inquire\_mod.F90

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

07 Jul 2014 - R. Yantosca - Removed routines now orphaned by HEMCO

22 Aug 2014 - R. Yantosca - Removed LATSOU, PRESOU, BESOU arrays, these  
 are now defined in the HEMCO code.

22 Aug 2014 - R. Yantosca - Remove XNUMOL\_\* parameters; these are obsolete

04 Nov 2014 - M. Yannetti - Added PRECISION\_MOD

30 Jan 2015 - E. Lundgren - Add new diagnostics structures for netCDF output.

02 May 2016 - R. Yantosca - Now define IDTRn, IDTPb, IDTBe locally

### 10.9.1 chemRnPbBe

Subroutine CHEMRnPbBe performs loss chemistry on 222Rn, 210Pb, and 7Be.

#### INTERFACE:

```

SUBROUTINE CHEMRnPbBe(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : AD01
USE DIAG_MOD, ONLY : AD02
#endif
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState

```

```
USE TIME_MOD, ONLY : GET_TS_CHEM
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(Optional), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object
```

**INPUT/OUTPUT PARAMETERS:**

```
TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

```
31 Oct 1999 - H. Liu - Initial version
(1) Now use F90 syntax (bmy, hyl, 3/22/99)
(2) Add FIRSTCHEM as an argument. Only compute the exponential terms
 when FIRSTCHEM = .TRUE., and save the values for later use
 (bmy, 3/24/99)
(3) Cosmetic changes (bmy, 10/13/99)
(4) Eliminate obsolete code and ND63 diagnostic (bmy, 4/12/00)
(5) Cosmetic changes (bmy, 7/12/00)
(6) Added to module "RnPbBe_mod.f". Also updated comments
 and made cosmetic changes. (bmy, 6/14/01)
(7) Add diagnostics for Rn/Be emissions. Also cleaned up some old code
 and added parallel DO-loops. Updated comments. (hyl, 8/6/02)
(8) Now make FIRSTCHEM a local SAVED variable. (bmy, 1/27/03)
(9) Now use function GET_TS_CHEM from "time_mod.f" (bmy, 2/11/03)
(10) Now references STT and N_TRACERS from "tracer_mod.f" (bmy, 7/20/04)
(11) Remove reference to CMN; it's obsolete. Now use inquiry functions
 from "tropopause_mod.f" to diagnose strat boxes. (bmy, 8/15/05)
08 Dec 2009 - R. Yantosca - Added ProTeX headers
08 Nov 2011 - R. Yantosca - Prevent out-of-bounds errors in diagnostics
09 Nov 2012 - M. Payer - Replaced all met field arrays with State_Met
 derived type object
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, State_Chm, RC
22 Aug 2014 - R. Yantosca - Copy emissions above the PBL to tracer array
22 Aug 2014 - R. Yantosca - Cosmetic changes, for clarity
04 Sep 2014 - R. Yantosca - Add minor changes for efficiency
04 Nov 2014 - M. Yannetti - Changed REAL*8 to REAL(fp)
12 Dec 2014 - M. Yannetti - Changed Hemco REAL*8 to REAL(hp)
 2 May 2016 - R. Yantosca - Now define IDTRn, IDTPb, IDTBe locally
22 Jun 2016 - R. Yantosca - Now use Ind_() to define species ID's
22 Jun 2016 - R. Yantosca - Rename species ID's to id_Rn, id_Pb, id_Be7
30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected
```

species ID from State\_Chm%Map\_Advect.

10 Aug 2016 - R. Yantosca - Remove temporary tracer-removal code  
 03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

---

## 10.10 Fortran: Module Interface rrtmg\_rad\_transfer\_mod.F

Module RRTMG\_RAD\_TRANSFER\_MOD contains arrays and routines for performing on-line radiative transfer in GEOS-Chem.

### INTERFACE:

```
MODULE RRTMG_RAD_TRANSFER_MOD
```

### USES:

```
USE CMN_FJX_MOD, ONLY : RTODAER, RTSSAER, RTASYMAER,
& WVAA, SPECMASK, LSPECRADMENU,
& NSPECRADMENU
USE CMN_SIZE_MOD, ONLY : IIPAR, JJPARG, LLPARG, NDUST, NAER
#if defined(BPCH_DIAG)
USE DIAG_MOD, ONLY : AD72 !RAD OUTPUT DIAGNOSTIC ARRAY
#endif
USE OMP_LIB
USE PARRRTM, ONLY : NBNDLW
USE PARRRSW, ONLY : NBNSW
```

```
IMPLICIT NONE
PRIVATE
```

```
!PUBLIC MEMBER FUNCTIONS
```

```
PUBLIC :: CLEANUP_SURFACE_RAD
PUBLIC :: INIT_SURFACE_RAD
PUBLIC :: READ_SURFACE_RAD
PUBLIC :: CLEANUP_MCICA_CLOUDS
PUBLIC :: INIT_MCICA_CLOUDS
```

### PUBLIC DATA MEMBERS:

```
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: ALBDIR(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: ALBDIF(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: EMISS (:,:,:)

REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CH4CLIM(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: N2OCLIM(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CFC11CLIM(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CFC12CLIM(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CCL4CLIM(:,:,:)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CFC22CLIM(:,:,:)

```

```

!MCICA cloud variables now stored for reuse
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CLDFMCL_LW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CIWPMCL_LW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CLWPMCL_LW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: TAUCMCL_LW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CLDFMCL_SW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CIWPMCL_SW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: CLWPMCL_SW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: TAUCMCL_SW(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: SSACMCL(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: ASMCML(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: FSFCMCL(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: REICMCL(:, :, :, :)
REAL*8, ALLOCATABLE, PUBLIC, TARGET :: RELQMCL(:, :, :, :)

```

## REVISION HISTORY:

```

18 Jun 2013 - D.A. Ridley - Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
17 May 2016 - M. Sulprizio- Add extra dimension to MCICA cloud arrays to
 allow for change from IIPAR*JJPAR to IIPAR,JJPAR
27 Jun 2016 - R. Yantosca - Now save species ID's in module variables
29 Nov 2016 - R. Yantosca - grid_mod.F90 is now gc_grid_mod.F90
18 May 2018 - C. Holmes - Unit fix for gas concentrations

```

### 10.10.1 do\_rrtmg\_rad\_transfer

INTERNAL SUBROUTINE

#### INTERFACE:

```

SUBROUTINE DO_RRTMG_RAD_TRANSFER(am_I_Root, THISDAY, THISMONTH,
& ICLD, ISPECMENU, ISEED,
& Input_Opt, State_Met, State_Chm,
& RC)

```

#### USES:

```

!-----
! Modules from GeosRad
!-----
USE MCICA_SUBCOL_GEN_LW, ONLY : MCICA_SUBCOL_LW
USE MCICA_SUBCOL_GEN_SW, ONLY : MCICA_SUBCOL_SW
USE PARKIND, ONLY : IM=>KIND_IM, RB=>KIND_RB
USE RRLW_CON, ONLY : GASCON, AVOGAD
USE PARRRTM, ONLY : NBNDLW, NGPTLW
USE PARRRSW, ONLY : NBNDLW, NGPTLW, NAEREC
USE RRTMG_LW_RAD, ONLY : RRTMG_LW

```

```

USE RRTMG_SW_RAD, ONLY : RRTMG_SW

!-----
! GEOS-Chem modules
!-----
USE CMN_FJX_MOD, ONLY : NSPECRAD ! NUMBER OF SPECIES FOR RT
USE CMN_FJX_MOD, ONLY : NASPECRAD ! NUMBER OF AEROSOL SPECIES
USE CMN_FJX_MOD, ONLY : SPECMASK, IRTWVSELECT
USE CMN_FJX_MOD, ONLY : ACOEF_RTWV, BCOEF_RTWV, CCOEF_RTWV
USE CMN_FJX_MOD, ONLY : WVAA, NWVAA
USE CMN_FJX_MOD, ONLY : NWVAAO
USE ErrCode_Mod
USE ERROR_MOD
USE GC_GRID_MOD, ONLY : GET_YMID
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AIRMW, PI, AVO
USE PRESSURE_MOD, ONLY : GET_PCENTER, GET_PEDGE
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_DAY_OF_YEAR, GET_HOUR
USE TOMS_MOD, ONLY : GET_OVERHEAD_03
USE UnitConv_Mod, ONLY : Convert_Spc_Units
#if defined(BPCH_DIAG)
USE DIAG_MOD, ONLY : AD72
#endif

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! root CPU?
INTEGER, INTENT(IN) :: THISDAY ! CURRENT DAY
INTEGER, INTENT(IN) :: THISMONTH ! CURRENT MONTH
INTEGER, INTENT(IN) :: ISPECMENU ! THE SPECIES BEING INCLUDED
 ! NEEDED FOR OUTPUT PURPOSES

INTEGER, INTENT(IN) :: ISEED
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
INTEGER, INTENT(INOUT) :: ICLD ! CLOUD FLAG FOR RRTMG
 ! 0-NOCLOUD, 1-GREY CLOUD

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Failure or success

```

**REMARKS:**

THIS ROUTINE PASSES INPUTS TO THE RRTMG DRIVER ROUTINE "RAD\_DRIVER"  
VIA THE ARGUMENT LIST. THIS PREVENTS CIRCULAR REFERENCES.

**REVISION HISTORY:**

- 17 AUG 2012 - R. YANTOSCA - INITIAL VERSION
- 15 Dec 2014 - M. Sulprizio- Moved radiation diagnostic from ND71 to ND72 to avoid conflicts with hourly max ppbv diagnostic.
- 15 Jan 2015 - M. Sulprizio- Added T\_CTM and P\_CTM arrays and calculate them in the same manner that we do in fast\_jx\_mod.F. Also moved and fixed calculation of O3\_CTM.
- 15 Jan 2015 - M. Sulprizio- Added ProTeX headers
- 13 Apr 2015 - R. Yantosca - Fixed inefficient loop ordering (should be J,I)
- 13 Apr 2015 - R. Yantosca - Add error check for JLOOP > 0 in a few places
- 23 Apr 2015 - R. Yantosca - Bug fix: YLAT was undefined; now corrected
- 23 Apr 2015 - R. Yantosca - Bug fix: all members of State\_Met now use (I,J,L) ordering. This is for the NASA GCM.
- 02 Jun 2015 - R. Yantosca - Bug fix: Add missing variables to OMP PRIVATE
- 03 Jun 2015 - R. Yantosca - Now use pointers to avoid array temporaries
- 12 Aug 2015 - E. Lundgren - Incoming tracer units are now [kg/kg] and are converted to [kg] for RRTMG
- 12 Aug 2015 - E. Lundgren - Now accept am\_I\_Root and RC as arguments
- 21 Dec 2015 - M. Sulprizio- Get air density directly from State\_Met object
- 22 Dec 2015 - M. Sulprizio- Replace CSPEC with State\_Chm%Species
- 19 Jan 2016 - E. Lundgren - Use global physical constants from physconstants rather than comode\_loop\_mod
- 17 May 2016 - M. Sulprizio- Remove NCOL, IJLOOP and change dimensions of arrays from IIPAR\*JJPAR to IIPAR,JJPAR
- 31 May 2016 - E. Lundgren - Replace Input\_Opt%TRACER\_MW\_G with emMW\_g from species database (emitted species g/mol)
- 16 Jun 2016 - K. Yu - Now define species ID's with the Ind\_ function
- 17 Jun 2016 - R. Yantosca - Only define species ID's on the first call
- 27 Jun 2016 - R. Yantosca - Bug fix: replace a couple of leftover IDO3 variables with the new id\_O3 species ID
- 30 Jun 2016 - M. Sulprizio- Replace ICH4 with id\_CH4 to remove dependence on comode\_loop\_mod.F
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.
- 28 Sep 2017 - E. Lundgren - Simplify unit conversions using wrapper routine
- 03 Jan 2018 - M. Sulprizio- Replace UCX CPP switch with Input\_Opt%LUCX

**10.10.2 set\_specmask**

Subroutine SET\_SPECMASK converts the species switches in the input.mod radiation section into the list of species that should be passed through to RRTMG. This must be done in a subtractive way, e.g. If we require the DRE of sulfate then the baseline will contain all species and the sulfate run will contain everything but sulfate, this way the contribution of sulfate can be inferred. Therefore, all species are initially set to 1 and their inclusion results in SPECMASK for the particular species being set to zero. (dar 10/2013)



**INTERFACE:**

```
SUBROUTINE SET_SPECMASK(ISPECRADMENU)
```

**USES:**

```
USE CMN_FJX_MOD, ONLY : SPECMASK, NSPECRAD, NASPECRAD,
& LSPECRADMENU
```

**INPUT PARAMETERS:**

```
INTEGER, INTENT(IN) :: ISPECRADMENU
```

**REVISION HISTORY:**

```
18 Jun 2013 - D.A. Ridley - Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
26 Jun 2015 - E. Lundgren - Fix typo in NXTRA definition
```

**10.10.3 read\_surface\_rad**

Subroutine READ\_SURFACE\_RAD gets the surface albedo and emissivity from data files processed from MODIS MCD43C3.5 and MOD11C2. Albedo is direct (black sky) and diffuse (white sky) and interpolated to the first 21 wavelengths of GADS, used in the RRTMG code. Emissivity has been interpolated to last 40 wavelengths of GADS! This routine is called from main.F when ITS\_TIME\_FOR\_SURFACE\_RAD() is true (every 8 days) DAR (10/2012)

**INTERFACE:**

```
SUBROUTINE READ_SURFACE_RAD(Input_Opt, FORCEREAD)
```

**USES:**

```
USE BPCH2_MOD, ONLY : GET_NAME_EXT_2D, GET_RES_EXT
USE BPCH2_MOD, ONLY : GET_TAUO, READ_BPCH2
USE CMN_FJX_MOD
USE CMN_SIZE_MOD ! SIZE PARAMETERS
USE Input_Opt_Mod, ONLY : OptInput
USE JULDAY_MOD, ONLY : JULDAY, CALDATE
USE TIME_MOD, ONLY : GET_YEAR, GET_MONTH
USE TIME_MOD, ONLY : GET_DAY, GET_DAY_OF_YEAR
USE TIME_MOD, ONLY : YMD_EXTRACT
```

**INPUT PARAMETERS:**

```
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
LOGICAL, INTENT(IN), OPTIONAL :: FORCEREAD ! Reset first-time flag?
```

**REMARKS:**

```
#####
NOTE: BINARY PUNCH INPUT IS BEING PHASED OUT. THIS DATA
WILL EVENTUALLY BE READ IN FROM netCDF FILES VIA HEMCO!
-- Bob Yantosca (05 Mar 2015)
#####
```

**REVISION HISTORY:**

```
18 Jun 2013 - D.A. Ridley - Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
10 Apr 2015 - R. Yantosca - Read from ExtData/CHEM_INPUTS/modis_surf_201210
```

**10.10.4 read\_strat\_clim**

Subroutine READ\_STRAT\_CLIM gets the stored N2O and CH4 profiles created based on TES climatology. These are global July mean profiles from TES that are interpolated to the 3D GEOS-Chem grid as a simple solution to account for stratospheric contribution of these species to radiative balance. This routine is called from main.F once at the start of the model run. DAR (12/2012)

**INTERFACE:**

```
SUBROUTINE READ_STRAT_CLIM(Input_Opt)
```

**USES:**

```
USE BPCH2_MOD, ONLY : GET_NAME_EXT_2D, GET_RES_EXT
USE BPCH2_MOD, ONLY : GET_TAUO, READ_BPCH2
USE CMN_FJX_MOD
USE CMN_SIZE_MOD ! SIZE PARAMETERS
USE Input_Opt_Mod, ONLY : OptInput
USE TIME_MOD, ONLY : GET_YEAR, GET_MONTH
USE TIME_MOD, ONLY : GET_DAY, GET_DAY_OF_YEAR
USE TRANSFER_MOD, ONLY : TRANSFER_3D
```

**INPUT PARAMETERS:**

```
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input options
```

**REMARKS:**

```
#####
NOTE: BINARY PUNCH INPUT IS BEING PHASED OUT. THIS DATA
WILL EVENTUALLY BE READ IN FROM netCDF FILES VIA HEMCO!
-- Bob Yantosca (10 Apr 2015)
#####
```

**REVISION HISTORY:**

```
18 Jun 2013 - D.A. Ridley - Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
10 Apr 2015 - R. Yantosca - Now read from ExtData/CHEM_INPUTS/RRTMG_201104/
```

### 10.10.5 init\_surface\_rad

Subroutine INIT\_SURFACE\_RAD initializes all allocatable module arrays.

#### INTERFACE:

```
SUBROUTINE INIT_SURFACE_RAD()
```

#### USES:

```
USE CMN_FJX_MOD
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ALLOC_ERR
```

#### REVISION HISTORY:

```
18 Jun 2013 - D.A. Ridley - Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
```

---

### 10.10.6 init\_strat\_clim

Subroutine INIT\_STRAT\_CLIM initializes all allocatable module arrays.

#### INTERFACE:

```
SUBROUTINE INIT_STRAT_CLIM
```

#### USES:

```
USE CMN_FJX_MOD
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ALLOC_ERR
```

#### REVISION HISTORY:

```
18 Jun 2013 - D.A. Ridley- Initial version
15 Jan 2015 - M. Sulprizio- Added ProTeX headers
```

---

### 10.10.7 init\_mcica\_clouds

Subroutine INIT\_MCICA\_CLOUDS initializes all allocatable module arrays.

#### INTERFACE:

```
SUBROUTINE INIT_MCICA_CLOUDS()
```

#### USES:

```
USE CMN_FJX_MOD
USE CMN_SIZE_MOD
USE ERROR_MOD, ONLY : ALLOC_ERR
USE PARRRTM, ONLY : NGPTLW
USE PARRRSW, ONLY : NGPTSW
```

#### REVISION HISTORY:

18 Jun 2013 - D.A. Ridley - Initial version  
15 Jan 2015 - M. Sulprizio- Added ProTeX headers  
17 May 2016 - M. Sulprizio- Change array dimensions from IIPAR\*JJPAR to  
IIPAR, JJPAR

---

#### 10.10.8 cleanup\_surface\_rad

Subroutine CLEANUP\_SURFACE\_RAD deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_SURFACE_RAD
```

#### REVISION HISTORY:

18 Jun 2013 - D.A. Ridley - Initial version  
15 Jan 2015 - M. Sulprizio- Added ProTeX headers

---

#### 10.10.9 cleanup\_strat\_clim

Subroutine CLEANUP\_STRAT\_CLIM deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_STRAT_CLIM
```

#### REVISION HISTORY:

18 Jun 2013 - D.A. Ridley - Initial version  
15 Jan 2015 - M. Sulprizio- Added ProTeX headers

---

#### 10.10.10 cleanup\_mcica\_clouds

Subroutine CLEANUP\_MCICA\_CLOUDS deallocates all module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_MCICA_CLOUDS
```

**REVISION HISTORY:**

18 Jun 2013 - D.A. Ridley - Initial version  
15 Jan 2015 - M. Sulprizio- Added ProTeX headers

---

**10.11 Fortran: Module Interface tagged\_co\_mod.F**

Module TAGGED\_CO\_MOD contains variables and routines used for the geographically tagged CO simulation.

**INTERFACE:**

```
MODULE TAGGED_CO_MOD
```

**USES:**

```
USE PhysConstants
USE PRECISION_MOD ! For GEOS-Chem Precision (fp)
```

```
IMPLICIT NONE
PRIVATE
```

**PRIVATE MEMBER FUNCTIONS:**

```
PRIVATE :: CALC_DIURNAL
```

**PUBLIC MEMBER FUNCTIONS:**

```
PUBLIC :: CHEM_TAGGED_CO
PUBLIC :: GET_ALPHA_ISOP
PUBLIC :: INIT_TAGGED_CO
PUBLIC :: CLEANUP_TAGGED_CO
```

**REMARKS:**

Tagged CO Species (you can modify these as needs be!)

```
=====
```

- (1 ) Total CO
- (2 ) CO from North American fossil fuel
- (3 ) CO from European fossil fuel
- (4 ) CO from Asian fossil fuel
- (5 ) CO from fossil fuel from everywhere else
- (6 ) CO from South American biomass burning
- (7 ) CO from African biomass burning
- (8 ) CO from Southeast Asian biomass burning
- (9 ) CO from Oceania biomass burning
- (10) CO from European biomass burning
- (11) CO from North American biomass burning
- (12) CO chemically produced from Methane
- (13) CO from Biofuel burning (whole world)

- (14) CO chemically produced from Isoprene
- (15) CO chemically produced from Monoterpenes
- (16) CO chemically produced from Methanol (CH3OH)
- (17) CO chemically produced from Acetone

## REVISION HISTORY:

- 28 Jul 2000- R. Yantosca - Initial version
- (1 ) Removed obsolete code from CHEM\_TAGGED\_CO (bmy, 12/21/00)
  - (2 ) Added CO sources from oxidation of biofuel VOC's, biomass burning VOC's, fossil fuel VOC's, and natural VOC's (bnd, bmy, 1/2/01)
  - (3 ) Added chemical P(CO) from CH3OH and MONOTERPENES (bnd, bmy, 1/2/01)
  - (4 ) Now cap SCALEYEAR at 1997 in "emiss\_tagged\_co" (bnd, bmy, 4/6/01)
  - (5 ) Removed obsolete commented-out code (bmy, 4/23/01)
  - (6 ) Added new module variables SUMACETCO, EMACET, CO\_PRODS, CO\_LOSS, ISOP96, MONO96, and MEOH96. Also added new module routines GET\_ALPHA\_ISOP, READ\_PCO\_LCO\_STRAT, GET\_PCO\_LCO\_STRAT, READ\_ACETONE, and READ\_BIOG\_FOR\_GEOS3. (bnd, bmy, 6/14/01)
  - (7 ) Now read files from DATA\_DIR/tagged\_CO\_200106/ (bmy, 6/19/01)
  - (8 ) Removed ISOP96, MONO96, and CH3OH96 since we now use the new GEOS-3 fields and no longer have to correct for the surface temperature. (bmy, 8/21/01)
  - (9 ) Bug fix: don't call GLOBAL\_NOX\_MOD in routine CHEM\_TAGGED\_CO unless a logical switch is set (bmy, 8/28/01)
  - (10) Updated comments (bmy, 9/6/01)
  - (11) Deleted obsolete code for 1998 GEOS-3 fix. Also now archive ND46 diagnostic as [atoms C/cm2/s] (bmy, 9/13/01)
  - (12) Bug fix in CHEM\_TAGGED\_CO: now save CO sources/sinks into the Total CO tracer (N=1). (qli, bmy, 9/21/01)
  - (13) Resize arrays of (IIPAR,JJPARG) to (IIPAR,JJPARG) (bmy, 9/28/01)
  - (14) Removed obsolete code from 9/28/01 (bmy, 10/23/01)
  - (15) Updated comments (bmy, 2/15/02)
  - (16) Removed double definition of SUMCH3OHCO (bmy, 3/20/02)
  - (17) Now use P(I,J) + PTOP instead of PS(I,J) (bmy, 4/11/02)
  - (18) Now divide module header into MODULE PRIVATE, MODULE VARIABLES, and MODULE ROUTINES sections. Updated comments (bmy, 5/28/02)
  - (19) Now references "pressure\_mod.f" (dsa, bdf, bmy, 8/21/02)
  - (20) Now reference AD, BXHEIGHT, T and SUNCOS from "dao\_mod.f". Also removed obsolete code from various routines. Now references ERROR\_STOP from "error\_mod.f". (bmy, 10/15/02)
  - (21) Now references "grid\_mod.f" and the new "time\_mod.f". (bmy, 2/3/03)
  - (22) Bug fix for NTAU in EMISS\_TAGGED\_CO. Bug fix for FILENAME in routine READ\_PCO\_LCO\_STRAT. (ave, bnd, bmy, 6/3/03)
  - (23) Updated arg list in call to EMISOP in EMISS\_TAGGED\_CO (bmy, 12/9/03)
  - (24) Now references "directory\_mod.f", "logical\_mod.f", "tracer\_mod.f". Now remove IJLOOP\_CO. (bmy, 7/20/04)
  - (25) Fixed bug in CHEM\_TAGGED\_CO (bmy, 3/7/05)
  - (26) Now reads data from both GEOS and GCAP grids. Now also references "tropopause\_mod.f". (bmy, 8/16/05)

(27) Now modified for new "biomass\_mod.f" (bmy, 4/5/06)  
 (28) BIOMASS(:, :, IDBCO) from "biomass\_mod.f" is now in units of  
       [molec CO/cm2/s]. Adjust unit conversion accordingly. (bmy, 9/27/06)  
 (29) Routines GET\_ALPHA\_ISOP, GET\_PCO\_LCO\_STRAT, READ\_PCO\_LCO\_STRAT,  
       READ\_ACETONE and INIT\_TAGGED\_CO are public now. Variable  
       EMACET is public now. (phs, 9/18/07)

13 Aug 2010 - R. Yantosca - Add modifications for MERRA (treat like GEOS-5)  
 08 Feb 2012 - R. Yantosca - Add modifications for GEOS-5.7.2 met  
 01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90  
 23 Oct 2012 - R. Yantosca - Update prod/loss for new GMI strat chem  
 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete  
 10 Apr 2014 - R. Yantosca - Now change CO\_PRODS and CO\_LOSS to REAL\*4  
 11 Jun 2014 - J.A. Fisher - Replace SUNCOS with SUNCOSmid for MEGAN  
 11 Jun 2014 - J.A. Fisher - Treat monoterpenes as in full chemistry  
 12 Jun 2014 - J.A. Fisher - Add option for non-local PBL mixing  
 10 Jul 2014 - J.A. Fisher - Add diurnal cycle for OH  
 10 Jul 2014 - J.A. Fisher - Change FIRSTEMISS to FIRSTTIME to allow  
       initialisation when emissions are off

20 Nov 2014 - M. Yannetti - Added PRECISION\_MOD  
 11 Mar 2015 - R. Yantosca - Add OH pointer for HEMCO data input  
 06 Jan 2016 - E. Lundgren - Use global physical parameters  
 21 Mar 2016 - R. Yantosca - Removed CO\_PRODS and CO\_LOSS, we now get  
       GMI strat P(CO) and L(CO) directly from HEMCO

21 Mar 2016 - R. Yantosca - Removed GET\_PCO\_LCO\_STRAT, it was obsolete  
 21 Mar 2016 - R. Yantosca - Get rid of READ\_PCO\_LCO\_STRAT, it's obsolete  
 28 Jun 2016 - R. Yantosca - Add extra error checks; remove EMACET array\  
 29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

---

### 10.11.1 chem\_tagged\_co

Subroutine CHEM\_TAGGED\_CO performs CO chemistry on geographically "tagged" CO species. Loss is via reaction with OH.

#### INTERFACE:

```

SUBROUTINE CHEM_TAGGED_CO(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_SIZE_MOD
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD
USE DIAG_MOD, ONLY : AD65
#endif
USE ErrCode_Mod
USE ERROR_MOD, ONLY : CHECK_VALUE
USE ERROR_MOD, ONLY : ERROR_STOP

```

```

USE GC_GRID_MOD, ONLY : GET_YMID
USE HCO_EmisList_Mod, ONLY : HCO_GetPtr
USE HCO_Error_Mod
USE HCO_Interface_Mod, ONLY : HcoState, GetHcoID
USE Input_Opt_Mod, ONLY : OptInput
USE PhysConstants, ONLY : AVO
USE State_Chm_Mod, ONLY : ChmState, Ind_
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_TS_EMIS
USE TIME_MOD, ONLY : GET_MONTH, GET_YEAR
USE TIME_MOD, ONLY : ITS_A_NEW_MONTH, ITS_A_NEW_YEAR

```

**INPUT PARAMETERS:**

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

**INPUT/OUTPUT PARAMETERS:**

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REVISION HISTORY:**

- 19 Oct 1999 - Q. Li, B. Duncan, B. Field - Initial version
- (1) Now do chemistry all the way to the model top.
  - (2) Use monthly mean OH fields for oxidation.
  - (3) Now reference the monthly mean OH array and the routine which reads it from disk in "global\_oh\_mod.f" (bmy, 7/28/00)
  - (4) Removed obsolete code from 10/6/00 (bmy, 12/21/00)
  - (5) Added P(CO) from CH3OH and MONOTERPENES. Also account for the variation of CH4 conc. w/ latitude and year (bnd, bmy, 1/2/01)
  - (6) Removed obsolete commented-out code (bmy, 4/23/01)
  - (7) Updated with new stuff from bnd. Also references module routines & arrays from "global\_nox\_mod.f" and "error\_mod.f". Removed THISMONTH as an argument since we can use the MONTH value from the "CMN" include file. (bmy, 6/14/01)
  - (8) Remove GEOS-3 fix for biogenic fields (ISOP96, MONO96, CH3OH96), since we now use updated met fields w/o the surface temperature problem (bmy, 8/21/01)
  - (9) Now only call GLOBAL\_NOX\_MOD to define the BNOX array, if switch ALPHA\_ISOP\_FROM\_NOX is set. The NOx concentrations used to compute the CO yield from isoprene are currently only at 4x5 (bmy, 8/28/01)
  - (10) Bug fix: now make sure to add CO production and loss into the



- STT(:, :, :, 1), which is the Total CO tracer. (qli, bmy, 9/21/01)
- (11) Updated comments. Bug fix: multiply CO\_OH (after using it to update tagged tracers) by GCO (the initial value of STT in molec/cm3) to convert it to an amount of CO lost by OH [molec/cm3]. (bmy, 2/19/02)
  - (12) Removed PS as an argument; use P(I,J) + PTOP instead of PS, in order to ensure that we use P and AD computed from the same pressure by AIRQNT. (bmy, 4/11/02)
  - (13) Now use GET\_PCENTER from "pressure\_mod.f" to compute the pressure at the midpoint of box (I,J,L). Also deleted obsolete, commented-out code. (dsa, bdf, bmy, 8/21/02)
  - (14) Now reference AD and T from "dao\_mod.f". Now make FIRSTCHEM a local SAVED variable. (bmy, 11/15/02)
  - (15) Now replace YLMID(J) with routine GET\_YMID of "grid\_mod.f". Now uses functions GET\_TS\_CHEM, GET\_MONTH, GET\_YEAR from the new "time\_mod.f". (bmy, 2/10/03)
  - (16) Now reference STT & N\_TRACERS from "tracer\_mod.f". Now references LSPLIT from "logical\_mod.f". Now references AD65 from "diag\_pl\_mod.f". Updated comments. (bmy, 7/20/04)
  - (17) Bug fix: re-insert ELSE between (1a-1) and (1a-2); it appears to have been mistakenly deleted. (bmy, 3/7/05)
  - (18) Now references ITS\_IN\_THE\_STRAT from "tropopause\_mod.f". Now remove reference to "CMN", it's obsolete. (bmy, 8/22/05)
  - (19) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
  - (20) Remove reference to "global\_ch4\_mod.f" (bmy, 5/31/06)
  - (21) Use newest JPL 2006 rate constant for CO+OH (jaf, jmao, 3/4/09)
- 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90
- 01 Mar 2012 - R. Yantosca - Now use GET\_YMID(I,J,L) from grid\_mod.F90
- 22 Oct 2012 - R. Yantosca - Now pass am\_I\_Root=.TRUE. to GET\_GLOBAL\_CH4
- 23 Oct 2012 - R. Yantosca - Added ProTeX headers
- 09 Nov 2012 - M. Payer - Replaced all met field arrays with State\_Met derived type object
- 25 Mar 2013 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Chm, RC
- 11 Apr 2013 - R. Yantosca - Now pass Input\_Opt to GET\_GLOBAL\_NOX
- 10 Jun 2013 - R. Yantosca - Avoid array temporaries in CHECK\_VALUE
- 23 Oct 2013 - R. Yantosca - Now pass objects to GET\_GLOBAL\_OH routine
- 10 Apr 2014 - R. Yantosca - Now use pointers to fields in State\_Met
- 10 Apr 2014 - R. Yantosca - Bug fix: DTSRCE should be DTCHEM
- 10 Jul 2014 - J.A. Fisher - Add diurnal cycle for OH
- 10 Jul 2014 - J.A. Fisher - Restore assignment of DTCHEM
- 10 Jul 2014 - J.A. Fisher - Remove unused DTSRCE, PCO, and AREA\_CM2
- 10 Jul 2014 - J.A. Fisher - Moved ALPHA definitions out of code
- 27 Aug 2014 - R. Yantosca - Bug fix: pass Input\_Opt to GET\_GLOBAL\_CH4
- 21 Oct 2014 - C. Keller - Now use GetHcoVal instead of Trac\_Tend array
- 26 Feb 2015 - E. Lundgren - Replace GET\_PCENTER with State\_Met%PMID.
- 11 Mar 2015 - R. Yantosca - Remove references to global\_nox\_mod.F
- 11 Mar 2015 - R. Yantosca - Now get OH via the HEMCO data structure
- 06 Jan 2016 - E. Lundgren - Use global physical parameters
- 21 Mar 2016 - R. Yantosca - Now define FMOL\_CO from the species database

29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts  
 28 Jun 2016 - R. Yantosca - Add error traps to make sure that the ISOP, ACET, MONX non-advected species are all found  
 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.  
 11 Oct 2016 - R. Yantosca - If emissions are turned off, do not update the SUM\*CO arrays, they will remain zeroed out.  
 12 Oct 2016 - R. Yantosca - Bug fix: convert OH from kg/m3 (which HEMCO brings in from disk) to molec/cm3.  
 27 Mar 2017 - R. Yantosca - Now compute ND65 diagnostic before applying the loss to the CO tagged species  
 24 Apr 2017 - M. Sulprizio- Update CO + OH (+M) rate constant for JPL 15-10  
 03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument  
 18 Jan 2018 - M. Sulprizio- Remove call to GET\_GLOBAL\_CH4 and obtain monthly CH4 concentrations from HEMCO instead

### 10.11.2 calc\_diurnal

Subroutine CALC\_DIRUNAL computes the sume of the cosine of the solar zenith angle over a 24 hour day as well as the total length of daylight to scale the offline OH concentrations.

#### INTERFACE:

```
SUBROUTINE CALC_DIURNAL
```

#### USES:

```
USE CMN_SIZE_MOD
USE GC_GRID_MOD, ONLY : GET_YMID_R
USE TIME_MOD, ONLY : ITS_A_NEW_DAY
USE TIME_MOD, ONLY : GET_MINUTE, GET_SECOND, GET_HOUR
USE TIME_MOD, ONLY : GET_TS_CHEM, GET_DAY_OF_YEAR, GET_LOCALTIME
```

#### REVISION HISTORY:

12 Mar 2014 - J. Fisher - Copied from OHN03TIME in carbon\_mod and COSSZA in dao\_mod  
 12 Feb 2018 - E. Lundgren - Updates for new timestep units (sec not min)

### 10.11.3 get\_alpha\_isop

Function GET\_ALPHA\_ISOP returns the CO yield from Isoprene (ALPHA\_ISOP) either as a function of NOx or as a constant.

#### INTERFACE:

```
FUNCTION GET_ALPHA_ISOP(FROM_NOX, NOX) RESULT(ALPHA_ISOP)
```

**USES:**

```
USE ERROR_MOD, ONLY : ERROR_STOP
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: FROM_NOX ! If =T, will take ALPHA_ISOP
 ! as f(NOx). If =F, will set
 ! ALPHA_ISOP to a constant
REAL(fp), INTENT(IN), OPTIONAL :: NOX ! NOx concentration [ppbv]
```

**RETURN VALUE:**

```
REAL(fp) :: ALPHA_ISOP ! CO yield from ISOP
```

**REVISION HISTORY:**

```
13 Jun 2001 - B. Duncan - Initial version
01 Oct 1995 - R. Yantosca - Initial version
(1) Now make NOx an optional argument (bmy, 8/28/01)
(2) Now reference ERROR_STOP from "error_mod.f" (bmy, 10/15/02)
(3) Updated comments (bmy, 7/20/04)
23 Oct 2012 - R. Yantosca - Added ProTeX headers
```

---

**10.11.4 init\_tagged\_co**

Subroutine INIT\_TAGGED\_CO allocates memory to module arrays.

**INTERFACE:**

```
SUBROUTINE INIT_TAGGED_CO(am_I_Root, Input_Opt, State_Diag, RC)
```

**USES:**

```
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Diag_Mod, ONLY : DgnState
```

**INPUT PARAMETERS:**

```
LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostic state object
```

**OUTPUT PARAMETERS:**

```
INTEGER, INTENT(OUT) :: RC ! Success or failure?
```

**REVISION HISTORY:**

- 19 Jul 2000 - R. Yantosca - Initial version
  - (1 ) Added ISOP96, MONO96, CH3OH96 for GEOS-3 (bnd, bmy, 6/14/01)
  - (2 ) Removed ISOP96, MONO96, CH3OH96 for GEOS-3, since the new GEOS-3 fields make these no longer necessary (bmy, 8/21/09)
  - (3 ) Now allocate BB\_REGION, FF\_REGION as (IIPAR,JJPAR) (bmy, 9/28/01)
  - (4 ) Removed obsolete code from 9/28/01 (bmy, 10/22/01)
  - (5 ) Now references ALLOC\_ERR from "error\_mod.f" (bmy, 10/15/02)
  - (6 ) Now remove IJLOOP\_CO (bmy, 7/20/04)
  - (7 ) Now public. Now references ITS\_A\_H2HD\_SIM from "tracer\_mod.f".  
Allocate needed variables if H2/HD simulation (phs, 9/18/07)
  - 23 Oct 2012 - R. Yantosca - Added ProTeX headers
  - 23 Oct 2012 - J. Fisher - Dimension CO\_PRODS and CO\_LOSSSS with  
(IIPAR,JJPAR,LLPAR) for new GMI strat chem
  - 10 Apr 2014 - R. Yantosca - Now call DEFINE\* routines to define the  
fossil fuel and biofuel regions
  - 10 Apr 2014 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, RC arguments
  - 21 Mar 2016 - R. Yantosca - Remove CO\_PRODS, CO\_LOSSSS arrays
  - 28 Jun 2016 - R. Yantosca - Remove EMACET array
  - 05 Dec 2017 - R. Yantosca - Now accept State\_Diag as an argument
- 

### 10.11.5 cleanup\_tagged\_co

Subroutine CLEANUP\_TAGGED\_CO deallocates memory from previously allocated module arrays.

#### INTERFACE:

```
SUBROUTINE CLEANUP_TAGGED_CO
```

#### REVISION HISTORY:

- 19 Jul 2000 - R. Yantosca - Initial version
  - (1 ) Added ISOP96, MONO96, CH3OH96 for GEOS-3 (bnd, bmy, 6/14/01)
  - (2 ) Removed ISOP96, MONO96, CH3OH96 for GEOS-3, since the new GEOS-3 fields make these no longer necessary (bmy, 8/21/09)
  - (3 ) Now remove IJLOOP\_CO (bmy, 7/20/04)
  - 23 Oct 2012 - R. Yantosca - Added ProTeX headers
  - 11 Mar 2015 - R. Yantosca - Now free the OH pointer
  - 21 Mar 2016 - R. Yantosca - Remove CO\_PRODS, CO\_LOSSSS
  - 21 Mar 2016 - R. Yantosca - Also nullify GMI\_PROD\_CO and GMI\_LOSS\_CO ptrs
  - 28 Jun 2016 - R. Yantosca - Remove EMACET array
- 

### 10.12 Fortran: Module Interface tagged\_o3\_mod.F

Module TAGGED\_O3\_MOD contains variables and routines to perform a tagged O3 simulation. P(O3) and L(O3) rates need to be archived from a full chemistry simulation before

you can run w/ Tagged O3.

**INTERFACE:**

MODULE TAGGED\_O3\_MOD

**USES:**

USE PRECISION\_MOD ! For GEOS-Chem Precision (fp, f4, f8)

IMPLICIT NONE  
PRIVATE

```

%%
%% If you want to the EXTENDED SIMULATION with all 13 tagged O3 species,
%% then uncomment this #ifdef statement. (bmy, 4/11/14)
#define USE_ALL_TAGO3_SPECIES 1
%%
%%

```

**PUBLIC MEMBER FUNCTIONS:**

PUBLIC :: CHEM\_TAGGED\_O3  
PUBLIC :: CLEANUP\_TAGGED\_O3  
PUBLIC :: INIT\_TAGGED\_O3

**PRIVATE MEMBER FUNCTIONS:**

PRIVATE :: GET\_REGIONAL\_PO3

**REMARKS:**

THE SIMPLE TAGGED O3 SIMULATION (default setting) HAS THESE ADVECTED SPECIES:

- 
- (1 ) O3 : Total O3
  - (2 ) O3Strt : Stratospheric O3

THE EXTENDED TAGGED O3 SIMULATION HAS THESE ADVECTED SPECIES:

- 
- (1 ) O3 : Total O3
  - (2 ) O3Strt : O3 from the Stratosphere (tropopause - atm top )
  - (3 ) O3Ut : O3 produced in Upper Trop (350 hPa - tropopause)
  - (4 ) O3Mt : O3 produced in Middle Trop (PBL top - 350 hPa )
  - (5 ) O3Row : O3 produced in Rest of World (surface - PBL top )
  - (6 ) O3PcBl : O3 produced in Pacific BL (surface - PBL top )
  - (7 ) O3NaBl : O3 produced in N. American BL (surface - PBL top )
  - (8 ) O3AtBl : O3 produced in Atlantic BL (surface - PBL top )
  - (9 ) O3EuBl : O3 produced in European BL (surface - PBL top )
  - (10) O3AfBl : O3 produced in N. African BL (surface - PBL top )
  - (11) O3AsBl : O3 produced in Asian (surface - PBL top )
  - (12) O3Init : O3 initial conditions (all levels )

(13) O3USA : O3 produced over the USA (all levels )

#### NOTES:

- 
- (1) The stratospheric O3 species must be species #2. This is due to how the Linoz stratospheric O3 chemistry scheme is written. We have accordingly reorganized the species numbers below.
- (2) The name "tagged\_ox\_mod.F" is historical. The Ox species in GEOS-Chem has now been replaced with O3. O3 usually makes up about 95% of Ox. The nomenclature "tagged Ox" is interchangeable with "tagged O3". As of Aug 2016, most uses of "tagged Ox" have been replaced with "tagged O3".
- (3) If using all of the tagged O3 species, note that that the sum of production of species #2 - #12 sums together to equal the production of species #1. In other words:

$$PP(I,J,L,1) \text{ will equal } SUM( PP(I,J,L,2:12) )$$

ALSO NOTE: The O3USA species is defined to include all of the O3 produced over the US (at all levels). Therefore, it should be treated separately from species 2 - 12.

- (4) When starting a long tagged O3 simulation, we recommend that you use a restart file where all species concentrations are set to zero. Then spin up for as many years as it takes to get into steady-state. This will ensure that the sum of tagged O3 species (#2 - #12) will equal the total O3 species (#1).

#### REVISION HISTORY:

- 20 Aug 2003 - A. Fiore - Initial version
- (1) Now accounts for GEOS-4 PBL being in meters (bmy, 1/15/04)
- (2) Bug fix: don't put function call in WRITE statement (bmy, 2/20/04)
- (3) Now bracket AD44 with an !\$OMP CRITICAL block (bmy, 3/24/04)
- (4) Now define regions w/ levels in GET\_REGIONAL\_POX (amf,rch,bmy,5/27/04)
- (5) Bug fix-avoid seg fault if PBLFRAC isn't allocated (bdf, bmy, 10/12/04)
- (6) Now reference "pbl\_mix\_mod.f" (bmy, 2/17/05)
- (7) Now make sure all USE statements are USE, ONLY (bmy, 10/3/05)
- (8) Now references XNUMOL from "tracer\_mod.f" (bmy, 10/25/05)
- (9) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
- (10) Modified for variable tropopause (phs, bmy, 1/19/07)
- (11) Now use LLTROP instead of LLTROP\_FIX everywhere (bmy, 12/4/07)
- (12) Now use LD65 instead of LLTROP everywhere (phs, 11/17/08)
- (13) Updates for LINOZ (dbj, jliu, bmy, 10/26/09)
- 19 Nov 2010 - R. Yantosca - Added ProTeX headers
- 28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
- 01 Mar 2012 - R. Yantosca - Now reference new grid\_mod.F90

14 Mar 2013 - M. Payer - Replace Ox with O3 as part of removal of NOx-Ox partitioning

20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete

23 Jan 2014 - M. Sulprizio- Now use LLTROP instead of LD65 or LLTROP\_FIX everywhere (hyl, bmy, 11/3/11)

11 Apr 2014 - R. Yantosca - Add #ifdef USE\_ALL\_TAGO3\_SPECIES to select between the simple and extended simulations

11 Apr 2014 - R. Yantosca - Updated comments

11 Apr 2014 - R. Yantosca - Now make the O3Strt tracer #2 for both the simple and extended tagged O3 simulations

11 Apr 2014 - R. Yantosca - Now make INIT\_TAGGED\_OX a public routine

17 Sep 2014 - C. Keller - Now use HEMCO for prod./loss rates. This makes subroutine READ\_POX\_LOX obsolete.

06 Nov 2014 - R. Yantosca - Removed code orphaned by HEMCO

21 Nov 2014 - M. Yannetti - Added PRECISION\_MOD

04 Mar 2015 - R. Yantosca - Declare pointer args to HCO\_GetPtr with REAL(f4)

16 Jun 2016 - M. Sulprizio- Replace IDTO3Strt from tracerid\_mod.F with a local definition (module variable id\_O3Strat)

12 Jul 2016 - R. Yantosca - Remove routine ADD\_STRAT\_POX, we now just directly add into State\_Chm%Species in strat\_chm\_mod.F90

10 Aug 2016 - M. Sulprizio- Rename from tagged\_ox\_mod.F to tagged\_o3\_mod.F

29 Nov 2016 - R. Yantosca - grid\_mod.F90 is now gc\_grid\_mod.F90

### 10.12.1 get\_regional\_po3

Subroutine GET\_REGIONAL\_PO3 returns the P(O3) for each of the tagged O3 species. Tagged O3 species are defined by both geographic location and altitude.

#### INTERFACE:

```
SUBROUTINE GET_REGIONAL_PO3(I, J, L, PP, State_Met)
```

#### USES:

```
USE CMN_SIZE_MOD ! Size parameters
USE GC_GRID_MOD, ONLY : GET_XMID, GET_YMID
USE PhysConstants ! SCALE_HEIGHT
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
#if defined(BPCH_DIAG)
USE CMN_DIAG_MOD ! ND44, ND65, LD65
#endif
```

#### INPUT PARAMETERS:

```
! GEOS-Chem grid box indices for lon, lat, alt
INTEGER, INTENT(IN) :: I, J, L
```

```
! Meteorology State object
TYPE(MetState), INTENT(IN) :: State_Met
```

### OUTPUT PARAMETERS:

```
! Array containing P(O3) for each tagged species
REAL(fp), INTENT(OUT) :: PP(IIPAR,JJPARG,LLTROP,N_TAGGED)
```

### REVISION HISTORY:

```
19 Aug 2003 - A. Fiore - Initial version
(1) Updated from the old routine "chemo3_split.f" (rch, bmy, 8/20/03)
(2) For GEOS-4, convert PBL from [m] to [hPa] w/ the hydrostatic law.
 Now references SCALE_HEIGHT from "CMN_GCTM". (bmy, 1/15/04)
(3) Now uses model levels instead of pressure in order to delineate
 between PBL, MT, and UT regions (amf, rch, bmy, 5/27/04)
(4) Now references ITS_IN_THE_TROP from "tropopause_mod.f". Now remove
 reference to "CMN", it's obsolete. (bmy, 8/22/05)
(5) Remove support for GEOS-1 and GEOS-STRAT met fields (bmy, 8/4/06)
(6) Resize the PP array from LLTROP to LLTROP_FIX (phs, 1/19/07)
(7) Now use LLTROP instead of LLTROP_FIX (bmy, 12/4/07)
(8) Now use LD65 instead of LLTROP (phs, 11/17/08)
08 Dec 2009 - R. Yantosca - Added ProTeX headers
28 Feb 2012 - R. Yantosca - Removed support for GEOS-3
01 Mar 2012 - R. Yantosca - Now use GET_XMID(I,J,L) from grid_mod.F90
01 Mar 2012 - R. Yantosca - Now use GET_YMID(I,J,L) from grid_mod.F90
26 Sep 2013 - R. Yantosca - Renamed GEOS_57 Cpp switch to GEOS_FP
23 Jan 2014 - M. Sulprizio- Now use LLTROP instead of LD65 (hyl,bmy,11/3/11)
24 Jul 2014 - R. Yantosca - Now compute BOXVL internally\
11 Aug 2015 - R. Yantosca - MERRA2 behaves the same way as GEOS-FP
25 Sep 2015 - E. Lundgren - Fix bug in setting X upper bound for ITS_IN_EUR
22 Oct 2015 - E. Lundgren - Fix PBLTOP and MTTOP levels for 750 hPa and
 350 hPa for GEOS-5, MERRA, GEOS-FP and MERRA2
```

### 10.12.2 chem\_tagged\_o3

Subroutine CHEM\_TAGGED\_O3 performs chemistry for several O3 species which are tagged by geographic and altitude regions.

### INTERFACE:

```
SUBROUTINE CHEM_TAGGED_O3(am_I_Root, Input_Opt, State_Met,
& State_Chm, State_Diag, RC)
```

### USES:

```
USE CMN_SIZE_MOD
```



```

USE ErrCode_Mod
USE ERROR_MOD, ONLY : ERROR_STOP
USE ERROR_MOD, ONLY : GEOS_CHEM_STOP
USE HCO_EMISLIST_MOD, ONLY : HCO_GetPtr
USE HCO_INTERFACE_MOD, ONLY : HcoState
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Diag_Mod, ONLY : DgnState
USE State_Met_Mod, ONLY : MetState
USE TIME_MOD, ONLY : GET_TS_CHEM
#if defined(BPCH_DIAG)
 USE CMN_DIAG_MOD
 USE DIAG_MOD, ONLY : AD65
#endif

```

```
IMPLICIT NONE
```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?
TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(MetState), INTENT(IN) :: State_Met ! Meteorology State object

```

#### INPUT/OUTPUT PARAMETERS:

```

TYPE(ChmState), INTENT(INOUT) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(INOUT) :: State_Diag ! Diagnostics State object

```

#### OUTPUT PARAMETERS:

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

#### !REMAKRS:

Dry deposition is now applied in mixing\_mod.F90. We have the application of 0x dry deposition from this routine, as well as the archival of the ND44 drydep diagnostic. (bmy, 6/15/15)

#### REVISION HISTORY:

- 20 Aug 2003 - R. Hudman - Initial version
- (1 ) Updated from the old routine "chemo3\_split.f" (rch, bmy, 8/20/03)
  - (2 ) Bug fix: don't put function call in WRITE statement (bmy, 2/20/04)
  - (3 ) Now use ND44\_TMP array to store vertical levels of drydep flux, then sum into AD44 array. This prevents numerical differences when using multiple processors. (bmy, 3/24/04)
  - (4 ) Now references LDYRD from "logical\_mod.f". Now references STT and N\_TRACERS from "tracer\_mod.f". Now references AD65 from "diag\_pl\_mod.f". Now uses ITS\_A\_NEW\_DAY from "time\_mod.f". (bmy, 7/20/04)
  - (5 ) Bug fix: Now avoid a SEG FAULT error if PBLFRAC isn't allocated. (bdf, bmy, 10/12/04)

- (6 ) Replace PBLFRAC from "drydep\_mod.f" with GET\_FRAC\_UNDER\_PBLTOP from "pbl\_mix\_mod.f". Now only sum ND44 diagnostic up to the maximum tropospheric level. (bmy, 2/17/05)
- (7 ) Resize PP, N D44\_TMP arrays from LLTROP to LLTROP\_FIX. Now only loop up to LLTROP\_FIX (phs, 1/19/07)
- (8 ) Now use LLTROP instead of LLTROP\_FIX (bmy, 12/4/07)
- (9 ) Now use LD65 instead of LLTROP (phs, 11/17/08)
- (10) Now only compute loss rate in troposphere (dbj, bmy, 10/26/09)
- 08 Dec 2009 - R. Yantosca - Added ProTeX headers
- 01 Mar 2012 - R. Yantosca - Now use GET\_AREA\_CM2(I,J,L) from grid\_mod.F90
- 14 Mar 2013 - M. Payer - Replace O<sub>x</sub> with O<sub>3</sub> as part of removal of NO<sub>x</sub>-O<sub>x</sub> partitioning
- 25 Mar 2013 - R. Yantosca - Now accept am\_I\_Root, Input\_Opt, State\_Chm, RC
- 20 Aug 2013 - R. Yantosca - Removed "define.h", this is now obsolete
- 23 Jan 2014 - M. Sulprizio- Now use LLTROP instead of LD65 (hyl,bmy,11/3/11)
- 11 Apr 2014 - R. Yantosca - Remove call to INIT\_TAGGED\_OX
- 24 Jul 2014 - R. Yantosca - Now compute BOXVL internally
- 17 Sep 2014 - C. Keller - Now use HEMCO for prod. and loss arrays
- 12 Jun 2015 - R. Yantosca - Now remove orphaned ND44 variables
- 15 Jun 2015 - R. Yantosca - Bug fix: PP doesn't have to be held PRIVATE
- 15 Jun 2015 - R. Yantosca - Updated comments for removal of drydep
- 29 Apr 2016 - R. Yantosca - Don't initialize pointers in declaration stmts
- 30 Jun 2016 - R. Yantosca - Remove instances of STT. Now get the advected species ID from State\_Chm%Map\_Advect.
- 03 Nov 2017 - R. Yantosca - Now accept State\_Diag as an argument

### 10.12.3 init\_tagged.o3

Subroutine INIT\_TAGGED\_O3 allocates and zeroes all module arrays.

#### INTERFACE:

```

SUBROUTINE INIT_TAGGED_O3(am_I_Root, Input_Opt,
& State_Chm, State_Diag, RC)

```

#### USES:

```

USE CMN_DIAG_MOD
USE CMN_SIZE_MOD
USE ErrCode_Mod
USE Input_Opt_Mod, ONLY : OptInput
USE State_Chm_Mod, ONLY : ChmState
USE State_Chm_Mod, ONLY : Ind_
USE State_Diag_Mod, ONLY : DgnState

```

#### INPUT PARAMETERS:

```

LOGICAL, INTENT(IN) :: am_I_Root ! Are we on the root CPU?

```

```

TYPE(OptInput), INTENT(IN) :: Input_Opt ! Input Options object
TYPE(ChmState), INTENT(IN) :: State_Chm ! Chemistry State object
TYPE(DgnState), INTENT(IN) :: State_Diag ! Diagnostics State object

```

**OUTPUT PARAMETERS:**

```

INTEGER, INTENT(OUT) :: RC ! Success or failure?

```

**REMARKS:**

This routine is now called from GC\_INIT\_EXTRA in GeosCore/input\_mod.F.

**REVISION HISTORY:**

```

20 Aug 2003 - R. Yantosca - Initial version
(1) Now reference N_TRACERS from "tracer_mod.f" (bmy, 7/20/04)
(2) Now use LD65 instead of LLTROP to dimension P24H, L24H (phs, 11/18/08)
08 Dec 2009 - R. Yantosca - Added ProTeX headers
25 Mar 2013 - R. Yantosca - Now accept am_I_Root, Input_Opt, RC args
23 Jan 2014 - M. Sulprizio- Now use LLTROP instead of LD65 to dimension P24H,
 L24H (hyl, bmy, 11/3/11)
20 Jun 2016 - R. Yantosca - Now define species ID's in the init phase
20 Jun 2016 - R. Yantosca - Add check to prevent array OOB errors
21 Jun 2016 - R. Yantosca - Rename IDT03Strt to id_03Strat
05 Dec 2017 - R. Yantosca - Now accept State_Diag as an argument

```

**10.12.4 cleanup\_tagged\_o3**

CLEANUP\_TAGGED\_O3 deallocates all module arrays.

**INTERFACE:**

```

SUBROUTINE CLEANUP_TAGGED_O3()

```

**REVISION HISTORY:**

```

20 Aug 2003 - R. Yantosca - Initial version
08 Dec 2009 - R. Yantosca - Added ProTeX headers

```