

Cartesian Partition Manager Library

1.0.0

作成 : Doxygen 1.8.0

Fri Jun 15 2012 17:15:59

Contents

1	構成索引	1
1.1	クラス階層	1
2	構成索引	3
2.1	構成	3
3	ファイル索引	5
3.1	ファイル一覧	5
4	クラス	7
4.1	クラス <code>cpm_ActiveSubDomainInfo</code>	7
4.1.1	説明	8
4.1.2	コンストラクタとデストラクタ	8
4.1.2.1	<code>cpm_ActiveSubDomainInfo</code>	8
4.1.2.2	<code>cpm_ActiveSubDomainInfo</code>	8
4.1.2.3	<code>~cpm_ActiveSubDomainInfo</code>	8
4.1.3	関数	8
4.1.3.1	<code>clear</code>	9
4.1.3.2	<code>GetBCID</code>	9
4.1.3.3	<code>GetPos</code>	9
4.1.3.4	<code>operator!=</code>	9
4.1.3.5	<code>operator==</code>	10
4.1.3.6	<code>Set</code>	10
4.1.3.7	<code>SetBCID</code>	10
4.1.3.8	<code>SetPos</code>	10
4.1.4	変数	11
4.1.4.1	<code>m_bcid</code>	11
4.1.4.2	<code>m_pos</code>	11
4.2	クラス <code>cpm_Base</code>	11
4.2.1	説明	12
4.2.2	コンストラクタとデストラクタ	12
4.2.2.1	<code>cpm_Base</code>	12

4.2.2.2	~cpm_Base	12
4.2.3	関数	12
4.2.3.1	cpm_strCompare	12
4.2.3.2	cpm_strCompareN	13
4.2.3.3	getCommNull	13
4.2.3.4	GetMemString	13
4.2.3.5	getRankNull	13
4.2.3.6	GetSpanTime	14
4.2.3.7	GetTime	14
4.2.3.8	GetWSpanTime	14
4.2.3.9	GetWTime	14
4.2.3.10	IsCommNull	15
4.2.3.11	IsRankNull	15
4.2.3.12	ReallIsDouble	15
4.2.3.13	VersionInfo	15
4.2.3.14	VersionInfo	16
4.3	クラス cpm_DomainInfo	16
4.3.1	説明	17
4.3.2	コンストラクタとデストラクタ	17
4.3.2.1	cpm_DomainInfo	17
4.3.2.2	~cpm_DomainInfo	17
4.3.3	関数	17
4.3.3.1	clear	17
4.3.3.2	GetOrigin	18
4.3.3.3	GetPitch	18
4.3.3.4	GetRegion	18
4.3.3.5	GetVoxNum	18
4.3.3.6	SetOrigin	19
4.3.3.7	SetPitch	19
4.3.3.8	SetRegion	19
4.3.3.9	SetVoxNum	19
4.3.4	変数	20
4.3.4.1	m_origin	20
4.3.4.2	m_pitch	20
4.3.4.3	m_region	20
4.3.4.4	m_voxNum	20
4.4	クラス cpm_GlobalDomainInfo	20
4.4.1	説明	22
4.4.2	コンストラクタとデストラクタ	22
4.4.2.1	cpm_GlobalDomainInfo	22

4.4.2.2	~cpm_GlobalDomainInfo	22
4.4.3	関数	22
4.4.3.1	AddSubDomain	22
4.4.3.2	clear	22
4.4.3.3	GetDivNum	23
4.4.3.4	GetSubDomainInfo	23
4.4.3.5	GetSubDomainNum	23
4.4.3.6	IsExistSubDomain	23
4.4.3.7	SetDivNum	24
4.4.4	変数	24
4.4.4.1	m_divNum	24
4.4.4.2	m_subDomainInfo	24
4.5	クラス cpm_LocalDomainInfo	24
4.5.1	説明	25
4.5.2	コンストラクタとデストラクタ	25
4.5.2.1	cpm_LocalDomainInfo	25
4.5.2.2	~cpm_LocalDomainInfo	26
4.5.3	関数	26
4.5.3.1	clear	26
4.6	クラス テンプレート cpm_ObjList< T >	26
4.6.1	説明	27
4.6.2	型定義	27
4.6.2.1	DelKeyList	27
4.6.2.2	ObjectMap	27
4.6.3	コンストラクタとデストラクタ	27
4.6.3.1	cpm_ObjList	27
4.6.3.2	~cpm_ObjList	27
4.6.4	関数	28
4.6.4.1	Add	28
4.6.4.2	Create	28
4.6.4.3	Delete	28
4.6.4.4	Get	28
4.6.5	変数	29
4.6.5.1	m_DelKeyList	29
4.6.5.2	m_newKey	29
4.6.5.3	m_ObjectMap	29
4.7	クラス cpm_ParaManager	29
4.7.1	説明	35
4.7.2	コンストラクタとデストラクタ	36
4.7.2.1	cpm_ParaManager	36

4.7.2.2	~cpm_ParaManager	36
4.7.3	関数	36
4.7.3.1	Abort	36
4.7.3.2	Allgather	36
4.7.3.3	Allgather	37
4.7.3.4	Allgatherv	37
4.7.3.5	Allgatherv	38
4.7.3.6	AllocDoubleS3D	38
4.7.3.7	AllocDoubleS4D	38
4.7.3.8	AllocDoubleS4DEx	39
4.7.3.9	AllocDoubleV3D	39
4.7.3.10	AllocDoubleV3DEx	39
4.7.3.11	AllocFloatS3D	40
4.7.3.12	AllocFloatS4D	40
4.7.3.13	AllocFloatS4DEx	40
4.7.3.14	AllocFloatV3D	41
4.7.3.15	AllocFloatV3DEx	41
4.7.3.16	AllocIntS3D	41
4.7.3.17	AllocIntS4D	42
4.7.3.18	AllocIntS4DEx	42
4.7.3.19	AllocIntV3D	42
4.7.3.20	AllocIntV3DEx	42
4.7.3.21	AllocRealS3D	43
4.7.3.22	AllocRealS4D	43
4.7.3.23	AllocRealS4DEx	43
4.7.3.24	AllocRealV3D	44
4.7.3.25	AllocRealV3DEx	44
4.7.3.26	Allreduce	44
4.7.3.27	Allreduce	45
4.7.3.28	Barrier	45
4.7.3.29	Bcast	46
4.7.3.30	Bcast	46
4.7.3.31	BndCommS3D	46
4.7.3.32	BndCommS3D	47
4.7.3.33	BndCommS3D_nowait	47
4.7.3.34	BndCommS3D_nowait	48
4.7.3.35	BndCommS4D	48
4.7.3.36	BndCommS4D	49
4.7.3.37	BndCommS4D_nowait	49
4.7.3.38	BndCommS4D_nowait	50

4.7.3.39	BndCommS4DEx	51
4.7.3.40	BndCommS4DEx	51
4.7.3.41	BndCommS4DEx_nowait	52
4.7.3.42	BndCommS4DEx_nowait	52
4.7.3.43	BndCommV3D	53
4.7.3.44	BndCommV3D	53
4.7.3.45	BndCommV3D_nowait	54
4.7.3.46	BndCommV3D_nowait	54
4.7.3.47	BndCommV3DEx	55
4.7.3.48	BndCommV3DEx	55
4.7.3.49	BndCommV3DEx_nowait	56
4.7.3.50	BndCommV3DEx_nowait	56
4.7.3.51	CalcCommSize	57
4.7.3.52	CopyArray	57
4.7.3.53	cpm_BndCommS3D_nowait	58
4.7.3.54	cpm_BndCommS4D_nowait	58
4.7.3.55	cpm_BndCommS4DEx_nowait	59
4.7.3.56	cpm_BndCommV3D_nowait	59
4.7.3.57	cpm_BndCommV3DEx_nowait	60
4.7.3.58	cpm_Irecv	60
4.7.3.59	cpm_Isend	61
4.7.3.60	cpm_Wait	61
4.7.3.61	cpm_wait_BndCommS3D	62
4.7.3.62	cpm_wait_BndCommS4D	62
4.7.3.63	cpm_wait_BndCommS4DEx	63
4.7.3.64	cpm_wait_BndCommV3D	63
4.7.3.65	cpm_wait_BndCommV3DEx	64
4.7.3.66	cpm_Waitall	64
4.7.3.67	CreateProcessGroup	65
4.7.3.68	DecideDivPattern	65
4.7.3.69	FindVoxelInfo	65
4.7.3.70	flush	66
4.7.3.71	flush	66
4.7.3.72	Gather	66
4.7.3.73	Gather	66
4.7.3.74	Gatherv	67
4.7.3.75	Gatherv	67
4.7.3.76	get_instance	68
4.7.3.77	get_instance	68
4.7.3.78	GetBCID	69

4.7.3.79	GetBndCommBuffer	69
4.7.3.80	GetBndCommBufferSize	69
4.7.3.81	GetDivNum	70
4.7.3.82	GetDivPos	70
4.7.3.83	GetGlobalOrigin	70
4.7.3.84	GetGlobalRegion	71
4.7.3.85	GetGlobalVoxelSize	71
4.7.3.86	GetLocalOrigin	71
4.7.3.87	GetLocalRegion	71
4.7.3.88	GetLocalVoxelSize	72
4.7.3.89	GetMPI_Comm	72
4.7.3.90	GetMPI_Datatype	72
4.7.3.91	GetMPI_Datatype	73
4.7.3.92	GetMPI_Op	73
4.7.3.93	GetMyRankID	74
4.7.3.94	GetNeighborRankID	74
4.7.3.95	GetNumRank	74
4.7.3.96	GetPeriodicRankID	74
4.7.3.97	GetPitch	75
4.7.3.98	GetVoxelHeadIndex	75
4.7.3.99	GetVoxelTailIndex	75
4.7.3.100	InitArray	76
4.7.3.101	Initialize	76
4.7.3.102	Initialize	76
4.7.3.103	Irecv	76
4.7.3.104	Irecv	77
4.7.3.105	Isend	77
4.7.3.106	Isend	78
4.7.3.107	IsParallel	78
4.7.3.108	IsParallel	78
4.7.3.109	packX	79
4.7.3.110	packXEx	79
4.7.3.111	packY	80
4.7.3.112	packYEx	80
4.7.3.113	packZ	81
4.7.3.114	packZEx	81
4.7.3.115	PeriodicCommS3D	82
4.7.3.116	PeriodicCommS3D	82
4.7.3.117	PeriodicCommS4D	83
4.7.3.118	PeriodicCommS4D	83

4.7.3.119	PeriodicCommS4DEx	84
4.7.3.120	PeriodicCommS4DEx	84
4.7.3.121	PeriodicCommV3D	85
4.7.3.122	PeriodicCommV3D	85
4.7.3.123	PeriodicCommV3DEx	86
4.7.3.124	PeriodicCommV3DEx	86
4.7.3.125	Recv	87
4.7.3.126	Recv	87
4.7.3.127	Send	88
4.7.3.128	Send	88
4.7.3.129	sendrecv	89
4.7.3.130	SetBndCommBuffer	89
4.7.3.131	unpackX	90
4.7.3.132	unpackXEx	90
4.7.3.133	unpackY	91
4.7.3.134	unpackYEx	91
4.7.3.135	unpackZ	92
4.7.3.136	unpackZEx	92
4.7.3.137	VoxelInit	93
4.7.3.138	VoxelInit	93
4.7.3.139	VoxelInit	94
4.7.3.140	Wait	94
4.7.3.141	wait_BndCommS3D	95
4.7.3.142	wait_BndCommS3D	95
4.7.3.143	wait_BndCommS4D	96
4.7.3.144	wait_BndCommS4D	96
4.7.3.145	wait_BndCommS4DEx	97
4.7.3.146	wait_BndCommS4DEx	97
4.7.3.147	wait_BndCommV3D	98
4.7.3.148	wait_BndCommV3D	98
4.7.3.149	wait_BndCommV3DEx	99
4.7.3.150	wait_BndCommV3DEx	99
4.7.3.151	Waitall	100
4.7.4	変数	100
4.7.4.1	m_bndCommInfoMap	100
4.7.4.2	m_nRank	100
4.7.4.3	m_procGrpList	100
4.7.4.4	m_rankNo	101
4.7.4.5	m_rankNoMap	101
4.7.4.6	m_reqList	101

4.7.4.7	<code>m_voxelInfoMap</code>	101
4.8	クラス <code>cpm_TextParser</code>	102
4.8.1	説明	102
4.8.2	コンストラクタとデストラクタ	103
4.8.2.1	<code>cpm_TextParser</code>	103
4.8.2.2	<code>~cpm_TextParser</code>	103
4.8.3	関数	103
4.8.3.1	<code>Read</code>	103
4.8.3.2	<code>readVector</code>	103
4.8.3.3	<code>readVector</code>	104
4.8.3.4	<code>readVector</code>	104
4.8.4	変数	104
4.8.4.1	<code>m_tp</code>	104
4.9	クラス <code>cpm_TextParserDomain</code>	105
4.9.1	説明	106
4.9.2	コンストラクタとデストラクタ	106
4.9.2.1	<code>cpm_TextParserDomain</code>	106
4.9.2.2	<code>~cpm_TextParserDomain</code>	106
4.9.3	関数	106
4.9.3.1	<code>Read</code>	106
4.9.3.2	<code>ReadDomainInfo</code>	107
4.9.3.3	<code>ReadMain</code>	107
4.9.3.4	<code>ReadSubdomainInfo</code>	107
4.10	クラス <code>cpm_VoxelInfo</code>	108
4.10.1	説明	109
4.10.2	コンストラクタとデストラクタ	110
4.10.2.1	<code>cpm_VoxelInfo</code>	110
4.10.2.2	<code>~cpm_VoxelInfo</code>	110
4.10.3	関数	110
4.10.3.1	<code>CreateLocalDomainInfo</code>	110
4.10.3.2	<code>CreateNeighborRankInfo</code>	110
4.10.3.3	<code>CreateRankMap</code>	111
4.10.3.4	<code>GetBCID</code>	111
4.10.3.5	<code>GetDivNum</code>	111
4.10.3.6	<code>GetDivPos</code>	111
4.10.3.7	<code>GetGlobalOrigin</code>	112
4.10.3.8	<code>GetGlobalRegion</code>	112
4.10.3.9	<code>GetGlobalVoxelSize</code>	112
4.10.3.10	<code>GetLocalOrigin</code>	112
4.10.3.11	<code>GetLocalRegion</code>	112

4.10.3.12	GetLocalVoxelSize	113
4.10.3.13	GetNeighborRankID	113
4.10.3.14	GetPeriodicRankID	113
4.10.3.15	GetPitch	113
4.10.3.16	GetVoxelHeadIndex	114
4.10.3.17	GetVoxelTailIndex	114
4.10.3.18	Init	114
4.10.4	フレンドと関連する関数	115
4.10.4.1	cpm_ParaManager	115
4.10.5	変数	115
4.10.5.1	m_comm	115
4.10.5.2	m_globalDomainInfo	115
4.10.5.3	m_localDomainInfo	115
4.10.5.4	m_neighborRankID	115
4.10.5.5	m_nRank	115
4.10.5.6	m_periodicRankID	115
4.10.5.7	m_rankMap	116
4.10.5.8	m_rankNo	116
4.10.5.9	m_voxelHeadIndex	116
4.10.5.10	m_voxelTailIndex	116
4.11	構造体 S_BNDCOMM_BUFFER	116
4.11.1	説明	117
4.11.2	コンストラクタとデストラクタ	117
4.11.2.1	S_BNDCOMM_BUFFER	117
4.11.2.2	~S_BNDCOMM_BUFFER	117
4.11.3	変数	117
4.11.3.1	m_bufX	117
4.11.3.2	m_bufY	117
4.11.3.3	m_bufZ	118
4.11.3.4	m_maxN	118
4.11.3.5	m_maxVC	118
4.11.3.6	m_nwX	118
4.11.3.7	m_nwY	118
4.11.3.8	m_nwZ	119
5	ファイル	121
5.1	cpm_Base.h	121
5.1.1	説明	122
5.1.2	マクロ定義	122
5.1.2.1	CPM_INLINE	122

5.2	cpm_Define.h	122
5.2.1	説明	124
5.2.2	マクロ定義	124
5.2.2.1	_IDX_S3D	124
5.2.2.2	_IDX_S4D	125
5.2.2.3	_IDX_S4DEX	125
5.2.2.4	_IDX_V3D	126
5.2.2.5	_IDX_V3DEX	126
5.2.2.6	REAL_BUF_TYPE	126
5.2.2.7	REAL_TYPE	126
5.2.3	列挙型	127
5.2.3.1	CPM_Datatype	127
5.2.3.2	cpm_DirFlag	127
5.2.3.3	cpm_ErrorCode	127
5.2.3.4	cpm_FaceFlag	129
5.2.3.5	CPM_Op	129
5.2.3.6	cpm_PMFlag	130
5.3	cpm_DomainInfo.cpp	130
5.3.1	説明	130
5.4	cpm_DomainInfo.h	131
5.4.1	説明	131
5.5	cpm_ObjList.h	132
5.5.1	説明	132
5.5.2	型定義	133
5.5.2.1	RankNoMap	133
5.6	cpm_ParaManager.cpp	133
5.6.1	説明	133
5.7	cpm_ParaManager.h	134
5.7.1	説明	134
5.7.2	型定義	135
5.7.2.1	BndCommInfoMap	135
5.7.2.2	RankNoMap	135
5.7.2.3	VoxelInfoMap	135
5.8	cpm_ParaManager_Alloc.cpp	135
5.8.1	説明	135
5.9	cpm_ParaManager_BndComm.h	136
5.9.1	説明	136
5.9.2	マクロ定義	136
5.9.2.1	_IDAFX	136
5.9.2.2	_IDXFY	137

5.9.2.3	<code>_IDXFZ</code>	137
5.10	<code>cpm_ParaManager_BndCommEx.h</code>	137
5.10.1	説明	137
5.10.2	マクロ定義	138
5.10.2.1	<code>_IDAFX</code>	138
5.10.2.2	<code>_IDXFY</code>	138
5.10.2.3	<code>_IDXFZ</code>	138
5.11	<code>cpm_ParaManager_frtIF.cpp</code>	139
5.11.1	説明	142
5.11.2	マクロ定義	142
5.11.2.1	<code>cpm_Abort_</code>	142
5.11.2.2	<code>cpm_Allgather_</code>	142
5.11.2.3	<code>cpm_Allgatherv_</code>	142
5.11.2.4	<code>cpm_Allreduce_</code>	142
5.11.2.5	<code>cpm_Barrier_</code>	142
5.11.2.6	<code>cpm_Bcast_</code>	142
5.11.2.7	<code>cpm_BndCommS3D_</code>	142
5.11.2.8	<code>cpm_BndCommS3D_nowait_</code>	142
5.11.2.9	<code>cpm_BndCommS4D_</code>	143
5.11.2.10	<code>cpm_BndCommS4D_nowait_</code>	143
5.11.2.11	<code>cpm_BndCommS4DEx_</code>	143
5.11.2.12	<code>cpm_BndCommS4DEx_nowait_</code>	143
5.11.2.13	<code>cpm_BndCommV3D_</code>	143
5.11.2.14	<code>cpm_BndCommV3D_nowait_</code>	143
5.11.2.15	<code>cpm_BndCommV3DEx_</code>	143
5.11.2.16	<code>cpm_BndCommV3DEx_nowait_</code>	143
5.11.2.17	<code>CPM_EXTERN</code>	143
5.11.2.18	<code>cpm_Gather_</code>	143
5.11.2.19	<code>cpm_Gatherv_</code>	143
5.11.2.20	<code>cpm_GetBCID_</code>	144
5.11.2.21	<code>cpm_GetDivNum_</code>	144
5.11.2.22	<code>cpm_GetDivPos_</code>	144
5.11.2.23	<code>cpm_GetGlobalOrigin_</code>	144
5.11.2.24	<code>cpm_GetGlobalRegion_</code>	144
5.11.2.25	<code>cpm_GetGlobalVoxelSize_</code>	144
5.11.2.26	<code>cpm_GetLocalOrigin_</code>	144
5.11.2.27	<code>cpm_GetLocalRegion_</code>	144
5.11.2.28	<code>cpm_GetLocalVoxelSize_</code>	144
5.11.2.29	<code>cpm_GetMyRankID_</code>	144
5.11.2.30	<code>cpm_GetNeighborRankID_</code>	144

5.11.2.31	<code>cpm_GetNumRank_</code>	144
5.11.2.32	<code>cpm_GetPeriodicRankID_</code>	145
5.11.2.33	<code>cpm_GetPitch_</code>	145
5.11.2.34	<code>cpm_GetVoxelHeadIndex_</code>	145
5.11.2.35	<code>cpm_GetVoxelTailIndex_</code>	145
5.11.2.36	<code>cpm_Initialize_</code>	145
5.11.2.37	<code>cpm_lrecv_</code>	145
5.11.2.38	<code>cpm_lsend_</code>	145
5.11.2.39	<code>cpm_IsParallel_</code>	145
5.11.2.40	<code>cpm_PeriodicCommS3D</code>	145
5.11.2.41	<code>cpm_PeriodicCommS4D</code>	145
5.11.2.42	<code>cpm_PeriodicCommS4DEx</code>	145
5.11.2.43	<code>cpm_PeriodicCommV3D</code>	145
5.11.2.44	<code>cpm_PeriodicCommV3DEx</code>	146
5.11.2.45	<code>cpm_Recv_</code>	146
5.11.2.46	<code>cpm_Send_</code>	146
5.11.2.47	<code>cpm_SetBndCommBuffer_</code>	146
5.11.2.48	<code>cpm_Voxellnit_</code>	146
5.11.2.49	<code>cpm_Voxellnit_nodiv_</code>	146
5.11.2.50	<code>cpm_Wait_</code>	146
5.11.2.51	<code>cpm_wait_BndCommS3D_</code>	146
5.11.2.52	<code>cpm_wait_BndCommS4D_</code>	146
5.11.2.53	<code>cpm_wait_BndCommS4DEx_</code>	146
5.11.2.54	<code>cpm_wait_BndCommV3D_</code>	146
5.11.2.55	<code>cpm_wait_BndCommV3DEx_</code>	146
5.11.2.56	<code>cpm_Waitall_</code>	147
5.11.3	関数	147
5.11.3.1	<code>cpm_Abort_</code>	147
5.11.3.2	<code>cpm_Allgather_</code>	147
5.11.3.3	<code>cpm_Allgatherv_</code>	147
5.11.3.4	<code>cpm_Allreduce_</code>	148
5.11.3.5	<code>cpm_Barrier_</code>	148
5.11.3.6	<code>cpm_Bcast_</code>	148
5.11.3.7	<code>cpm_BndCommS3D_</code>	149
5.11.3.8	<code>cpm_BndCommS3D_nowait_</code>	149
5.11.3.9	<code>cpm_BndCommS4D_</code>	150
5.11.3.10	<code>cpm_BndCommS4D_nowait_</code>	150
5.11.3.11	<code>cpm_BndCommS4DEx_</code>	151
5.11.3.12	<code>cpm_BndCommS4DEx_nowait_</code>	151
5.11.3.13	<code>cpm_BndCommV3D_</code>	152

5.11.3.14	cpm_BndCommV3D_nowait_	152
5.11.3.15	cpm_BndCommV3DEx_	153
5.11.3.16	cpm_BndCommV3DEx_nowait_	153
5.11.3.17	cpm_Gather_	153
5.11.3.18	cpm_Gatherv_	154
5.11.3.19	cpm_GetBCID_	154
5.11.3.20	cpm_GetDivNum_	155
5.11.3.21	cpm_GetDivPos_	155
5.11.3.22	cpm_GetGlobalOrigin_	155
5.11.3.23	cpm_GetGlobalRegion_	155
5.11.3.24	cpm_GetGlobalVoxelSize_	156
5.11.3.25	cpm_GetLocalOrigin_	156
5.11.3.26	cpm_GetLocalRegion_	156
5.11.3.27	cpm_GetLocalVoxelSize_	157
5.11.3.28	cpm_GetMyRankID_	157
5.11.3.29	cpm_GetNeighborRankID_	157
5.11.3.30	cpm_GetNumRank_	157
5.11.3.31	cpm_GetPeriodicRankID_	158
5.11.3.32	cpm_GetPitch_	158
5.11.3.33	cpm_GetVoxelHeadIndex_	158
5.11.3.34	cpm_GetVoxelTailIndex_	159
5.11.3.35	cpm_Initialize_	159
5.11.3.36	cpm_Irecv_	159
5.11.3.37	cpm_Isend_	160
5.11.3.38	cpm_IsParallel_	160
5.11.3.39	cpm_PeriodicComms3D_	160
5.11.3.40	cpm_PeriodicComms4D_	161
5.11.3.41	cpm_PeriodicComms4DEx_	161
5.11.3.42	cpm_PeriodicCommV3D_	162
5.11.3.43	cpm_PeriodicCommV3DEx_	162
5.11.3.44	cpm_Recv_	163
5.11.3.45	cpm_Send_	163
5.11.3.46	cpm_SetBndCommBuffer_	164
5.11.3.47	cpm_Voxellnit_	164
5.11.3.48	cpm_Voxellnit_nodiv_	164
5.11.3.49	cpm_Wait_	165
5.11.3.50	cpm_wait_BndComms3D_	165
5.11.3.51	cpm_wait_BndComms4D_	166
5.11.3.52	cpm_wait_BndComms4DEx_	166
5.11.3.53	cpm_wait_BndCommV3D_	167

5.11.3.54 cpm_wait_BndCommV3DEx_	167
5.11.3.55 cpm_Waitall_	167
5.12 cpm_ParaManager_inline.h	168
5.12.1 説明	168
5.13 cpm_ParaManager_MPI.cpp	168
5.13.1 説明	169
5.14 cpm_TextParser.cpp	169
5.14.1 説明	169
5.15 cpm_TextParser.h	169
5.15.1 説明	170
5.16 cpm_TextParserDomain.cpp	171
5.16.1 説明	171
5.17 cpm_TextParserDomain.h	171
5.17.1 説明	172
5.18 cpm_Version.h	173
5.18.1 説明	173
5.18.2 マクロ定義	173
5.18.2.1 CPM_REVISION	173
5.18.2.2 CPM_VERSION_NO	173
5.19 cpm_VoxelInfo.cpp	174
5.19.1 説明	174
5.20 cpm_VoxelInfo.h	174
5.20.1 説明	175

Chapter 1

構成索引

1.1 クラス階層

この継承一覧はおおまかにはソートされていますが、完全にアルファベット順でソートされてはいません。

cpm_Base	11
cpm_ActiveSubDomainInfo	7
cpm_LocalDomainInfo	24
cpm_DomainInfo	16
cpm_GlobalDomainInfo	20
cpm_LocalDomainInfo	24
cpm_ObjList< T >	26
cpm_ParaManager	29
cpm_TextParser	102
cpm_TextParserDomain	105
cpm_VoxelInfo	108
S_BNDCOMM_BUFFER	116

Chapter 2

構成索引

2.1 構成

クラス、構造体、共用体、インタフェースの説明です。

cpm_ActiveSubDomainInfo	7
cpm_Base	11
cpm_DomainInfo	16
cpm_GlobalDomainInfo	20
cpm_LocalDomainInfo	24
cpm_ObjList< T >	26
cpm_ParaManager	29
cpm_TextParser	102
cpm_TextParserDomain	105
cpm_VoxelInfo	108
S_BNDCOMM_BUFFER	116

Chapter 3

ファイル索引

3.1 ファイル一覧

これはファイル一覧です。

cpm_Base.h	121
cpm_Define.h	122
cpm_DomainInfo.cpp	130
cpm_DomainInfo.h	131
cpm_ObjList.h	132
cpm_ParaManager.cpp	133
cpm_ParaManager.h	134
cpm_ParaManager_Alloc.cpp	135
cpm_ParaManager_BndComm.h	136
cpm_ParaManager_BndCommEx.h	137
cpm_ParaManager_frtIF.cpp	139
cpm_ParaManager_inline.h	168
cpm_ParaManager_MPI.cpp	168
cpm_TextParser.cpp	169
cpm_TextParser.h	169
cpm_TextParserDomain.cpp	171
cpm_TextParserDomain.h	171
cpm_Version.h	173
cpm_VoxelInfo.cpp	174
cpm_VoxelInfo.h	174

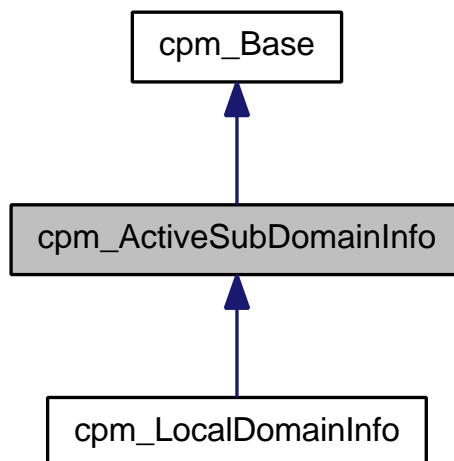
Chapter 4

クラス

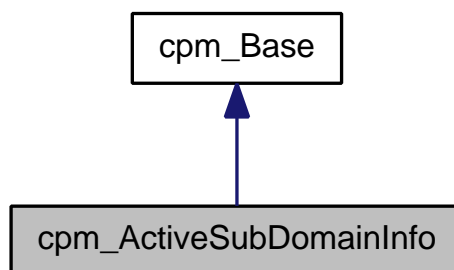
4.1 クラス cpm_ActiveSubDomainInfo

```
#include <cpm_DomainInfo.h>
```

cpm_ActiveSubDomainInfo に対する継承グラフ



cpm_ActiveSubDomainInfo のコラボレーション図



Public メソッド

- [cpm_ActiveSubDomainInfo \(\)](#)
- [cpm_ActiveSubDomainInfo \(int pos\[3\], int bcid\[6\]\)](#)

- virtual `~cpm_ActiveSubDomainInfo()`
- virtual void `clear()`
- void `Set`(int pos[3], int bcid[6])
- void `SetPos`(int pos[3])
- const int * `GetPos`() const
- void `SetBCID`(int bcid[6])
- const int * `GetBCID`() const
- bool `operator==`(cpm_ActiveSubDomainInfo dom)
- bool `operator!=`(cpm_ActiveSubDomainInfo dom)

Private 変数

- int `m_pos` [3]
領域分割内での位置
- int `m_bcid` [6]
6面の境界条件ID

4.1.1 説明

CPMのサブ領域情報クラス

cpm_DomainInfo.h の 97 行で定義されています。

4.1.2 コンストラクタとデストラクタ

4.1.2.1 cpm_ActiveSubDomainInfo::cpm_ActiveSubDomainInfo ()

デフォルトコンストラクタ

cpm_DomainInfo.cpp の 118 行で定義されています。

参照先 `clear()`.

4.1.2.2 cpm_ActiveSubDomainInfo::cpm_ActiveSubDomainInfo (int pos[3], int bcid[6])

コンストラクタ

引数

<code>in</code>	<code>pos</code>	領域分割内での位置
<code>in</code>	<code>bcid</code>	面の境界条件ID

cpm_DomainInfo.cpp の 126 行で定義されています。

参照先 `Set()`.

4.1.2.3 cpm_ActiveSubDomainInfo::~cpm_ActiveSubDomainInfo () [virtual]

デストラクタ

cpm_DomainInfo.cpp の 134 行で定義されています。

4.1.3 関数

4.1.3.1 `void cpm_ActiveSubDomainInfo::clear () [virtual]`

情報のクリア

`cpm_LocalDomainInfo` で再定義されています。

`cpm_DomainInfo.cpp` の 141 行で定義されています。

参照先 `m_bcid`, `m_pos`, `X_MINUS`, `X_PLUS`, `Y_MINUS`, `Y_PLUS`, `Z_MINUS`, と `Z_PLUS`.

参照元 `cpm_ActiveSubDomainInfo()`.

4.1.3.2 `const int * cpm_ActiveSubDomainInfo::GetBCID () const`

BCID の取得

戻り値

境界条件ID 情報整数配列のポインタ

`cpm_DomainInfo.cpp` の 197 行で定義されています。

参照先 `m_bcid`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, と `cpm_VoxelInfo::GetBCID()`.

4.1.3.3 `const int * cpm_ActiveSubDomainInfo::GetPos () const`

位置の取得

戻り値

位置情報整数配列のポインタ

`cpm_DomainInfo.cpp` の 176 行で定義されています。

参照先 `m_pos`.

参照元 `cpm_VoxelInfo::CreateNeighborRankInfo()`, `cpm_VoxelInfo::CreateRankMap()`, と `cpm_VoxelInfo::GetDivPos()`.

4.1.3.4 `bool cpm_ActiveSubDomainInfo::operator!=(cpm_ActiveSubDomainInfo dom)`

比較演算子

引数

<code>in</code>	<code>dom</code>	比較対象の活性サブドメイン情報
-----------------	------------------	-----------------

戻り値

<code>true</code>	違う位置情報を持つ
<code>false</code>	同じ位置情報を持つ

`cpm_DomainInfo.cpp` の 216 行で定義されています。

参照先 `m_pos`.

4.1.3.5 bool cpm_ActiveSubDomainInfo::operator==(cpm_ActiveSubDomainInfo dom)

比較演算子

引数

<i>in</i>	<i>dom</i>	比較対象の活性サブドメイン情報
-----------	------------	-----------------

戻り値

<i>true</i>	同じ位置情報を持つ
<i>false</i>	違う位置情報を持つ

cpm_DomainInfo.cpp の 205 行で定義されています。

参照先 m_pos.

4.1.3.6 void cpm_ActiveSubDomainInfo::Set (int pos[3], int bcid[6])

値のセット

引数

<i>in</i>	<i>pos</i>	領域分割内での位置
<i>in</i>	<i>bcid</i>	面の境界条件ID

cpm_DomainInfo.cpp の 157 行で定義されています。

参照先 SetBCID(), と SetPos().

参照元 cpm_ActiveSubDomainInfo().

4.1.3.7 void cpm_ActiveSubDomainInfo::SetBCID (int bcid[6])

境界条件ID のセット

引数

<i>in</i>	<i>bcid</i>	面の境界条件ID
-----------	-------------	----------

cpm_DomainInfo.cpp の 184 行で定義されています。

参照先 m_bcid, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadSubdomainInfo(), と Set().

4.1.3.8 void cpm_ActiveSubDomainInfo::SetPos (int pos[3])

位置のセット

引数

<i>in</i>	<i>pos</i>	領域分割内での位置
-----------	------------	-----------

cpm_DomainInfo.cpp の 166 行で定義されています。

参照先 m_pos.

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadSubdomainInfo(), と Set().

4.1.4 変数

4.1.4.1 int cpm_ActiveSubDomainInfo::m_bcid[6] [private]

6 面の境界条件ID

cpm_DomainInfo.h の 170 行で定義されています。

参照元 clear(), GetBCID(), と SetBCID().

4.1.4.2 int cpm_ActiveSubDomainInfo::m_pos[3] [private]

領域分割内での位置

cpm_DomainInfo.h の 169 行で定義されています。

参照元 clear(), GetPos(), operator!=(), operator==(), と SetPos().

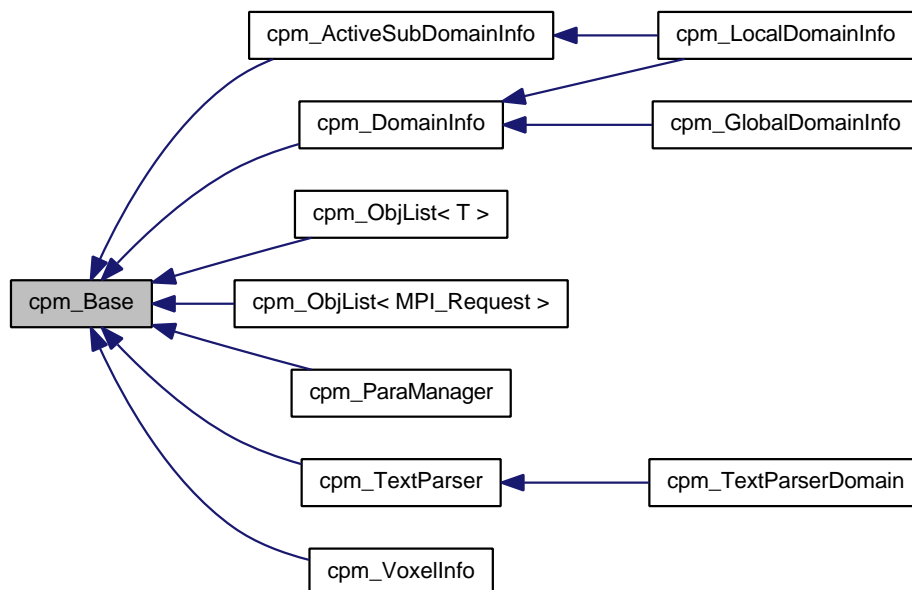
このクラスの説明は次のファイルから生成されました:

- [cpm_DomainInfo.h](#)
- [cpm_DomainInfo.cpp](#)

4.2 クラス cpm_Base

```
#include <cpm_Base.h>
```

cpm_Base に対する継承グラフ



Public メソッド

- `CPM_INLINE int cpm_strCompare (std::string str1, std::string str2, bool ignorecase=true)`
- `CPM_INLINE int cpm_strCompareN (std::string str1, std::string str2, size_t num, bool ignorecase=true)`

Static Public メソッド

- `static CPM_INLINE int getRankNull ()`

- static `CPM_INLINE` bool `IsRankNull` (int rankNo)
- static `CPM_INLINE` MPI_Comm `getCommNull` ()
- static `CPM_INLINE` bool `IsCommNull` (MPI_Comm comm)
- static `CPM_INLINE` bool `ReallsDouble` ()
- static `CPM_INLINE` double `GetTime` ()
- static `CPM_INLINE` double `GetSpanTime` (double before)
- static `CPM_INLINE` double `GetWTime` ()
- static `CPM_INLINE` double `GetWSpanTime` (double before)
- static `CPM_INLINE` std::string `GetMemString` (size_t mem)
- static `CPM_INLINE` void `VersionInfo` ()
- static `CPM_INLINE` void `VersionInfo` (std::ostream &ofs)

Protected メソッド

- `cpm_Base` ()
- virtual `~cpm_Base` ()

4.2.1 説明

CPM のベースクラス

`cpm_Base.h` の 45 行で定義されています。

4.2.2 コンストラクタとデストラクタ

4.2.2.1 `cpm_Base::cpm_Base()` [`inline`, `protected`]

コンストラクタ

`cpm_Base.h` の 239 行で定義されています。

4.2.2.2 `virtual cpm_Base::~~cpm_Base()` [`inline`, `protected`, `virtual`]

デストラクタ

`cpm_Base.h` の 242 行で定義されています。

4.2.3 関数

4.2.3.1 `CPM_INLINE int cpm_Base::cpm_strCompare (std::string str1, std::string str2, bool ignorecase = true)` [`inline`]

文字列の比較

引数

in	<i>str1</i>	文字列 1
in	<i>str2</i>	文字列 2
in	<i>ignorecase</i>	true=大文字小文字を区別しない、false=区別する

戻り値

0	一致する
0 以外	一致しない

cpm_Base.h の 205 行で定義されています。

参照元 cpm_strCompareN(), cpm_TextParserDomain::ReadDomainInfo(), と cpm_TextParserDomain::ReadSubdomainInfo().

4.2.3.2 CPM_INLINE int cpm_Base::cpm_strCompareN (std::string str1, std::string str2, size_t num, bool ignorecase =true) [inline]

文字列の比較 (文字数指定)

引数

in	str1	文字列 1
in	str2	文字列 2
in	num	比較する文字数 (先頭から)
in	ignorecase	true=大文字小文字を区別しない、false=区別する

戻り値

0	一致する
0 以外	一致しない

cpm_Base.h の 227 行で定義されています。

参照先 cpm_strCompare().

参照元 cpm_TextParserDomain::ReadSubdomainInfo().

4.2.3.3 static CPM_INLINE MPI_Comm cpm_Base::getCommNull () [inline, static]

NULL の MPI_Comm を取得

戻り値

NULL の MPI_Comm

cpm_Base.h の 72 行で定義されています。

参照元 cpm_ParaManager::GetMPI_Comm().

4.2.3.4 static CPM_INLINE std::string cpm_Base::GetMemString (size_t mem) [inline, static]

メモリ量の文字列を返す

引数

in	mem	メモリ量 (byte)
----	-----	-------------

戻り値

メモリ量の文字列

cpm_Base.h の 146 行で定義されています。

4.2.3.5 static CPM_INLINE int cpm_Base::getRankNull () [inline, static]

NULL のランク番号を取得

戻り値

NULL のランク番号

cpm_Base.h の 54 行で定義されています。

参照元 cpm_VoxelInfo::cpm_VoxelInfo(), cpm_VoxelInfo::CreateNeighborRankInfo(), cpm_VoxelInfo::CreateRankMap(), cpm_ParaManager::GetMyRankID(), cpm_ParaManager::PeriodicCommS4D(), と cpm_ParaManager::PeriodicCommS4DEx().

4.2.3.6 static CPM_INLINE double cpm_Base::GetSpanTime (double *before*) [inline, static]

経過時刻の取得 (gettimeofday 版)

引数

<i>in</i>	<i>before</i>	計測開始時刻
-----------	---------------	--------

戻り値

計測開始時刻からの経過時刻

cpm_Base.h の 117 行で定義されています。

参照先 GetTime().

4.2.3.7 static CPM_INLINE double cpm_Base::GetTime () [inline, static]

時刻の取得 (gettimeofday 版) 時刻

cpm_Base.h の 102 行で定義されています。

参照元 GetSpanTime().

4.2.3.8 static CPM_INLINE double cpm_Base::GetWSpanTime (double *before*) [inline, static]

経過時刻の取得 (MPI_Wtime 版)

引数

<i>in</i>	<i>before</i>	計測開始時刻
-----------	---------------	--------

戻り値

計測開始時刻からの経過時刻

cpm_Base.h の 136 行で定義されています。

参照先 GetWTime().

4.2.3.9 static CPM_INLINE double cpm_Base::GetWTime () [inline, static]

時刻の取得 (MPI_Wtime 版) 時刻

cpm_Base.h の 126 行で定義されています。

参照元 GetWSpanTime().

4.2.3.10 static CPM_INLINE bool cpm_Base::IsCommNull (MPI_Comm comm) [inline, static]

NULL のMPI_Comm かどうかを確認

戻り値

<i>true</i>	NULL
<i>false</i>	NULL ではない

cpm_Base.h の 81 行で定義されています。

参照元 cpm_ParaManager::Allgather(), cpm_ParaManager::Allgatherv(), cpm_ParaManager::Allreduce(), cpm_ParaManager::Barrier(), cpm_ParaManager::Bcast(), cpm_ParaManager::CreateProcessGroup(), cpm_ParaManager::Gather(), cpm_ParaManager::Gatherv(), cpm_ParaManager::GetMyRankID(), cpm_ParaManager::GetNumRank(), cpm_VoxellInfo::Init(), cpm_ParaManager::Irecv(), cpm_ParaManager::Isend(), cpm_ParaManager::Recv(), cpm_ParaManager::Send(), と cpm_ParaManager::VoxellInit().

4.2.3.11 static CPM_INLINE bool cpm_Base::IsRankNull (int rankNo) [inline, static]

NULL のランクかどうかを確認

戻り値

<i>true</i>	NULL
<i>false</i>	NULL ではない

cpm_Base.h の 63 行で定義されています。

参照元 cpm_ParaManager::packX(), cpm_ParaManager::packXEx(), cpm_ParaManager::packY(), cpm_ParaManager::packYEx(), cpm_ParaManager::packZ(), cpm_ParaManager::packZEx(), cpm_ParaManager::sendrecv(), cpm_ParaManager::unpackX(), cpm_ParaManager::unpackXEx(), cpm_ParaManager::unpackY(), cpm_ParaManager::unpackYEx(), cpm_ParaManager::unpackZ(), と cpm_ParaManager::unpackZEx().

4.2.3.12 static CPM_INLINE bool cpm_Base::ReallIsDouble () [inline, static]

実数型REAL_TYPE が倍精度かどうか確認

戻り値

<i>true</i>	倍精度
<i>false</i>	単精度

cpm_Base.h の 91 行で定義されています。

参照先 REAL_TYPE.

参照元 cpm_ParaManager::GetMPI_Datatype().

4.2.3.13 static CPM_INLINE void cpm_Base::VersionInfo () [inline, static]

バージョンを出力する

引数

<i>ofs</i>	出力ストリーム
------------	---------

cpm_Base.h の 180 行で定義されています。

4.2.3.14 `static CPM_INLINE void cpm_Base::VersionInfo (std::ostream & ofs) [inline, static]`

バージョンを出力する

引数

<code>ofs</code>	出力ストリーム
------------------	---------

cpm_Base.h の 189 行で定義されています。

参照先 CPM_VERSION_NO.

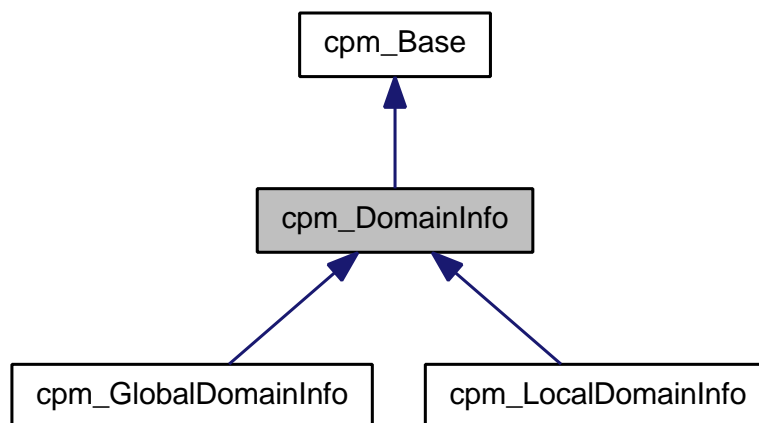
このクラスの説明は次のファイルから生成されました:

- [cpm_Base.h](#)

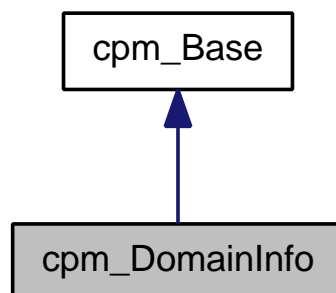
4.3 クラス cpm_DomainInfo

```
#include <cpm_DomainInfo.h>
```

cpm_DomainInfo に対する継承グラフ



cpm_DomainInfo のコラボレーション図



Public メソッド

- [cpm_DomainInfo \(\)](#)
- [virtual ~cpm_DomainInfo \(\)](#)
- [virtual void clear \(\)](#)

- void `SetOrigin` (`REAL_TYPE` org[3])
- const `REAL_TYPE` * `GetOrigin` () const
- void `SetPitch` (`REAL_TYPE` pch[3])
- const `REAL_TYPE` * `GetPitch` () const
- void `SetRegion` (`REAL_TYPE` rgn[3])
- const `REAL_TYPE` * `GetRegion` () const
- void `SetVoxNum` (int vox[3])
- const int * `GetVoxNum` () const

Private 変数

- `REAL_TYPE` m_origin [3]
原点
- `REAL_TYPE` m_pitch [3]
ピッチ
- `REAL_TYPE` m_region [3]
空間サイズ
- int m_voxNum [3]
VOXEL 数

4.3.1 説明

CPM の領域情報クラス

`cpm_DomainInfo.h` の 22 行で定義されています。

4.3.2 コンストラクタとデストラクタ

4.3.2.1 `cpm_DomainInfo::cpm_DomainInfo ()`

コンストラクタ

`cpm_DomainInfo.cpp` の 18 行で定義されています。

参照先 `clear()`.

4.3.2.2 `cpm_DomainInfo::~~cpm_DomainInfo ()` [virtual]

デストラクタ

`cpm_DomainInfo.cpp` の 26 行で定義されています。

4.3.3 関数

4.3.3.1 `void cpm_DomainInfo::clear ()` [virtual]

情報のクリア

`cpm_LocalDomainInfo`, と `cpm_GlobalDomainInfo` で再定義されています。

`cpm_DomainInfo.cpp` の 33 行で定義されています。

参照先 m_origin, m_pitch, m_region, と m_voxNum.

参照元 `cpm_DomainInfo()`.

4.3.3.2 `const REAL_TYPE * cpm_DomainInfo::GetOrigin () const`

原点の取得

戻り値

原点情報実数配列のポインタ

cpm_DomainInfo.cpp の 57 行で定義されています。

参照先 m_origin.

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_VoxelInfo::GetGlobalOrigin(), と cpm_VoxelInfo::GetLocalOrigin().

4.3.3.3 `const REAL_TYPE * cpm_DomainInfo::GetPitch () const`

ピッチの取得

戻り値

ピッチ情報実数配列のポインタ

cpm_DomainInfo.cpp の 75 行で定義されています。

参照先 m_pitch.

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), と cpm_VoxelInfo::GetPitch().

4.3.3.4 `const REAL_TYPE * cpm_DomainInfo::GetRegion () const`

空間サイズの取得

戻り値

空間サイズ情報実数配列のポインタ

cpm_DomainInfo.cpp の 93 行で定義されています。

参照先 m_region.

参照元 cpm_VoxelInfo::GetGlobalRegion(), と cpm_VoxelInfo::GetLocalRegion().

4.3.3.5 `const int * cpm_DomainInfo::GetVoxNum () const`

VOXEL 数の取得

戻り値

VOXEL 数情報実数配列のポインタ

cpm_DomainInfo.cpp の 111 行で定義されています。

参照先 m_voxNum.

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_VoxelInfo::GetGlobalVoxelSize(), と cpm_VoxelInfo::GetLocalVoxelSize().

4.3.3.6 `void cpm_DomainInfo::SetOrigin (REAL_TYPE org[3])`

原点のセット

引数

<code>in</code>	<code>org</code>	原点情報
-----------------	------------------	------

`cpm_DomainInfo.cpp` の 47 行で定義されています。

参照先 `m_origin`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_TextParserDomain::ReadDomainInfo()`, と `cpm_ParManager::VoxelInit()`.

4.3.3.7 `void cpm_DomainInfo::SetPitch (REAL_TYPE pch[3])`

ピッチのセット

引数

<code>in</code>	<code>pch</code>	ピッチ情報
-----------------	------------------	-------

`cpm_DomainInfo.cpp` の 65 行で定義されています。

参照先 `m_pitch`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_TextParserDomain::ReadDomainInfo()`, と `cpm_ParManager::VoxelInit()`.

4.3.3.8 `void cpm_DomainInfo::SetRegion (REAL_TYPE rgn[3])`

空間サイズのセット

引数

<code>in</code>	<code>rgn</code>	空間サイズ情報
-----------------	------------------	---------

`cpm_DomainInfo.cpp` の 83 行で定義されています。

参照先 `m_region`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_TextParserDomain::ReadDomainInfo()`, と `cpm_ParManager::VoxelInit()`.

4.3.3.9 `void cpm_DomainInfo::SetVoxNum (int vox[3])`

VOXEL 数のセット

引数

<code>in</code>	<code>vox</code>	VOXEL 数情報
-----------------	------------------	-----------

`cpm_DomainInfo.cpp` の 101 行で定義されています。

参照先 `m_voxNum`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_TextParserDomain::ReadDomainInfo()`, と `cpm_ParManager::VoxelInit()`.

4.3.4 変数

4.3.4.1 `REAL_TYPE cpm_DomainInfo::m_origin[3]` [private]

原点

`cpm_DomainInfo.h` の 88 行で定義されています。

参照元 `clear()`, `GetOrigin()`, と `SetOrigin()`.

4.3.4.2 `REAL_TYPE cpm_DomainInfo::m_pitch[3]` [private]

ピッチ

`cpm_DomainInfo.h` の 89 行で定義されています。

参照元 `clear()`, `GetPitch()`, と `SetPitch()`.

4.3.4.3 `REAL_TYPE cpm_DomainInfo::m_region[3]` [private]

空間サイズ

`cpm_DomainInfo.h` の 90 行で定義されています。

参照元 `clear()`, `GetRegion()`, と `SetRegion()`.

4.3.4.4 `int cpm_DomainInfo::m_voxNum[3]` [private]

VOXEL 数

`cpm_DomainInfo.h` の 91 行で定義されています。

参照元 `clear()`, `GetVoxNum()`, と `SetVoxNum()`.

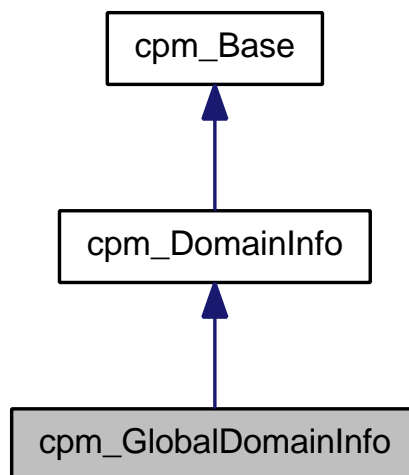
このクラスの説明は次のファイルから生成されました:

- [cpm_DomainInfo.h](#)
- [cpm_DomainInfo.cpp](#)

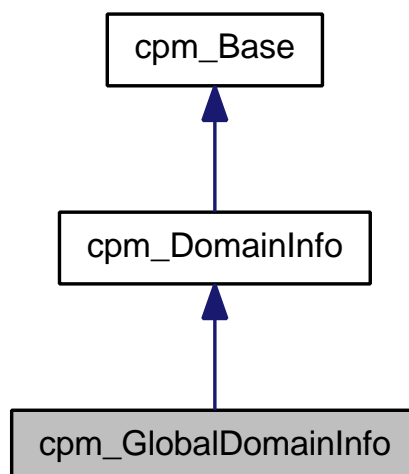
4.4 クラス `cpm_GlobalDomainInfo`

```
#include <cpm_DomainInfo.h>
```

`cpm_GlobalDomainInfo` に対する継承グラフ



`cpm_GlobalDomainInfo` のコラボレーション図



Public メソッド

- `cpm_GlobalDomainInfo` ()
- virtual `~cpm_GlobalDomainInfo` ()
- virtual void `clear` ()
- void `SetDivNum` (int div[3])
- const int * `GetDivNum` () const
- bool `IsExistSubDomain` (`cpm_ActiveSubDomainInfo` subDomain)
- bool `AddSubDomain` (`cpm_ActiveSubDomainInfo` subDomain)
- int `GetSubDomainNum` () const
- const `cpm_ActiveSubDomainInfo` * `GetSubDomainInfo` (size_t idx) const

Private 変数

- int `m_divNum` [3]
領域分割数

- `std::vector`
`< cpm_ActiveSubDomainInfo > m_subDomainInfo`
 活性サブドメイン情報

4.4.1 説明

CPM の全体領域情報クラス

`cpm_DomainInfo.h` の 176 行で定義されています。

4.4.2 コンストラクタとデストラクタ

4.4.2.1 `cpm_GlobalDomainInfo::cpm_GlobalDomainInfo ()`

コンストラクタ

`cpm_DomainInfo.cpp` の 226 行で定義されています。

参照先 `clear()`.

4.4.2.2 `cpm_GlobalDomainInfo::~cpm_GlobalDomainInfo () [virtual]`

デストラクタ

`cpm_DomainInfo.cpp` の 234 行で定義されています。

4.4.3 関数

4.4.3.1 `bool cpm_GlobalDomainInfo::AddSubDomain (cpm_ActiveSubDomainInfo subDomain)`

活性サブドメイン情報の追加

引数

<code>in</code>	<code>subDomain</code>	追加する活性サブドメイン情報
-----------------	------------------------	----------------

戻り値

<code>true</code>	追加した
<code>false</code>	追加に失敗 (同じ領域分割位置で追加済み)

`cpm_DomainInfo.cpp` の 285 行で定義されています。

参照先 `IsExistSubDomain()`, と `m_subDomainInfo`.

参照元 `cpm_TextParserDomain::ReadSubdomainInfo()`, と `cpm_ParaManager::VoxellInit()`.

4.4.3.2 `void cpm_GlobalDomainInfo::clear () [virtual]`

情報のクリア

`cpm_DomainInfo` を再定義しています。

`cpm_DomainInfo.cpp` の 241 行で定義されています。

参照先 `m_divNum`, と `m_subDomainInfo`.

参照元 `cpm_GlobalDomainInfo()`.

4.4.3.3 `const int * cpm_GlobalDomainInfo::GetDivNum () const`

領域分割数の取得

戻り値

領域分割数整数配列のポインタ

`cpm_DomainInfo.cpp` の 264 行で定義されています。

参照先 `m_divNum`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_VoxelInfo::CreateNeighborRankInfo()`, `cpm_VoxelInfo::CreateRankMap()`, `cpm_VoxelInfo::GetDivNum()`, と `cpm_TextParserDomain::ReadSubdomainInfo()`.

4.4.3.4 `const cpm_ActiveSubDomainInfo * cpm_GlobalDomainInfo::GetSubDomainInfo (size_t idx) const`

活性サブドメイン情報を取得

引数

<code>in</code>	<code>idx</code>	登録順番号
-----------------	------------------	-------

戻り値

活性サブドメイン情報ポインタ

`cpm_DomainInfo.cpp` の 306 行で定義されています。

参照先 `GetSubDomainNum()`, と `m_subDomainInfo`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, と `cpm_VoxelInfo::CreateRankMap()`.

4.4.3.5 `int cpm_GlobalDomainInfo::GetSubDomainNum () const`

活性サブドメインの数を取得

戻り値

活性サブドメインの数

`cpm_DomainInfo.cpp` の 298 行で定義されています。

参照先 `m_subDomainInfo`.

参照元 `cpm_VoxelInfo::CreateLocalDomainInfo()`, `cpm_VoxelInfo::CreateRankMap()`, `GetSubDomainInfo()`, と `cpm_ParaManager::VoxelInit()`.

4.4.3.6 `bool cpm_GlobalDomainInfo::IsExistSubDomain (cpm_ActiveSubDomainInfo subDomain)`

活性サブドメイン情報の存在チェック

引数

<code>in</code>	<code>subDomain</code>	チェックする活性サブドメイン情報
-----------------	------------------------	------------------

戻り値

<i>true</i>	存在する
<i>false</i>	存在しない

cpm_DomainInfo.cpp の 272 行で定義されています。

参照先 m_subDomainInfo.

参照元 AddSubDomain().

4.4.3.7 void cpm_GlobalDomainInfo::SetDivNum (int *div*[3])

領域分割数のセット

引数

<i>in</i>	<i>div</i>	領域分割数
-----------	------------	-------

cpm_DomainInfo.cpp の 254 行で定義されています。

参照先 m_divNum.

参照元 cpm_TextParserDomain::ReadDomainInfo(), と cpm_ParaManager::VoxelInit().

4.4.4 変数

4.4.4.1 int cpm_GlobalDomainInfo::m_divNum[3] [private]

領域分割数

cpm_DomainInfo.h の 237 行で定義されています。

参照元 clear(), GetDivNum(), と SetDivNum().

4.4.4.2 std::vector<cpm_ActiveSubDomainInfo> cpm_GlobalDomainInfo::m_subDomainInfo [private]

活性サブドメイン情報

cpm_DomainInfo.h の 238 行で定義されています。

参照元 AddSubDomain(), clear(), GetSubDomainInfo(), GetSubDomainNum(), と IsExistSubDomain().

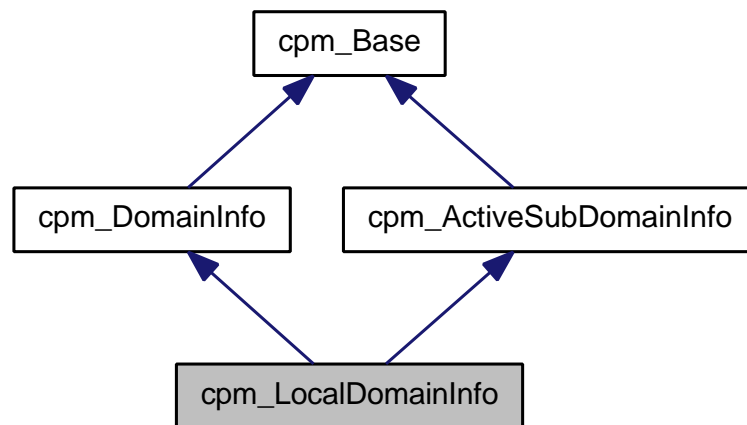
このクラスの説明は次のファイルから生成されました:

- [cpm_DomainInfo.h](#)
- [cpm_DomainInfo.cpp](#)

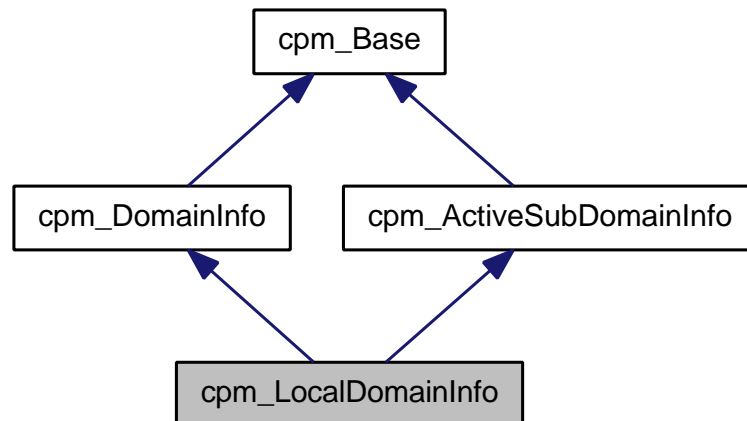
4.5 クラス cpm_LocalDomainInfo

```
#include <cpm_DomainInfo.h>
```


`cpm_LocalDomainInfo` に対する継承グラフ



`cpm_LocalDomainInfo` のコラボレーション図



Public メソッド

- [cpm_LocalDomainInfo](#) ()
- virtual [~cpm_LocalDomainInfo](#) ()
- virtual void [clear](#) ()

4.5.1 説明

CPM のローカル領域情報クラス

`cpm_DomainInfo.h` の 244 行で定義されています。

4.5.2 コンストラクタとデストラクタ

4.5.2.1 `cpm_LocalDomainInfo::cpm_LocalDomainInfo` ()

コンストラクタ

`cpm_DomainInfo.cpp` の 314 行で定義されています。

4.5.2.2 `cpm_LocalDomainInfo::~~cpm_LocalDomainInfo ()` [virtual]

デストラクタ

`cpm_DomainInfo.cpp` の 321 行で定義されています。

4.5.3 関数

4.5.3.1 `void cpm_LocalDomainInfo::clear ()` [virtual]

情報のクリア

`cpm_DomainInfo` を再定義しています。

`cpm_DomainInfo.cpp` の 328 行で定義されています。

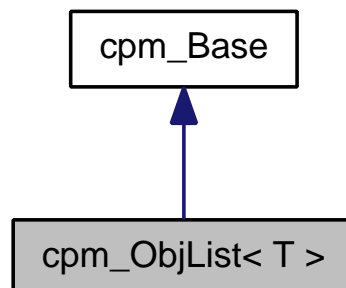
このクラスの説明は次のファイルから生成されました:

- [cpm_DomainInfo.h](#)
- [cpm_DomainInfo.cpp](#)

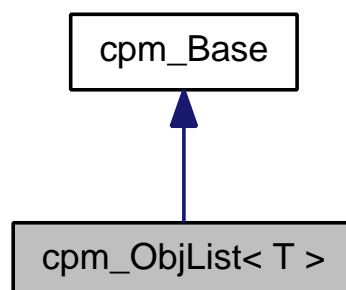
4.6 クラス テンプレート `cpm_ObjList< T >`

```
#include <cpm_ObjList.h>
```

`cpm_ObjList< T >` に対する継承グラフ



`cpm_ObjList< T >` のコラボレーション図



Public メソッド

- [cpm_ObjList \(\)](#)
- [~cpm_ObjList \(\)](#)

- `T * Create ()`
- `int Add (T *obj)`
- `cpm_ErrorCode Delete (int key)`
- `T * Get (int key)`

Private 型

- `typedef std::map< int, void * > ObjectMap`
- `typedef std::list< int > DelKeyList`

Private 変数

- `ObjectMap m_ObjectMap`
- `DelKeyList m_DelKeyList`
- `int m_newKey`

4.6.1 説明

```
template<class T>class cpm_ObjList< T >
```

CPM の汎用オブジェクト管理クラス

`cpm_ObjList.h` の 29 行で定義されています。

4.6.2 型定義

```
4.6.2.1 template<class T> typedef std::list<int> cpm_ObjList< T >::DelKeyList [private]
```

削除済み登録番号のリスト

`cpm_ObjList.h` の 43 行で定義されています。

```
4.6.2.2 template<class T> typedef std::map<int, void*> cpm_ObjList< T >::ObjectMap [private]
```

オブジェクトのマップ

`cpm_ObjList.h` の 39 行で定義されています。

4.6.3 コンストラクタとデストラクタ

```
4.6.3.1 template<class T> cpm_ObjList< T >::cpm_ObjList ( ) [inline]
```

コンストラクタ

`cpm_ObjList.h` の 56 行で定義されています。

```
4.6.3.2 template<class T> cpm_ObjList< T >::~~cpm_ObjList ( ) [inline]
```

デストラクタ

`cpm_ObjList.h` の 64 行で定義されています。

4.6.4 関数

4.6.4.1 `template<class T> int cpm_ObjList< T >::Add (T * obj) [inline]`

オブジェクトの追加

引数

<i>in</i>	<i>obj</i>	追加するオブジェクト
-----------	------------	------------

戻り値

登録番号 (負のとき登録失敗)

cpm_ObjList.h の 89 行で定義されています。

参照元 cpm_ParaManager::cpm_BndCommS3D_nowait(), cpm_ParaManager::cpm_BndCommS4D_nowait(), cpm_ParaManager::cpm_BndCommS4DEx_nowait(), cpm_ParaManager::cpm_BndCommV3D_nowait(), cpm_ParaManager::cpm_BndCommV3DEx_nowait(), cpm_ParaManager::cpm_Irecv(), と cpm_ParaManager::cpm_Isend().

4.6.4.2 `template<class T> T* cpm_ObjList< T >::Create () [inline]`

オブジェクトの生成 デフォルトコンストラクタが必要

戻り値

生成したオブジェクトのポインタ

cpm_ObjList.h の 79 行で定義されています。

参照元 cpm_ParaManager::cpm_BndCommS3D_nowait(), cpm_ParaManager::cpm_BndCommS4D_nowait(), cpm_ParaManager::cpm_BndCommS4DEx_nowait(), cpm_ParaManager::cpm_BndCommV3D_nowait(), cpm_ParaManager::cpm_BndCommV3DEx_nowait(), cpm_ParaManager::cpm_Irecv(), と cpm_ParaManager::cpm_Isend().

4.6.4.3 `template<class T> cpm_ErrorCode cpm_ObjList< T >::Delete (int key) [inline]`

オブジェクトの削除

引数

<i>in</i>	<i>key</i>	Add の戻り値である登録番号
-----------	------------	-----------------

戻り値

CPM 終了コード (0,CPM_SUCCESS=正常終了)

cpm_ObjList.h の 119 行で定義されています。

参照元 cpm_ParaManager::cpm_Wait(), cpm_ParaManager::cpm_wait_BndCommS3D(), cpm_ParaManager::cpm_wait_BndCommS4D(), cpm_ParaManager::cpm_wait_BndCommS4DEx(), cpm_ParaManager::cpm_wait_BndCommV3D(), cpm_ParaManager::cpm_wait_BndCommV3DEx(), と cpm_ParaManager::cpm_Waitall().

4.6.4.4 `template<class T> T* cpm_ObjList< T >::Get (int key) [inline]`

オブジェクトの取得

引数

in	key	Add の戻り値である登録番号
----	-----	-----------------

戻り値

オブジェクトのポインタ

cpm_ObjList.h の 138 行で定義されています。

参照元 cpm_ParaManager::cpm_Wait(), cpm_ParaManager::cpm_wait_BndCommS3D(), cpm_ParaManager::cpm_wait_BndCommS4D(), cpm_ParaManager::cpm_wait_BndCommS4DEx(), cpm_ParaManager::cpm_wait_BndCommV3D(), cpm_ParaManager::cpm_wait_BndCommV3DEx(), cpm_ParaManager::cpm_Waitall(), と cpm_ObjList< MPI_Request >::Delete().

4.6.5 変数

4.6.5.1 `template<class T> DelKeyList cpm_ObjList< T >::m_DelKeyList [private]`

cpm_ObjList.h の 44 行で定義されています。

参照元 cpm_ObjList< MPI_Request >::Add(), cpm_ObjList< MPI_Request >::cpm_ObjList(), cpm_ObjList< MPI_Request >::Delete(), と cpm_ObjList< MPI_Request >::~~cpm_ObjList().

4.6.5.2 `template<class T> int cpm_ObjList< T >::m_newKey [private]`

使用可能な登録番号

cpm_ObjList.h の 47 行で定義されています。

参照元 cpm_ObjList< MPI_Request >::Add(), と cpm_ObjList< MPI_Request >::cpm_ObjList().

4.6.5.3 `template<class T> ObjectMap cpm_ObjList< T >::m_ObjectMap [private]`

cpm_ObjList.h の 40 行で定義されています。

参照元 cpm_ObjList< MPI_Request >::Add(), cpm_ObjList< MPI_Request >::cpm_ObjList(), cpm_ObjList< MPI_Request >::Delete(), cpm_ObjList< MPI_Request >::Get(), と cpm_ObjList< MPI_Request >::~~cpm_ObjList().

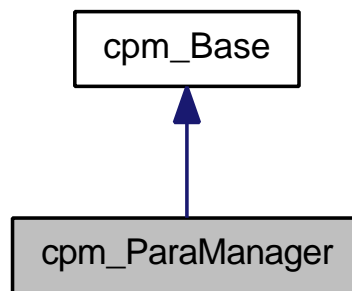
このクラスの説明は次のファイルから生成されました:

- [cpm_ObjList.h](#)

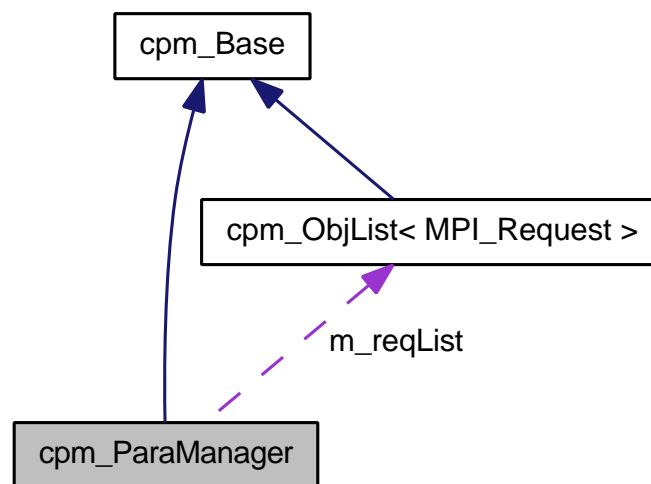
4.7 クラス cpm_ParaManager

```
#include <cpm_ParaManager.h>
```

cpm_ParaManager に対する継承グラフ



cpm_ParaManager のコラボレーション図



Public メソッド

- [cpm_ErrorCode Initialize \(\)](#)
- [cpm_ErrorCode Initialize \(int &argc, char **&argv\)](#)
- [bool IsParallel \(\)](#)
- [bool IsParallel \(\) const](#)
- [cpm_ErrorCode Voxellnit \(cpm_GlobalDomainInfo *domainInfo, size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [cpm_ErrorCode Voxellnit \(int div\[3\], int vox\[3\], REAL_TYPE origin\[3\], REAL_TYPE pitch\[3\], int obcid\[6\], size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [cpm_ErrorCode Voxellnit \(int vox\[3\], REAL_TYPE origin\[3\], REAL_TYPE pitch\[3\], int obcid\[6\], size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [int CreateProcessGroup \(int nproc, int *proclist, int parentProcGrpNo=0\)](#)
- [const cpm_VoxelInfo * FindVoxelInfo \(int procGrpNo=0\)](#)
- [const int * GetDivNum \(int procGrpNo=0\)](#)
- [const REAL_TYPE * GetPitch \(int procGrpNo=0\)](#)
- [const int * GetGlobalVoxelSize \(int procGrpNo=0\)](#)
- [const REAL_TYPE * GetGlobalOrigin \(int procGrpNo=0\)](#)
- [const REAL_TYPE * GetGlobalRegion \(int procGrpNo=0\)](#)
- [const int * GetLocalVoxelSize \(int procGrpNo=0\)](#)
- [const REAL_TYPE * GetLocalOrigin \(int procGrpNo=0\)](#)
- [const REAL_TYPE * GetLocalRegion \(int procGrpNo=0\)](#)

- const int * [GetDivPos](#) (int procGrpNo=0)
- const int * [GetBCID](#) (int procGrpNo=0)
- const int * [GetVoxelHeadIndex](#) (int procGrpNo=0)
- const int * [GetVoxelTailIndex](#) (int procGrpNo=0)
- const int * [GetNeighborRankID](#) (int procGrpNo=0)
- const int * [GetPeriodicRankID](#) (int procGrpNo=0)
- int [GetMyRankID](#) (int procGrpNo=0)
- int [GetNumRank](#) (int procGrpNo=0)
- MPI_Comm [GetMPI_Comm](#) (int procGrpNo=0)
- void [Abort](#) (int errorcode)
- [cpm_ErrorCode Barrier](#) (int procGrpNo=0)
- [cpm_ErrorCode Wait](#) (MPI_Request *request)
- [cpm_ErrorCode Waitall](#) (int count, MPI_Request requests[])
- template<class T >
[CPM_INLINE cpm_ErrorCode Bcast](#) (T *buf, int count, int root, int procGrpNo=0)
- [cpm_ErrorCode Bcast](#) (MPI_Datatype dtype, void *buf, int count, int root, int procGrpNo=0)
- template<class T >
[CPM_INLINE cpm_ErrorCode Send](#) (T *buf, int count, int dest, int procGrpNo=0)
- [cpm_ErrorCode Send](#) (MPI_Datatype dtype, void *buf, int count, int dest, int procGrpNo=0)
- template<class T >
[CPM_INLINE cpm_ErrorCode Recv](#) (T *buf, int count, int source, int procGrpNo=0)
- [cpm_ErrorCode Recv](#) (MPI_Datatype dtype, void *buf, int count, int source, int procGrpNo=0)
- template<class T >
[CPM_INLINE cpm_ErrorCode Isend](#) (T *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)
- [cpm_ErrorCode Isend](#) (MPI_Datatype dtype, void *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)
- template<class T >
[CPM_INLINE cpm_ErrorCode Irecv](#) (T *buf, int count, int source, MPI_Request *request, int procGrpNo=0)
- [cpm_ErrorCode Irecv](#) (MPI_Datatype dtype, void *buf, int count, int source, MPI_Request *request, int procGrpNo=0)
- template<class T >
[CPM_INLINE cpm_ErrorCode Allreduce](#) (T *sendbuf, T *recvbuf, int count, MPI_Op op, int procGrpNo=0)
- [cpm_ErrorCode Allreduce](#) (MPI_Datatype dtype, void *sendbuf, void *recvbuf, int count, MPI_Op op, int procGrpNo=0)
- template<class Ts , class Tr >
[CPM_INLINE cpm_ErrorCode Gather](#) (Ts *sendbuf, int sendcnt, Tr *recvbuf, int recvcnt, int root, int procGrpNo=0)
- [cpm_ErrorCode Gather](#) (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int recvcnt, int root, int procGrpNo=0)
- template<class Ts , class Tr >
[CPM_INLINE cpm_ErrorCode Allgather](#) (Ts *sendbuf, int sendcnt, Tr *recvbuf, int recvcnt, int procGrpNo=0)
- [cpm_ErrorCode Allgather](#) (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int recvcnt, int procGrpNo=0)
- template<class Ts , class Tr >
[CPM_INLINE cpm_ErrorCode Gatherv](#) (Ts *sendbuf, int sendcnt, Tr *recvbuf, int *recvcnts, int *displs, int root, int procGrpNo=0)
- [cpm_ErrorCode Gatherv](#) (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int *recvcnts, int *displs, int root, int procGrpNo=0)
- template<class Ts , class Tr >
[CPM_INLINE cpm_ErrorCode Allgatherv](#) (Ts *sendbuf, int sendcnt, Tr *recvbuf, int *recvcnts, int *displs, int procGrpNo=0)
- [cpm_ErrorCode Allgatherv](#) (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int *recvcnts, int *displs, int procGrpNo=0)
- [cpm_ErrorCode cpm_Wait](#) (int reqNo)
- [cpm_ErrorCode cpm_Waitall](#) (int count, int reqNoList[])
- [cpm_ErrorCode cpm_Isend](#) (void *buf, int count, int datatype, int dest, int *reqNo, int procGrpNo=0)

- [cpm_ErrorCode cpm_lrecv](#) (void *buf, int count, int datatype, int source, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_BndCommsS3D_nowait](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_BndCommV3D_nowait](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_BndCommsS4D_nowait](#) (void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_wait_BndCommsS3D](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_wait_BndCommV3D](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_wait_BndCommsS4D](#) (void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_BndCommV3DEx_nowait](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_BndCommsS4DEx_nowait](#) (void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_wait_BndCommV3DEx](#) (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode cpm_wait_BndCommsS4DEx](#) (void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- [cpm_ErrorCode SetBndCommBuffer](#) (size_t maxVC, size_t maxN, int procGrpNo=0)
- [size_t GetBndCommBufferSize](#) (int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommsS3D](#) (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- [cpm_ErrorCode BndCommsS3D](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommV3D](#) (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- [cpm_ErrorCode BndCommV3D](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommsS4D](#) (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0)
- [cpm_ErrorCode BndCommsS4D](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommsS3D_nowait](#) (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- [cpm_ErrorCode BndCommsS3D_nowait](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommV3D_nowait](#) (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- [cpm_ErrorCode BndCommV3D_nowait](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode BndCommsS4D_nowait](#) (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- [cpm_ErrorCode BndCommsS4D_nowait](#) (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
[CPM_INLINE cpm_ErrorCode wait_BndCommsS3D](#) (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)

- `cpm_ErrorCode wait_BndCommS3D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode wait_BndCommV3D` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode wait_BndCommV3D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode wait_BndCommS4D` (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode wait_BndCommS4D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode PeriodicCommS3D` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `cpm_ErrorCode PeriodicCommS3D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode PeriodicCommV3D` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `cpm_ErrorCode PeriodicCommV3D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode PeriodicCommS4D` (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `cpm_ErrorCode PeriodicCommS4D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode BndCommV3DEx` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `cpm_ErrorCode BndCommV3DEx` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode BndCommS4DEx` (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `cpm_ErrorCode BndCommS4DEx` (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode BndCommV3DEx_nowait` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode BndCommV3DEx_nowait` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode BndCommS4DEx_nowait` (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode BndCommS4DEx_nowait` (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode wait_BndCommV3DEx` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode wait_BndCommV3DEx` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode wait_BndCommS4DEx` (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode wait_BndCommS4DEx` (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)

- `template<class T >`
`CPM_INLINE cpm_ErrorCode PeriodicCommV3DEx` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_`,
`comm`, `cpm_DirFlag dir`, `cpm_PMFlag pm`, `int procGrpNo=0`)
- `cpm_ErrorCode PeriodicCommV3DEx` (`MPI_Datatype dtype`, `void *array`, `int imax`, `int jmax`, `int kmax`, `int vc`,
`int vc_comm`, `cpm_DirFlag dir`, `cpm_PMFlag pm`, `int procGrpNo=0`)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode PeriodicCommS4DEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`,
`int vc_comm`, `cpm_DirFlag dir`, `cpm_PMFlag pm`, `int procGrpNo=0`)
- `cpm_ErrorCode PeriodicCommS4DEx` (`MPI_Datatype dtype`, `void *array`, `int nmax`, `int imax`, `int jmax`, `int`
`kmax`, `int vc`, `int vc_comm`, `cpm_DirFlag dir`, `cpm_PMFlag pm`, `int procGrpNo=0`)
- `template<class T >`
`void InitArray` (`T *array`, `size_t size`)
- `template<class T >`
`void CopyArray` (`T *source`, `T *dist`, `size_t size`)
- `REAL_TYPE * AllocRealS3D` (`int vc`, `int procGrpNo=0`)
- `double * AllocDoubleS3D` (`int vc`, `int procGrpNo=0`)
- `float * AllocFloatS3D` (`int vc`, `int procGrpNo=0`)
- `int * AllocIntS3D` (`int vc`, `int procGrpNo=0`)
- `REAL_TYPE * AllocRealV3D` (`int vc`, `int procGrpNo=0`)
- `double * AllocDoubleV3D` (`int vc`, `int procGrpNo=0`)
- `float * AllocFloatV3D` (`int vc`, `int procGrpNo=0`)
- `int * AllocIntV3D` (`int vc`, `int procGrpNo=0`)
- `REAL_TYPE * AllocRealV3DEx` (`int vc`, `int procGrpNo=0`)
- `double * AllocDoubleV3DEx` (`int vc`, `int procGrpNo=0`)
- `float * AllocFloatV3DEx` (`int vc`, `int procGrpNo=0`)
- `int * AllocIntV3DEx` (`int vc`, `int procGrpNo=0`)
- `REAL_TYPE * AllocRealS4D` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `double * AllocDoubleS4D` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `float * AllocFloatS4D` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `int * AllocIntS4D` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `REAL_TYPE * AllocRealS4DEx` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `double * AllocDoubleS4DEx` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `float * AllocFloatS4DEx` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `int * AllocIntS4DEx` (`int nmax`, `int vc`, `int procGrpNo=0`)
- `void flush` (`std::ostream &out`, `int procGrpNo=0`)
- `void flush` (`FILE *fp`, `int procGrpNo=0`)

Static Public メソッド

- `static cpm_ParaManager * get_instance` ()
- `static cpm_ParaManager * get_instance` (`int &argc`, `char **&argv`)
- `template<class T >`
`static CPM_INLINE MPI_Datatype GetMPI_Datatype` (`T *ptr`)
- `static MPI_Datatype GetMPI_Datatype` (`int datatype`)
- `static MPI_Op GetMPI_Op` (`int op`)

Private メソッド

- `cpm_ParaManager` ()
- `virtual ~cpm_ParaManager` ()
- `cpm_ErrorCode DecideDivPattern` (`int divNum`, `int voxSize[3]`, `int divPttn[3]`) `const`
- `unsigned long long CalcCommSize` (`unsigned long long iDiv`, `unsigned long long jDiv`, `unsigned long long`
`kDiv`, `unsigned long long voxsize[3]`) `const`
- `CPM_INLINE S_BND COMM_BUFFER * GetBndCommBuffer` (`int procGrpNo=0`)

- `template<class T >`
`cpm_ErrorCode packX` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackX` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode packY` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackY` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode packZ` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackZ` (`T *array`, `int imax`, `int jmax`, `int kmax`, `int nmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode packXEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackXEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode packYEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackYEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode packZEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *sendm`, `T *sendp`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackZEx` (`T *array`, `int nmax`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `T *recv`, `T *recv`, `int nIDm`, `int nIDp`)
- `template<class T >`
`cpm_ErrorCode sendrecv` (`T *sendm`, `T *recv`, `T *sendp`, `T *recv`, `size_t nw`, `MPI_Request *req`, `int nIDm`, `int nIDr`, `int nIDp`, `int nIDr`, `int nIDp`, `int nIDp`, `int nIDp`, `int nIDp`, `int nIDp`)

Private 変数

- `int m_nRank`
- `int m_rankNo`
- `std::vector< MPI_Comm > m_procGrpList`
- `VoxelInfoMap m_voxelInfoMap`
- `RankNoMap m_rankNoMap`
- `BndCommInfoMap m_bndCommInfoMap`
- `cpm_ObjList< MPI_Request > m_reqList`

4.7.1 説明

CPM の並列管理クラス

- 現時点ではユーザがインスタンスすることを許していない

- `get_instance` 静的関数を用いて唯一のインスタンスを取得する

`cpm_ParaManager.h` の 74 行で定義されています。

4.7.2 コンストラクタとデストラクタ

4.7.2.1 `cpm_ParaManager::cpm_ParaManager()` [private]

コンストラクタ

`cpm_ParaManager.cpp` の 48 行で定義されています。

参照先 `m_bndCommInfoMap`, `m_nRank`, `m_procGrpList`, `m_rankNo`, `m_rankNoMap`, と `m_voxelInfoMap`.

4.7.2.2 `cpm_ParaManager::~cpm_ParaManager()` [private, virtual]

デストラクタ

`cpm_ParaManager.cpp` の 69 行で定義されています。

参照先 `m_bndCommInfoMap`, `m_procGrpList`, `m_rankNoMap`, と `m_voxelInfoMap`.

4.7.3 関数

4.7.3.1 `void cpm_ParaManager::Abort(int errorcode)`

Abort

- `MPI_Abort` のインターフェイス

引数

in	<i>errorcode</i>	<code>MPI_Abort</code> に渡すエラーコード
----	------------------	----------------------------------

`cpm_ParaManager_MPI.cpp` の 146 行で定義されています。

参照元 `cpm_Abort()`, と `VoxelInit()`.

4.7.3.2 `template<class Ts, class Tr> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Allgather(Ts * sendbuf, int sendcnt, Tr * recvbuf, int recvcnt, int procGrpNo = 0)`

Allgather

- `MPI_Allgather` のインターフェイス

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	受信データのサイズ
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_ParaManager_inline.h` の 202 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allgather_().

4.7.3.3 **cpm_ErrorCode cpm_ParaManager::Allgather** (MPI_Datatype *stype*, void * *sendbuf*, int *sendcnt*, MPI_Datatype *rtype*, void * *recvbuf*, int *recvcnt*, int *procGrpNo* = 0)

Allgather

- MPI_Allgather のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>stype</i>	送信データのMPI_Datatype
in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>rtype</i>	受信データのMPI_Datatype
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	送信データのサイズ
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 432 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_ALLGATHER, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.4 **template<class Ts, class Tr > CPM_INLINE cpm_ErrorCode cpm_ParaManager::Allgatherv** (Ts * *sendbuf*, int *sendcnt*, Tr * *recvbuf*, int * *recvcnts*, int * *displs*, int *procGrpNo* = 0)

Allgatherv

- MPI_Allgatherv のインターフェイス

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 246 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allgatherv_().

4.7.3.5 `cpm_ErrorCode cpm_ParaManager::Allgatherv (MPI_Datatype stype, void * sendbuf, int sendcnt, MPI_Datatype rtype, void * recvbuf, int * recvcnts, int * displs, int procGrpNo = 0)`

Allgatherv

- MPI_Allgatherv のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>stype</i>	送信データのMPI_Datatype
in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>rtype</i>	受信データのMPI_Datatype
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 491 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_ALLGATHERV, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.6 `double * cpm_ParaManager::AllocDoubleS3D (int vc, int procGrpNo = 0)`

配列確保 double(imax,jmax,kmax)

引数

in	<i>vc</i>	仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 76 行で定義されています。

参照先 AllocDoubleS4D().

4.7.3.7 `double * cpm_ParaManager::AllocDoubleS4D (int nmax, int vc, int procGrpNo = 0)`

配列確保 double(imax,jmax,kmax,nmax)

引数

in	<i>vc</i>	仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 32 行で定義されています。

参照先 `GetLocalVoxelSize()`.

参照元 `AllocDoubleS3D()`, `AllocDoubleS4DEx()`, `AllocDoubleV3D()`, と `AllocDoubleV3DEx()`.

4.7.3.8 `double * cpm_ParaManager::AllocDoubleS4DEx (int nmax, int vc, int procGrpNo = 0)`

配列確保 `double(nmax,imax,jmax,kmax)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 172 行で定義されています。

参照先 `AllocDoubleS4D()`.

4.7.3.9 `double * cpm_ParaManager::AllocDoubleV3D (int vc, int procGrpNo = 0)`

配列確保 `double(imax,jmax,kmax,3)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 108 行で定義されています。

参照先 `AllocDoubleS4D()`.

4.7.3.10 `double * cpm_ParaManager::AllocDoubleV3DEx (int vc, int procGrpNo = 0)`

配列確保 `double(3,imax,jmax,kmax)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 140 行で定義されています。

参照先 AllocDoubleS4D().

4.7.3.11 float * cpm_ParaManager::AllocFloatS3D (int *vc*, int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 84 行で定義されています。

参照先 AllocFloatS4D().

4.7.3.12 float * cpm_ParaManager::AllocFloatS4D (int *nmax*, int *vc*, int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax,nmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 44 行で定義されています。

参照先 GetLocalVoxelSize().

参照元 AllocFloatS3D(), AllocFloatS4DEx(), AllocFloatV3D(), と AllocFloatV3DEx().

4.7.3.13 float * cpm_ParaManager::AllocFloatS4DEx (int *nmax*, int *vc*, int *procGrpNo* = 0)

配列確保 float(nmax,imax,jmax,kmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 180 行で定義されています。

参照先 `AllocFloatS4D()`。

4.7.3.14 `float * cpm_ParaManager::AllocFloatV3D (int vc, int procGrpNo = 0)`

配列確保 `float(imax,jmax,kmax,3)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 116 行で定義されています。

参照先 `AllocFloatS4D()`。

4.7.3.15 `float * cpm_ParaManager::AllocFloatV3DEx (int vc, int procGrpNo = 0)`

配列確保 `float(3,imax,jmax,kmax)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 148 行で定義されています。

参照先 `AllocFloatS4D()`。

4.7.3.16 `int * cpm_ParaManager::AllocIntS3D (int vc, int procGrpNo = 0)`

配列確保 `int(imax,jmax,kmax)`

引数

<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 92 行で定義されています。

参照先 `AllocIntS4D()`。

4.7.3.17 int * cpm_ParaManager::AllocIntS4D (int nmax, int vc, int procGrpNo = 0)

配列確保 int(imax,jmax,kmax,nmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 56 行で定義されています。

参照先 GetLocalVoxelSize().

参照元 AllocIntS3D(), AllocIntS4DEx(), AllocIntV3D(), と AllocIntV3DEx().

4.7.3.18 int * cpm_ParaManager::AllocIntS4DEx (int nmax, int vc, int procGrpNo = 0)

配列確保 int(nmax,imax,jmax,kmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 188 行で定義されています。

参照先 AllocIntS4D().

4.7.3.19 int * cpm_ParaManager::AllocIntV3D (int vc, int procGrpNo = 0)

配列確保 int(imax,jmax,kmax,3)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 124 行で定義されています。

参照先 AllocIntS4D().

4.7.3.20 int * cpm_ParaManager::AllocIntV3DEx (int vc, int procGrpNo = 0)

配列確保 int(3,imax,jmax,kmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 156 行で定義されています。

参照先 AllocIntS4D().

4.7.3.21 REAL_TYPE * cpm_ParaManager::AllocRealS3D (int vc, int procGrpNo = 0)

配列確保 [REAL_TYPE\(imax,jmax,kmax\)](#)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 68 行で定義されています。

参照先 AllocRealS4D().

4.7.3.22 REAL_TYPE * cpm_ParaManager::AllocRealS4D (int nmax, int vc, int procGrpNo = 0)

配列確保 [REAL_TYPE\(imax,jmax,kmax,nmax\)](#)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 20 行で定義されています。

参照先 GetLocalVoxelSize(), と REAL_TYPE.

参照元 AllocRealS3D(), AllocRealS4DEx(), AllocRealV3D(), と AllocRealV3DEx().

4.7.3.23 REAL_TYPE * cpm_ParaManager::AllocRealS4DEx (int nmax, int vc, int procGrpNo = 0)

配列確保 [REAL_TYPE\(nmax,imax,jmax,kmax\)](#)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 164 行で定義されています。

参照先 AllocRealS4D().

4.7.3.24 `REAL_TYPE * cpm_ParaManager::AllocRealV3D (int vc, int procGrpNo = 0)`

配列確保 `REAL_TYPE(imax,jmax,kmax,3)`

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 100 行で定義されています。

参照先 AllocRealS4D().

4.7.3.25 `REAL_TYPE * cpm_ParaManager::AllocRealV3DEx (int vc, int procGrpNo = 0)`

配列確保 `REAL_TYPE(3,imax,jmax,kmax)`

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 132 行で定義されています。

参照先 AllocRealS4D().

4.7.3.26 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::Allreduce (T * sendbuf, T * recvbuf, int count, MPI_Op op, int procGrpNo = 0)`

Allreduce

- MPI_Allreduce のインターフェイス

引数

in	sendbuf	送信データ
out	recvbuf	受信データ
in	count	送受信データのサイズ
in	op	オペレータ
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 163 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allreduce_().

4.7.3.27 cpm_ErrorCode cpm_ParaManager::Allreduce (MPI_Datatype dtype, void * sendbuf, void * recvbuf, int count, MPI_Op op, int procGrpNo = 0)

Allreduce

- MPI_Allreduce のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	dtype	送信データのMPI_Datatype
in	sendbuf	送信データ
out	recvbuf	受信データ
in	count	送受信データのサイズ
in	op	オペレータ
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 376 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_ALLREDUCE, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.28 cpm_ErrorCode cpm_ParaManager::Barrier (int procGrpNo = 0)

Barrier

- MPI_Barrier のインターフェイス

引数

in	procGrpNo	プロセスグループ番号 (省略時=0)
----	-----------	--------------------

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 156 行で定義されています。

参照先 CPM_ERROR_MPI_BARRIER, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

参照元 cpm_Barrier_(), と flush().

4.7.3.29 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Bcast (T * buf, int count, int root, int procGrpNo = 0)`

Bcast

- MPI_Bcast のインターフェイス

引数

	<i>inout]</i>	buf 送受信バッファ
<i>in</i>	<i>count</i>	送信バッファのサイズ (ワード数)
<i>in</i>	<i>root</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 78 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Bcast_().

4.7.3.30 `cpm_ErrorCode cpm_ParaManager::Bcast (MPI_Datatype dtype, void * buf, int count, int root, int procGrpNo = 0)`

Bcast

- MPI_Bcast のインターフェイス
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	送信バッファのMPI_Datatype
	<i>inout]</i>	buf 送受信バッファ
<i>in</i>	<i>count</i>	送信バッファのサイズ (ワード数)
<i>in</i>	<i>root</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 234 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_BCAST, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.31 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndComms3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 43 行で定義されています。

参照先 BndCommS4D().

参照元 cpm_BndCommS3D_().

4.7.3.32 cpm_ErrorCode cpm_ParaManager::BndCommS3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)

袖通信 (Scalar3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax) の形式の配列の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 520 行で定義されています。

参照先 BndCommS4D().

4.7.3.33 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS3D_nowait (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)

非同期版袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS3D をコールする

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 165 行で定義されています。

参照先 BndCommS4D_nowait().

参照元 cpm_BndCommS3D_nowait().

4.7.3.34 `cpm_ErrorCode cpm_ParaManager::BndCommS3D_nowait (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Scalar3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS3D をコールする

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 582 行で定義されています。

参照先 BndCommS4D_nowait().

4.7.3.35 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4D (T * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_BndComm.h` の 61 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, packX(), packY(), packZ(), sendrecv(), unpackX(), unpackY(), unpackZ(), Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 BndCommS3D(), BndCommS4D(), BndCommV3D(), と `cpm_BndCommS4D_()`.

4.7.3.36 `cpm_ErrorCode cpm_ParaManager::BndCommS4D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Scalar4D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 538 行で定義されています。

参照先 BndCommS4D(), と CPM_ERROR_MPI_INVALID_DATATYPE.

4.7.3.37 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4D_nowait (T * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4D をコールする

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 185 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, packX(), packY(), packZ(), sendrecv(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 BndCommS3D_nowait(), BndCommS4D_nowait(), BndCommV3D_nowait(), と cpm_BndCommS4D_nowait().

4.7.3.38 cpm_ErrorCode cpm_ParaManager::BndCommS4D_nowait (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)

非同期版袖通信 (Scalar4D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nmax) の形式の配列の非同期袖通信を行う
- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4D をコールする

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 600 行で定義されています。

参照先 BndCommS4D_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

4.7.3.39 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4DEx (T * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 58 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, packXEx(), packYEx(), packZEx(), sendrecv(), unpackXEx(), unpackYEx(), unpackZEx(), Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 BndCommS4DEx(), BndCommV3DEx(), と cpm_BndCommS4DEx_().

4.7.3.40 `cpm_ErrorCode cpm_ParaManager::BndCommS4DEx (MPI_Datatype dtype, void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Scalar4DEx 版, MPI_Datatype 指定)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 777 行で定義されています。

参照先 BndCommS4DEx(), と CPM_ERROR_MPI_INVALID_DATATYPE.

4.7.3.41 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4DEx_nowait (T * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4DEx をコールする

引数

in	array	袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	req	MPI リクエスト
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 171 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, packXEx(), packYEx(), packZEx(), sendrecv(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 BndCommS4DEx_nowait(), BndCommV3DEx_nowait(), と cpm_BndCommS4DEx_nowait().

4.7.3.42 `cpm_ErrorCode cpm_ParaManager::BndCommS4DEx_nowait (MPI_Datatype dtype, void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Scalar4DEx 版, MPI_Datatype 指定)

- (nmax,imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4DEx をコールする

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 830 行で定義されています。

参照先 BndCommsS4DEx_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

4.7.3.43 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 52 行で定義されています。

参照先 BndCommsS4D().

参照元 cpm_BndCommV3D_().

4.7.3.44 `cpm_ErrorCode cpm_ParaManager::BndCommV3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Vector3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 529 行で定義されています。

参照先 BndCommS4D().

4.7.3.45 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3D_nowait (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommV3D をコールする

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 175 行で定義されています。

参照先 BndCommS4D_nowait().

参照元 cpm_BndCommV3D_nowait().

4.7.3.46 `cpm_ErrorCode cpm_ParaManager::BndCommV3D_nowait (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Vector3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,3) の形式の配列の非同期袖通信を行う
- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す

- wait、展開は wait_BndCommV3D をコールする

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 591 行で定義されています。

参照先 BndCommsS4D_nowait().

4.7.3.47 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3DEx (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 49 行で定義されています。

参照先 BndCommsS4DEx().

参照元 cpm_BndCommV3DEx_().

4.7.3.48 `cpm_ErrorCode cpm_ParaManager::BndCommV3DEx (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo = 0)`

袖通信 (Vector3DEx 版, MPI_Datatype 指定)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 768 行で定義されています。

参照先 BndCommS4DEx().

4.7.3.49 `template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3DEx_nowait (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommV3DEx をコールする

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 161 行で定義されています。

参照先 BndCommS4DEx_nowait().

参照元 cpm_BndCommV3DEx_nowait().

4.7.3.50 `cpm_ErrorCode cpm_ParaManager::BndCommV3DEx_nowait (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信 (Vector3DEx 版, MPI_Datatype 指定)

- (3,imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す

- wait、展開は wait_BndCommV3DEx をコールする

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 821 行で定義されています。

参照先 BndCommS4DEx_nowait().

4.7.3.51 unsigned long long cpm_ParaManager::CalcCommSize (unsigned long long *iDiv*, unsigned long long *jDiv*, unsigned long long *kDiv*, unsigned long long *voxsize[3]*) const [private]

I,J,K 分割を行った時の通信点数の総数を取得する

引数

in	<i>iDiv</i>	i 方向領域分割数
in	<i>jDiv</i>	j 方向領域分割数
in	<i>kDiv</i>	k 方向領域分割数
in	<i>voxSize</i>	空間全体のボクセル数

戻り値

袖通信点数

cpm_ParaManager.cpp の 445 行で定義されています。

参照元 DecideDivPattern().

4.7.3.52 template<class T> CPM_INLINE void cpm_ParaManager::CopyArray (T* *source*, T* *dist*, size_t *size*)

配列のコピー

引数

in	<i>source</i>	コピー元の配列のポインタ
out	<i>dist</i>	コピー先の配列のポインタ
in	<i>size</i>	配列サイズ

cpm_ParaManager_inline.h の 32 行で定義されています。

4.7.3.53 **cpm_ErrorCode cpm_ParaManager::cpm_BndCommS3D_nowait** (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_BndCommS3D_nowait

- BndCommS3D_nowait のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2696 行で定義されています。

参照先 cpm_ObjList< T >::Add(), BndCommS3D_nowait(), cpm_BndCommS4D_nowait(), CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_REGIST_OBJKEY, CPM_SUCCESS, cpm_ObjList< T >::Create(), GetMPI_Datatype(), と m_reqList.

参照元 cpm_BndCommS3D_nowait().

4.7.3.54 **cpm_ErrorCode cpm_ParaManager::cpm_BndCommS4D_nowait** (void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_BndCommS4D_nowait

- BndCommS4D_nowait のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	nmax	配列サイズ (成分数)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2780 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `BndCommS4D_nowait()`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, と `m_reqList`.

参照元 `cpm_BndCommS3D_nowait()`, `cpm_BndCommS4D_nowait_()`, と `cpm_BndCommV3D_nowait()`.

4.7.3.55 `cpm_ErrorCode cpm_ParaManager::cpm_BndCommS4DEx_nowait (void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)`

`cpm_BndCommS4DEx_nowait`

- `BndCommS4DEx_nowait` のインターフェイス
- Fortran インターフェイス用

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (<code>cpm_fparam.fi</code> 参照)
out	<i>reqNo</i>	リクエスト番号配列 (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2994 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `BndCommS4DEx_nowait()`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, と `m_reqList`.

参照元 `cpm_BndCommS4DEx_nowait_()`, と `cpm_BndCommV3DEx_nowait()`.

4.7.3.56 `cpm_ErrorCode cpm_ParaManager::cpm_BndCommV3D_nowait (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)`

`cpm_BndCommV3D_nowait`

- `BndCommV3D_nowait` のインターフェイス
- Fortran インターフェイス用

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (<code>cpm_fparam.fi</code> 参照)
out	<i>reqNo</i>	リクエスト番号配列 (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2738 行で定義されています。

参照先 cpm_ObjList< T >::Add(), BndCommV3D_nowait(), cpm_BndCommS4D_nowait(), CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_REGIST_OBJKEY, CPM_SUCCESS, cpm_ObjList< T >::Create(), GetMPI_Datatype(), と m_reqList.

参照元 cpm_BndCommV3D_nowait().

4.7.3.57 cpm_ErrorCode cpm_ParaManager::cpm_BndCommV3DEx_nowait (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_BndCommV3DEx_nowait

- BndCommV3DEx_nowait のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2952 行で定義されています。

参照先 cpm_ObjList< T >::Add(), BndCommV3DEx_nowait(), cpm_BndCommS4DEx_nowait(), CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_REGIST_OBJKEY, CPM_SUCCESS, cpm_ObjList< T >::Create(), GetMPI_Datatype(), と m_reqList.

参照元 cpm_BndCommV3DEx_nowait().

4.7.3.58 cpm_ErrorCode cpm_ParaManager::cpm_Irecv (void * buf, int count, int datatype, int source, int * reqNo, int procGrpNo = 0)

cpm_Irecv

- MPI_Irecv のインターフェイス
- Fortran インターフェイス用

引数

out	buf	受信データ
in	count	受信データのサイズ
in	datatype	受信データのデータタイプ (cpm_fparam.fi 参照)
in	source	送信元のランク番号 (procGrpNo 内でのランク番号)
out	reqNo	リクエスト番号 (Fortran 用)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2664 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, `Irecv()`, と `m_reqList`.

参照元 `cpm_Irecv()`.

4.7.3.59 `cpm_ErrorCode cpm_ParaManager::cpm_Isend (void * buf, int count, int datatype, int dest, int * reqNo, int procGrpNo = 0)`

`cpm_Isend`

- `MPI_Isend` のインターフェイス
- Fortran インターフェイス用

引数

in	<i>buf</i>	送信データ
in	<i>count</i>	送信データのサイズ
in	<i>datatype</i>	受信データのデータタイプ (<code>cpm_fparam.fi</code> 参照)
in	<i>dest</i>	送信先のランク番号 (<code>procGrpNo</code> 内でのランク番号)
out	<i>reqNo</i>	リクエスト番号 (Fortran 用)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2627 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, `Isend()`, と `m_reqList`.

参照元 `cpm_Isend()`.

4.7.3.60 `cpm_ErrorCode cpm_ParaManager::cpm_Wait (int reqNo)`

`cpm_Wait`

- `MPI_Wait` のインターフェイス
- Fortran インターフェイス用

引数

in	<i>reqNo</i>	リクエスト番号
----	--------------	---------

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2563 行で定義されています。

参照先 `CPM_ERROR_INVALID_OBJKEY`, `CPM_ERROR_MPI_WAIT`, `cpm_ObjList< T >::Delete()`, `cpm_ObjList< T >::Get()`, と `m_reqList`.

参照元 `cpm_Wait()`.

4.7.3.61 **cpm_ErrorCode cpm_ParaManager::cpm_wait_BndCommS3D** (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_wait_BndCommS3D

- wait_BndCommS3D のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2818 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_wait_BndCommS4D(), cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommS3D().

参照元 cpm_wait_BndCommS3D().

4.7.3.62 **cpm_ErrorCode cpm_ParaManager::cpm_wait_BndCommS4D** (void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_wait_BndCommS4D

- wait_BndCommS4D のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	nmax	配列サイズ (成分数)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2910 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommS4D().

参照元 cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D(), と cpm_wait_BndCommV3D().

4.7.3.63 cpm_ErrorCode cpm_ParaManager::cpm_wait_BndCommS4DEx (void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_wait_BndCommS4DEx

- wait_BndCommS4DEx のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 3078 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommS4DEx().

参照元 cpm_wait_BndCommS4DEx(), と cpm_wait_BndCommV3DEx().

4.7.3.64 cpm_ErrorCode cpm_ParaManager::cpm_wait_BndCommV3D (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_wait_BndCommV3D

- wait_BndCommV3D のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2864 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_wait_BndCommS4D(), cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommV3D().

参照元 cpm_wait_BndCommV3D_().

4.7.3.65 cpm_ErrorCode cpm_ParaManager::cpm_wait_BndCommV3DEx (void * array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int * reqNo, int procGrpNo = 0)

cpm_wait_BndCommV3DEx

- wait_BndCommV3DEx のインターフェイス
- Fortran インターフェイス用

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	datatype	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	reqNo	リクエスト番号配列 (サイズ 12)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 3032 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_wait_BndCommS4DEx(), cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommV3DEx().

参照元 cpm_wait_BndCommV3DEx_().

4.7.3.66 cpm_ErrorCode cpm_ParaManager::cpm_Waitall (int count, int reqNoList[])

cpm_Waitall

- MPI_Waitall のインターフェイス

引数

in	count	リクエストの数
in	reqNoList	リクエスト番号のリスト

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2586 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_WAITALL, CPM_SUCCESS, cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), と m_reqList.

参照元 cpm_Waitall().

4.7.3.67 `int cpm_ParaManager::CreateProcessGroup (int nproc, int * proclst, int parentProcGrpNo = 0)`

プロセスグループの作成

- ・ 指定されたプロセスリストを使用してプロセスグループを生成する

引数

in	<i>nproc</i>	使用するプロセスの数
in	<i>proclst</i>	使用するプロセスのリスト (親プロセスグループでのランク番号)
in	<i>parentProcGrpNo</i>	親とするプロセスグループ番号 (省略時 0)

戻り値

0 以上	生成されたプロセスグループ番号
-1	エラー

cpm_ParaManager.cpp の 473 行で定義されています。

参照先 GetMPI_Comm(), cpm_Base::IsCommNull(), と m_procGrpList.

4.7.3.68 `cpm_ErrorCode cpm_ParaManager::DecideDivPattern (int divNum, int voxSize[3], int divPttn[3]) const`
[private]

並列プロセス数から I,J,K 方向の分割数を取得する

引数

in	<i>divNum</i>	ランク数
in	<i>voxSize</i>	空間全体のボクセル数
out	<i>divPttn</i>	領域分割数

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 356 行で定義されています。

参照先 CalcCommSize(), CPM_ERROR_INVALID_PTR, CPM_ERROR_INVALID_VOXELSIZE, CPM_ERROR_MPI_DIMSCREATE, と CPM_SUCCESS.

参照元 Voxellnit().

4.7.3.69 `const cpm_VoxellInfo * cpm_ParaManager::FindVoxellInfo (int procGrpNo = 0)`

VOXEL 空間マップを検索

引数

in	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
----	------------------	--------------------

戻り値

VOXEL 空間情報ポインタ

cpm_ParaManager.cpp の 515 行で定義されています。

参照先 m_voxelInfoMap.

参照元 GetBCID(), GetDivNum(), GetDivPos(), GetGlobalOrigin(), GetGlobalRegion(), GetGlobalVoxelSize(), GetLocalOrigin(), GetLocalRegion(), GetLocalVoxelSize(), GetNeighborRankID(), GetPeriodicRankID(), GetPitch(), GetVoxelHeadIndex(), と GetVoxelTailIndex().

4.7.3.70 void cpm_ParaManager::flush (std::ostream & out, int procGrpNo = 0)

flush

cpm_ParaManager.cpp の 803 行で定義されています。

参照先 Barrier().

参照元 Initialize().

4.7.3.71 void cpm_ParaManager::flush (FILE * fp, int procGrpNo = 0)

flush

cpm_ParaManager.cpp の 816 行で定義されています。

参照先 Barrier().

4.7.3.72 template<class Ts, class Tr> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Gather (Ts * sendbuf, int sendcnt, Tr * recvbuf, int recvcnt, int root, int procGrpNo = 0)

Gather

- MPI_Gather のインターフェイス

引数

in	sendbuf	送信データ
in	sendcnt	送信データのサイズ
out	recvbuf	受信データ
in	recvcnt	送信データのサイズ
in	root	受信するランク番号 (procGrpNo 内でのランク番号)
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 180 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Gather_().

4.7.3.73 cpm_ErrorCode cpm_ParaManager::Gather (MPI_Datatype stype, void * sendbuf, int sendcnt, MPI_Datatype rtype, void * recvbuf, int recvcnt, int root, int procGrpNo = 0)

Gather

- `MPI_Gather` のインターフェイス
- `MPI_Datatype` を指定するバージョン

引数

in	<i>stype</i>	送信データのMPI_Datatype
in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>rtype</i>	受信データのMPI_Datatype
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	送信データのサイズ
in	<i>root</i>	受信するランク番号 (<code>procGrpNo</code> 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_ParaManager_MPI.cpp` の 403 行で定義されています。

参照先 `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_MPI_GATHER`, `CPM_ERROR_NOT_IN_PROCGROUP`, `CPM_SUCCESS`, `GetMPI_Comm()`, と `cpm_Base::IsCommNull()`.

4.7.3.74 `template<class Ts, class Tr> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Gatherv (Ts * sendbuf, int sendcnt, Tr * recvbuf, int * recvcnts, int * displs, int root, int procGrpNo = 0)`

Gatherv

- `MPI_Gatherv` のインターフェイス

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>root</i>	受信するランク番号 (<code>procGrpNo</code> 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_ParaManager_inline.h` の 224 行で定義されています。

参照先 `CPM_ERROR_MPI_INVALID_DATATYPE`, と `GetMPI_Datatype()`.

参照元 `cpm_Gatherv_()`.

4.7.3.75 `cpm_ErrorCode cpm_ParaManager::Gatherv (MPI_Datatype stype, void * sendbuf, int sendcnt, MPI_Datatype rtype, void * recvbuf, int * recvcnts, int * displs, int root, int procGrpNo = 0)`

Gatherv

- `MPI_Gatherv` のインターフェイス
- `MPI_Datatype` を指定するバージョン

引数

in	<i>stype</i>	送信データのMPI_Datatype
in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>rtype</i>	受信データのMPI_Datatype
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>root</i>	受信するランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 461 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_GATHERV, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.76 cpm_ParaManager * cpm_ParaManager::get_instance () [static]

唯一のインスタンスの取得

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 19 行で定義されています。

参照元 cpm_Abort_(), cpm_Allgather_(), cpm_Allgatherv_(), cpm_Allreduce_(), cpm_Barrier_(), cpm_Bcast_(), cpm_BndCommS3D_(), cpm_BndCommS3D_nowait_(), cpm_BndCommS4D_(), cpm_BndCommS4D_nowait_(), cpm_BndCommS4DEx_(), cpm_BndCommS4DEx_nowait_(), cpm_BndCommV3D_(), cpm_BndCommV3D_nowait_(), cpm_BndCommV3DEx_(), cpm_BndCommV3DEx_nowait_(), cpm_Gather_(), cpm_Gatherv_(), cpm_GetBCID_(), cpm_GetDivNum_(), cpm_GetDivPos_(), cpm_GetGlobalOrigin_(), cpm_GetGlobalRegion_(), cpm_GetGlobalVoxelSize_(), cpm_GetLocalOrigin_(), cpm_GetLocalRegion_(), cpm_GetLocalVoxelSize_(), cpm_GetMyRankID_(), cpm_GetNeighborRankID_(), cpm_GetNumRank_(), cpm_GetPeriodicRankID_(), cpm_GetPitch_(), cpm_GetVoxelHeadIndex_(), cpm_GetVoxelTailIndex_(), cpm_Initialize_(), cpm_Irecv_(), cpm_Isend_(), cpm_IsParallel_(), cpm_PeriodicCommS3D_(), cpm_PeriodicCommS4D_(), cpm_PeriodicCommS4DEx_(), cpm_PeriodicCommV3D_(), cpm_PeriodicCommV3DEx_(), cpm_Recv_(), cpm_Send_(), cpm_SetBndCommBuffer_(), cpm_Voxellnit_(), cpm_Voxellnit_nodiv_(), cpm_Wait_(), cpm_wait_BndCommS3D_(), cpm_wait_BndCommS4D_(), cpm_wait_BndCommS4DEx_(), cpm_wait_BndCommV3D_(), cpm_wait_BndCommV3DEx_(), と cpm_Waitall_().

4.7.3.77 cpm_ParaManager * cpm_ParaManager::get_instance (int & argc, char **& argv) [static]

唯一のインスタンスの取得 (initialize 処理も実行)

引数

in	<i>argc</i>	プログラム実行時引数の数
in	<i>argv</i>	プログラム実行時引数

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 31 行で定義されています。

参照先 CPM_SUCCESS, と Initialize().

4.7.3.78 const int * cpm_ParaManager::GetBCID (int *procGrpNo* = 0)

自ランクのBCID を取得

引数

in	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
----	------------------	--------------------

戻り値

自ランクのBCID 整数配列のポインタ

cpm_ParaManager.cpp の 633 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetBCID().

参照元 cpm_GetBCID_().

4.7.3.79 CPM_INLINE S_BNDCOMM_BUFFER* cpm_ParaManager::GetBndCommBuffer (int *procGrpNo* = 0) [inline, private]

袖通信バッファの取得

- ・ 袖通信バッファ情報の取得

引数

in	<i>procGrpNo</i>	プロセスグループ番号
----	------------------	------------

戻り値

袖通信バッファ情報のポインタ

cpm_ParaManager.h の 1780 行で定義されています。

参照先 m_bndCommInfoMap.

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndComm-BufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), wait_BndCommS4D(), と wait_BndCommS4DEx().

4.7.3.80 size_t cpm_ParaManager::GetBndCommBufferSize (int *procGrpNo* = 0)

袖通信バッファサイズの取得

- ・ 袖通信バッファとして確保されている配列サイズ (byte) を返す

引数

in	<i>procGrpNo</i>	プロセスグループ番号 (負の場合、全プロセスグループでのトータルを返す)
----	------------------	--------------------------------------

戻り値

バッファサイズ (byte)

cpm_ParaManager.cpp の 760 行で定義されています。

参照先 GetBndCommBuffer(), m_bndCommInfoMap, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, と REAL_BUF_TYPE.

4.7.3.81 `const int * cpm_ParaManager::GetDivNum (int procGrpNo = 0)`

領域分割数を取得

引数

<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号 (省略時=0)
-----------------	------------------------	--------------------

戻り値

領域分割数整数配列のポインタ

cpm_ParaManager.cpp の 525 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetDivNum().

参照元 cpm_GetDivNum_().

4.7.3.82 `const int * cpm_ParaManager::GetDivPos (int procGrpNo = 0)`

自ランクの領域分割位置を取得

引数

<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号 (省略時=0)
-----------------	------------------------	--------------------

戻り値

自ランクの領域分割位置整数配列のポインタ

cpm_ParaManager.cpp の 621 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetDivPos().

参照元 cpm_GetDivPos_().

4.7.3.83 `const REAL_TYPE * cpm_ParaManager::GetGlobalOrigin (int procGrpNo = 0)`

全体空間の原点を取得

引数

<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号 (省略時=0)
-----------------	------------------------	--------------------

戻り値

全体空間の原点実数配列のポインタ

cpm_ParaManager.cpp の 561 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetGlobalOrigin().

参照元 cpm_GetGlobalOrigin_().

4.7.3.84 `const REAL_TYPE * cpm_ParaManager::GetGlobalRegion (int procGrpNo = 0)`

全体空間サイズを取得

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

全体空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 573 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetGlobalRegion().

参照元 cpm_GetGlobalRegion_().

4.7.3.85 `const int * cpm_ParaManager::GetGlobalVoxelSize (int procGrpNo = 0)`

全体ボクセル数を取得

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

全体ボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 549 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetGlobalVoxelSize().

参照元 cpm_GetGlobalVoxelSize_().

4.7.3.86 `const REAL_TYPE * cpm_ParaManager::GetLocalOrigin (int procGrpNo = 0)`

自ランクの空間原点を取得

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

自ランクの空間原点実数配列のポインタ

cpm_ParaManager.cpp の 597 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetLocalOrigin().

参照元 cpm_GetLocalOrigin_(), と cpm_GetLocalRegion_().

4.7.3.87 `const REAL_TYPE * cpm_ParaManager::GetLocalRegion (int procGrpNo = 0)`

自ランクの空間サイズを取得

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

自ランクの空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 609 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetLocalRegion().

4.7.3.88 `const int * cpm_ParaManager::GetLocalVoxelSize (int procGrpNo = 0)`

自ランクのボクセル数を取得

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

ローカルボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 585 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetLocalVoxelSize().

参照元 AllocDoubleS4D(), AllocFloatS4D(), AllocIntS4D(), AllocRealS4D(), cpm_GetLocalVoxelSize_(), と SetBndCommBuffer().

4.7.3.89 `MPI_Comm cpm_ParaManager::GetMPI_Comm (int procGrpNo = 0)`

MPI コミュニケータの取得

- MPI_COMM_NULL が返ってきた場合は、1. プロセスグループが存在しない、2. プロセスグループに自ランクが含まれていない、のいずれか

引数

<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)
-----------	------------------	--------------------

戻り値

MPI コミュニケータ

cpm_ParaManager_MPI.cpp の 131 行で定義されています。

参照先 cpm_Base::getCommNull(), と m_procGrpList.

参照元 Allgather(), Allgatherv(), Allreduce(), Barrier(), Bcast(), CreateProcessGroup(), Gather(), Gatherv(), Irecv(), Isend(), Recv(), Send(), と VoxelInit().

4.7.3.90 `template<class T > CPM_INLINE MPI_Datatype cpm_ParaManager::GetMPI_Datatype (T * ptr)`
`[static]`

MPI_Datatype を取得

引数

<code>in</code>	<code>ptr</code>	取得したいデータのポインタ
-----------------	------------------	---------------

戻り値

`MPI_Datatype`

`cpm_ParaManager_inline.h` の 42 行で定義されています。

参照元 `Allgather()`, `Allgatherv()`, `Allreduce()`, `Bcast()`, `cpm_Allgather_()`, `cpm_Allgatherv_()`, `cpm_Allreduce_()`, `cpm_Bcast_()`, `cpm_BndCommsS3D_()`, `cpm_BndCommsS3D_nowait_()`, `cpm_BndCommsS4D_()`, `cpm_BndCommS4D_nowait_()`, `cpm_BndCommsS4DEx_()`, `cpm_BndCommsS4DEx_nowait_()`, `cpm_BndCommV3D_()`, `cpm_BndCommV3D_nowait_()`, `cpm_BndCommV3DEx_()`, `cpm_BndCommV3DEx_nowait_()`, `cpm_Gather_()`, `cpm_Gatherv_()`, `cpm_Irecv_()`, `cpm_Isend_()`, `cpm_PeriodicCommsS3D_()`, `cpm_PeriodicCommsS4D_()`, `cpm_PeriodicCommS4DEx_()`, `cpm_PeriodicCommV3D_()`, `cpm_PeriodicCommV3DEx_()`, `cpm_Recv_()`, `cpm_Send_()`, `cpm_wait_BndCommsS3D_()`, `cpm_wait_BndCommS4D_()`, `cpm_wait_BndCommS4DEx_()`, `cpm_wait_BndCommV3D_()`, `cpm_wait_BndCommV3DEx_()`, `Gather()`, `Gatherv()`, `Irecv()`, `Isend()`, `Recv()`, と `Send()`.

4.7.3.91 `MPI_Datatype cpm_ParaManager::GetMPI_Datatype (int datatype)` [static]

`MPI_Datatype` を取得

- Fortran データタイプから `MPI_Datatype` を取得

引数

<code>in</code>	<code>datatype</code>	取得したいデータのポインタ
-----------------	-----------------------	---------------

戻り値

`MPI_Datatype`

`cpm_ParaManager_MPI.cpp` の 20 行で定義されています。

参照先 `CPM_CHAR`, `CPM_DOUBLE`, `CPM_FLOAT`, `CPM_INT`, `CPM_LONG`, `CPM_LONG_DOUBLE`, `CPM_REAL`, `CPM_SHORT`, `CPM_UNSIGNED`, `CPM_UNSIGNED_CHAR`, `CPM_UNSIGNED_LONG`, `CPM_UNSIGNED_SHORT`, と `cpm_Base::ReallsDouble()`.

4.7.3.92 `MPI_Op cpm_ParaManager::GetMPI_Op (int op)` [static]

`MPI_Op` を取得

- Fortran オペレータタイプから `MPI_Op` を取得

引数

<code>in</code>	<code>op</code>	取得したいデータのポインタ
-----------------	-----------------	---------------

戻り値

`MPI_Op`

`cpm_ParaManager_MPI.cpp` の 54 行で定義されています。

参照先 `CPM_BAND`, `CPM_BOR`, `CPM_BXOR`, `CPM_LAND`, `CPM_LOR`, `CPM_LXOR`, `CPM_MAX`, `CPM_MIN`, `CPM_PROD`, と `CPM_SUM`.

参照元 `cpm_Allreduce_()`.

4.7.3.93 `int cpm_ParaManager::GetMyRankID (int procGrpNo = 0)`

ランク番号の取得

- MPI_PROC_NULL が返ってきた場合は、 1. プロセスグループが存在しない、 2. プロセスグループに自ランクが含まれていない、 のいずれか

引数

<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号 (省略時=0)
-----------------	------------------------	--------------------

戻り値

ランク番号

`cpm_ParaManager_MPI.cpp` の 75 行で定義されています。

参照先 `cpm_Base::getRankNull()`, `cpm_Base::isCommNull()`, と `m_procGrpList`.

参照元 `cpm_GetMyRankID_()`.

4.7.3.94 `const int * cpm_ParaManager::GetNeighborRankID (int procGrpNo = 0)`

自ランクの隣接ランク番号を取得

戻り値

自ランクの隣接ランク番号整数配列のポインタ

`cpm_ParaManager.cpp` の 669 行で定義されています。

参照先 `FindVoxelInfo()`, と `cpm_VoxelInfo::GetNeighborRankID()`.

参照元 `BndCommS4D()`, `BndCommS4D_nowait()`, `BndCommS4DEx()`, `BndCommS4DEx_nowait()`, `cpm_GetNeighborRankID_()`, `wait_BndCommS4D()`, と `wait_BndCommS4DEx()`.

4.7.3.95 `int cpm_ParaManager::GetNumRank (int procGrpNo = 0)`

ランク数の取得

- プロセスグループのランク数を取得する

引数

<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号 (省略時 0)
-----------------	------------------------	--------------------

戻り値

ランク数

`cpm_ParaManager_MPI.cpp` の 103 行で定義されています。

参照先 `cpm_Base::isCommNull()`, と `m_procGrpList`.

参照元 `cpm_GetNumRank_()`, と `VoxelInit()`.

4.7.3.96 `const int * cpm_ParaManager::GetPeriodicRankID (int procGrpNo = 0)`

自ランクの周期境界の隣接ランク番号を取得

戻り値

自ランクの周期境界の隣接ランク番号整数配列のポインタ

cpm_ParaManager.cpp の 681 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetPeriodicRankID().

参照元 cpm_GetPeriodicRankID_(), PeriodicComms4D(), と PeriodicComms4DEx().

4.7.3.97 const REAL_TYPE * cpm_ParaManager::GetPitch (int procGrpNo = 0)

ピッチを取得

引数

in	procGrpNo	プロセスグループ番号 (省略時=0)
----	-----------	--------------------

戻り値

ピッチ実数配列のポインタ

cpm_ParaManager.cpp の 537 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetPitch().

参照元 cpm_GetPitch_().

4.7.3.98 const int * cpm_ParaManager::GetVoxelHeadIndex (int procGrpNo = 0)

自ランクの始点VOXEL の全体空間でのインデクスを取得

- ・ 全体空間の先頭インデクスを 0 としたC 型のインデクス

戻り値

自ランクの始点インデクス整数配列のポインタ

cpm_ParaManager.cpp の 645 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetVoxelHeadIndex().

参照元 cpm_GetVoxelHeadIndex_().

4.7.3.99 const int * cpm_ParaManager::GetVoxelTailIndex (int procGrpNo = 0)

自ランクの終点VOXEL の全体空間でのインデクスを取得

- ・ 全体空間の先頭インデクスを 0 としたC 型のインデクス

戻り値

自ランクの終点インデクス整数配列のポインタ

cpm_ParaManager.cpp の 657 行で定義されています。

参照先 FindVoxelInfo(), と cpm_VoxelInfo::GetVoxelTailIndex().

参照元 cpm_GetVoxelTailIndex_().

4.7.3.100 `template<class T> CPM_INLINE void cpm_ParaManager::InitArray (T * array, size_t size)`

配列の初期化処理

引数

out	array	初期化する配列のポインタ
in	size	配列サイズ

cpm_ParaManager_inline.h の 22 行で定義されています。

4.7.3.101 `cpm_ErrorCode cpm_ParaManager::Initialize ()`

初期化処理 (MPI_Init は実行済みの場合)

- MPI_Init は既に実行済みである必要がある
- 並列数、自ランク番号を取得

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 117 行で定義されています。

参照先 CPM_ERROR_MPI, CPM_ERROR_NO_MPI_INIT, CPM_SUCCESS, flush(), IsParallel(), m_nRank, と m_rankNo.

参照元 cpm_Initialize_(), get_instance(), と Initialize().

4.7.3.102 `cpm_ErrorCode cpm_ParaManager::Initialize (int & argc, char **& argv)`

初期化処理 (MPI_Init も実行する)

- MPI_Init が実行されていない場合、実行する
- 並列数、自ランク番号を取得

引数

in	argc	プログラム実行時引数の数
in	argv	プログラム実行時引数

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 163 行で定義されています。

参照先 CPM_ERROR_MPI, Initialize(), m_nRank, と m_rankNo.

4.7.3.103 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Irecv (T * buf, int count, int source, MPI_Request * request, int procGrpNo = 0)`

Irecv

- MPI_Irecv のインターフェイス

引数

out	<i>buf</i>	受信データ
in	<i>count</i>	受信データのサイズ
in	<i>source</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
out	<i>request</i>	リクエストハンドル
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 146 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Irecv(), と sendrecv().

4.7.3.104 `cpm_ErrorCode cpm_ParaManager::Irecv (MPI_Datatype dtype, void * buf, int count, int source, MPI.Request * request, int procGrpNo = 0)`

Irecv

- MPI_Irecv のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	送信データのMPI_Datatype
out	<i>buf</i>	受信データ
in	<i>count</i>	受信データのサイズ
in	<i>source</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
out	<i>request</i>	リクエストハンドル
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 347 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_IRECV, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.105 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Isend (T * buf, int count, int dest, MPI.Request * request, int procGrpNo = 0)`

Isend

- MPI_Isend のインターフェイス

引数

in	<i>buf</i>	送信データ
in	<i>count</i>	送信データのサイズ
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
out	<i>request</i>	リクエストハンドル
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 129 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Isend(), と sendrecv().

4.7.3.106 `cpm_ErrorCode cpm_ParaManager::Isend (MPI_Datatype dtype, void * buf, int count, int dest, MPI.Request * request, int procGrpNo = 0)`

Isend

- MPI_Isend のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	送信データのMPI_Datatype
in	<i>buf</i>	送信データ
in	<i>count</i>	送信データのサイズ
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
out	<i>request</i>	リクエストハンドル
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 318 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_ISEND, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.107 `bool cpm_ParaManager::IsParallel ()`

並列実行であるかチェックする 並列実行であっても、並列数が 1 のときは false となる

戻り値

<i>true</i>	並列実行
<i>false</i>	逐次実行

cpm_ParaManager.cpp の 187 行で定義されています。

参照先 m_nRank.

参照元 cpm_IsParallel_(), と Initialize().

4.7.3.108 `bool cpm_ParaManager::IsParallel () const`

並列実行であるかチェックする (const)

- 並列実行であっても、並列数が 1 のときは false となる

戻り値

<i>true</i>	並列実行
<i>false</i>	逐次実行

cpm_ParaManager.cpp の 199 行で定義されています。

参照先 m_nRank.

4.7.3.109 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packX(T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のX 方向送信バッファのセット

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>sendm</i>	マイナス方向の送信バッファ
out	<i>sendp</i>	プラス方向の送信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 542 行で定義されています。

参照先 _IDX_S4D, _IDXFX, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommsS4D(), BndCommsS4D_nowait(), と PeriodicCommsS4D().

4.7.3.110 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packXEx(T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEx,Vector3DEx 版) のX 方向送信バッファのセット

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>sendm</i>	マイナス方向の送信バッファ
out	<i>sendp</i>	プラス方向の送信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 508 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4DEX(), BndCommS4DEX_nowait(), と PeriodicCommS4DEX().

4.7.3.111 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packY (T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のY 方向送信バッファのセット

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	nmax	配列サイズ (成分数)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	sendm	マイナス方向の送信バッファ
out	sendp	プラス方向の送信バッファ
in	nIDm	マイナス方向の隣接ランク番号
in	nIDp	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 602 行で定義されています。

参照先 _IDX_S4D, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4D(), BndCommS4D_nowait(), と PeriodicCommS4D().

4.7.3.112 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packYEx (T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEX,Vector3DEX 版) のY 方向送信バッファのセット

引数

in	array	袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	sendm	マイナス方向の送信バッファ
out	sendp	プラス方向の送信バッファ
in	nIDm	マイナス方向の隣接ランク番号
in	nIDp	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 568 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4DEx(), BndCommS4DEx_nowait(), と PeriodicCommS4DEx().

4.7.3.113 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packZ (T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のZ 方向送信バッファのセット

引数

in	array	袖通信をする配列の先頭ポインタ
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	nmax	配列サイズ (成分数)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	sendm	マイナス方向の送信バッファ
out	sendp	プラス方向の送信バッファ
in	nIDm	マイナス方向の隣接ランク番号
in	nIDp	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 662 行で定義されています。

参照先 _IDX_S4D, _IDXFZ, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4D(), BndCommS4D_nowait(), と PeriodicCommS4D().

4.7.3.114 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::packZEx (T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* sendm, T* sendp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEx,Vector3DEx 版) のZ 方向送信バッファのセット

引数

in	array	袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	sendm	マイナス方向の送信バッファ
out	sendp	プラス方向の送信バッファ
in	nIDm	マイナス方向の隣接ランク番号
in	nIDp	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 628 行で定義されています。

参照先 _IDX_S4DEX, _IDXFZ, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4DEx(), BndCommS4DEx_nowait(), と PeriodicCommS4DEx().

4.7.3.115 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommsS3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 372 行で定義されています。

参照先 PeriodicCommsS4D().

参照元 cpm_PeriodicCommsS3D_().

4.7.3.116 `cpm_ErrorCode cpm_ParaManager::PeriodicCommsS3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 706 行で定義されています。

参照先 PeriodicCommS4D().

4.7.3.117 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommS4D (T * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の周期境界方向の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
<i>in</i>	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 392 行で定義されています。

参照先 BOTH, CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_PERIODIC_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetPeriodicRankID(), cpm_Base::getRankNull(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, MINUS2PLUS, packX(), packY(), packZ(), PLUS2MINUS, sendrecv(), unpackX(), unpackY(), unpackZ(), Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_PeriodicCommS4D_(), PeriodicCommS3D(), PeriodicCommS4D(), と PeriodicCommV3D().

4.7.3.118 `cpm_ErrorCode cpm_ParaManager::PeriodicCommS4D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar4D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nmax) の形式の配列の周期境界方向の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)

in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 724 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4D().

4.7.3.119 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommS4DEx (T * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 358 行で定義されています。

参照先 BOTH, CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_PERIODIC_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetPeriodicRankID(), cpm_Base::getRankNull(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, MINUS2PLUS, packXEx(), packYEx(), packZEx(), PLUS2MINUS, sendrecv(), unpackXEx(), unpackYEx(), unpackZEx(), Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_PeriodicCommS4DEx_(), PeriodicCommS4DEx(), と PeriodicCommV3DEx().

4.7.3.120 `cpm_ErrorCode cpm_ParaManager::PeriodicCommS4DEx (MPI_Datatype dtype, void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar4DEx 版, MPI_Datatype 指定)

- (nmax,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- `MPI_Datatype` を指定するバージョン

引数

<code>in</code>	<code>dtype</code>	袖通信データのMPI_Datatype
	<code>inout[]</code>	array 袖通信をする配列の先頭ポインタ
<code>in</code>	<code>nmax</code>	配列サイズ (成分数)
<code>in</code>	<code>imax</code>	配列サイズ (I 方向)
<code>in</code>	<code>jmax</code>	配列サイズ (J 方向)
<code>in</code>	<code>kmax</code>	配列サイズ (K 方向)
<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>vc_comm</code>	通信する仮想セル数
<code>in</code>	<code>dir</code>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
<code>in</code>	<code>pm</code>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 936 行で定義されています。

参照先 `CPM_ERROR_MPI_INVALID_DATATYPE`, と `PeriodicCommS4DEx()`.

4.7.3.121 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommV3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の周期境界方向の袖通信を行う

引数

	<code>inout[]</code>	array 袖通信をする配列の先頭ポインタ
<code>in</code>	<code>imax</code>	配列サイズ (I 方向)
<code>in</code>	<code>jmax</code>	配列サイズ (J 方向)
<code>in</code>	<code>kmax</code>	配列サイズ (K 方向)
<code>in</code>	<code>vc</code>	仮想セル数
<code>in</code>	<code>vc_comm</code>	通信する仮想セル数
<code>in</code>	<code>dir</code>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
<code>in</code>	<code>pm</code>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
<code>in</code>	<code>procGrpNo</code>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_BndComm.h` の 382 行で定義されています。

参照先 `PeriodicCommS4D()`.

参照元 `cpm_PeriodicCommV3D_()`.

4.7.3.122 `cpm_ErrorCode cpm_ParaManager::PeriodicCommV3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Vector3D 版, `MPI_Datatype` 指定)

- (imax,jmax,kmax,3) の形式の配列の周期境界方向の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 715 行で定義されています。

参照先 PeriodicComms4D().

4.7.3.123 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommV3DEx (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 348 行で定義されています。

参照先 PeriodicComms4DEx().

参照元 cpm_PeriodicCommV3DEx_().

4.7.3.124 `cpm_ErrorCode cpm_ParaManager::PeriodicCommV3DEx (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Vector3DEx 版, MPI_Datatype 指定)

- (3,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 927 行で定義されています。

参照先 PeriodicCommsS4DEx().

4.7.3.125 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Recv (T * buf, int count, int source, int procGrpNo = 0)`

Recv

- MPI_Recv のインターフェイス

引数

out	<i>buf</i>	受信データ
in	<i>count</i>	受信データのサイズ
in	<i>source</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 112 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Recv().

4.7.3.126 `cpm_ErrorCode cpm_ParaManager::Recv (MPI_Datatype dtype, void * buf, int count, int source, int procGrpNo = 0)`

Recv

- MPI_Recv のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	送信データのMPI_Datatype
out	<i>buf</i>	受信データ
in	<i>count</i>	受信データのサイズ
in	<i>source</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 289 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_SEND, CPM_ERROR_NOT_IN_PROCGROUP, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

4.7.3.127 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::Send (T * buf, int count, int dest, int procGrpNo = 0)`

Send

- MPI_Send のインターフェイス

引数

in	<i>buf</i>	送信データ
in	<i>count</i>	送信データのサイズ
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 95 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Send_().

4.7.3.128 `cpm_ErrorCode cpm_ParaManager::Send (MPI_Datatype dtype, void * buf, int count, int dest, int procGrpNo = 0)`

Send

- MPI_Send のインターフェイス
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	送信データのMPI_Datatype
in	<i>buf</i>	送信データ
in	<i>count</i>	送信データのサイズ
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 261 行で定義されています。

参照先 `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_MPI_SEND`, `CPM_ERROR_NOT_IN_PROCGROUP`, `CPM_SUCCESS`, `GetMPI_Comm()`, と `cpm_Base::IsCommNull()`.

```
4.7.3.129 template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::sendrecv ( T * sendm, T * recvm,
T * sendp, T * recvp, size_t nw, MPI_Request * req, int nIDsm, int nIDrm, int nIDsp, int nIDrp, int procGrpNo = 0 )
[private]
```

1 方向 (プラス、マイナス) の双方向袖通信処理

引数

in	<i>sendm</i>	マイナス方向の送信バッファ
in	<i>sendp</i>	プラス方向の送信バッファ
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nw</i>	送受信サイズ
out	<i>req</i>	MPI_Request 配列のポインタ (サイズ 4)
in	<i>nIDsm</i>	マイナス方向受信用の隣接ランク番号
in	<i>nIDrm</i>	マイナス方向送信用の隣接ランク番号
in	<i>nIDsp</i>	プラス方向受信用の隣接ランク番号
in	<i>nIDrp</i>	プラス方向送信用の隣接ランク番号
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_BndComm.h` の 722 行で定義されています。

参照先 `CPM_SUCCESS`, `Irecv()`, `Isend()`, と `cpm_Base::IsRankNull()`.

参照元 `BndCommsS4D()`, `BndCommsS4D_nowait()`, `BndCommsS4DEx()`, `BndCommsS4DEx_nowait()`, `PeriodicCommsS4D()`, と `PeriodicCommsS4DEx()`.

```
4.7.3.130 cpm_ErrorCode cpm_ParaManager::SetBndCommBuffer ( size_t maxVC, size_t maxN, int procGrpNo = 0 )
```

袖通信バッファのセット

- 6face 分の送受信バッファを確保する

引数

in	<i>maxVC</i>	送受信バッファの最大袖数
in	<i>maxN</i>	送受信バッファの最大成分数
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 693 行で定義されています。

参照先 CPM_ERROR_BNDCOMM, CPM_ERROR_BNDCOMM_VOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetLocalVoxelSize(), m_bndCommInfoMap, S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_maxN, S_BNDCOMM_BUFFER::m_maxVC, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, と REAL_BUF_TYPE.

参照元 cpm_SetBndCommBuffer(), と VoxelInit().

4.7.3.131 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackX (T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のX 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 572 行で定義されています。

参照先 _IDX_S4D, _IDXFX, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4D(), PeriodicCommS4D(), と wait_BndCommS4D().

4.7.3.132 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackXEx (T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEx,Vector3DEx 版) のX 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 538 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4DEx(), PeriodicCommS4DEx(), と wait_BndCommS4DEx().

4.7.3.133 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackY (T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のY 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 632 行で定義されています。

参照先 _IDX_S4D, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4D(), PeriodicCommS4D(), と wait_BndCommS4D().

4.7.3.134 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackYEx (T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEx,Vector3DEx 版) のY 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 598 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4DEX(), PeriodicCommS4DEX(), と wait_BndCommS4DEX().

4.7.3.135 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackZ (T* array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar3D,4D,Vector3D 版) のZ 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 692 行で定義されています。

参照先 _IDX_S4D, _IDXFZ, CPM_SUCCESS, と cpm_Base::IsRankNull().

参照元 BndCommS4D(), PeriodicCommS4D(), と wait_BndCommS4D().

4.7.3.136 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::unpackZEx (T* array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T* recvm, T* recvp, int nIDm, int nIDp) [private]`

袖通信 (Scalar4DEX,Vector3DEX 版) のZ 方向受信バッファを元に戻す

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_BndCommEx.h` の 658 行で定義されています。

参照先 `_IDX_S4DEX`, `_IDXFZ`, `CPM_SUCCESS`, と `cpm_Base::IsRankNull()`.

参照元 `BndCommS4DEX()`, `PeriodicCommS4DEX()`, と `wait_BndCommS4DEX()`.

4.7.3.137 `cpm_ErrorCode cpm_ParaManager::Voxellnit (cpm_GlobalDomainInfo * domainInfo, size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割

- 既に作成済みの領域分割情報を用いた領域分割処理

引数

<code>in</code>	<code>domainInfo</code>	領域分割情報
<code>in</code>	<code>maxVC</code>	最大の袖数 (袖通信用)
<code>in</code>	<code>maxN</code>	最大の成分数 (袖通信用)
<code>in</code>	<code>procGrpNo</code>	領域分割を行うプロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 211 行で定義されています。

参照先 `Abort()`, `CPM_ERROR_ALREADY_VOXELINIIT`, `CPM_ERROR_INSERT_VOXELMAP`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_MISMATCH_NP_SUBDOMAIN`, `CPM_ERROR_MPI_INVALID_COMM`, `CPM_SUCCESS`, `GetMPI_Comm()`, `cpm_GlobalDomainInfo::GetSubDomainNum()`, `cpm_VoxellInfo::Init()`, `cpm_Base::IsCommNull()`, `m_procGrpList`, `m_voxellInfoMap`, と `SetBndCommBuffer()`.

参照元 `cpm_Voxellnit_()`, `cpm_Voxellnit_nodiv_()`, と `Voxellnit()`.

4.7.3.138 `cpm_ErrorCode cpm_ParaManager::Voxellnit (int div[3], int vox[3], REAL_TYPE origin[3], REAL_TYPE pitch[3], int obcid[6], size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割

- 領域分割の各種情報を引数で渡して領域分割を行う
- プロセスグループの全てのランクが活性ドメインになる
- I,J,K 方向の領域分割数を指定するバージョン

引数

<code>in</code>	<code>div</code>	領域分割数
<code>in</code>	<code>vox</code>	空間全体のボクセル数
<code>in</code>	<code>origin</code>	空間全体の原点
<code>in</code>	<code>pitch</code>	ボクセルピッチ
<code>in</code>	<code>obcid</code>	全体空間の外部境界条件ID
<code>in</code>	<code>maxVC</code>	最大の袖数 (袖通信用)
<code>in</code>	<code>maxN</code>	最大の成分数 (袖通信用)
<code>in</code>	<code>procGrpNo</code>	領域分割を行うプロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 279 行で定義されています。

参照先 cpm_GlobalDomainInfo::AddSubDomain(), CPM_SUCCESS, DecideDivPattern(), GetNumRank(), REAL_TYPE, cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), Voxellnit(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

4.7.3.139 cpm_ErrorCode cpm_ParaManager::Voxellnit (int vox[3], REAL_TYPE origin[3], REAL_TYPE pitch[3], int obcid[6], size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)

領域分割

- ・ 領域分割の各種情報を引数で渡して領域分割を行う
- ・ プロセスグループの全てのランクが活性ドメインになる
- ・ 並列数=プロセスグループの並列数とし、内部で自動的に領域分割をするバージョン

引数

in	vox	空間全体のボクセル数
in	origin	空間全体の原点
in	pitch	ボクセルピッチ
in	obcid	全体空間の外部境界条件ID
in	maxVC	最大の袖数 (袖通信用)
in	maxN	最大の成分数 (袖通信用)
in	procGrpNo	領域分割を行うプロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 334 行で定義されています。

参照先 CPM_SUCCESS, DecideDivPattern(), GetNumRank(), と Voxellnit().

4.7.3.140 cpm_ErrorCode cpm_ParaManager::Wait (MPI_Request * request)

Wait

- ・ MPI_Wait のインターフェイス

引数

in	request	リクエストハンドル
----	---------	-----------

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 177 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_REQUEST, CPM_ERROR_MPI_WAIT, と CPM_SUCCESS.

4.7.3.141 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommsS3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 274 行で定義されています。

参照先 wait_BndCommsS4D().

参照元 cpm_wait_BndCommsS3D().

4.7.3.142 `cpm_ErrorCode cpm_ParaManager::wait_BndCommsS3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 644 行で定義されています。

参照先 wait_BndCommsS4D().

4.7.3.143 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommsS4D (T * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 294 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, unpackX(), unpackY(), unpackZ(), Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_wait_BndCommsS4D(), wait_BndCommsS3D(), wait_BndCommsS4D(), と wait_BndCommV3D().

4.7.3.144 `cpm_ErrorCode cpm_ParaManager::wait_BndCommsS4D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar4D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う
- MPI_Datatype を指定するバージョン

引数

in	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 662 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と wait_BndCommS4D().

4.7.3.145 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommS4DEx (T * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 270 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, unpackXEx(), unpackYEx(), unpackZEx(), Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_wait_BndCommS4DEx(), wait_BndCommS4DEx(), と wait_BndCommV3DEx().

4.7.3.146 `cpm_ErrorCode cpm_ParaManager::wait_BndCommS4DEx (MPI_Datatype dtype, void * array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar4DEx 版, MPI_Datatype 指定)

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>nmax</i>	配列サイズ (成分数)
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 883 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と wait_BndCommS4DEx().

4.7.3.147 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommV3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 284 行で定義されています。

参照先 wait_BndCommS4D().

参照元 cpm_wait_BndCommV3D().

4.7.3.148 `cpm_ErrorCode cpm_ParaManager::wait_BndCommV3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Vector3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 653 行で定義されています。

参照先 wait_BndComms4D().

4.7.3.149 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommV3DEx (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 260 行で定義されています。

参照先 wait_BndCommS4DEx().

参照元 cpm_wait_BndCommV3DEx().

4.7.3.150 `cpm_ErrorCode cpm_ParaManager::wait_BndCommV3DEx (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Vector3DEx 版, MPI_Datatype 指定)

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- MPI_Datatype を指定するバージョン

引数

<i>in</i>	<i>dtype</i>	袖通信データのMPI_Datatype
	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
<i>in</i>	<i>imax</i>	配列サイズ (I 方向)
<i>in</i>	<i>jmax</i>	配列サイズ (J 方向)
<i>in</i>	<i>kmax</i>	配列サイズ (K 方向)
<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<i>vc_comm</i>	通信する仮想セル数
<i>in</i>	<i>req</i>	MPI リクエスト
<i>in</i>	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 874 行で定義されています。

参照先 wait_BndComms4DEx().

4.7.3.151 cpm_ErrorCode cpm_ParaManager::Waitall (int count, MPI_Request requests[])

Waitall

- MPI_Waitall のインターフェイス

引数

in	count	リクエストの数
in	requests	リクエストハンドル配列

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 201 行で定義されています。

参照先 CPM_ERROR_MPI_WAITALL, と CPM_SUCCESS.

参照元 BndComms4D(), BndComms4DEx(), PeriodicComms4D(), PeriodicComms4DEx(), wait_BndComms4D(), と wait_BndComms4DEx().

4.7.4 変数

4.7.4.1 BndCommInfoMap cpm_ParaManager::m_bndCommInfoMap [private]

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManager.h の 2058 行で定義されています。

参照元 cpm_ParaManager(), GetBndCommBuffer(), GetBndCommBufferSize(), SetBndCommBuffer(), と ~cpm_ParaManager().

4.7.4.2 int cpm_ParaManager::m_nRank [private]

プロセス並列数

cpm_ParaManager.h の 2031 行で定義されています。

参照元 cpm_ParaManager(), Initialize(), と IsParallel().

4.7.4.3 std::vector<MPI_Comm> cpm_ParaManager::m_procGrpList [private]

プロセスグループのリスト

- VOXEL 空間番号をインデクスとしたVOXEL 空間のMPI コミュニケータを格納
- vector のインデクス=プロセスグループ番号とする
- [0] には必ずMPI_COMM_WORLD を格納
- 自ランクが含まれるプロセスグループのみを管理する (同じプロセスグループでもプロセス毎に異なるプロセスグループ番号になる場合もある)

`cpm_ParaManager.h` の 2043 行で定義されています。

参照元 `cpm_ParaManager()`, `CreateProcessGroup()`, `GetMPI_Comm()`, `GetMyRankID()`, `GetNumRank()`, `Voxel-Init()`, と `~cpm_ParaManager()`.

4.7.4.4 `int cpm_ParaManager::m_rankNo` [private]

MPI_COMM_WORLD での自ランク番号

`cpm_ParaManager.h` の 2034 行で定義されています。

参照元 `cpm_ParaManager()`, と `Initialize()`.

4.7.4.5 `RankNoMap cpm_ParaManager::m_rankNoMap` [private]

プロセスグループ毎のランク番号マップ

- VOXEL 空間番号をキーとしたランク番号マップ (未使用)

`cpm_ParaManager.h` の 2054 行で定義されています。

参照元 `cpm_ParaManager()`, と `~cpm_ParaManager()`.

4.7.4.6 `cpm_ObjList<MPI.Request> cpm_ParaManager::m_reqList` [private]

MPI_Request の管理マップ

- Fortran インターフェイス用

`cpm_ParaManager.h` の 2063 行で定義されています。

参照元 `cpm_BndCommsS3D_nowait()`, `cpm_BndCommsS4D_nowait()`, `cpm_BndCommS4DEx_nowait()`, `cpm_BndCommV3D_nowait()`, `cpm_BndCommV3DEx_nowait()`, `cpm_lrecv()`, `cpm_lsend()`, `cpm_Wait()`, `cpm_wait_BndCommS3D()`, `cpm_wait_BndCommS4D()`, `cpm_wait_BndCommS4DEx()`, `cpm_wait_BndCommV3D()`, `cpm_wait_BndCommV3DEx()`, と `cpm_Waitall()`.

4.7.4.7 `VoxelInfoMap cpm_ParaManager::m_voxelInfoMap` [private]

プロセスグループ毎のVOXEL 空間情報マップ

- VOXEL 空間番号をキーとしたVOXEL 空間情報マップ
- 自ランクが含まれるVOXEL 空間のみを管理する

`cpm_ParaManager.h` の 2049 行で定義されています。

参照元 `cpm_ParaManager()`, `FindVoxelInfo()`, `VoxelInit()`, と `~cpm_ParaManager()`.

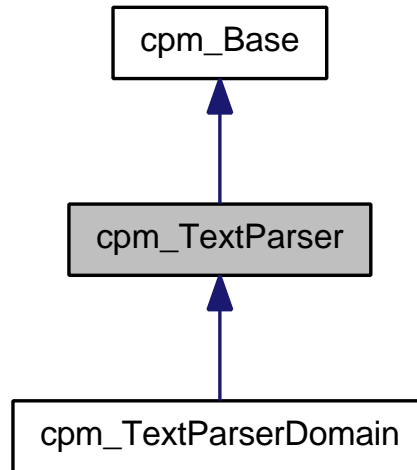
このクラスの説明は次のファイルから生成されました:

- [cpm_ParaManager.h](#)
- [cpm_ParaManager.cpp](#)
- [cpm_ParaManager_Alloc.cpp](#)
- [cpm_ParaManager_frtIF.cpp](#)
- [cpm_ParaManager_MPI.cpp](#)
- [cpm_ParaManager_BndComm.h](#)
- [cpm_ParaManager_BndCommEx.h](#)
- [cpm_ParaManager_inline.h](#)

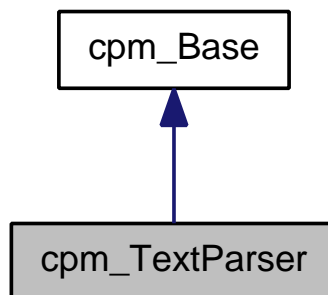
4.8 クラス cpm_TextParser

```
#include <cpm_TextParser.h>
```

cpm_TextParser に対する継承グラフ



cpm_TextParser のコラボレーション図



Protected メソッド

- [cpm_TextParser \(\)](#)
- virtual [~cpm_TextParser \(\)](#)
- int [Read](#) (std::string filename)
- int [readVector](#) (std::string label, float *vec, const int nvec)
- int [readVector](#) (std::string label, double *vec, const int nvec)
- int [readVector](#) (std::string label, int *vec, const int nvec)

Protected 変数

- TextParser * [m_tp](#)

4.8.1 説明

CPM のテキストパーサークラス

cpm_TextParser.h の 22 行で定義されています。

4.8.2 コンストラクタとデストラクタ

4.8.2.1 `cpm_TextParser::cpm_TextParser()` [protected]

コンストラクタ

`cpm_TextParser.cpp` の 18 行で定義されています。

参照先 `m_tp`.

4.8.2.2 `cpm_TextParser::~cpm_TextParser()` [protected, virtual]

デストラクタ

`cpm_TextParser.cpp` の 27 行で定義されています。

参照先 `m_tp`.

4.8.3 関数

4.8.3.1 `int cpm_TextParser::Read(std::string filename)` [protected]

読み込み処理

- ・ユーザは直接コールできない

引数

in	<i>filename</i>	読み込むファイル名
----	-----------------	-----------

戻り値

`TextParser` クラスの終了コード

`cpm_TextParser.cpp` の 34 行で定義されています。

参照先 `m_tp`.

参照元 `cpm_TextParserDomain::ReadMain()`.

4.8.3.2 `int cpm_TextParser::readVector(std::string label, float * vec, const int nvec)` [protected]

ベクトルデータの読み込み (単精度実数版)

引数

in	<i>label</i>	ベクトルデータのテキストラベル
out	<i>vec</i>	読み込んだベクトルデータ (サイズは <i>nvec</i> 確保されている必要がある)
in	<i>nvec</i>	読み込んだベクトルデータの数

戻り値

1000 未満	テキストパーサのエラーコード
<code>CPM_ERROR_TP_NOVECTOR(2001)</code>	指定ラベルがベクトルデータではない
<code>CPM_ERROR_TP_VECTOR_SIZE(2002)</code>	ベクトルデータのサイズが <i>nvec</i> と一致しない

`cpm_TextParser.cpp` の 53 行で定義されています。

参照先 `m_tp`.

参照元 `cpm_TextParserDomain::ReadDomainInfo()`, と `cpm_TextParserDomain::ReadSubdomainInfo()`.

4.8.3.3 `int cpm_TextParser::readVector (std::string label, double * vec, const int nvec)` [protected]

ベクトルデータの読み込み (倍精度実数版)

引数

in	label	ベクトルデータのテキストラベル
out	vec	読み込んだベクトルデータ (サイズは <code>nvec</code> 確保されている必要がある)
in	nvec	読み込んだベクトルデータの数

戻り値

1000 未満	テキストパーサのエラーコード
<code>CPM_ERROR_TP_NOVECTOR(2001)</code>	指定ラベルがベクトルデータではない
<code>CPM_ERROR_TP_VECTOR_SIZE(2002)</code>	ベクトルデータのサイズが <code>nvec</code> と一致しない

`cpm_TextParser.cpp` の 88 行で定義されています。

参照先 `CPM_ERROR_TP_NOVECTOR`, `CPM_ERROR_TP_VECTOR_SIZE`, と `m_tp`.

4.8.3.4 `int cpm_TextParser::readVector (std::string label, int * vec, const int nvec)` [protected]

ベクトルデータの読み込み (整数版)

引数

in	label	ベクトルデータのテキストラベル
out	vec	読み込んだベクトルデータ (サイズは <code>nvec</code> 確保されている必要がある)
in	nvec	読み込んだベクトルデータの数

戻り値

1000 未満	テキストパーサのエラーコード
<code>CPM_ERROR_TP_NOVECTOR(2001)</code>	指定ラベルがベクトルデータではない
<code>CPM_ERROR_TP_VECTOR_SIZE(2002)</code>	ベクトルデータのサイズが <code>nvec</code> と一致しない

`cpm_TextParser.cpp` の 123 行で定義されています。

参照先 `m_tp`.

4.8.4 変数

4.8.4.1 `TextParser* cpm_TextParser::m_tp` [protected]

テキストパーサークラスのインスタンス

`cpm_TextParser.h` の 91 行で定義されています。

参照元 `cpm_TextParser()`, `Read()`, `cpm_TextParserDomain::ReadDomainInfo()`, `cpm_TextParserDomain::Read-`

SubdomainInfo(), readVector(), と ~cpm_TextParser().

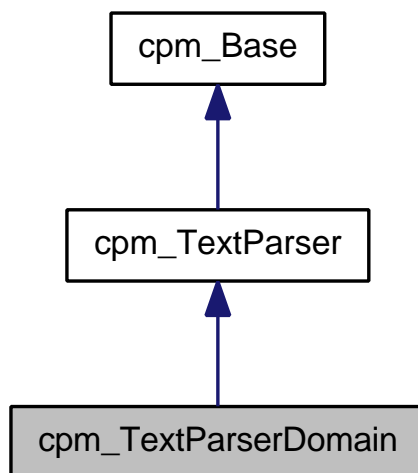
このクラスの説明は次のファイルから生成されました:

- [cpm_TextParser.h](#)
- [cpm_TextParser.cpp](#)

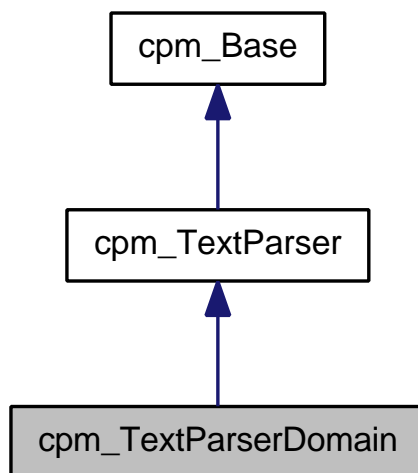
4.9 クラス cpm_TextParserDomain

```
#include <cpm_TextParserDomain.h>
```

cpm_TextParserDomain に対する継承グラフ



cpm_TextParserDomain のコラボレーション図



Public メソッド

- [cpm_TextParserDomain \(\)](#)
- [virtual ~cpm_TextParserDomain \(\)](#)

Static Public メソッド

- static `cpm_GlobalDomainInfo * Read` (std::string filename, int &errorcode)

Private メソッド

- `cpm_GlobalDomainInfo * ReadMain` (std::string filename, int &errorcode)
- int `ReadDomainInfo` (cpm_GlobalDomainInfo *dInfo)
- int `ReadSubdomainInfo` (cpm_GlobalDomainInfo *dInfo)

4.9.1 説明

CPM の領域情報テキストパーサークラス

cpm_TextParserDomain.h の 22 行で定義されています。

4.9.2 コンストラクタとデストラクタ

4.9.2.1 cpm_TextParserDomain::cpm_TextParserDomain ()

コンストラクタ

cpm_TextParserDomain.cpp の 18 行で定義されています。

4.9.2.2 cpm_TextParserDomain::~cpm_TextParserDomain () [virtual]

デストラクタ

cpm_TextParserDomain.cpp の 25 行で定義されています。

4.9.3 関数

4.9.3.1 cpm_GlobalDomainInfo * cpm_TextParserDomain::Read (std::string filename, int & errorcode) [static]

読み込み処理

- TextParser クラスを用いて領域分割情報ファイルを読み込む
- TextParser クラスのインスタンスはクリア (remove) される

引数

in	<i>filename</i>	読み込むファイル名
out	<i>errorcode</i>	CPM エラーコード

戻り値

領域情報ポインタ

cpm_TextParserDomain.cpp の 32 行で定義されています。

参照先 ReadMain().

4.9.3.2 `int cpm_TextParserDomain::ReadDomainInfo (cpm_GlobalDomainInfo * dInfo) [private]`

DomainInfo の読み込み

引数

<i>inout</i>	dInfo 領域情報
--------------	------------

戻り値

CPM エラーコード

`cpm_TextParserDomain.cpp` の 82 行で定義されています。

参照先 `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_TP_INVALID_G_DIV`, `CPM_ERROR_TP_INVALID_G_ORG`, `CPM_ERROR_TP_INVALID_G_PITCH`, `CPM_ERROR_TP_INVALID_G_VOXEL`, `cpm_Base::cpm_strCompare()`, `CPM_SUCCESS`, `cpm_TextParser::m_tp`, `cpm_TextParser::readVector()`, `REAL_TYPE`, `cpm_GlobalDomainInfo::SetDivNum()`, `cpm_DomainInfo::SetOrigin()`, `cpm_DomainInfo::SetPitch()`, `cpm_DomainInfo::SetRegion()`, と `cpm_DomainInfo::SetVoxNum()`.

参照元 `ReadMain()`.

4.9.3.3 `cpm_GlobalDomainInfo * cpm_TextParserDomain::ReadMain (std::string filename, int & errorcode) [private]`

読み込み処理のメイン

- TextParser クラスを用いて領域分割情報ファイルを読み込む
- TextParser クラスのインスタンスはクリア (remove) される

引数

<i>in</i>	<i>filename</i>	読み込むファイル名
<i>out</i>	<i>errorcode</i>	CPM エラーコード

戻り値

領域情報ポインタ

`cpm_TextParserDomain.cpp` の 44 行で定義されています。

参照先 `CPM_ERROR_INVALID_PTR`, `cpm_TextParser::Read()`, `ReadDomainInfo()`, と `ReadSubdomainInfo()`.

参照元 `Read()`.

4.9.3.4 `int cpm_TextParserDomain::ReadSubdomainInfo (cpm_GlobalDomainInfo * dInfo) [private]`

ActiveSubDomains の読み込み

引数

<i>inout</i>	dInfo 領域情報
--------------	------------

戻り値

CPM エラーコード

`cpm_TextParserDomain.cpp` の 224 行で定義されています。

参照先 `cpm_GlobalDomainInfo::AddSubDomain()`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_TP_INVALID_BCID`, `CPM_ERROR_TP_INVALID_G_DIV`, `CPM_ERROR_TP_INVALID_POS`, `cpm_Base::cpm_strCompare()`, `cpm_Base::cpm_strCompareN()`, `CPM_SUCCESS`, `cpm_GlobalDomainInfo::GetDivNum()`, `cpm_TextParser::m_tp`, `cpm_TextParser::readVector()`, `cpm_ActiveSubDomainInfo::SetBCID()`, と `cpm_ActiveSubDomainInfo::SetPos()`.

参照元 `ReadMain()`.

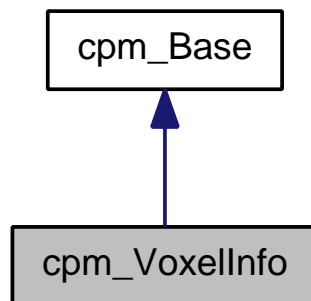
このクラスの説明は次のファイルから生成されました:

- [cpm_TextParserDomain.h](#)
- [cpm_TextParserDomain.cpp](#)

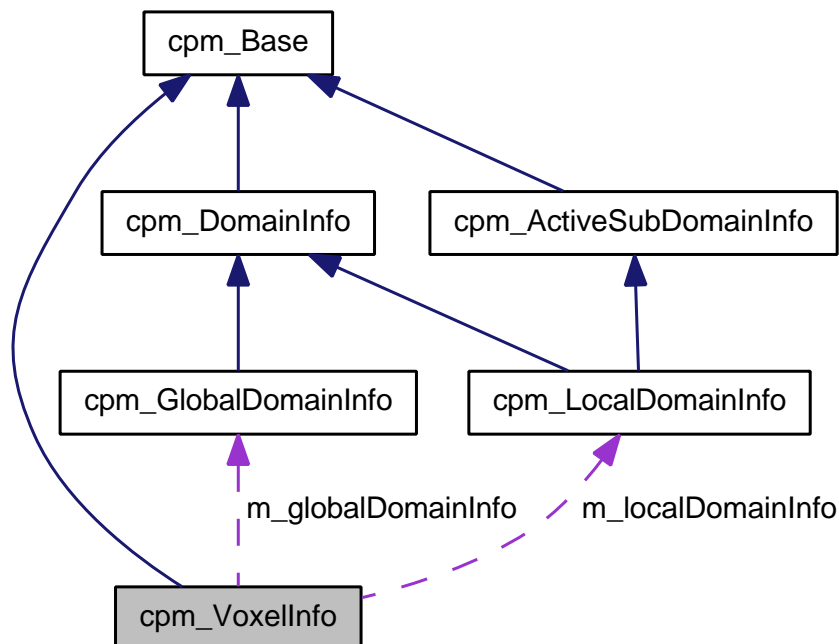
4.10 クラス `cpm_VoxelInfo`

```
#include <cpm_VoxelInfo.h>
```

`cpm_VoxelInfo` に対する継承グラフ



`cpm_VoxelInfo` のコラボレーション図



Private メソッド

- `cpm_VoxelInfo` ()
- `virtual ~cpm_VoxelInfo` ()
- `cpm_ErrorCode Init` (MPI_Comm comm, `cpm_GlobalDomainInfo *dInfo`)
- `bool CreateRankMap` ()
- `bool CreateNeighborRankInfo` ()
- `bool CreateLocalDomainInfo` ()
- `const int * GetDivNum` () const
- `const REAL_TYPE * GetPitch` () const
- `const int * GetGlobalVoxelSize` () const
- `const REAL_TYPE * GetGlobalOrigin` () const
- `const REAL_TYPE * GetGlobalRegion` () const
- `const int * GetLocalVoxelSize` () const
- `const REAL_TYPE * GetLocalOrigin` () const
- `const REAL_TYPE * GetLocalRegion` () const
- `const int * GetDivPos` () const
- `const int * GetBCID` () const
- `const int * GetVoxelHeadIndex` () const
- `const int * GetVoxelTailIndex` () const
- `const int * GetNeighborRankID` () const
- `const int * GetPeriodicRankID` () const

Private 変数

- `cpm_GlobalDomainInfo m_globalDomainInfo`
空間全体の領域情報
- `cpm_LocalDomainInfo m_localDomainInfo`
自ランクの領域情報
- `int m_voxelHeadIndex` [3]
自ランクの始点ボクセルインデックス
- `int m_voxelTailIndex` [3]
自ランクの終点ボクセルインデックス
- `MPI_Comm m_comm`
MPI コミュニケータ
- `int m_nRank`
コミュニケータ内のランク数 (=プロセス並列数)
- `int m_rankNo`
コミュニケータ内でのランク番号
- `int m_neighborRankID` [6]
隣接ランク番号 (外部境界は負の値)
- `int m_periodicRankID` [6]
周期境界の隣接ランク番号
- `int * m_rankMap`
ランクマップ

フレンド

- `class cpm_ParaManager`

4.10.1 説明

CPM のVOXEL 空間情報管理クラス

`cpm_VoxelInfo.h` の 23 行で定義されています。

4.10.2 コンストラクタとデストラクタ

4.10.2.1 `cpm_VoxelInfo::cpm_VoxelInfo ()` [private]

コンストラクタ

`cpm_VoxelInfo.cpp` の 18 行で定義されています。

参照先 `cpm_Base::getRankNull()`, `m_comm`, `m_neighborRankID`, `m_nRank`, `m_periodicRankID`, `m_rankMap`, `m_rankNo`, `m_voxelHeadIndex`, と `m_voxelTailIndex`.

4.10.2.2 `cpm_VoxelInfo::~cpm_VoxelInfo ()` [private, virtual]

デストラクタ

`cpm_VoxelInfo.cpp` の 43 行で定義されています。

参照先 `m_rankMap`.

4.10.3 関数

4.10.3.1 `bool cpm_VoxelInfo::CreateLocalDomainInfo ()` [private]

ローカル領域情報を生成

戻り値

<i>true</i>	正常終了
<i>false</i>	エラー

`cpm_VoxelInfo.cpp` の 158 行で定義されています。

参照先 `_IDX_S3D`, `cpm_ActiveSubDomainInfo::GetBCID()`, `cpm_GlobalDomainInfo::GetDivNum()`, `cpm_DomainInfo::GetOrigin()`, `cpm_DomainInfo::GetPitch()`, `cpm_GlobalDomainInfo::GetSubDomainInfo()`, `cpm_GlobalDomainInfo::GetSubDomainNum()`, `cpm_DomainInfo::GetVoxNum()`, `m_globalDomainInfo`, `m_localDomainInfo`, `m_rankMap`, `m_rankNo`, `m_voxelHeadIndex`, `m_voxelTailIndex`, `REAL_TYPE`, `cpm_ActiveSubDomainInfo::SetBCID()`, `cpm_DomainInfo::SetOrigin()`, `cpm_DomainInfo::SetPitch()`, `cpm_ActiveSubDomainInfo::SetPos()`, `cpm_DomainInfo::SetRegion()`, と `cpm_DomainInfo::SetVoxNum()`.

参照元 `Init()`.

4.10.3.2 `bool cpm_VoxelInfo::CreateNeighborRankInfo ()` [private]

隣接ランク情報を生成

戻り値

<i>true</i>	正常終了
<i>false</i>	エラー

`cpm_VoxelInfo.cpp` の 278 行で定義されています。

参照先 `_IDX_S3D`, `cpm_GlobalDomainInfo::GetDivNum()`, `cpm_ActiveSubDomainInfo::GetPos()`, `cpm_Base::getRankNull()`, `m_globalDomainInfo`, `m_localDomainInfo`, `m_neighborRankID`, `m_periodicRankID`, `m_rankMap`, `m_rankNo`, `X_MINUS`, `X_PLUS`, `Y_MINUS`, `Y_PLUS`, `Z_MINUS`, と `Z_PLUS`.

参照元 `Init()`.

4.10.3.3 `bool cpm_VoxelInfo::CreateRankMap () [private]`

ランクマップを生成

戻り値

<code>true</code>	正常終了
<code>false</code>	エラー

`cpm_VoxelInfo.cpp` の 95 行で定義されています。

参照先 `_IDX_S3D`, `cpm_GlobalDomainInfo::GetDivNum()`, `cpm_ActiveSubDomainInfo::GetPos()`, `cpm_Base::getRankNull()`, `cpm_GlobalDomainInfo::GetSubDomainInfo()`, `cpm_GlobalDomainInfo::GetSubDomainNum()`, `m_globalDomainInfo`, と `m_rankMap`.

参照元 `Init()`.

4.10.3.4 `const int * cpm_VoxelInfo::GetBCID () const [private]`

自ランクのBCID を取得

戻り値

自ランクのBCID 整数配列のポインタ

`cpm_VoxelInfo.cpp` の 456 行で定義されています。

参照先 `cpm_ActiveSubDomainInfo::GetBCID()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetBCID()`.

4.10.3.5 `const int * cpm_VoxelInfo::GetDivNum () const [private]`

領域分割数を取得

戻り値

領域分割数整数配列のポインタ

`cpm_VoxelInfo.cpp` の 384 行で定義されています。

参照先 `cpm_GlobalDomainInfo::GetDivNum()`, と `m_globalDomainInfo`.

参照元 `cpm_ParaManager::GetDivNum()`.

4.10.3.6 `const int * cpm_VoxelInfo::GetDivPos () const [private]`

自ランクの領域分割位置を取得

戻り値

自ランクの領域分割位置整数配列のポインタ

`cpm_VoxelInfo.cpp` の 448 行で定義されています。

参照先 `cpm_ActiveSubDomainInfo::GetPos()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetDivPos()`.

4.10.3.7 `const REAL_TYPE * cpm_VoxelInfo::GetGlobalOrigin () const [private]`

全体空間の原点を取得

戻り値

全体空間の原点実数配列のポインタ

cpm_VoxelInfo.cpp の 408 行で定義されています。

参照先 cpm_DomainInfo::GetOrigin(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetGlobalOrigin().

4.10.3.8 `const REAL_TYPE * cpm_VoxelInfo::GetGlobalRegion () const [private]`

全体空間サイズを取得

戻り値

全体空間サイズ実数配列のポインタ

cpm_VoxelInfo.cpp の 416 行で定義されています。

参照先 cpm_DomainInfo::GetRegion(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetGlobalRegion().

4.10.3.9 `const int * cpm_VoxelInfo::GetGlobalVoxelSize () const [private]`

全体ボクセル数を取得

戻り値

全体ボクセル数整数配列のポインタ

cpm_VoxelInfo.cpp の 400 行で定義されています。

参照先 cpm_DomainInfo::GetVoxNum(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetGlobalVoxelSize().

4.10.3.10 `const REAL_TYPE * cpm_VoxelInfo::GetLocalOrigin () const [private]`

自ランクの空間原点を取得

戻り値

自ランクの空間原点実数配列のポインタ

cpm_VoxelInfo.cpp の 432 行で定義されています。

参照先 cpm_DomainInfo::GetOrigin(), と m_localDomainInfo.

参照元 cpm_ParaManager::GetLocalOrigin().

4.10.3.11 `const REAL_TYPE * cpm_VoxelInfo::GetLocalRegion () const [private]`

自ランクの空間サイズを取得

戻り値

自ランクの空間サイズ実数配列のポインタ

`cpm_VoxelInfo.cpp` の 440 行で定義されています。

参照先 `cpm_DomainInfo::GetRegion()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetLocalRegion()`.

4.10.3.12 `const int * cpm_VoxelInfo::GetLocalVoxelSize () const` [private]

自ランクのボクセル数を取得

戻り値

自ランクのボクセル数整数配列のポインタ

`cpm_VoxelInfo.cpp` の 424 行で定義されています。

参照先 `cpm_DomainInfo::GetVoxNum()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetLocalVoxelSize()`.

4.10.3.13 `const int * cpm_VoxelInfo::GetNeighborRankID () const` [private]

自ランクの隣接ランク番号を取得

戻り値

自ランクの隣接ランク番号整数配列のポインタ

`cpm_VoxelInfo.cpp` の 480 行で定義されています。

参照先 `m_neighborRankID`.

参照元 `cpm_ParaManager::GetNeighborRankID()`.

4.10.3.14 `const int * cpm_VoxelInfo::GetPeriodicRankID () const` [private]

自ランクの周期境界の隣接ランク番号を取得

戻り値

自ランクの周期境界の隣接ランク番号整数配列のポインタ

`cpm_VoxelInfo.cpp` の 488 行で定義されています。

参照先 `m_periodicRankID`.

参照元 `cpm_ParaManager::GetPeriodicRankID()`.

4.10.3.15 `const REAL_TYPE * cpm_VoxelInfo::GetPitch () const` [private]

ピッチを取得

戻り値

ピッチ実数配列のポインタ

cpm_VoxelInfo.cpp の 392 行で定義されています。

参照先 cpm_DomainInfo::GetPitch(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetPitch().

4.10.3.16 `const int * cpm_VoxelInfo::GetVoxelHeadIndex () const [private]`

自ランクの始点VOXEL の全体空間でのインデクスを取得

戻り値

自ランクの始点インデクス整数配列のポインタ

cpm_VoxelInfo.cpp の 464 行で定義されています。

参照先 m_voxelHeadIndex.

参照元 cpm_ParaManager::GetVoxelHeadIndex().

4.10.3.17 `const int * cpm_VoxelInfo::GetVoxelTailIndex () const [private]`

自ランクの終点VOXEL の全体空間でのインデクスを取得

戻り値

自ランクの終点インデクス整数配列のポインタ

cpm_VoxelInfo.cpp の 472 行で定義されています。

参照先 m_voxelTailIndex.

参照元 cpm_ParaManager::GetVoxelTailIndex().

4.10.3.18 `cpm_ErrorCode cpm_VoxelInfo::Init (MPI_Comm comm, cpm_GlobalDomainInfo * dInfo) [private]`

CPM 領域分割情報の生成

- MPI_COMM_WORLD を使用した領域を生成する。

引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>dInfo</i>	領域分割情報
in	<i>maxVC</i>	最大の袖数 (袖通信用)
in	<i>maxN</i>	最大の成分数 (袖通信用)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfo.cpp の 51 行で定義されています。

参照先 CPM_ERROR_CREATE_LOCALDOMAIN, CPM_ERROR_CREATE_NEIGHBOR, CPM_ERROR_CREATE_RANKMAP, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, Create-

`LocalDomainInfo()`, `CreateNeighborRankInfo()`, `CreateRankMap()`, `cpm_Base::IsCommNull()`, `m_comm`, `m_globalDomainInfo`, `m_nRank`, と `m_rankNo`.

参照元 `cpm_ParaManager::VoxelInit()`.

4.10.4 フレンドと関連する関数

4.10.4.1 `friend class cpm_ParaManager [friend]`

`cpm_VoxelInfo.h` の 25 行で定義されています。

4.10.5 変数

4.10.5.1 `MPI_Comm cpm_VoxelInfo::m_comm [private]`

MPI コミュニケーター

`cpm_VoxelInfo.h` の 155 行で定義されています。

参照元 `cpm_VoxelInfo()`, と `Init()`.

4.10.5.2 `cpm_GlobalDomainInfo cpm_VoxelInfo::m_globalDomainInfo [private]`

空間全体の領域情報

`cpm_VoxelInfo.h` の 147 行で定義されています。

参照元 `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, `CreateRankMap()`, `GetDivNum()`, `GetGlobalOrigin()`, `GetGlobalRegion()`, `GetGlobalVoxelSize()`, `GetPitch()`, と `Init()`.

4.10.5.3 `cpm_LocalDomainInfo cpm_VoxelInfo::m_localDomainInfo [private]`

自ランクの領域情報

`cpm_VoxelInfo.h` の 150 行で定義されています。

参照元 `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, `GetBCID()`, `GetDivPos()`, `GetLocalOrigin()`, `GetLocalRegion()`, と `GetLocalVoxelSize()`.

4.10.5.4 `int cpm_VoxelInfo::m_neighborRankID[6] [private]`

隣接ランク番号 (外部境界は負の値)

`cpm_VoxelInfo.h` の 158 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateNeighborRankInfo()`, と `GetNeighborRankID()`.

4.10.5.5 `int cpm_VoxelInfo::m_nRank [private]`

コミュニケーター内のランク数 (=プロセス並列数)

`cpm_VoxelInfo.h` の 156 行で定義されています。

参照元 `cpm_VoxelInfo()`, と `Init()`.

4.10.5.6 `int cpm_VoxelInfo::m_periodicRankID[6] [private]`

周期境界の隣接ランク番号

cpm_VoxelInfo.h の 159 行で定義されています。

参照元 cpm_VoxelInfo(), CreateNeighborRankInfo(), と GetPeriodicRankID().

4.10.5.7 int* cpm_VoxelInfo::m_rankMap [private]

ランクマップ

cpm_VoxelInfo.h の 161 行で定義されています。

参照元 cpm_VoxelInfo(), CreateLocalDomainInfo(), CreateNeighborRankInfo(), CreateRankMap(), と ~cpm_VoxelInfo().

4.10.5.8 int cpm_VoxelInfo::m_rankNo [private]

コミュニケーター内でのランク番号

cpm_VoxelInfo.h の 157 行で定義されています。

参照元 cpm_VoxelInfo(), CreateLocalDomainInfo(), CreateNeighborRankInfo(), と Init().

4.10.5.9 int cpm_VoxelInfo::m_voxelHeadIndex[3] [private]

自ランクの始点ボクセルインデックス

cpm_VoxelInfo.h の 151 行で定義されています。

参照元 cpm_VoxelInfo(), CreateLocalDomainInfo(), と GetVoxelHeadIndex().

4.10.5.10 int cpm_VoxelInfo::m_voxelTailIndex[3] [private]

自ランクの終点ボクセルインデックス

cpm_VoxelInfo.h の 152 行で定義されています。

参照元 cpm_VoxelInfo(), CreateLocalDomainInfo(), と GetVoxelTailIndex().

このクラスの説明は次のファイルから生成されました:

- [cpm_VoxelInfo.h](#)
- [cpm_VoxelInfo.cpp](#)

4.11 構造体 S_BNDCOMM_BUFFER

```
#include <cpm_ParaManager.h>
```

Public メソッド

- [S_BNDCOMM_BUFFER \(\)](#)
- [~S_BNDCOMM_BUFFER \(\)](#)

Public 変数

- [size_t m_maxVC](#)
最大袖数
- [size_t m_maxN](#)

- 最大成分数
- `size_t m_nwX`
バッファサイズ
- `size_t m_nwY`
バッファサイズ
- `size_t m_nwZ`
バッファサイズ
- `REAL_BUF_TYPE * m_bufX [4]`
バッファ
- `REAL_BUF_TYPE * m_bufY [4]`
バッファ
- `REAL_BUF_TYPE * m_bufZ [4]`
バッファ

4.11.1 説明

袖通信バッファ情報

cpm_ParaManager.h の 33 行で定義されています。

4.11.2 コンストラクタとデストラクタ

4.11.2.1 S_BNDCOMM_BUFFER::S_BNDCOMM_BUFFER () [inline]

cpm_ParaManager.h の 44 行で定義されています。

参照先 `m_bufX`, `m_bufY`, `m_bufZ`, `m_maxN`, `m_maxVC`, `m_nwX`, `m_nwY`, と `m_nwZ`.

4.11.2.2 S_BNDCOMM_BUFFER::~S_BNDCOMM_BUFFER () [inline]

cpm_ParaManager.h の 56 行で定義されています。

参照先 `m_bufX`, `m_bufY`, と `m_bufZ`.

4.11.3 変数

4.11.3.1 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufX[4]

バッファ

cpm_ParaManager.h の 40 行で定義されています。

参照元 `cpm_ParaManager::BndCommS4D()`, `cpm_ParaManager::BndCommS4D_nowait()`, `cpm_ParaManager::BndCommS4DEx()`, `cpm_ParaManager::BndCommS4DEx_nowait()`, `cpm_ParaManager::PeriodicCommS4D()`, `cpm_ParaManager::PeriodicCommS4DEx()`, `S_BNDCOMM_BUFFER()`, `cpm_ParaManager::SetBndCommBuffer()`, `cpm_ParaManager::wait_BndCommS4D()`, `cpm_ParaManager::wait_BndCommS4DEx()`, と `~S_BNDCOMM_BUFFER()`.

4.11.3.2 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufY[4]

バッファ

cpm_ParaManager.h の 41 行で定義されています。

参照元 `cpm_ParaManager::BndCommS4D()`, `cpm_ParaManager::BndCommS4D_nowait()`, `cpm_ParaManager::BndCommS4DEx()`, `cpm_ParaManager::BndCommS4DEx_nowait()`, `cpm_ParaManager::PeriodicCommS4D()`, `cpm_ParaManager::PeriodicCommS4DEx()`, `S_BNDCOMM_BUFFER()`, `cpm_ParaManager::SetBndCommBuffer()`, `cpm_ParaManager::wait_BndCommS4D()`, `cpm_ParaManager::wait_BndCommS4DEx()`, と `~S_BNDCOMM_BUFFER()`.

4.11.3.3 `REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufZ[4]`

バッファ

`cpm_ParaManager.h` の 42 行で定義されています。

参照元 `cpm_ParaManager::BndCommS4D()`, `cpm_ParaManager::BndCommS4D_nowait()`, `cpm_ParaManager::BndCommS4DEx()`, `cpm_ParaManager::BndCommS4DEx_nowait()`, `cpm_ParaManager::PeriodicCommS4D()`, `cpm_ParaManager::PeriodicCommS4DEx()`, `S_BNDCOMM_BUFFER()`, `cpm_ParaManager::SetBndCommBuffer()`, `cpm_ParaManager::wait_BndCommS4D()`, `cpm_ParaManager::wait_BndCommS4DEx()`, と `~S_BNDCOMM_BUFFER()`.

4.11.3.4 `size_t S_BNDCOMM_BUFFER::m_maxN`

最大成分数

`cpm_ParaManager.h` の 36 行で定義されています。

参照元 `S_BNDCOMM_BUFFER()`, と `cpm_ParaManager::SetBndCommBuffer()`.

4.11.3.5 `size_t S_BNDCOMM_BUFFER::m_maxVC`

最大袖数

`cpm_ParaManager.h` の 35 行で定義されています。

参照元 `S_BNDCOMM_BUFFER()`, と `cpm_ParaManager::SetBndCommBuffer()`.

4.11.3.6 `size_t S_BNDCOMM_BUFFER::m_nwX`

バッファサイズ

`cpm_ParaManager.h` の 37 行で定義されています。

参照元 `cpm_ParaManager::BndCommS4D()`, `cpm_ParaManager::BndCommS4D_nowait()`, `cpm_ParaManager::BndCommS4DEx()`, `cpm_ParaManager::BndCommS4DEx_nowait()`, `cpm_ParaManager::GetBndCommBufferSize()`, `cpm_ParaManager::PeriodicCommS4D()`, `cpm_ParaManager::PeriodicCommS4DEx()`, `S_BNDCOMM_BUFFER()`, `cpm_ParaManager::SetBndCommBuffer()`, `cpm_ParaManager::wait_BndCommS4D()`, と `cpm_ParaManager::wait_BndCommS4DEx()`.

4.11.3.7 `size_t S_BNDCOMM_BUFFER::m_nwY`

バッファサイズ

`cpm_ParaManager.h` の 38 行で定義されています。

参照元 `cpm_ParaManager::BndCommS4D()`, `cpm_ParaManager::BndCommS4D_nowait()`, `cpm_ParaManager::BndCommS4DEx()`, `cpm_ParaManager::BndCommS4DEx_nowait()`, `cpm_ParaManager::GetBndCommBufferSize()`, `cpm_ParaManager::PeriodicCommS4D()`, `cpm_ParaManager::PeriodicCommS4DEx()`, `S_BNDCOMM_BUFFER()`, `cpm_ParaManager::SetBndCommBuffer()`, `cpm_ParaManager::wait_BndCommS4D()`, と `cpm_ParaManager::wait_BndCommS4DEx()`.

4.11.3.8 size_t S_BNDCOMM_BUFFER::m_nwZ

バッファサイズ

cpm_ParaManager.h の 39 行で定義されています。

参照元 cpm_ParaManager::BndCommS4D(), cpm_ParaManager::BndCommS4D_nowait(), cpm_ParaManager::BndCommS4DEx(), cpm_ParaManager::BndCommS4DEx_nowait(), cpm_ParaManager::GetBndCommBufferSize(), cpm_ParaManager::PeriodicCommS4D(), cpm_ParaManager::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManager::SetBndCommBuffer(), cpm_ParaManager::wait_BndCommS4D(), と cpm_ParaManager::wait_BndCommS4DEx()。

この構造体の説明は次のファイルから生成されました:

- [cpm_ParaManager.h](#)

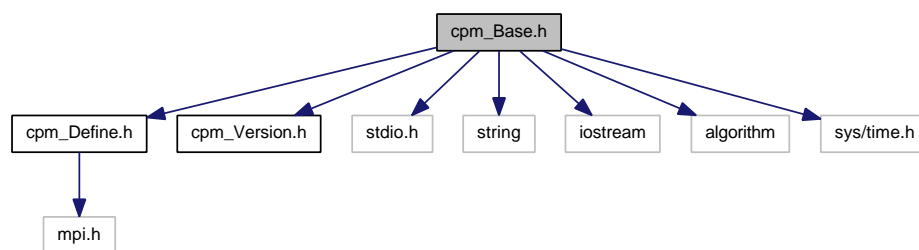
Chapter 5

ファイル

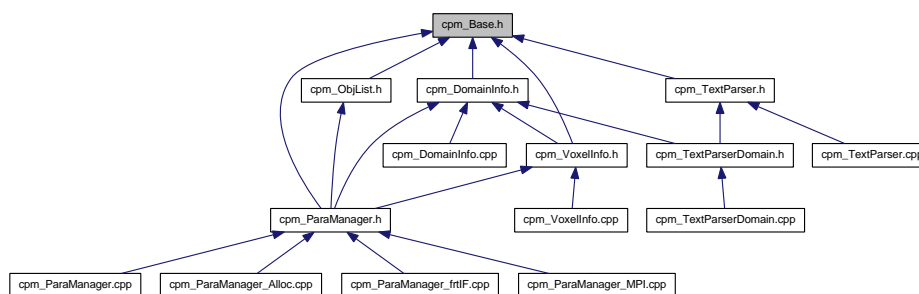
5.1 cpm_Base.h

```
#include "cpm_Define.h"  
#include "cpm_Version.h"  
#include <stdio.h>  
#include <string>  
#include <iostream>  
#include <algorithm>  
#include <sys/time.h>
```

cpm_Base.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class `cpm_Base`

マクロ定義

- `#define CPM_INLINE inline`

5.1.1 説明

CPM のベースクラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

`cpm_Base.h` で定義されています。

5.1.2 マクロ定義

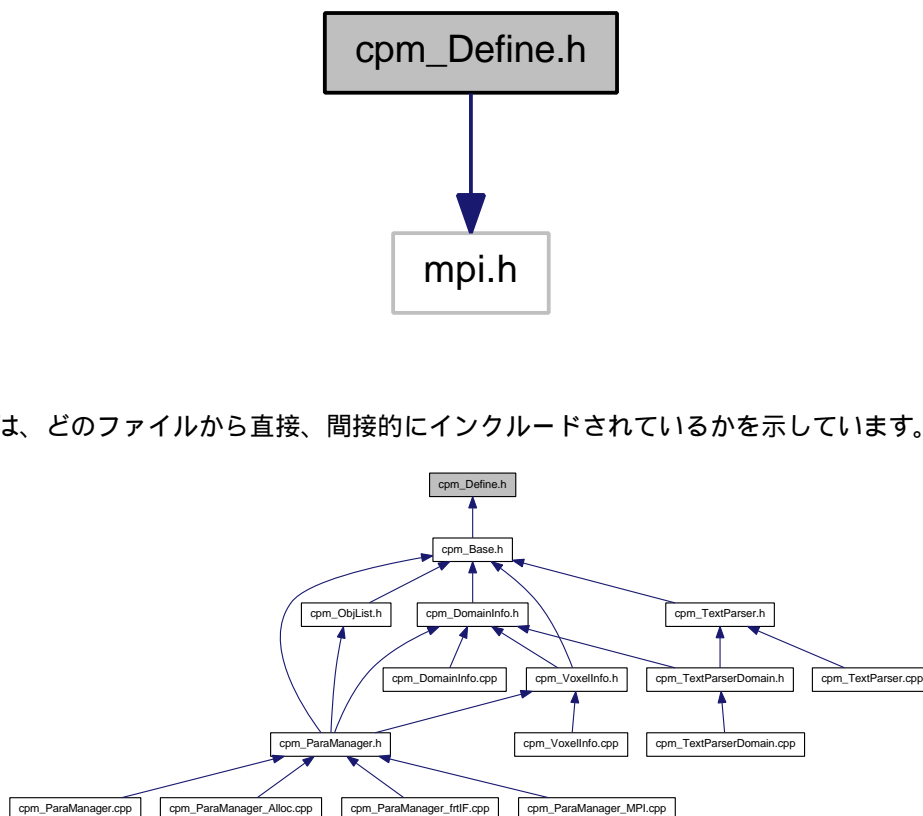
5.1.2.1 `#define CPM_INLINE inline`

`cpm_Base.h` の 38 行で定義されています。

5.2 `cpm_Define.h`

```
#include "mpi.h"
```

`cpm_Define.h` のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

マクロ定義

- #define REAL_TYPE float
- #define REAL_BUF_TYPE REAL_TYPE
- #define _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC)
- #define _IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)
- #define _IDX_V3D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC))
- #define _IDX_S4DEX(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)
- #define _IDX_V3DEX(_N, _I, _J, _K, _NI, _NJ, _NK, _VC) (_IDX_S4DEX(_N, _I, _J, _K, _NI, _NJ, _NK, _VC))

列挙型

- enum cpm_FaceFlag {
X_MINUS = 0, Y_MINUS = 1, Z_MINUS = 2, X_PLUS = 3,
Y_PLUS = 4, Z_PLUS = 5 }
- enum cpm_DirFlag { X_DIR = 0, Y_DIR = 1, Z_DIR = 2 }
- enum cpm_PMFlag { PLUS2MINUS = 0, MINUS2PLUS = 1, BOTH = 2 }
- enum cpm_ErrorCode {
CPM_SUCCESS = 0, CPM_ERROR = 1000, CPM_ERROR_PM_INSTANCE = 1001, CPM_ERROR_INVALID_PTR = 1002,
CPM_ERROR_INVALID_DOMAIN_NO = 1003, CPM_ERROR_INVALID_OBJKEY = 1004, CPM_ERROR_REGIST_OBJKEY = 1005,
CPM_ERROR_TEXTPARSER = 2000,
CPM_ERROR_NO_TEXTPARSER = 2001, CPM_ERROR_TP_NOVECTOR = 2002, CPM_ERROR_TP_VECTOR_SIZE = 2003,
CPM_ERROR_TP_INVALID_G_ORG = 2004,
CPM_ERROR_TP_INVALID_G_VOXEL = 2005, CPM_ERROR_TP_INVALID_G_PITCH = 2006,
CPM_ERROR_TP_INVALID_G_RGN = 2007, CPM_ERROR_TP_INVALID_G_DIV = 2008,
CPM_ERROR_TP_INVALID_POS = 2009, CPM_ERROR_TP_INVALID_BCID = 2010, CPM_ERROR_VOXELINIT = 3000,
CPM_ERROR_NOT_IN_PROCGROUP = 3001,
CPM_ERROR_ALREADY_VOXELINIT = 3002, CPM_ERROR_MISMATCH_NP_SUBDOMAIN = 3003,
CPM_ERROR_CREATE_RANKMAP = 3004, CPM_ERROR_CREATE_NEIGHBOR = 3005,
CPM_ERROR_CREATE_LOCALDOMAIN = 3006, CPM_ERROR_INSERT_VOXELMAP = 3007,
CPM_ERROR_CREATE_PROCGROUP = 3008, CPM_ERROR_INVALID_VOXELSIZE = 3009,
CPM_ERROR_GET_INFO = 4000, CPM_ERROR_GET_DIVNUM = 4001, CPM_ERROR_GET_PITCH = 4002,
CPM_ERROR_GET_GLOBALVOXELSIZE = 4003,
CPM_ERROR_GET_GLOBALORIGIN = 4004, CPM_ERROR_GET_GLOBALREGION = 4005, CPM_ERROR_GET_LOCALV = 4006,
CPM_ERROR_GET_LOCALORIGIN = 4007,
CPM_ERROR_GET_LOCALREGION = 4008, CPM_ERROR_GET_DIVPOS = 4009, CPM_ERROR_GET_BCID = 4010,
CPM_ERROR_GET_HEADINDEX = 4011,
CPM_ERROR_GET_TAILINDEX = 4012, CPM_ERROR_GET_NEIGHBOR_RANK = 4013, CPM_ERROR_GET_PERIODIC_F = 4014,
CPM_ERROR_GET_MYRANK = 4015,
CPM_ERROR_GET_NUMRANK = 4016, CPM_ERROR_MPI = 9000, CPM_ERROR_NO_MPI_INIT = 9001,
CPM_ERROR_MPI_BARRIER = 9003,
CPM_ERROR_MPI_BCAST = 9004, CPM_ERROR_MPI_SEND = 9005, CPM_ERROR_MPI_RECV = 9006,
CPM_ERROR_MPI_ISEND = 9007,
CPM_ERROR_MPI_IRECV = 9008, CPM_ERROR_MPI_WAIT = 9009, CPM_ERROR_MPI_WAITALL = 9010,
CPM_ERROR_MPI_ALLREDUCE = 9011,
CPM_ERROR_MPI_GATHER = 9012, CPM_ERROR_MPI_ALLGATHER = 9013, CPM_ERROR_MPI_GATHERV = 9014,
CPM_ERROR_MPI_ALLGATHERV = 9015,
CPM_ERROR_MPI_DIMSCREATE = 9016, CPM_ERROR_BNDCOMM = 9500, CPM_ERROR_BNDCOMM_VOXELSIZE = 9501,
CPM_ERROR_BNDCOMM_BUFFER = 9502,
CPM_ERROR_BNDCOMM_BUFFERLENGTH = 9503, CPM_ERROR_PERIODIC = 9600, CPM_ERROR_PERIODIC_INVALID = 9601,
CPM_ERROR_PERIODIC_INVALID_PM = 9602,
CPM_ERROR_MPI_INVALID_COMM = 9100, CPM_ERROR_MPI_INVALID_DATATYPE = 9101,
CPM_ERROR_MPI_INVALID_OPERATOR = 9102, CPM_ERROR_MPI_INVALID_REQUEST = 9103 }

- enum CPM_Datatype {
CPM_CHAR = 1, CPM_UNSIGNED_CHAR = 2, CPM_BYTE = 3, CPM_SHORT = 4,
CPM_UNSIGNED_SHORT = 5, CPM_INT = 6, CPM_UNSIGNED = 7, CPM_LONG = 8,
CPM_UNSIGNED_LONG = 9, CPM_FLOAT = 10, CPM_DOUBLE = 11, CPM_LONG_DOUBLE = 12,
CPM_REAL = 52 }
- enum CPM_Op {
CPM_MAX = 100, CPM_MIN = 101, CPM_SUM = 102, CPM_PROD = 103,
CPM_LAND = 104, CPM_BAND = 105, CPM_LOR = 106, CPM_BOR = 107,
CPM_LXOR = 108, CPM_BXOR = 109, CPM_MINLOC = 110, CPM_MAXLOC = 111 }

5.2.1 説明

CPM の定義マクロ記述ヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_Define.h](#) で定義されています。

5.2.2 マクロ定義

5.2.2.1 #define _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC)

値:

```
( size_t(_K+_VC) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

3次元インデクス (i,j,k) -> 1次元インデクス変換マクロ

引数

in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1次元インデクス

[cpm_Define.h](#) の 56 行で定義されています。

参照元 [cpm_VoxelInfo::CreateLocalDomainInfo\(\)](#), [cpm_VoxelInfo::CreateNeighborRankInfo\(\)](#), と [cpm_VoxelInfo::CreateRankMap\(\)](#).

5.2.2.2 #define _IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)

値:

```
( size_t(_N) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) * size_t(_NK+2*_VC) \
+ _IDX_S3D(_I,_J,_K,_NI,_NJ,_NK,_VC) \
)
```

4次元インデクス (i,j,k,n) -> 1次元インデクス変換マクロ

引数

in	_I	i 方向インデクス
in	_J	j 方向インデクス
in	_K	k 方向インデクス
in	_N	成分インデクス
in	_NI	i 方向インデクスサイズ
in	_NJ	j 方向インデクスサイズ
in	_NK	k 方向インデクスサイズ
in	_VC	仮想セル数

戻り値

1次元インデクス

cpm_Define.h の 73 行で定義されています。

参照元 cpm_ParaManager::packX(), cpm_ParaManager::packY(), cpm_ParaManager::packZ(), cpm_ParaManager::unpackX(), cpm_ParaManager::unpackY(), と cpm_ParaManager::unpackZ().

5.2.2.3 #define _IDX_S4DEX(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)

値:

```
( size_t(_NN) * _IDX_S3D(_I,_J,_K,_NI,_NJ,_NK,_VC) \
+ size_t(_N) )
```

4次元インデクス (n,i,j,k) -> 1次元インデクス変換マクロ

引数

in	_N	成分インデクス
in	_I	i 方向インデクス
in	_J	j 方向インデクス
in	_K	k 方向インデクス
in	_NN	成分数
in	_NI	i 方向インデクスサイズ
in	_NJ	j 方向インデクスサイズ
in	_NK	k 方向インデクスサイズ
in	_VC	仮想セル数

戻り値

1次元インデクス

cpm_Define.h の 102 行で定義されています。

参照元 cpm_ParaManager::packXEx(), cpm_ParaManager::packYEx(), cpm_ParaManager::packZEx(), cpm_ParaManager::unpackXEx(), cpm_ParaManager::unpackYEx(), と cpm_ParaManager::unpackZEx().

5.2.2.4 `#define _IDX_V3D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)(_IDX_S4D(_I,_J,_K,_N,_NI,_NJ,_NK,_VC))`

3次元インデクス (i,j,k,3) -> 1次元インデクス変換マクロ

引数

in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_N</code>	成分インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

cpm_Define.h の 88 行で定義されています。

5.2.2.5 `#define _IDX_V3DEX(_N, _I, _J, _K, _NI, _NJ, _NK, _VC)(_IDX_S4DEX(_N,_I,_J,_K,3,_NI,_NJ,_NK,_VC))`

3次元インデクス (3,i,j,k) -> 1次元インデクス変換マクロ

引数

in	<code>_N</code>	成分インデクス
in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

cpm_Define.h の 116 行で定義されています。

5.2.2.6 `#define REAL_BUF_TYPE REAL_TYPE`

袖通信バッファの型指定

- ・ デフォルトでは、`REAL_BUF_TYPE=REAL_TYPE`
- ・ コンパイル時オプション-D_BUFSIZE_DOUBLE_を付与することで `REAL_BUF_TYPE=double` になる
- ・ コンパイル時オプション-D_BUFSIZE_LONG_DOUBLE_を付与することで `REAL_BUF_TYPE=long double` になる

cpm_Define.h の 42 行で定義されています。

参照元 `cpm_ParaManager::GetBndCommBufferSize()`, と `cpm_ParaManager::SetBndCommBuffer()`.

5.2.2.7 `#define REAL_TYPE float`

実数型の指定

- ・ デフォルトでは、`REAL_TYPE=float`
- ・ コンパイル時オプション-D_REAL_IS_DOUBLE_を付与することで `REAL_TYPE=double` になる

cpm_Define.h の 27 行で定義されています。

参照元 cpm_ParaManager::AllocRealS4D(), cpm_GetGlobalOrigin_(), cpm_GetGlobalRegion_(), cpm_GetLocalOrigin_(), cpm_GetLocalRegion_(), cpm_GetPitch_(), cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadDomainInfo(), cpm_Base::ReallsDouble(), と cpm_ParaManager::VoxelInit().

5.2.3 列挙型

5.2.3.1 enum CPM_Datatype

fortran 用のデータタイプ

列挙型の値:

CPM_CHAR char
CPM_UNSIGNED_CHAR unsigned char
CPM_BYTE byte(not support)
CPM_SHORT short
CPM_UNSIGNED_SHORT unsigned short
CPM_INT int
CPM_UNSIGNED unsigned
CPM_LONG long
CPM_UNSIGNED_LONG unsigned long
CPM_FLOAT float
CPM_DOUBLE double
CPM_LONG_DOUBLE long double
CPM_REAL REAL_TYPE.

cpm_Define.h の 232 行で定義されています。

5.2.3.2 enum cpm_DirFlag

軸方向フラグ

列挙型の値:

X_DIR X direction.
Y_DIR Y direction.
Z_DIR Z direction.

cpm_Define.h の 130 行で定義されています。

5.2.3.3 enum cpm_ErrorCode

CPM のエラーコード

列挙型の値:

CPM_SUCCESS 正常終了
CPM_ERROR その他のエラー
CPM_ERROR_PM_INSTANCE 並列管理クラス cpm_ParaManager のインスタンス失敗
CPM_ERROR_INVALID_PTR ポインタのエラー

CPM_ERROR_INVALID_DOMAIN_NO 領域番号が不正

CPM_ERROR_INVALID_OBJKEY 指定登録番号のオブジェクトが存在しない

CPM_ERROR_REGIST_OBJKEY オブジェクト登録に失敗:

CPM_ERROR_TEXTPARSER テキストパーサーに関するエラー

CPM_ERROR_NO_TEXTPARSER テキストパーサーを組み込んでいない

CPM_ERROR_TP_NOVECTOR 領域分割情報ファイルのベクトルデータ読み込みエラー

CPM_ERROR_TP_VECTOR_SIZE 領域分割情報ファイルのベクトルデータのサイズが不正

CPM_ERROR_TP_INVALID_G_ORG 領域分割情報ファイルのドメイン原点情報が不正

CPM_ERROR_TP_INVALID_G_VOXEL 領域分割情報ファイルのドメインVOXEL 数情報が不正

CPM_ERROR_TP_INVALID_G_PITCH 領域分割情報ファイルのドメインピッチ情報が不正

CPM_ERROR_TP_INVALID_G_RGN 領域分割情報ファイルのドメイン空間サイズ情報が不正

CPM_ERROR_TP_INVALID_G_DIV 領域分割情報ファイルのドメイン領域分割数情報が不正

CPM_ERROR_TP_INVALID_POS 領域分割情報ファイルのサブドメイン位置情報が不正

CPM_ERROR_TP_INVALID_BCID 領域分割情報ファイルのサブドメインBCID 情報が不正

CPM_ERROR_VOXELINIT Voxellnit でエラー

CPM_ERROR_NOT_IN_PROCGROUP 自ランクがプロセスグループに含まれていない

CPM_ERROR_ALREADY_VOXELINIT 指定されたプロセスグループが既に領域分割済み:

CPM_ERROR_MISMATCH_NP_SUBDOMAIN 並列数とサブドメイン数が一致していない

CPM_ERROR_CREATE_RANKMAP ランクマップ生成に失敗

CPM_ERROR_CREATE_NEIGHBOR 隣接ランク情報生成に失敗

CPM_ERROR_CREATE_LOCALDOMAIN ローカル領域情報生成に失敗

CPM_ERROR_INSERT_VOXELMAP 領域情報のマップへの登録失敗

CPM_ERROR_CREATE_PROCGROUP プロセスグループ生成に失敗

CPM_ERROR_INVALID_VOXELSIZE VOXEL 数が不正

CPM_ERROR_GET_INFO 情報取得系関数でエラー

CPM_ERROR_GET_DIVNUM 領域分割数の取得エラー

CPM_ERROR_GET_PITCH ピッチの取得エラー

CPM_ERROR_GET_GLOBALVOXELSIZE 全体ボクセル数の取得エラー

CPM_ERROR_GET_GLOBALORIGIN 全体空間の原点の取得エラー

CPM_ERROR_GET_GLOBALREGION 全体空間サイズの取得エラー

CPM_ERROR_GET_LOCALVOXELSIZE 自ランクのボクセル数の取得エラー

CPM_ERROR_GET_LOCALORIGIN 自ランクの空間原点の取得エラー

CPM_ERROR_GET_LOCALREGION 自ランクの空間サイズの取得エラー

CPM_ERROR_GET_DIVPOS 自ランクの領域分割位置の取得エラー

CPM_ERROR_GET_BCID 自ランクのBCID の取得エラー

CPM_ERROR_GET_HEADINDEX 始点インデックスの取得エラー

CPM_ERROR_GET_TAILINDEX 終点インデックスの取得エラー

CPM_ERROR_GET_NEIGHBOR_RANK 隣接ランク番号の取得エラー

CPM_ERROR_GET_PERIODIC_RANK 周期境界位置の隣接ランク番号の取得エラー

CPM_ERROR_GET_MYRANK ランク番号の取得エラー

CPM_ERROR_GET_NUMRANK ランク数の取得エラー

CPM_ERROR_MPI MPI のエラー

CPM_ERROR_NO_MPI_INIT MPI_Init がコールされていない

CPM_ERROR_MPI_BARRIER MPI_Barrier でエラー

CPM_ERROR_MPI_BCAST MPI_Bcast でエラー

CPM_ERROR_MPI_SEND MPI_Send でエラー
CPM_ERROR_MPI_RECV MPI_Recv でエラー
CPM_ERROR_MPI_ISEND MPI_Isend でエラー
CPM_ERROR_MPI_Irecv MPI_Irecv でエラー
CPM_ERROR_MPI_WAIT MPI_Wait でエラー
CPM_ERROR_MPI_WAITALL MPI_Waitall でエラー
CPM_ERROR_MPI_ALLREDUCE MPI_Allreduce でエラー
CPM_ERROR_MPI_GATHER MPI_Gather でエラー
CPM_ERROR_MPI_ALLGATHER MPI_Allgather でエラー
CPM_ERROR_MPI_GATHERV MPI_Gatherv でエラー
CPM_ERROR_MPI_ALLGATHERV MPI_Allgatherv でエラー
CPM_ERROR_MPI_DIMSCREATE MPI_Dims_create でエラー
CPM_ERROR_BNDCOMM BndComm でエラー
CPM_ERROR_BNDCOMM_VOXELSIZE VoxelSize 取得でエラー
CPM_ERROR_BNDCOMM_BUFFER 袖通信バッファ取得でエラー
CPM_ERROR_BNDCOMM_BUFFERLENGTH 袖通信バッファサイズが足りない
CPM_ERROR_PERIODIC PeriodicComm でエラー
CPM_ERROR_PERIODIC_INVALID_DIR 不正な軸方向フラグが指定された
CPM_ERROR_PERIODIC_INVALID_PM 不正な正負方向フラグが指定された
CPM_ERROR_MPI_INVALID_COMM MPI コミュニケータが不正
CPM_ERROR_MPI_INVALID_DATATYPE 対応しない型が指定された
CPM_ERROR_MPI_INVALID_OPERATOR 対応しないオペレータが指定された
CPM_ERROR_MPI_INVALID_REQUEST 不正なリクエストが指定された

cpm_Define.h の 146 行で定義されています。

5.2.3.4 enum cpm_FaceFlag

面フラグ

列挙型の値:

X_MINUS -X face
Y_MINUS -Y face
Z_MINUS -Z face
X_PLUS +X face
Y_PLUS +Y face
Z_PLUS +Z face

cpm_Define.h の 119 行で定義されています。

5.2.3.5 enum CPM_Op

fortran 用のオペレータ

列挙型の値:

CPM_MAX 最大値
CPM_MIN 最小値

CPM_SUM 和
CPM_PROD 積
CPM LAND 論理積
CPM_BAND ビット演算の積
CPM_LOR 論理和
CPM_BOR ビット演算の和
CPM_LXOR 排他的論理和
CPM_BXOR ビット演算の排他的論理和
CPM_MINLOC 最大値と位置 (not support)
CPM_MAXLOC 最小値と位置 (not support)

cpm_Define.h の 259 行で定義されています。

5.2.3.6 enum cpm_PMFlag

方向フラグ

列挙型の値:

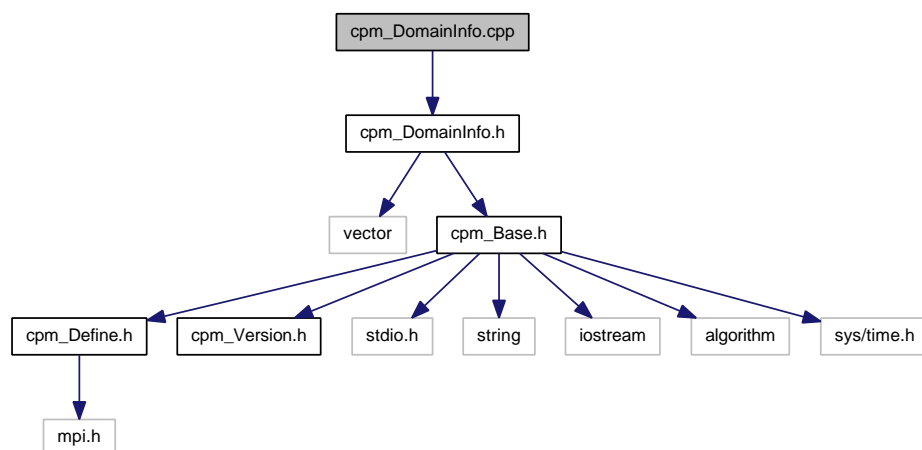
PLUS2MINUS plus -> minus direction
MINUS2PLUS minus -> plus direction
BOTH plus <-> minus direction

cpm_Define.h の 138 行で定義されています。

5.3 cpm_DomainInfo.cpp

```
#include "cpm_DomainInfo.h"
```

cpm_DomainInfo.cpp のインクルード依存関係図



5.3.1 説明

DomainInfo クラスのソースファイル

作者

University of Tokyo

日付

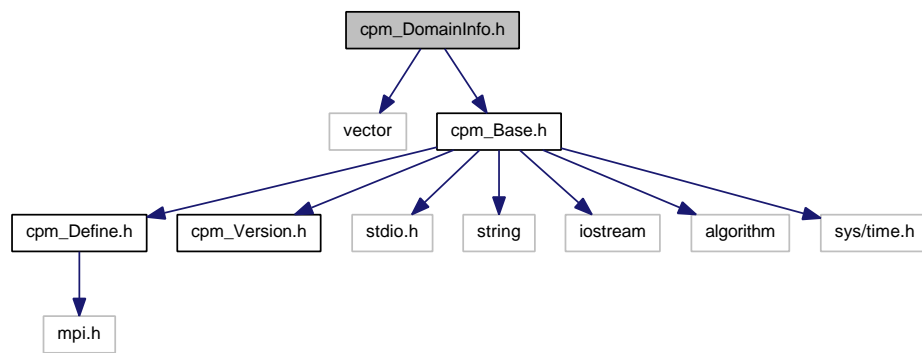
2012/05/31

[cpm_DomainInfo.cpp](#) で定義されています。

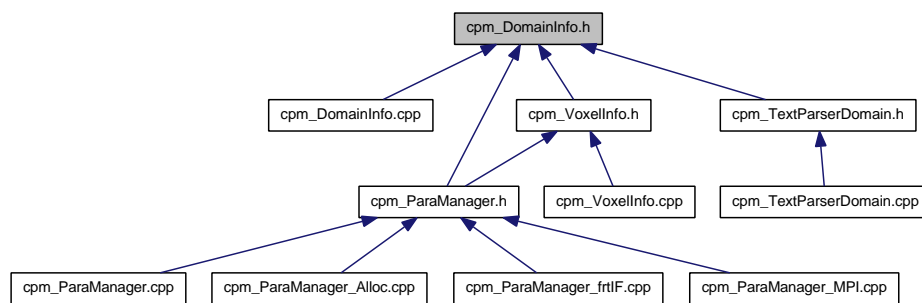
5.4 cpm_DomainInfo.h

```
#include <vector>
#include "cpm_Base.h"
```

cpm_DomainInfo.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class [cpm_DomainInfo](#)
- class [cpm_ActiveSubDomainInfo](#)
- class [cpm_GlobalDomainInfo](#)
- class [cpm_LocalDomainInfo](#)

5.4.1 説明

領域情報クラスのヘッダーファイル

作者

University of Tokyo

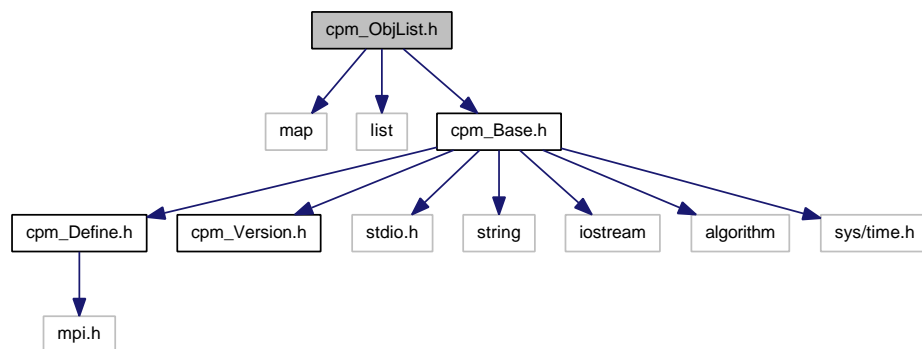
日付

2012/05/31

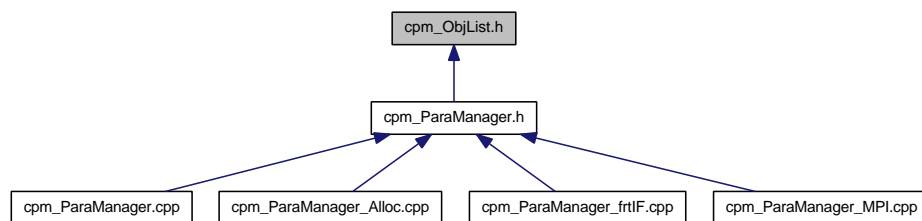
[cpm_DomainInfo.h](#) で定義されています。

5.5 cpm_ObjList.h

```
#include <map>
#include <list>
#include "cpm_Base.h"
cpm_ObjList.h のインクルード依存関係図
```



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class [cpm_ObjList< T >](#)

型定義

- typedef std::map< int, int * > [RankNoMap](#)

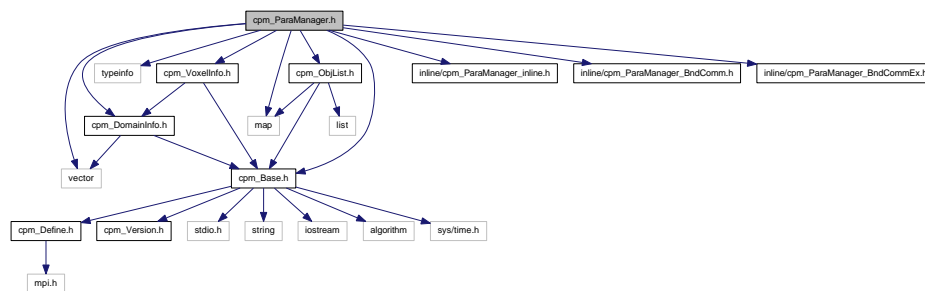
5.5.1 説明

汎用オブジェクトの管理クラスのヘッダーファイル

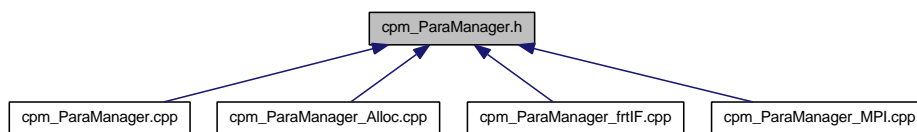
5.7 cpm_ParaManager.h

```
#include <map>
#include <vector>
#include <typeinfo>
#include "cpm_Base.h"
#include "cpm_DomainInfo.h"
#include "cpm_VoxelInfo.h"
#include "cpm_ObjList.h"
#include "inline/cpm_ParaManager_inline.h"
#include "inline/cpm_ParaManager_BndComm.h"
#include "inline/cpm_ParaManager_BndCommEx.h"
```

cpm_ParaManager.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- struct [S_BNDCOMM_BUFFER](#)
- class [cpm_ParaManager](#)

型定義

- typedef std::map< int,
 [cpm_VoxelInfo](#) * > [VoxelInfoMap](#)
- typedef std::map< int, int * > [RankNoMap](#)
- typedef std::map< int,
 [S_BNDCOMM_BUFFER](#) * > [BndCommInfoMap](#)

5.7.1 説明

パラレルマネージャクラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_ParaManager.h](#) で定義されています。

5.7.2 型定義

5.7.2.1 typedef std::map<int, S_BNDCOMM_BUFFER*> BndCommInfoMap

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManager.h の 68 行で定義されています。

5.7.2.2 typedef std::map<int, int*> RankNoMap

プロセスグループ毎のランク番号マップ

cpm_ParaManager.h の 30 行で定義されています。

5.7.2.3 typedef std::map<int, cpm_VoxelInfo*> VoxelInfoMap

プロセスグループ毎のVOXEL 空間情報管理マップ

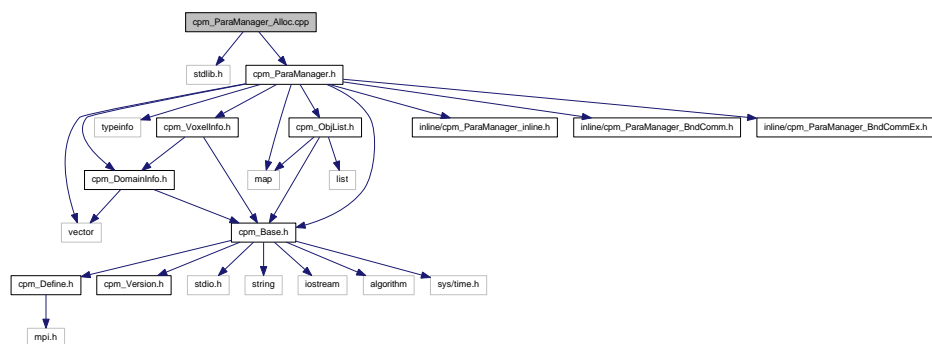
cpm_ParaManager.h の 27 行で定義されています。

5.8 cpm_ParaManager_Alloc.cpp

```
#include <stdlib.h>
```

```
#include "cpm_ParaManager.h"
```

cpm_ParaManager_Alloc.cpp のインクルード依存関係図



5.8.1 説明

パラレルマネージャクラスのソースファイル

作者

University of Tokyo

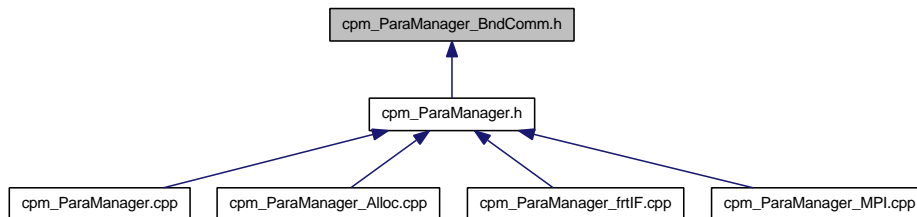
日付

2012/05/31

[cpm_ParaManager_Alloc.cpp](#) で定義されています。

5.9 cpm_ParaManager_BndComm.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



マクロ定義

- `#define _IDXFX(_I, _J, _K, _N, _IS, _NJ, _NK, _VC)`
- `#define _IDXFY(_I, _J, _K, _N, _NI, _JS, _NK, _VC)`
- `#define _IDXFZ(_I, _J, _K, _N, _NI, _NJ, _KS, _VC)`

5.9.1 説明

パラレルマネージャクラスのインラインヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_ParaManager_BndComm.h](#) で定義されています。

5.9.2 マクロ定義

5.9.2.1 `#define _IDXFX(_I, _J, _K, _N, _IS, _NJ, _NK, _VC)`

値:

```
( size_t(_N)      * size_t(_VC) * size_t(_NJ+2*_VC) * size_t(_NK+2*_VC) \
+ size_t(_K+_VC) * size_t(_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_VC) \
+ size_t(_I-(_IS)) \
)
```

`cpm_ParaManager_BndComm.h` の 18 行で定義されています。

参照元 `cpm_ParaManager::packX()`, と `cpm_ParaManager::unpackX()`.

5.9.2.2 #define _IDXFY(_I, _J, _K, _N, _NI, _JS, _NK, _VC)

値:

```
( size_t(_N)          * size_t(_NI+2*_VC) * size_t(_VC) * size_t(_NK+2*_VC) \
+ size_t(_K+_VC)      * size_t(_NI+2*_VC) * size_t(_VC) \
+ size_t(_J-(_JS))    * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

cpm_ParaManager_BndComm.h の 25 行で定義されています。

参照元 cpm_ParaManager::packY(), と cpm_ParaManager::unpackY().

5.9.2.3 #define _IDXFZ(_I, _J, _K, _N, _NI, _NJ, _KS, _VC)

値:

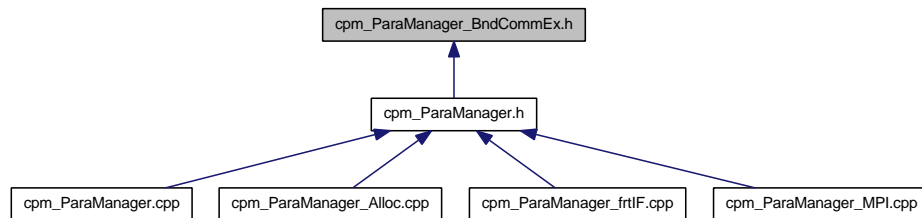
```
( size_t(_N)          * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) * size_t(_VC) \
+ size_t(_K-(_KS))    * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC)      * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

cpm_ParaManager_BndComm.h の 32 行で定義されています。

参照元 cpm_ParaManager::packZ(), と cpm_ParaManager::unpackZ().

5.10 cpm_ParaManager_BndCommEx.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



マクロ定義

- #define _IDXF(_N, _I, _J, _K, _NN, _IS, _NJ, _NK, _VC)
- #define _IDXFY(_N, _I, _J, _K, _NN, _NI, _JS, _NK, _VC)
- #define _IDXFZ(_N, _I, _J, _K, _NN, _NI, _NJ, _KS, _VC)

5.10.1 説明

パラレルマネージャクラスのインラインヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_ParaManager_BndCommEx.h](#) で定義されています。

5.10.2 マクロ定義

5.10.2.1 #define _IDAFX(_N, _I, _J, _K, _NN, _IS, _NJ, _NK, _VC)

値:

```
( size_t(_NN) \
* ( size_t(_K+_VC) * size_t(_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_VC) \
+ size_t(_I-(_IS)) \
) \
+ size_t(_N) \
)
```

[cpm_ParaManager_BndCommEx.h](#) の 18 行で定義されています。

参照元 [cpm_ParaManager::packXEx\(\)](#), と [cpm_ParaManager::unpackXEx\(\)](#).

5.10.2.2 #define _IDXFY(_N, _I, _J, _K, _NN, _NI, _JS, _NK, _VC)

値:

```
( size_t(_NN) \
* ( size_t(_K+_VC) * size_t(_NI+2*_VC) * size_t(_VC) \
+ size_t(_J-(_JS)) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
) \
+ size_t(_N) \
)
```

[cpm_ParaManager_BndCommEx.h](#) の 27 行で定義されています。

参照元 [cpm_ParaManager::packYEx\(\)](#), と [cpm_ParaManager::unpackYEx\(\)](#).

5.10.2.3 #define _IDXFZ(_N, _I, _J, _K, _NN, _NI, _NJ, _KS, _VC)

値:

```
( size_t(_NN) \
* ( size_t(_K-(_KS)) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
) \
+ size_t(_N) \
)
```

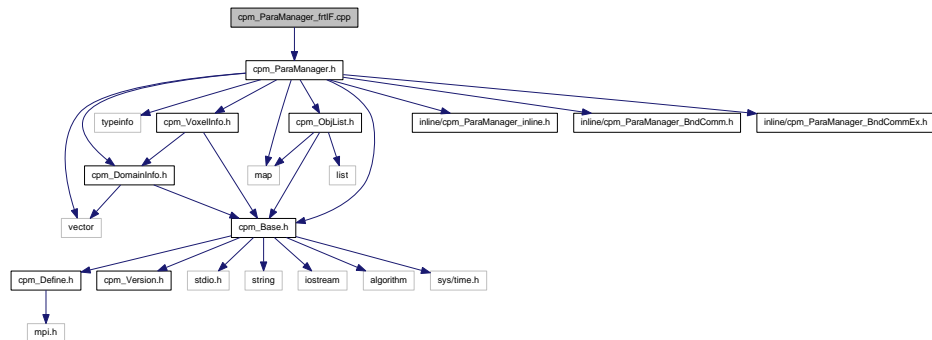
[cpm_ParaManager_BndCommEx.h](#) の 36 行で定義されています。

参照元 [cpm_ParaManager::packZEx\(\)](#), と [cpm_ParaManager::unpackZEx\(\)](#).

5.11 cpm_ParaManager_frtIF.cpp

```
#include "cpm_ParaManager.h"
```

cpm_ParaManager_frtIF.cpp のインクルード依存関係図



マクロ定義

- #define CPM_EXTERN extern "C"
- #define cpm_Initialize_ cpm_initialize_
- #define cpm_Voxellnit_ cpm_voxelinit_
- #define cpm_Voxellnit_nodiv_ cpm_voxelinit_nodiv_
- #define cpm_IsParallel_ cpm_isparallel_
- #define cpm_GetDivNum_ cpm_getdivnum_
- #define cpm_GetPitch_ cpm_getpitch_
- #define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_
- #define cpm_GetGlobalOrigin_ cpm_getglobalorigin_
- #define cpm_GetGlobalRegion_ cpm_getglobalregion_
- #define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_
- #define cpm_GetLocalOrigin_ cpm_getlocalorigin_
- #define cpm_GetLocalRegion_ cpm_getlocalregion_
- #define cpm_GetDivPos_ cpm_getdivpos_
- #define cpm_GetBCID_ cpm_getbcid_
- #define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_
- #define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_
- #define cpm_GetNeighborRankID_ cpm_getneighborrankid_
- #define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_
- #define cpm_GetMyRankID_ cpm_getmyrankid_
- #define cpm_GetNumRank_ cpm_getnumrank_
- #define cpm_Abort_ cpm_abort_
- #define cpm_Barrier_ cpm_barrier_
- #define cpm_Wait_ cpm_wait_
- #define cpm_Waitall_ cpm_waitall_
- #define cpm_Bcast_ cpm_bcast_
- #define cpm_Send_ cpm_send_
- #define cpm_Recv_ cpm_recv_
- #define cpm_Isend_ cpm_isend_
- #define cpm_Irecv_ cpm_irecv_
- #define cpm_Allreduce_ cpm_allreduce_
- #define cpm_Gather_ cpm_gather_
- #define cpm_Allgather_ cpm_allgather_
- #define cpm_Gatherv_ cpm_gatherv_
- #define cpm_Allgatherv_ cpm_allgatherv_

- `#define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_`
- `#define cpm_BndCommsS3D_ cpm_bndcomms3d_`
- `#define cpm_BndCommV3D_ cpm_bndcommv3d_`
- `#define cpm_BndCommsS4D_ cpm_bndcomms4d_`
- `#define cpm_BndCommsS3D_nowait_ cpm_bndcomms3d_nowait_`
- `#define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_`
- `#define cpm_BndCommsS4D_nowait_ cpm_bndcomms4d_nowait_`
- `#define cpm_wait_BndCommsS3D_ cpm_wait_bndcomms3d_`
- `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`
- `#define cpm_wait_BndCommsS4D_ cpm_wait_bndcomms4d_`
- `#define cpm_BndCommV3DEx_ cpm_bndcommv3dex_`
- `#define cpm_BndCommsS4DEx_ cpm_bndcomms4dex_`
- `#define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_`
- `#define cpm_BndCommsS4DEx_nowait_ cpm_bndcomms4dex_nowait_`
- `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`
- `#define cpm_wait_BndCommsS4DEx_ cpm_wait_bndcomms4dex_`
- `#define cpm_PeriodicCommsS3D_ cpm_periodiccomms3d_`
- `#define cpm_PeriodicCommV3D_ cpm_periodiccommv3d_`
- `#define cpm_PeriodicCommsS4D_ cpm_periodiccomms4d_`
- `#define cpm_PeriodicCommV3DEx_ cpm_periodiccommv3dex_`
- `#define cpm_PeriodicCommsS4DEx_ cpm_periodiccomms4dex_`

関数

- `CPM_EXTERN void cpm_Initialize_ (int *ierr)`
- `CPM_EXTERN void cpm_Voxellnit_ (int *div, int *vox, REAL_TYPE *origin, REAL_TYPE *pitch, int *obcid, int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Voxellnit_nodiv_ (int *vox, REAL_TYPE *origin, REAL_TYPE *pitch, int *obcid, int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_IsParallel_ (int *ipara, int *ierr)`
- `CPM_EXTERN void cpm_GetDivNum_ (int *div, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetPitch_ (REAL_TYPE *pch, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalVoxelSize_ (int *wsz, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalOrigin_ (REAL_TYPE *worg, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalRegion_ (REAL_TYPE *wrgn, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalVoxelSize_ (int *lsz, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalOrigin_ (REAL_TYPE *lorg, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalRegion_ (REAL_TYPE *lrgn, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetDivPos_ (int *pos, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetBCID_ (int *bcid, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int *idx, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetVoxelTailIndex_ (int *idx, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetNeighborRankID_ (int *nID, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetPeriodicRankID_ (int *nID, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetMyRankID_ (int *id, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetNumRank_ (int *nrank, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Abort_ (int *errorcode)`
- `CPM_EXTERN void cpm_Barrier_ (int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Wait_ (int *reqNo, int *ierr)`
- `CPM_EXTERN void cpm_Waitall_ (int *count, int *reqlist, int *ierr)`
- `CPM_EXTERN void cpm_Bcast_ (void *buf, int *count, int *datatype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Send_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Recv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *ierr)`

- `CPM_EXTERN void cpm_Isend_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *reqNo, int *ierr)`
- `CPM_EXTERN void cpm_Irecv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *reqNo, int *ierr)`
- `CPM_EXTERN void cpm_Allreduce_ (void *sendbuf, void *recvbuf, int *count, int *datatype, int *op, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Gather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Allgather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Gatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Allgatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_SetBndCommBuffer_ (int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS4D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS3D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommsS4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommsS3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommsS4DEx_nowait_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3DEx_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommsS4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommsS4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommsS3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommsS4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`

5.11.1 説明

パラレルマネージャクラスのFortran インターフェイスのソースファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_ParaManager_frtIF.cpp](#) で定義されています。

5.11.2 マクロ定義

5.11.2.1 `#define cpm_Abort_ cpm_abort_`

`cpm_ParaManager_frtIF.cpp` の 45 行で定義されています。

5.11.2.2 `#define cpm_Allgather_ cpm_allgather_`

`cpm_ParaManager_frtIF.cpp` の 56 行で定義されています。

5.11.2.3 `#define cpm_Allgatherv_ cpm_allgatherv_`

`cpm_ParaManager_frtIF.cpp` の 58 行で定義されています。

5.11.2.4 `#define cpm_Allreduce_ cpm_allreduce_`

`cpm_ParaManager_frtIF.cpp` の 54 行で定義されています。

5.11.2.5 `#define cpm_Barrier_ cpm_barrier_`

`cpm_ParaManager_frtIF.cpp` の 46 行で定義されています。

5.11.2.6 `#define cpm_Bcast_ cpm_bcast_`

`cpm_ParaManager_frtIF.cpp` の 49 行で定義されています。

5.11.2.7 `#define cpm_BndComms3D_ cpm_bndcomms3d_`

`cpm_ParaManager_frtIF.cpp` の 60 行で定義されています。

5.11.2.8 `#define cpm_BndComms3D_nowait_ cpm_bndcomms3d_nowait_`

`cpm_ParaManager_frtIF.cpp` の 63 行で定義されています。

5.11.2.9 #define cpm_BndCommsS4D_ cpm_bndcomms4d_

cpm_ParaManager_frtIF.cpp の 62 行で定義されています。

参照元 cpm_BndCommS3D_(), と cpm_BndCommV3D_().

5.11.2.10 #define cpm_BndCommsS4D_nowait_ cpm_bndcomms4d_nowait_

cpm_ParaManager_frtIF.cpp の 65 行で定義されています。

参照元 cpm_BndCommS3D_nowait_(), と cpm_BndCommV3D_nowait_().

5.11.2.11 #define cpm_BndCommsS4DEx_ cpm_bndcomms4dex_

cpm_ParaManager_frtIF.cpp の 70 行で定義されています。

参照元 cpm_BndCommV3DEx_().

5.11.2.12 #define cpm_BndCommsS4DEx_nowait_ cpm_bndcomms4dex_nowait_

cpm_ParaManager_frtIF.cpp の 72 行で定義されています。

参照元 cpm_BndCommV3DEx_nowait_().

5.11.2.13 #define cpm_BndCommV3D_ cpm_bndcommv3d_

cpm_ParaManager_frtIF.cpp の 61 行で定義されています。

5.11.2.14 #define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_

cpm_ParaManager_frtIF.cpp の 64 行で定義されています。

5.11.2.15 #define cpm_BndCommV3DEx_ cpm_bndcommv3dex_

cpm_ParaManager_frtIF.cpp の 69 行で定義されています。

5.11.2.16 #define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_

cpm_ParaManager_frtIF.cpp の 71 行で定義されています。

5.11.2.17 #define CPM_EXTERN extern "C"

extern 宣言

cpm_ParaManager_frtIF.cpp の 17 行で定義されています。

5.11.2.18 #define cpm_Gather_ cpm_gather_

cpm_ParaManager_frtIF.cpp の 55 行で定義されています。

5.11.2.19 #define cpm_Gatherv_ cpm_gatherv_

cpm_ParaManager_frtIF.cpp の 57 行で定義されています。

5.11.2.20 `#define cpm_GetBCID_ cpm_getbcid_`

`cpm_ParaManager_frtIF.cpp` の 38 行で定義されています。

5.11.2.21 `#define cpm_GetDivNum_ cpm_getdivnum_`

`cpm_ParaManager_frtIF.cpp` の 29 行で定義されています。

5.11.2.22 `#define cpm_GetDivPos_ cpm_getdivpos_`

`cpm_ParaManager_frtIF.cpp` の 37 行で定義されています。

5.11.2.23 `#define cpm_GetGlobalOrigin_ cpm_getglobalorigin_`

`cpm_ParaManager_frtIF.cpp` の 32 行で定義されています。

5.11.2.24 `#define cpm_GetGlobalRegion_ cpm_getglobalregion_`

`cpm_ParaManager_frtIF.cpp` の 33 行で定義されています。

5.11.2.25 `#define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_`

`cpm_ParaManager_frtIF.cpp` の 31 行で定義されています。

5.11.2.26 `#define cpm_GetLocalOrigin_ cpm_getlocalorigin_`

`cpm_ParaManager_frtIF.cpp` の 35 行で定義されています。

5.11.2.27 `#define cpm_GetLocalRegion_ cpm_getlocalregion_`

`cpm_ParaManager_frtIF.cpp` の 36 行で定義されています。

5.11.2.28 `#define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_`

`cpm_ParaManager_frtIF.cpp` の 34 行で定義されています。

5.11.2.29 `#define cpm_GetMyRankID_ cpm_getmyrankid_`

`cpm_ParaManager_frtIF.cpp` の 43 行で定義されています。

5.11.2.30 `#define cpm_GetNeighborRankID_ cpm_getneighborrankid_`

`cpm_ParaManager_frtIF.cpp` の 41 行で定義されています。

5.11.2.31 `#define cpm_GetNumRank_ cpm_getnumrank_`

`cpm_ParaManager_frtIF.cpp` の 44 行で定義されています。

5.11.2.32 `#define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_`

cpm_ParaManager_frtIF.cpp の 42 行で定義されています。

5.11.2.33 `#define cpm_GetPitch_ cpm_getpitch_`

cpm_ParaManager_frtIF.cpp の 30 行で定義されています。

5.11.2.34 `#define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_`

cpm_ParaManager_frtIF.cpp の 39 行で定義されています。

5.11.2.35 `#define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_`

cpm_ParaManager_frtIF.cpp の 40 行で定義されています。

5.11.2.36 `#define cpm_Initialize_ cpm_initialize_`

cpm_ParaManager_frtIF.cpp の 25 行で定義されています。

5.11.2.37 `#define cpm_Irecv_ cpm_irecv_`

cpm_ParaManager_frtIF.cpp の 53 行で定義されています。

5.11.2.38 `#define cpm_Isend_ cpm_isend_`

cpm_ParaManager_frtIF.cpp の 52 行で定義されています。

5.11.2.39 `#define cpm_IsParallel_ cpm_isparallel_`

cpm_ParaManager_frtIF.cpp の 28 行で定義されています。

5.11.2.40 `#define cpm_PeriodicCommsS3D cpm_periodiccomms3d_`

cpm_ParaManager_frtIF.cpp の 75 行で定義されています。

5.11.2.41 `#define cpm_PeriodicCommsS4D cpm_periodiccomms4d_`

cpm_ParaManager_frtIF.cpp の 77 行で定義されています。

5.11.2.42 `#define cpm_PeriodicCommsS4DEx cpm_periodiccomms4dex_`

cpm_ParaManager_frtIF.cpp の 79 行で定義されています。

5.11.2.43 `#define cpm_PeriodicCommV3D cpm_periodiccommv3d_`

cpm_ParaManager_frtIF.cpp の 76 行で定義されています。

5.11.2.44 `#define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_`

cpm_ParaManager_frtIF.cpp の 78 行で定義されています。

5.11.2.45 `#define cpm_Recv_ cpm_recv_`

cpm_ParaManager_frtIF.cpp の 51 行で定義されています。

5.11.2.46 `#define cpm_Send_ cpm_send_`

cpm_ParaManager_frtIF.cpp の 50 行で定義されています。

5.11.2.47 `#define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_`

cpm_ParaManager_frtIF.cpp の 59 行で定義されています。

5.11.2.48 `#define cpm_Voxellnit_ cpm_voxellnit_`

cpm_ParaManager_frtIF.cpp の 26 行で定義されています。

5.11.2.49 `#define cpm_Voxellnit_nodiv_ cpm_voxellnit_nodiv_`

cpm_ParaManager_frtIF.cpp の 27 行で定義されています。

5.11.2.50 `#define cpm_Wait_ cpm_wait_`

cpm_ParaManager_frtIF.cpp の 47 行で定義されています。

5.11.2.51 `#define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_`

cpm_ParaManager_frtIF.cpp の 66 行で定義されています。

5.11.2.52 `#define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_`

cpm_ParaManager_frtIF.cpp の 68 行で定義されています。

参照元 `cpm_wait_BndCommS3D_()`, と `cpm_wait_BndCommV3D_()`.

5.11.2.53 `#define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_`

cpm_ParaManager_frtIF.cpp の 74 行で定義されています。

参照元 `cpm_wait_BndCommV3DEx_()`.

5.11.2.54 `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`

cpm_ParaManager_frtIF.cpp の 67 行で定義されています。

5.11.2.55 `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`

cpm_ParaManager_frtIF.cpp の 73 行で定義されています。

5.11.2.56 `#define cpm_Waitall_ cpm_waitall_`

cpm_ParaManager_frtIF.cpp の 48 行で定義されています。

5.11.3 関数

5.11.3.1 CPM_EXTERN void cpm_Abort_(int * *errorcode*)

Abort

- Abort のFortran インターフェイス関数

引数

in	<i>errorcode</i>	MPI_Abort に渡すエラーコード
----	------------------	---------------------

cpm_ParaManager_frtIF.cpp の 961 行で定義されています。

参照先 cpm_ParaManager::Abort(), と cpm_ParaManager::get_instance().

5.11.3.2 CPM_EXTERN void cpm_Allgather_(void * *sendbuf*, int * *sendcnt*, int * *sendtype*, void * *recvbuf*, int * *recvcnt*, int * *recvtype*, int * *procGrpNo*, int * *ierr*)

MPI_Allgather のFortran インターフェイス

- MPI_Allgather のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	受信データのサイズ
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1378 行で定義されています。

参照先 cpm_ParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_
Datatype().

5.11.3.3 CPM_EXTERN void cpm_Allgatherv_(void * *sendbuf*, int * *sendcnt*, int * *sendtype*, void * *recvbuf*, int * *recvcnts*, int * *displs*, int * *recvtype*, int * *procGrpNo*, int * *ierr*)

MPI_Allgatherv のFortran インターフェイス

- MPI_Allgatherv のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ

in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1480 行で定義されています。

参照先 cpm_ParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_
Datatype().

**5.11.3.4 CPM_EXTERN void cpm_Allreduce_ (void * *sendbuf*, void * *recvbuf*, int * *count*, int * *datatype*, int * *op*, int
* *procGrpNo*, int * *ierr*)**

MPI_Allreduce のFortran インターフェイス

- MPI_Allreduce のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
out	<i>recvbuf</i>	受信データ
in	<i>count</i>	送受信データのサイズ
in	<i>datatype</i>	送受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>op</i>	オペレータ
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1277 行で定義されています。

参照先 cpm_ParaManager::Allreduce(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_MPI_INVALID_OPERATOR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_
instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::GetMPI_Op().

5.11.3.5 CPM_EXTERN void cpm_Barrier_ (int * *procGrpNo*, int * *ierr*)

Barrier

- Barrier のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 989 行で定義されています。

参照先 cpm_ParaManager::Barrier(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_
ParaManager::get_instance().

5.11.3.6 CPM_EXTERN void cpm_Bcast_ (void * *buf*, int * *count*, int * *datatype*, int * *root*, int * *procGrpNo*, int * *ierr*)

Bcast

- Bcast のFortran インターフェイス関数

引数

	<i>inout]</i>	buf 送受信バッファ
in	<i>count</i>	送信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>root</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1078 行で定義されています。

参照先 cpm_ParaManager::Bcast(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.7 CPM_EXTERN void cpm_BndComms3D_(void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の袖通信を行う
- BndComms3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1608 行で定義されています。

参照先 cpm_ParaManager::BndComms3D(), cpm_BndComms3D_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.8 CPM_EXTERN void cpm_BndComms3D_nowait_(void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の袖通信を行う
- BndComms3D_nowait のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1750 行で定義されています。

参照先 cpm_ParaManager::cpm_BndCommS3D_nowait(), cpm_BndCommS4D_nowait_, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

5.11.3.9 CPM_EXTERN void cpm_BndCommS4D_ (void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommS4D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1563 行で定義されています。

参照先 cpm_ParaManager::BndCommS4D(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.10 CPM_EXTERN void cpm_BndCommS4D_nowait_ (void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommS4D_nowait のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1710 行で定義されています。

参照先 cpm_ParaManager::cpm_BndCommS4D_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

5.11.3.11 CPM_EXTERN void cpm_BndCommS4DEx (void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommS4DEx のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1973 行で定義されています。

参照先 cpm_ParaManager::BndCommS4DEx(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.12 CPM_EXTERN void cpm_BndCommS4DEx_nowait_ (void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommS4DEx_nowait のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2070 行で定義されています。

参照先 `cpm_ParaManager::cpm_BndCommS4DEx_nowait()`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_PM_INSTANCE`, と `cpm_ParaManager::get_instance()`.

5.11.3.13 CPM_EXTERN void cpm_BndCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う
- BndCommV3D のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 1658 行で定義されています。

参照先 `cpm_ParaManager::BndCommV3D()`, `cpm_BndCommS4D_`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_PM_INSTANCE`, `cpm_ParaManager::get_instance()`, と `cpm_ParaManager::GetMPI_Datatype()`.

5.11.3.14 CPM_EXTERN void cpm_BndCommV3D_nowait_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う
- BndCommV3D_nowait のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 1796 行で定義されています。

参照先 `cpm_BndCommS4D_nowait_`, `cpm_ParaManager::cpm_BndCommV3D_nowait()`, `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_PM_INSTANCE`, と `cpm_ParaManager::get_instance()`.

5.11.3.15 **CPM_EXTERN** void **cpm_BndCommV3DEx_**(void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

袖通信 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommV3D のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2018 行で定義されています。

参照先 cpm_ParaManager::BndCommV3DEx(), cpm_BndCommS4DEx_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.16 **CPM_EXTERN** void **cpm_BndCommV3DEx_nowait_**(void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

非同期版袖通信 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommV3D_nowait のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2110 行で定義されています。

参照先 cpm_BndCommS4DEx_nowait_, cpm_ParaManager::cpm_BndCommV3DEx_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

5.11.3.17 **CPM_EXTERN** void **cpm_Gather_**(void * *sendbuf*, int * *sendcnt*, int * *sendtype*, void * *recvbuf*, int * *recvcnt*, int * *recvtype*, int * *root*, int * *procGrpNo*, int * *ierr*)

MPI_Gather のFortran インターフェイス

- MPI_Gather のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	受信データのサイズ
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>root</i>	受信するランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1328 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gather(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.18 CPM_EXTERN void cpm_Gatherv_ (void * *sendbuf*, int * *sendcnt*, int * *sendtype*, void * *recvbuf*, int * *recvcnts*, int * *displs*, int * *recvtype*, int * *root*, int * *procGrpNo*, int * *ierr*)

MPI_Gatherv のFortran インターフェイス

- MPI_Gatherv のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>root</i>	受信するランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1429 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gatherv(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

5.11.3.19 CPM_EXTERN void cpm_GetBCID_ (int * *bcid*, int * *procGrpNo*, int * *ierr*)

自ランクのBCID を取得

- GetBCID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>bcid</i>	自ランクのBCID(6word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 666 行で定義されています。

参照先 CPM_ERROR_GET_BCID, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetBCID().

5.11.3.20 CPM_EXTERN void cpm_GetDivNum_ (int * div, int * procGrpNo, int * ierr)

領域分割数を取得

- GetDivNum のFortran インターフェイス関数

引数

out	div	領域分割数 (3word の整数配列)
in	procGrpNo	プロセスグループ番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 288 行で定義されています。

参照先 CPM_ERROR_GET_DIVNUM, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetDivNum().

5.11.3.21 CPM_EXTERN void cpm_GetDivPos_ (int * pos, int * procGrpNo, int * ierr)

自ランクの領域分割位置を取得

- GetDivPos のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	pos	自ランクの領域分割位置 (3word の整数配列)
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 624 行で定義されています。

参照先 CPM_ERROR_GET_DIVPOS, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetDivPos().

5.11.3.22 CPM_EXTERN void cpm_GetGlobalOrigin_ (REAL_TYPE * worg, int * procGrpNo, int * ierr)

全体空間の原点を取得

- GetGlobalOrigin のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	worg	全体空間の原点 (3word の実数配列)
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 414 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALORIGIN, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), cpm_ParaManager::GetGlobalOrigin(), と REAL_TYPE.

5.11.3.23 CPM_EXTERN void cpm_GetGlobalRegion_ (REAL_TYPE * wrgn, int * procGrpNo, int * ierr)

全体空間サイズを取得

- GetGlobalRegion のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>wrgn</i>	全体空間サイズ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 456 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), cpm_ParaManager::GetGlobalRegion(), と REAL_TYPE.

5.11.3.24 CPM_EXTERN void cpm_GetGlobalVoxelSize_ (int * *wsz*, int * *procGrpNo*, int * *ierr*)

全体ボクセル数を取得

- GetGlobalVoxelSize のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>wsz</i>	全体ボクセル数 (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 372 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalVoxelSize().

5.11.3.25 CPM_EXTERN void cpm_GetLocalOrigin_ (REAL_TYPE * *lorg*, int * *procGrpNo*, int * *ierr*)

自ランクの空間原点を取得

- GetLocalOrigin のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lorg</i>	自ランクの空間原点 (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 540 行で定義されています。

参照先 CPM_ERROR_GET_LOCALORIGIN, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), cpm_ParaManager::GetLocalOrigin(), と REAL_TYPE.

5.11.3.26 CPM_EXTERN void cpm_GetLocalRegion_ (REAL_TYPE * *lrgn*, int * *procGrpNo*, int * *ierr*)

自ランクの空間サイズを取得

- GetLocalRegion のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lrgn</i>	自ランクの空間サイズ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 582 行で定義されています。

参照先 CPM_ERROR_GET_LOCALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), cpm_ParaManager::GetLocalOrigin(), と REAL_TYPE.

5.11.3.27 CPM_EXTERN void cpm_GetLocalVoxelSize_ (int * *lsz*, int * *procGrpNo*, int * *ierr*)

自ランクのボクセル数を取得

- GetLocalVoxelSize のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lsz</i>	自ランクのボクセル数 (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 498 行で定義されています。

参照先 CPM_ERROR_GET_LOCALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalVoxelSize().

5.11.3.28 CPM_EXTERN void cpm_GetMyRankID_ (int * *id*, int * *procGrpNo*, int * *ierr*)

ランク番号の取得

- GetMyRankID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>id</i>	ランク番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 887 行で定義されています。

参照先 CPM_ERROR_GET_MYRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMyRankID().

5.11.3.29 CPM_EXTERN void cpm_GetNeighborRankID_ (int * *nID*, int * *procGrpNo*, int * *ierr*)

自ランクの隣接ランク番号を取得

- GetNeighborRankID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>nID</i>	自ランクの隣接ランク番号 (6word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 797 行で定義されています。

参照先 CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetNeighborRankID().

5.11.3.30 CPM_EXTERN void cpm_GetNumRank_ (int * *nrank*, int * *procGrpNo*, int * *ierr*)

ランク数の取得

- GetNumRank のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>nrank</i>	ランク数
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 925 行で定義されています。

参照先 CPM_ERROR_GET_NUMRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetNumRank().

5.11.3.31 CPM_EXTERN void cpm_GetPeriodicRankID_(int * *nID*, int * *procGrpNo*, int * *ierr*)

自ランクの周期境界の隣接ランク番号を取得

- GetPeriodicRankID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>nID</i>	自ランクの周期境界の隣接ランク番号 6word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 842 行で定義されています。

参照先 CPM_ERROR_GET_PERIODIC_RANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetPeriodicRankID().

5.11.3.32 CPM_EXTERN void cpm_GetPitch_(REAL_TYPE * *pch*, int * *procGrpNo*, int * *ierr*)

ピッチを取得

- GetPitch のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>pch</i>	ピッチ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 330 行で定義されています。

参照先 CPM_ERROR_GET_PITCH, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), cpm_ParaManager::GetPitch(), と REAL_TYPE.

5.11.3.33 CPM_EXTERN void cpm_GetVoxelHeadIndex_(int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの始点VOXEL の全体空間でのインデクスを取得

- GetVoxelHeadIndex のFortran インターフェイス関数
- 全体空間の先頭インデクスを 0 としたC 型のインデクス

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>idx</i>	自ランクの始点VOXEL インデクス (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 712 行で定義されています。

参照先 CPM_ERROR_GET_HEADINDEX, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetVoxelHeadIndex().

5.11.3.34 CPM_EXTERN void cpm_GetVoxelTailIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの終点VOXEL の全体空間でのインデクスを取得

- GetVoxelTailIndex のFortran インターフェイス関数
- 全体空間の先頭インデクスを 0 としたC 型のインデクス

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>idx</i>	自ランクの終点VOXEL インデクス (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 755 行で定義されています。

参照先 CPM_ERROR_GET_TAILINDEX, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetVoxelTailIndex().

5.11.3.35 CPM_EXTERN void cpm_Initialize_ (int * *ierr*)

初期化処理 (MPI_Init は実行済みの場合)

- Initialize のFortran インターフェイス関数
- Fortran でMPI_Init がコールされている必要がある

引数

out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)
-----	-------------	--------------------------------------

cpm_ParaManager_frtIF.cpp の 146 行で定義されています。

参照先 CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::Initialize().

5.11.3.36 CPM_EXTERN void cpm_Irecv_ (void * *buf*, int * *count*, int * *datatype*, int * *source*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Irecv

- Irecv のFortran インターフェイス関数

引数

	<i>inout</i>]	buf 受信バッファ
in	<i>count</i>	受信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>source</i>	送信元先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
in	<i>reqNo</i>	リクエスト番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1238 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_lrecv(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

5.11.3.37 CPM_EXTERN void cpm_Isend_ (void * *buf*, int * *count*, int * *datatype*, int * *dest*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Isend

- Isend のFortran インターフェイス関数

引数

	<i>inout</i>]	buf 送信バッファ
in	<i>count</i>	送信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>reqNo</i>	リクエスト番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1199 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Isend(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

5.11.3.38 CPM_EXTERN void cpm_IsParallel_ (int * *ipara*, int * *ierr*)

並列実行であるかチェックする

- IsParallel のFortran インターフェイス関数

引数

out	<i>ipara</i>	並列実行フラグ (1=並列実行、1 以外=逐次実行)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 254 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::IsParallel().

5.11.3.39 CPM_EXTERN void cpm_PeriodicCommS3D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *dir*, int * *pm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

周期境界袖通信 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS3D のFortran インターフェイス関数

引数

	<i>inout</i>]	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)

in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2307 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS3D().

5.11.3.40 CPM_EXTERN void cpm_PeriodicCommS4D_ (void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

周期境界袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS4D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2244 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_(), と cpm_PeriodicCommV3D_().

5.11.3.41 CPM_EXTERN void cpm_PeriodicCommS4DEx_ (void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

周期境界袖通信 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS4DEx のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)

in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2446 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_().

5.11.3.42 CPM_EXTERN void cpm_PeriodicCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

周期境界袖通信 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommV3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2376 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3D().

5.11.3.43 CPM_EXTERN void cpm_PeriodicCommV3DEx_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

周期境界袖通信 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommV3DEx のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2509 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4DEx(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3DEx().

5.11.3.44 CPM_EXTERN void cpm_Recv_(void * buf, int * count, int * datatype, int * source, int * procGrpNo, int * ierr)

Recv

- Recv のFortran インターフェイス関数

引数

	<i>inout]</i>	buf 受信バッファ
in	<i>count</i>	受信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>source</i>	送信元のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1158 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Recv().

5.11.3.45 CPM_EXTERN void cpm_Send_(void * buf, int * count, int * datatype, int * dest, int * procGrpNo, int * ierr)

Send

- Send のFortran インターフェイス関数

引数

	<i>inout]</i>	buf 送信バッファ
in	<i>count</i>	送信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1118 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Send().

5.11.3.46 CPM_EXTERN void cpm_SetBndCommBuffer_ (int * maxVC, int * maxN, int * procGrpNo, int * ierr)

袖通信バッファのセット (Fortran インターフェイス)

- ・ 袖通信バッファ確保処理のFortran インターフェイス関数

引数

in	maxVC	送受信バッファの最大袖数
in	maxN	送受信バッファの最大成分数
in	procGrpNo	プロセスグループ番号
out	ierr	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1526 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::SetBndCommBuffer().

5.11.3.47 CPM_EXTERN void cpm_Voxellnit_ (int * div, int * vox, REAL_TYPE * origin, REAL_TYPE * pitch, int * obcid, int * maxVC, int * maxN, int * procGrpNo, int * ierr)

領域分割

- ・ Voxellnit のFortran インターフェイス関数
- ・ 領域分割の各種情報を引数で渡して領域分割を行う
- ・ プロセスグループの全てのランクが活性ドメインになる
- ・ 領域分割数を指定する

引数

in	div	領域分割数 (サイズ 3)
in	vox	空間全体のボクセル数 (サイズ 3)
in	origin	空間全体の原点 (サイズ 3)
in	pitch	ボクセルピッチ (サイズ 3)
in	obcid	全体空間の外部境界条件ID(サイズ 6)
in	maxVC	最大の袖数 (袖通信用)
in	maxN	最大の成分数 (袖通信用)
in	procGrpNo	領域分割を行うプロセスグループ番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 183 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::Voxellnit().

5.11.3.48 CPM_EXTERN void cpm_Voxellnit_nodiv_ (int * vox, REAL_TYPE * origin, REAL_TYPE * pitch, int * obcid, int * maxVC, int * maxN, int * procGrpNo, int * ierr)

領域分割

- ・ Voxellnit のFortran インターフェイス関数
- ・ 領域分割の各種情報を引数で渡して領域分割を行う

- ・ プロセスグループの全てのランクが活性ドメインになる
- ・ プロセスグループのランク数で自動領域分割

引数

in	<i>vox</i>	空間全体のボクセル数 (サイズ 3)
in	<i>origin</i>	空間全体の原点 (サイズ 3)
in	<i>pitch</i>	ボクセルピッチ (サイズ 3)
in	<i>obcid</i>	全体空間の外部境界条件ID(サイズ 6)
in	<i>maxVC</i>	最大の袖数 (袖通信用)
in	<i>maxN</i>	最大の成分数 (袖通信用)
in	<i>procGrpNo</i>	領域分割を行うプロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 223 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::VoxelInit().

5.11.3.49 CPM_EXTERN void cpm_Wait_ (int * reqNo, int * ierr)

Wait

- ・ Wait のFortran インターフェイス関数

引数

in	<i>reqNo</i>	リクエスト番号 (0 以上の整数)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1017 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Wait(), と cpm_ParaManager::get_instance().

5.11.3.50 CPM_EXTERN void cpm_wait_BndCommS3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信の wait、展開 (Scalar3D 版) のFortran インターフェイス

- ・ (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- ・ wait_BndCommS3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1883 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D_, と cpm_ParaManager::get_instance().

5.11.3.51 CPM_EXTERN void cpm_wait_BndCommS4D_ (void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信の wait、展開 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1843 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS4D(), と cpm_ParaManager::get_instance().

5.11.3.52 CPM_EXTERN void cpm_wait_BndCommS4DEx_ (void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信の wait、展開 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4DEx のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2157 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS4DEx(), と cpm_ParaManager::get_instance().

5.11.3.53 CPM_EXTERN void cpm_wait_BndCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信の wait、展開 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3D のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1928 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_wait_BndCommS4D_, cpm_ParaManager::cpm_wait_BndCommV3D(), と cpm_ParaManager::get_instance().

5.11.3.54 CPM_EXTERN void cpm_wait_BndCommV3DEx_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

非同期版袖通信の wait、展開 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3DEx のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2197 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_wait_BndCommS4DEx_, cpm_ParaManager::cpm_wait_BndCommV3DEx(), と cpm_ParaManager::get_instance().

5.11.3.55 CPM_EXTERN void cpm_Waitall_ (int * count, int * reqlist, int * ierr)

Waitall

- Waitall のFortran インターフェイス関数

引数

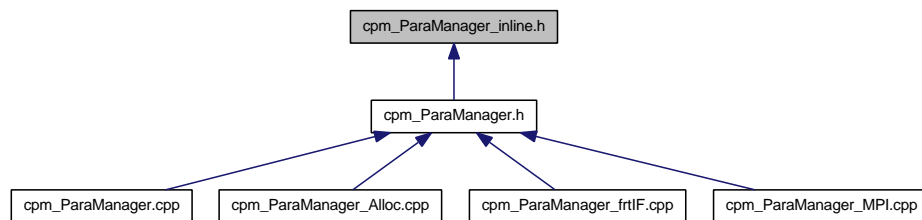
in	<i>count</i>	リクエストの数
in	<i>reqlist</i>	リクエスト番号のリスト (0 以上の整数)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1046 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Waitall(), と cpm_ParaManager::get_instance().

5.12 cpm_ParaManager_inline.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



5.12.1 説明

パラレルマネージャクラスの inline 関数ヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

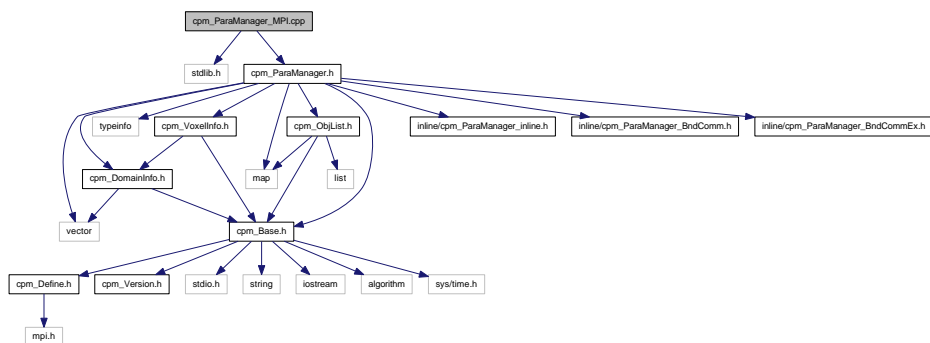
[cpm_ParaManager_inline.h](#) で定義されています。

5.13 cpm_ParaManager_MPI.cpp

```
#include "stdlib.h"
```

```
#include "cpm_ParaManager.h"
```

cpm_ParaManager_MPI.cpp のインクルード依存関係図



5.13.1 説明

パラレルマネージャクラスのMPI インターフェイス関数ソースファイル

作者

University of Tokyo

日付

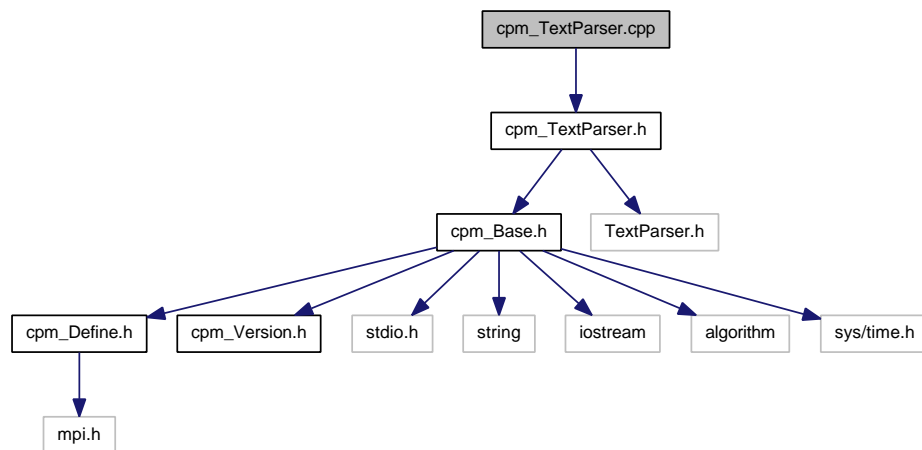
2012/05/31

[cpm_ParaManager_MPI.cpp](#) で定義されています。

5.14 cpm_TextParser.cpp

```
#include "cpm_TextParser.h"
```

cpm_TextParser.cpp のインクルード依存関係図



5.14.1 説明

TextParser クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

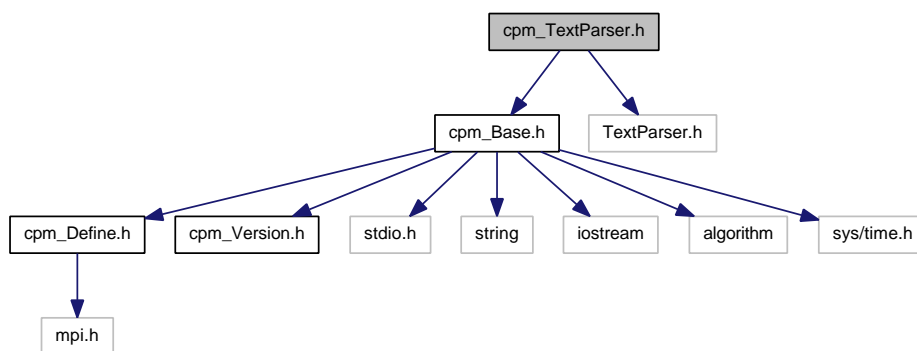
[cpm_TextParser.cpp](#) で定義されています。

5.15 cpm_TextParser.h

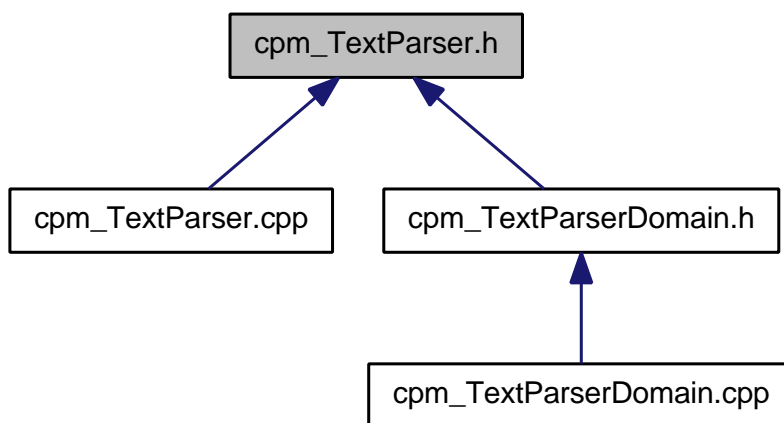
```
#include "cpm_Base.h"
```

```
#include "TextParser.h"
```

cpm_TextParser.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class `cpm_TextParser`

5.15.1 説明

テキストパーサークラスのヘッダーファイル

作者

University of Tokyo

日付

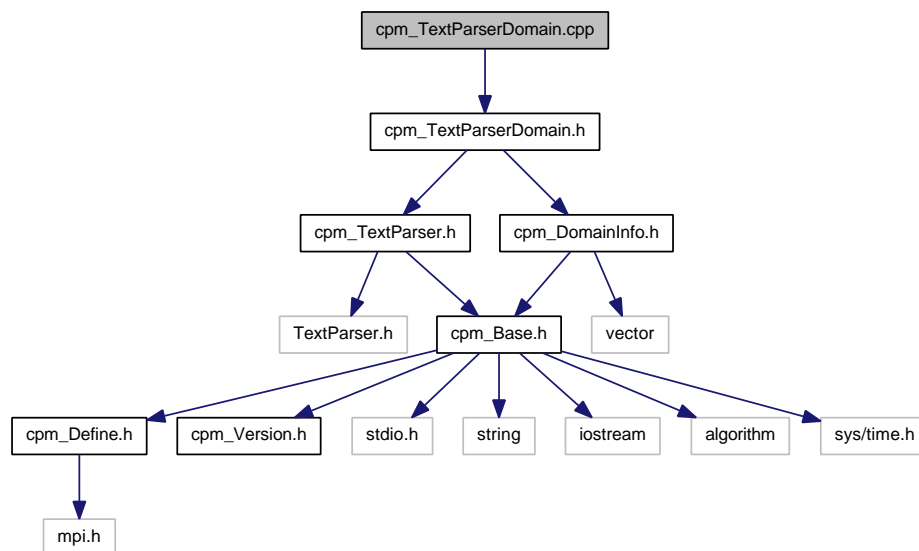
2012/05/31

`cpm_TextParser.h` で定義されています。

5.16 cpm_TextParserDomain.cpp

```
#include "cpm_TextParserDomain.h"
```

cpm_TextParserDomain.cpp のインクルード依存関係図



5.16.1 説明

CPM 領域情報のTextParser クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

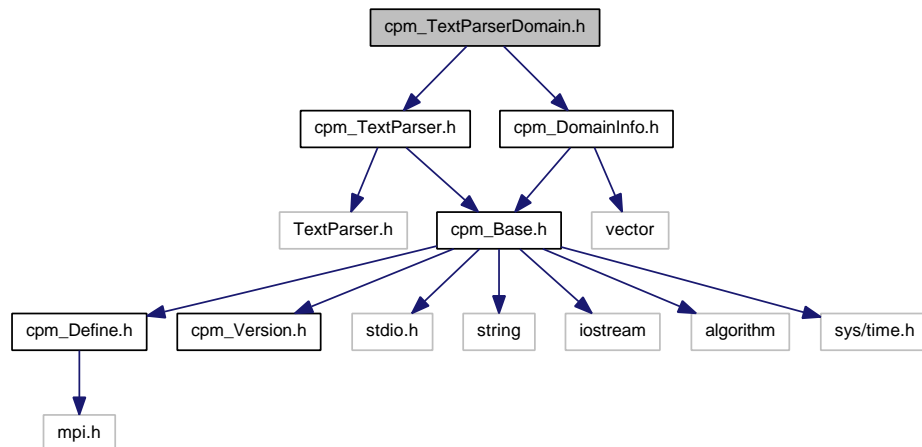
[cpm_TextParserDomain.cpp](#) で定義されています。

5.17 cpm_TextParserDomain.h

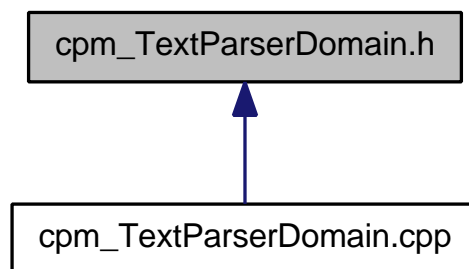
```
#include "cpm_TextParser.h"
```

```
#include "cpm_DomainInfo.h"
```

cpm_TextParserDomain.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class [cpm_TextParserDomain](#)

5.17.1 説明

領域情報のテキストパーサークラスのヘッダーファイル

作者

University of Tokyo

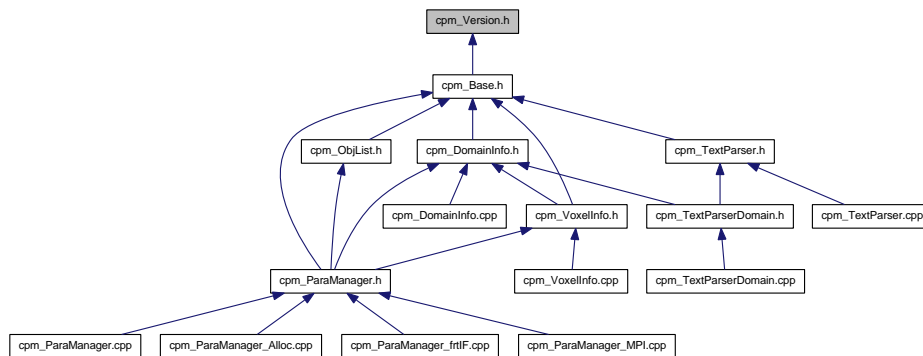
日付

2012/05/31

[cpm_TextParserDomain.h](#) で定義されています。

5.18 cpm_Version.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



マクロ定義

- `#define CPM_VERSION_NO "1.0.0"`
- `#define CPM_REVISION "20120630_0900"`

5.18.1 説明

CPM バージョン情報のヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_Version.h](#) で定義されています。

5.18.2 マクロ定義

5.18.2.1 `#define CPM_REVISION "20120630_0900"`

CPM ライブラリのリビジョン

`cpm_Version.h` の 22 行で定義されています。

5.18.2.2 `#define CPM_VERSION_NO "1.0.0"`

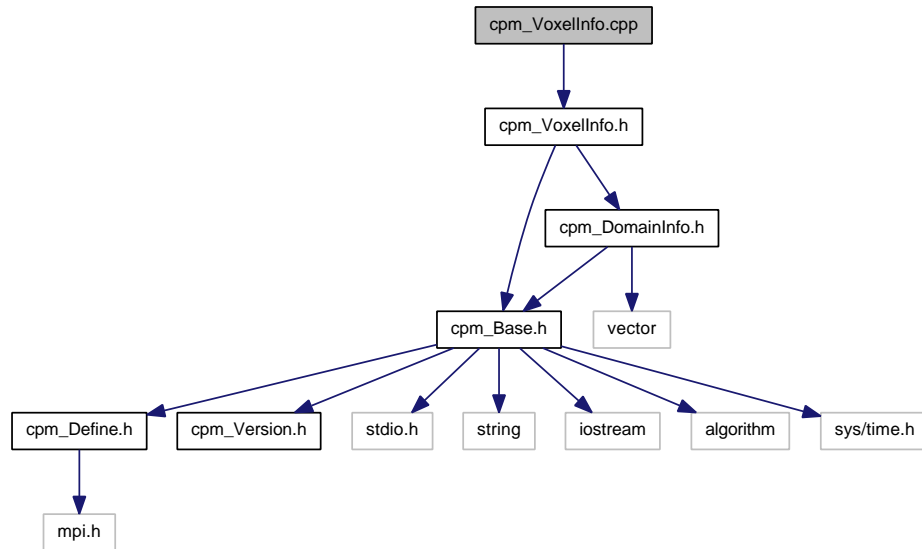
CPM ライブラリのバージョン

`cpm_Version.h` の 19 行で定義されています。

参照元 `cpm_Base::VersionInfo()`.

5.19 cpm_VoxelInfo.cpp

```
#include "cpm_VoxelInfo.h"
cpm_VoxelInfo.cpp のインクルード依存関係図
```



5.19.1 説明

VOXEL 空間情報クラスのソースファイル

作者

University of Tokyo

日付

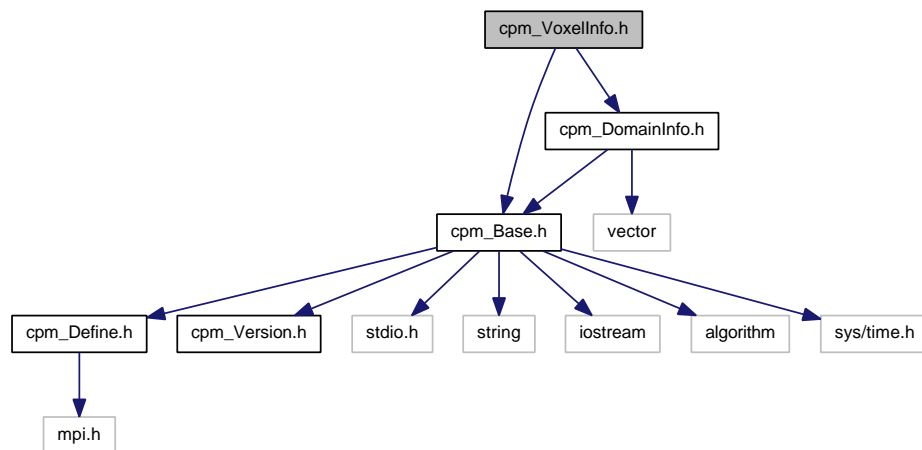
2012/05/31

[cpm_VoxelInfo.cpp](#) で定義されています。

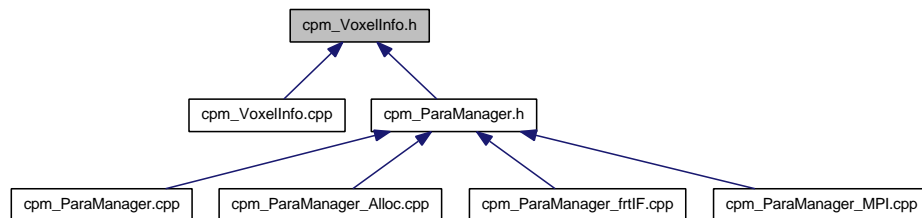
5.20 cpm_VoxelInfo.h

```
#include "cpm_Base.h"
#include "cpm_DomainInfo.h"
```

cpm_VoxelInfo.h のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



構成

- class [cpm_VoxelInfo](#)

5.20.1 説明

VOXEL 空間情報クラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_VoxelInfo.h](#) で定義されています。

Index

- ~S_BNDCOMM_BUFFER
 - S_BNDCOMM_BUFFER, [117](#)
- ~cpm_ActiveSubDomainInfo
 - cpm_ActiveSubDomainInfo, [8](#)
- ~cpm_Base
 - cpm_Base, [12](#)
- ~cpm_DomainInfo
 - cpm_DomainInfo, [17](#)
- ~cpm_GlobalDomainInfo
 - cpm_GlobalDomainInfo, [22](#)
- ~cpm_LocalDomainInfo
 - cpm_LocalDomainInfo, [25](#)
- ~cpm_ObjList
 - cpm_ObjList, [27](#)
- ~cpm_ParaManager
 - cpm_ParaManager, [36](#)
- ~cpm_TextParser
 - cpm_TextParser, [103](#)
- ~cpm_TextParserDomain
 - cpm_TextParserDomain, [106](#)
- ~cpm_VoxelInfo
 - cpm_VoxelInfo, [110](#)
- _IDXXFX
 - cpm_ParaManager_BndComm.h, [136](#)
 - cpm_ParaManager_BndCommEx.h, [138](#)
- _IDXXFY
 - cpm_ParaManager_BndComm.h, [136](#)
 - cpm_ParaManager_BndCommEx.h, [138](#)
- _IDXXFZ
 - cpm_ParaManager_BndComm.h, [137](#)
 - cpm_ParaManager_BndCommEx.h, [138](#)
- _IDX_S3D
 - cpm_Define.h, [124](#)
- _IDX_S4D
 - cpm_Define.h, [124](#)
- _IDX_S4DEX
 - cpm_Define.h, [125](#)
- _IDX_V3D
 - cpm_Define.h, [125](#)
- _IDX_V3DEX
 - cpm_Define.h, [126](#)
- Abort
 - cpm_ParaManager, [36](#)
- Add
 - cpm_ObjList, [28](#)
- AddSubDomain
 - cpm_GlobalDomainInfo, [22](#)
- Allgather
 - cpm_ParaManager, [36, 37](#)
- Allgatherv
 - cpm_ParaManager, [37](#)
- AllocDoubleS3D
 - cpm_ParaManager, [38](#)
- AllocDoubleS4D
 - cpm_ParaManager, [38](#)
- AllocDoubleS4DEX
 - cpm_ParaManager, [39](#)
- AllocDoubleV3D
 - cpm_ParaManager, [39](#)
- AllocDoubleV3DEX
 - cpm_ParaManager, [39](#)
- AllocFloatS3D
 - cpm_ParaManager, [40](#)
- AllocFloatS4D
 - cpm_ParaManager, [40](#)
- AllocFloatS4DEX
 - cpm_ParaManager, [40](#)
- AllocFloatV3D
 - cpm_ParaManager, [41](#)
- AllocFloatV3DEX
 - cpm_ParaManager, [41](#)
- AllocIntS3D
 - cpm_ParaManager, [41](#)
- AllocIntS4D
 - cpm_ParaManager, [41](#)
- AllocIntS4DEX
 - cpm_ParaManager, [42](#)
- AllocIntV3D
 - cpm_ParaManager, [42](#)
- AllocIntV3DEX
 - cpm_ParaManager, [42](#)
- AllocRealS3D
 - cpm_ParaManager, [43](#)
- AllocRealS4D
 - cpm_ParaManager, [43](#)
- AllocRealS4DEX
 - cpm_ParaManager, [43](#)
- AllocRealV3D
 - cpm_ParaManager, [44](#)
- AllocRealV3DEX
 - cpm_ParaManager, [44](#)
- Allreduce
 - cpm_ParaManager, [44, 45](#)
- BOTH
 - cpm_Define.h, [130](#)
- Barrier
 - cpm_ParaManager, [45](#)
- Bcast

- cpm_ParaManager, [45, 46](#)
- BndCommInfoMap
 - cpm_ParaManager.h, [135](#)
- BndCommS3D
 - cpm_ParaManager, [46, 47](#)
- BndCommS3D_nowait
 - cpm_ParaManager, [47, 48](#)
- BndCommS4D
 - cpm_ParaManager, [48, 49](#)
- BndCommS4D_nowait
 - cpm_ParaManager, [49, 50](#)
- BndCommS4DEx
 - cpm_ParaManager, [51](#)
- BndCommS4DEx_nowait
 - cpm_ParaManager, [52](#)
- BndCommV3D
 - cpm_ParaManager, [53](#)
- BndCommV3D_nowait
 - cpm_ParaManager, [54](#)
- BndCommV3DEx
 - cpm_ParaManager, [55](#)
- BndCommV3DEx_nowait
 - cpm_ParaManager, [56](#)
- CPM_BAND
 - cpm_Define.h, [130](#)
- CPM_BOR
 - cpm_Define.h, [130](#)
- CPM_BXOR
 - cpm_Define.h, [130](#)
- CPM_BYTE
 - cpm_Define.h, [127](#)
- CPM_CHAR
 - cpm_Define.h, [127](#)
- CPM_DOUBLE
 - cpm_Define.h, [127](#)
- CPM_Datatype
 - cpm_Define.h, [127](#)
- CPM_ERROR
 - cpm_Define.h, [127](#)
- CPM_ERROR_ALREADY_VOXELINIIT
 - cpm_Define.h, [128](#)
- CPM_ERROR_BNDCOMM
 - cpm_Define.h, [129](#)
- CPM_ERROR_BNDCOMM_BUFFER
 - cpm_Define.h, [129](#)
- CPM_ERROR_BNDCOMM_BUFFERLENGTH
 - cpm_Define.h, [129](#)
- CPM_ERROR_BNDCOMM_VOXELSIZE
 - cpm_Define.h, [129](#)
- CPM_ERROR_CREATE_LOCALDOMAIN
 - cpm_Define.h, [128](#)
- CPM_ERROR_CREATE_NEIGHBOR
 - cpm_Define.h, [128](#)
- CPM_ERROR_CREATE_PROCGROUP
 - cpm_Define.h, [128](#)
- CPM_ERROR_CREATE_RANKMAP
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_BCID
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_DIVNUM
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_DIVPOS
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_GLOBALORIGIN
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_GLOBALREGION
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_GLOBALVOXELSIZE
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_HEADINDEX
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_INFO
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_LOCALORIGIN
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_LOCALREGION
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_LOCALVOXELSIZE
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_MYRANK
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_NEIGHBOR_RANK
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_NUMRANK
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_PERIODIC_RANK
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_PITCH
 - cpm_Define.h, [128](#)
- CPM_ERROR_GET_TAILINDEX
 - cpm_Define.h, [128](#)
- CPM_ERROR_INSERT_VOXELMAP
 - cpm_Define.h, [128](#)
- CPM_ERROR_INVALID_DOMAIN_NO
 - cpm_Define.h, [127](#)
- CPM_ERROR_INVALID_OBJKEY
 - cpm_Define.h, [128](#)
- CPM_ERROR_INVALID_PTR
 - cpm_Define.h, [127](#)
- CPM_ERROR_INVALID_VOXELSIZE
 - cpm_Define.h, [128](#)
- CPM_ERROR_MISMATCH_NP_SUBDOMAIN
 - cpm_Define.h, [128](#)
- CPM_ERROR_MPI
 - cpm_Define.h, [128](#)
- CPM_ERROR_MPI_ALLGATHER
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_ALLGATHERV
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_ALLREDUCE
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_BARRIER
 - cpm_Define.h, [128](#)
- CPM_ERROR_MPI_BCAST
 - cpm_Define.h, [128](#)
- CPM_ERROR_MPI_DIMSCREATE

- cpm_Define.h, [129](#)
- CPM_ERROR_MPI_GATHER
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_GATHERV
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_INVALID_COMM
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_INVALID_DATATYPE
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_INVALID_OPERATOR
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_INVALID_REQUEST
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_Irecv
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_Isend
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_RECV
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_SEND
 - cpm_Define.h, [128](#)
- CPM_ERROR_MPI_WAIT
 - cpm_Define.h, [129](#)
- CPM_ERROR_MPI_WAITALL
 - cpm_Define.h, [129](#)
- CPM_ERROR_NO_MPI_INIT
 - cpm_Define.h, [128](#)
- CPM_ERROR_NO_TEXTPARSER
 - cpm_Define.h, [128](#)
- CPM_ERROR_NOT_IN_PROCGROUP
 - cpm_Define.h, [128](#)
- CPM_ERROR_PERIODIC
 - cpm_Define.h, [129](#)
- CPM_ERROR_PERIODIC_INVALID_DIR
 - cpm_Define.h, [129](#)
- CPM_ERROR_PERIODIC_INVALID_PM
 - cpm_Define.h, [129](#)
- CPM_ERROR_PM_INSTANCE
 - cpm_Define.h, [127](#)
- CPM_ERROR_REGIST_OBJKEY
 - cpm_Define.h, [128](#)
- CPM_ERROR_TEXTPARSER
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_BCID
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_G_DIV
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_G_ORG
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_G_PITCH
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_G_RGN
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_G_VOXEL
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_INVALID_POS
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_NOVECTOR
 - cpm_Define.h, [128](#)
- CPM_ERROR_TP_VECTOR_SIZE
 - cpm_Define.h, [128](#)
- CPM_ERROR_VOXELINIT
 - cpm_Define.h, [128](#)
- CPM_EXTERN
 - cpm_ParaManager_frtIf.cpp, [143](#)
- CPM_FLOAT
 - cpm_Define.h, [127](#)
- CPM_INLINE
 - cpm_Base.h, [122](#)
- CPM_INT
 - cpm_Define.h, [127](#)
- CPM_LAND
 - cpm_Define.h, [130](#)
- CPM_LONG
 - cpm_Define.h, [127](#)
- CPM_LONG_DOUBLE
 - cpm_Define.h, [127](#)
- CPM_LOR
 - cpm_Define.h, [130](#)
- CPM_LXOR
 - cpm_Define.h, [130](#)
- CPM_MAX
 - cpm_Define.h, [129](#)
- CPM_MAXLOC
 - cpm_Define.h, [130](#)
- CPM_MIN
 - cpm_Define.h, [129](#)
- CPM_MINLOC
 - cpm_Define.h, [130](#)
- CPM_Op
 - cpm_Define.h, [129](#)
- CPM_PROD
 - cpm_Define.h, [130](#)
- CPM_REAL
 - cpm_Define.h, [127](#)
- CPM_REVISION
 - cpm_Version.h, [173](#)
- CPM_SHORT
 - cpm_Define.h, [127](#)
- CPM_SUCCESS
 - cpm_Define.h, [127](#)
- CPM_SUM
 - cpm_Define.h, [129](#)
- CPM_UNSIGNED
 - cpm_Define.h, [127](#)
- CPM_UNSIGNED_CHAR
 - cpm_Define.h, [127](#)
- CPM_UNSIGNED_LONG
 - cpm_Define.h, [127](#)
- CPM_UNSIGNED_SHORT
 - cpm_Define.h, [127](#)
- CPM_VERSION_NO
 - cpm_Version.h, [173](#)
- CalcCommSize
 - cpm_ParaManager, [57](#)
- clear

- cpm_ActiveSubDomainInfo, 8
- cpm_DomainInfo, 17
- cpm_GlobalDomainInfo, 22
- cpm_LocalDomainInfo, 26
- CopyArray
 - cpm_ParaManager, 57
- cpm_Abort_
 - cpm_ParaManager_frtIF.cpp, 142, 147
- cpm_ActiveSubDomainInfo, 7
 - ~cpm_ActiveSubDomainInfo, 8
 - clear, 8
 - cpm_ActiveSubDomainInfo, 8
 - cpm_ActiveSubDomainInfo, 8
 - GetBCID, 9
 - GetPos, 9
 - m_bcid, 11
 - m_pos, 11
 - operator==, 9
 - Set, 10
 - SetBCID, 10
 - SetPos, 10
- cpm_Allgather_
 - cpm_ParaManager_frtIF.cpp, 142, 147
- cpm_Allgatherv_
 - cpm_ParaManager_frtIF.cpp, 142, 147
- cpm_Allreduce_
 - cpm_ParaManager_frtIF.cpp, 142, 148
- cpm_Barrier
 - cpm_ParaManager_frtIF.cpp, 142, 148
- cpm_Base, 11
 - ~cpm_Base, 12
 - cpm_Base, 12
 - cpm_strCompare, 12
 - cpm_strCompareN, 13
 - cpm_Base, 12
 - getCommNull, 13
 - GetMemString, 13
 - getRankNull, 13
 - GetSpanTime, 14
 - GetTime, 14
 - GetWSpanTime, 14
 - GetWTime, 14
 - IsCommNull, 14
 - IsRankNull, 15
 - ReallIsDouble, 15
 - VersionInfo, 15
- cpm_Base.h, 121
 - CPM_INLINE, 122
- cpm_Bcast_
 - cpm_ParaManager_frtIF.cpp, 142, 148
- cpm_BndCommS3D_
 - cpm_ParaManager_frtIF.cpp, 142, 149
- cpm_BndCommS3D_nowait
 - cpm_ParaManager, 57
- cpm_BndCommS3D_nowait_
 - cpm_ParaManager_frtIF.cpp, 142, 149
- cpm_BndCommS4D_
 - cpm_ParaManager_frtIF.cpp, 142, 150
- cpm_BndCommS4D_nowait
 - cpm_ParaManager, 58
- cpm_BndCommS4D_nowait_
 - cpm_ParaManager_frtIF.cpp, 143, 150
- cpm_BndCommS4DEx_
 - cpm_ParaManager_frtIF.cpp, 143, 151
- cpm_BndCommS4DEx_nowait
 - cpm_ParaManager, 59
- cpm_BndCommS4DEx_nowait_
 - cpm_ParaManager_frtIF.cpp, 143, 151
- cpm_BndCommV3D_
 - cpm_ParaManager_frtIF.cpp, 143, 152
- cpm_BndCommV3D_nowait
 - cpm_ParaManager, 59
- cpm_BndCommV3D_nowait_
 - cpm_ParaManager_frtIF.cpp, 143, 152
- cpm_BndCommV3DEx_
 - cpm_ParaManager_frtIF.cpp, 143, 152
- cpm_BndCommV3DEx_nowait
 - cpm_ParaManager, 60
- cpm_BndCommV3DEx_nowait_
 - cpm_ParaManager_frtIF.cpp, 143, 153
- cpm_Define.h, 122
 - _IDX_S3D, 124
 - _IDX_S4D, 124
 - _IDX_S4DEX, 125
 - _IDX_V3D, 125
 - _IDX_V3DEX, 126
 - BOTH, 130
 - CPM_BAND, 130
 - CPM_BOR, 130
 - CPM_BXOR, 130
 - CPM_BYTE, 127
 - CPM_CHAR, 127
 - CPM_DOUBLE, 127
 - CPM_Datatype, 127
 - CPM_ERROR, 127
 - CPM_ERROR_ALREADY_VOXELINIIT, 128
 - CPM_ERROR_BNDCOMM, 129
 - CPM_ERROR_BNDCOMM_BUFFER, 129
 - CPM_ERROR_BNDCOMM_BUFFERLENGTH, 129
 - CPM_ERROR_BNDCOMM_VOXELSIZE, 129
 - CPM_ERROR_CREATE_LOCALDOMAIN, 128
 - CPM_ERROR_CREATE_NEIGHBOR, 128
 - CPM_ERROR_CREATE_PROCGROUP, 128
 - CPM_ERROR_CREATE_RANKMAP, 128
 - CPM_ERROR_GET_BCID, 128
 - CPM_ERROR_GET_DIVNUM, 128
 - CPM_ERROR_GET_DIVPOS, 128
 - CPM_ERROR_GET_GLOBALORIGIN, 128
 - CPM_ERROR_GET_GLOBALREGION, 128
 - CPM_ERROR_GET_GLOBALVOXELSIZE, 128
 - CPM_ERROR_GET_HEADINDEX, 128
 - CPM_ERROR_GET_INFO, 128
 - CPM_ERROR_GET_LOCALORIGIN, 128
 - CPM_ERROR_GET_LOCALREGION, 128
 - CPM_ERROR_GET_LOCALVOXELSIZE, 128

- CPM_ERROR_GET_MYRANK, 128
- CPM_ERROR_GET_NEIGHBOR_RANK, 128
- CPM_ERROR_GET_NUMRANK, 128
- CPM_ERROR_GET_PERIODIC_RANK, 128
- CPM_ERROR_GET_PITCH, 128
- CPM_ERROR_GET_TAILINDEX, 128
- CPM_ERROR_INSERT_VOXELMAP, 128
- CPM_ERROR_INVALID_DOMAIN_NO, 127
- CPM_ERROR_INVALID_OBJKEY, 128
- CPM_ERROR_INVALID_PTR, 127
- CPM_ERROR_INVALID_VOXELSIZE, 128
- CPM_ERROR_MISMATCH_NP_SUBDOMAIN, 128
- CPM_ERROR_MPI, 128
- CPM_ERROR_MPI_ALLGATHER, 129
- CPM_ERROR_MPI_ALLGATHERV, 129
- CPM_ERROR_MPI_ALLREDUCE, 129
- CPM_ERROR_MPI_BARRIER, 128
- CPM_ERROR_MPI_BCAST, 128
- CPM_ERROR_MPI_DIMSCREATE, 129
- CPM_ERROR_MPI_GATHER, 129
- CPM_ERROR_MPI_GATHERV, 129
- CPM_ERROR_MPI_INVALID_COMM, 129
- CPM_ERROR_MPI_INVALID_DATATYPE, 129
- CPM_ERROR_MPI_INVALID_OPERATOR, 129
- CPM_ERROR_MPI_INVALID_REQUEST, 129
- CPM_ERROR_MPI_Irecv, 129
- CPM_ERROR_MPI_Isend, 129
- CPM_ERROR_MPI_RECV, 129
- CPM_ERROR_MPI_SEND, 128
- CPM_ERROR_MPI_WAIT, 129
- CPM_ERROR_MPI_WAITALL, 129
- CPM_ERROR_NO_MPI_INIT, 128
- CPM_ERROR_NO_TEXTPARSER, 128
- CPM_ERROR_NOT_IN_PROCGROUP, 128
- CPM_ERROR_PERIODIC, 129
- CPM_ERROR_PERIODIC_INVALID_DIR, 129
- CPM_ERROR_PERIODIC_INVALID_PM, 129
- CPM_ERROR_PM_INSTANCE, 127
- CPM_ERROR_REGIST_OBJKEY, 128
- CPM_ERROR_TEXTPARSER, 128
- CPM_ERROR_TP_INVALID_BCID, 128
- CPM_ERROR_TP_INVALID_G_DIV, 128
- CPM_ERROR_TP_INVALID_G_ORG, 128
- CPM_ERROR_TP_INVALID_G_PITCH, 128
- CPM_ERROR_TP_INVALID_G_RGN, 128
- CPM_ERROR_TP_INVALID_G_VOXEL, 128
- CPM_ERROR_TP_INVALID_POS, 128
- CPM_ERROR_TP_NOVECTOR, 128
- CPM_ERROR_TP_VECTOR_SIZE, 128
- CPM_ERROR_VOXELINIT, 128
- CPM_FLOAT, 127
- CPM_INT, 127
- CPM_LAND, 130
- CPM_LONG, 127
- CPM_LONG_DOUBLE, 127
- CPM_LOR, 130
- CPM_LXOR, 130
- CPM_MAX, 129
- CPM_MAXLOC, 130
- CPM_MIN, 129
- CPM_MINLOC, 130
- CPM_Op, 129
- CPM_PROD, 130
- CPM_REAL, 127
- CPM_SHORT, 127
- CPM_SUCCESS, 127
- CPM_SUM, 129
- CPM_UNSIGNED, 127
- CPM_UNSIGNED_CHAR, 127
- CPM_UNSIGNED_LONG, 127
- CPM_UNSIGNED_SHORT, 127
- cpm_DirFlag, 127
- cpm_ErrorCode, 127
- cpm_FaceFlag, 129
- cpm_PMFlag, 130
- MINUS2PLUS, 130
- PLUS2MINUS, 130
- REAL_BUF_TYPE, 126
- REAL_TYPE, 126
- X_DIR, 127
- X_MINUS, 129
- X_PLUS, 129
- Y_DIR, 127
- Y_MINUS, 129
- Y_PLUS, 129
- Z_DIR, 127
- Z_MINUS, 129
- Z_PLUS, 129
- cpm_DirFlag
 - cpm_Define.h, 127
- cpm_DomainInfo, 16
 - ~cpm_DomainInfo, 17
 - clear, 17
 - cpm_DomainInfo, 17
 - cpm_DomainInfo, 17
 - GetOrigin, 17
 - GetPitch, 18
 - GetRegion, 18
 - GetVoxNum, 18
 - m_origin, 20
 - m_pitch, 20
 - m_region, 20
 - m_voxNum, 20
 - SetOrigin, 18
 - SetPitch, 19
 - SetRegion, 19
 - SetVoxNum, 19
- cpm_DomainInfo.cpp, 130
- cpm_DomainInfo.h, 131
- cpm_ErrorCode
 - cpm_Define.h, 127
- cpm_FaceFlag
 - cpm_Define.h, 129
- cpm_Gather_
 - cpm_ParaManager_frtIF.cpp, 143, 153

- cpm_Gatherv_
 - cpm_ParaManager_frtIF.cpp, 143, 154
- cpm_GetBCID_
 - cpm_ParaManager_frtIF.cpp, 143, 154
- cpm_GetDivNum_
 - cpm_ParaManager_frtIF.cpp, 144, 155
- cpm_GetDivPos_
 - cpm_ParaManager_frtIF.cpp, 144, 155
- cpm_GetGlobalOrigin_
 - cpm_ParaManager_frtIF.cpp, 144, 155
- cpm_GetGlobalRegion_
 - cpm_ParaManager_frtIF.cpp, 144, 155
- cpm_GetGlobalVoxelSize_
 - cpm_ParaManager_frtIF.cpp, 144, 156
- cpm_GetLocalOrigin_
 - cpm_ParaManager_frtIF.cpp, 144, 156
- cpm_GetLocalRegion_
 - cpm_ParaManager_frtIF.cpp, 144, 156
- cpm_GetLocalVoxelSize_
 - cpm_ParaManager_frtIF.cpp, 144, 157
- cpm_GetMyRankID_
 - cpm_ParaManager_frtIF.cpp, 144, 157
- cpm_GetNeighborRankID_
 - cpm_ParaManager_frtIF.cpp, 144, 157
- cpm_GetNumRank_
 - cpm_ParaManager_frtIF.cpp, 144, 157
- cpm_GetPeriodicRankID_
 - cpm_ParaManager_frtIF.cpp, 144, 158
- cpm_GetPitch_
 - cpm_ParaManager_frtIF.cpp, 145, 158
- cpm_GetVoxelHeadIndex_
 - cpm_ParaManager_frtIF.cpp, 145, 158
- cpm_GetVoxelTailIndex_
 - cpm_ParaManager_frtIF.cpp, 145, 159
- cpm_GlobalDomainInfo, 20
 - ~cpm_GlobalDomainInfo, 22
 - AddSubDomain, 22
 - clear, 22
 - cpm_GlobalDomainInfo, 22
 - cpm_GlobalDomainInfo, 22
 - GetDivNum, 22
 - GetSubDomainInfo, 23
 - GetSubDomainNum, 23
 - IsExistSubDomain, 23
 - m_divNum, 24
 - m_subDomainInfo, 24
 - SetDivNum, 24
- cpm_Initialize_
 - cpm_ParaManager_frtIF.cpp, 145, 159
- cpm_Irecv
 - cpm_ParaManager, 60
- cpm_Irecv_
 - cpm_ParaManager_frtIF.cpp, 145, 159
- cpm_IsParallel_
 - cpm_ParaManager_frtIF.cpp, 145, 160
- cpm_Isend
 - cpm_ParaManager, 61
- cpm_Isend_
 - cpm_ParaManager_frtIF.cpp, 145, 160
- cpm_LocalDomainInfo, 24
 - ~cpm_LocalDomainInfo, 25
 - clear, 26
 - cpm_LocalDomainInfo, 25
 - cpm_LocalDomainInfo, 25
- cpm_ObjList
 - ~cpm_ObjList, 27
 - Add, 28
 - cpm_ObjList, 27
 - cpm_ObjList, 27
 - Create, 28
 - DelKeyList, 27
 - Delete, 28
 - Get, 28
 - m_DelKeyList, 29
 - m_ObjectMap, 29
 - m_newKey, 29
 - ObjectMap, 27
- cpm_ObjList< T >, 26
- cpm_ObjList.h, 132
 - RankNoMap, 133
- cpm_PMFlag
 - cpm_Define.h, 130
- cpm_ParaManager, 29
 - ~cpm_ParaManager, 36
 - Abort, 36
 - Allgather, 36, 37
 - Allgatherv, 37
 - AllocDoubleS3D, 38
 - AllocDoubleS4D, 38
 - AllocDoubleS4DEx, 39
 - AllocDoubleV3D, 39
 - AllocDoubleV3DEx, 39
 - AllocFloatS3D, 40
 - AllocFloatS4D, 40
 - AllocFloatS4DEx, 40
 - AllocFloatV3D, 41
 - AllocFloatV3DEx, 41
 - AllocIntS3D, 41
 - AllocIntS4D, 41
 - AllocIntS4DEx, 42
 - AllocIntV3D, 42
 - AllocIntV3DEx, 42
 - AllocRealS3D, 43
 - AllocRealS4D, 43
 - AllocRealS4DEx, 43
 - AllocRealV3D, 44
 - AllocRealV3DEx, 44
 - Allreduce, 44, 45
 - Barrier, 45
 - Bcast, 45, 46
 - BndCommS3D, 46, 47
 - BndCommS3D_nowait, 47, 48
 - BndCommS4D, 48, 49
 - BndCommS4D_nowait, 49, 50
 - BndCommS4DEx, 51
 - BndCommS4DEx_nowait, 52

- BndCommV3D, 53
- BndCommV3D_nowait, 54
- BndCommV3DEx, 55
- BndCommV3DEx_nowait, 56
- CalcCommSize, 57
- CopyArray, 57
- cpm_BndCommsS3D_nowait, 57
- cpm_BndCommsS4D_nowait, 58
- cpm_BndCommsS4DEx_nowait, 59
- cpm_BndCommV3D_nowait, 59
- cpm_BndCommV3DEx_nowait, 60
- cpm_Irecv, 60
- cpm_Isend, 61
- cpm_ParaManager, 36
- cpm_Wait, 61
- cpm_Waitall, 64
- cpm_wait_BndCommS3D, 61
- cpm_wait_BndCommS4D, 62
- cpm_wait_BndCommS4DEx, 63
- cpm_wait_BndCommV3D, 63
- cpm_wait_BndCommV3DEx, 64
- cpm_ParaManager, 36
- cpm_VoxelInfo, 115
- CreateProcessGroup, 65
- DecideDivPattern, 65
- FindVoxelInfo, 65
- flush, 66
- Gather, 66
- Gatherv, 67
- get_instance, 68
- GetBCID, 69
- GetBndCommBuffer, 69
- GetBndCommBufferSize, 69
- GetDivNum, 70
- GetDivPos, 70
- GetGlobalOrigin, 70
- GetGlobalRegion, 70
- GetGlobalVoxelSize, 71
- GetLocalOrigin, 71
- GetLocalRegion, 71
- GetLocalVoxelSize, 72
- GetMPI_Comm, 72
- GetMPI_Datatype, 72, 73
- GetMPI_Op, 73
- GetMyRankID, 73
- GetNeighborRankID, 74
- GetNumRank, 74
- GetPeriodicRankID, 74
- GetPitch, 75
- GetVoxelHeadIndex, 75
- GetVoxelTailIndex, 75
- InitArray, 75
- Initialize, 76
- Irecv, 76, 77
- IsParallel, 78
- Isend, 77, 78
- m_bndCommInfoMap, 100
- m_nRank, 100
- m_procGrpList, 100
- m_rankNo, 101
- m_rankNoMap, 101
- m_reqList, 101
- m_voxelInfoMap, 101
- packX, 79
- packXEx, 79
- packY, 80
- packYEx, 80
- packZ, 81
- packZEx, 81
- PeriodicCommsS3D, 82
- PeriodicCommsS4D, 83
- PeriodicCommsS4DEx, 84
- PeriodicCommV3D, 85
- PeriodicCommV3DEx, 86
- Recv, 87
- Send, 88
- sendrecv, 89
- SetBndCommBuffer, 89
- unpackX, 90
- unpackXEx, 90
- unpackY, 91
- unpackYEx, 91
- unpackZ, 92
- unpackZEx, 92
- VoxelInit, 93, 94
- Wait, 94
- wait_BndCommsS3D, 94, 95
- wait_BndCommsS4D, 95, 96
- wait_BndCommsS4DEx, 97
- wait_BndCommV3D, 98
- wait_BndCommV3DEx, 99
- Waitall, 100
- cpm_ParaManager.cpp, 133
- cpm_ParaManager.h, 134
 - BndCommInfoMap, 135
 - RankNoMap, 135
 - VoxelInfoMap, 135
- cpm_ParaManager_Alloc.cpp, 135
- cpm_ParaManager_BndComm.h, 136
 - _IDXXF, 136
 - _IDXXFY, 136
 - _IDXXFZ, 137
- cpm_ParaManager_BndCommEx.h, 137
 - _IDXXF, 138
 - _IDXXFY, 138
 - _IDXXFZ, 138
- cpm_ParaManager_MPI.cpp, 168
- cpm_ParaManager_frtIF.cpp, 139
 - CPM_EXTERN, 143
 - cpm_Abort_, 142, 147
 - cpm_Allgather_, 142, 147
 - cpm_Allgatherv_, 142, 147
 - cpm_Allreduce_, 142, 148
 - cpm_Barrier_, 142, 148
 - cpm_Bcast_, 142, 148
 - cpm_BndCommS3D_, 142, 149

- cpm_BndCommsS3D_nowait_, 142, 149
- cpm_BndCommsS4D_, 142, 150
- cpm_BndCommsS4D_nowait_, 143, 150
- cpm_BndCommsS4DEx_, 143, 151
- cpm_BndCommsS4DEx_nowait_, 143, 151
- cpm_BndCommV3D_, 143, 152
- cpm_BndCommV3D_nowait_, 143, 152
- cpm_BndCommV3DEx_, 143, 152
- cpm_BndCommV3DEx_nowait_, 143, 153
- cpm_Gather_, 143, 153
- cpm_Gatherv_, 143, 154
- cpm_GetBCID_, 143, 154
- cpm_GetDivNum_, 144, 155
- cpm_GetDivPos_, 144, 155
- cpm_GetGlobalOrigin_, 144, 155
- cpm_GetGlobalRegion_, 144, 155
- cpm_GetGlobalVoxelSize_, 144, 156
- cpm_GetLocalOrigin_, 144, 156
- cpm_GetLocalRegion_, 144, 156
- cpm_GetLocalVoxelSize_, 144, 157
- cpm_GetMyRankID_, 144, 157
- cpm_GetNeighborRankID_, 144, 157
- cpm_GetNumRank_, 144, 157
- cpm_GetPeriodicRankID_, 144, 158
- cpm_GetPitch_, 145, 158
- cpm_GetVoxelHeadIndex_, 145, 158
- cpm_GetVoxelTailIndex_, 145, 159
- cpm_Initialize_, 145, 159
- cpm_Irecv_, 145, 159
- cpm_IsParallel_, 145, 160
- cpm_Isend_, 145, 160
- cpm_PeriodicCommsS3D_, 145
- cpm_PeriodicCommsS3D_, 160
- cpm_PeriodicCommsS4D_, 145
- cpm_PeriodicCommsS4D_, 161
- cpm_PeriodicCommsS4DEx_, 145
- cpm_PeriodicCommsS4DEx_, 161
- cpm_PeriodicCommV3D_, 145
- cpm_PeriodicCommV3D_, 162
- cpm_PeriodicCommV3DEx_, 145
- cpm_PeriodicCommV3DEx_, 162
- cpm_Recv_, 146, 163
- cpm_Send_, 146, 163
- cpm_SetBndCommBuffer_, 146, 164
- cpm_Voxellnit_, 146, 164
- cpm_Voxellnit_nodiv_, 146, 164
- cpm_Wait_, 146, 165
- cpm_Waitall_, 146, 167
- cpm_wait_BndCommsS3D_, 146, 165
- cpm_wait_BndCommsS4D_, 146, 166
- cpm_wait_BndCommsS4DEx_, 146, 166
- cpm_wait_BndCommV3D_, 146, 166
- cpm_wait_BndCommV3DEx_, 146, 167
- cpm_ParaManager_inline.h, 168
- cpm_PeriodicCommS3D_
 - cpm_ParaManager_frtlF.cpp, 145
- cpm_PeriodicCommS3D_
 - cpm_ParaManager_frtlF.cpp, 160
- cpm_PeriodicCommS4D_
 - cpm_ParaManager_frtlF.cpp, 145
- cpm_PeriodicCommS4D_
 - cpm_ParaManager_frtlF.cpp, 161
- cpm_PeriodicCommS4DEx_
 - cpm_ParaManager_frtlF.cpp, 145
- cpm_PeriodicCommS4DEx_
 - cpm_ParaManager_frtlF.cpp, 161
- cpm_PeriodicCommV3D_
 - cpm_ParaManager_frtlF.cpp, 145
- cpm_PeriodicCommV3D_
 - cpm_ParaManager_frtlF.cpp, 162
- cpm_PeriodicCommV3DEx_
 - cpm_ParaManager_frtlF.cpp, 145
- cpm_PeriodicCommV3DEx_
 - cpm_ParaManager_frtlF.cpp, 162
- cpm_Recv_
 - cpm_ParaManager_frtlF.cpp, 146, 163
- cpm_Send_
 - cpm_ParaManager_frtlF.cpp, 146, 163
- cpm_SetBndCommBuffer_
 - cpm_ParaManager_frtlF.cpp, 146, 164
- cpm_TextParser, 102
 - ~cpm_TextParser, 103
 - cpm_TextParser, 103
 - cpm_TextParser, 103
 - m_tp, 104
 - Read, 103
 - readVector, 103, 104
- cpm_TextParser.cpp, 169
- cpm_TextParser.h, 169
- cpm_TextParserDomain, 105
 - ~cpm_TextParserDomain, 106
 - cpm_TextParserDomain, 106
 - cpm_TextParserDomain, 106
 - Read, 106
 - ReadDomainInfo, 106
 - ReadMain, 107
 - ReadSubdomainInfo, 107
- cpm_TextParserDomain.cpp, 171
- cpm_TextParserDomain.h, 171
- cpm_Version.h, 173
 - CPM_REVISION, 173
 - CPM_VERSION_NO, 173
- cpm_VoxellInfo, 108
 - ~cpm_VoxellInfo, 110
 - cpm_ParaManager, 115
 - cpm_VoxellInfo, 110
 - cpm_VoxellInfo, 110
 - CreateLocalDomainInfo, 110
 - CreateNeighborRankInfo, 110
 - CreateRankMap, 110
 - GetBCID, 111
 - GetDivNum, 111
 - GetDivPos, 111
 - GetGlobalOrigin, 111
 - GetGlobalRegion, 112
 - GetGlobalVoxelSize, 112

- GetLocalOrigin, 112
- GetLocalRegion, 112
- GetLocalVoxelSize, 113
- GetNeighborRankID, 113
- GetPeriodicRankID, 113
- GetPitch, 113
- GetVoxelHeadIndex, 114
- GetVoxelTailIndex, 114
- Init, 114
- m_comm, 115
- m_globalDomainInfo, 115
- m_localDomainInfo, 115
- m_nRank, 115
- m_neighborRankID, 115
- m_periodicRankID, 115
- m_rankMap, 116
- m_rankNo, 116
- m_voxelHeadIndex, 116
- m_voxelTailIndex, 116
- cpm_VoxelInfo.cpp, 174
- cpm_VoxelInfo.h, 174
- cpm_VoxelInit_
 - cpm_ParaManager_frtIF.cpp, 146, 164
- cpm_VoxelInit_nodiv_
 - cpm_ParaManager_frtIF.cpp, 146, 164
- cpm_Wait
 - cpm_ParaManager, 61
- cpm_Wait_
 - cpm_ParaManager_frtIF.cpp, 146, 165
- cpm_Waitall
 - cpm_ParaManager, 64
- cpm_Waitall_
 - cpm_ParaManager_frtIF.cpp, 146, 167
- cpm_strCompare
 - cpm_Base, 12
- cpm_strCompareN
 - cpm_Base, 13
- cpm_wait_BndCommsS3D
 - cpm_ParaManager, 61
- cpm_wait_BndCommsS3D_
 - cpm_ParaManager_frtIF.cpp, 146, 165
- cpm_wait_BndCommsS4D
 - cpm_ParaManager, 62
- cpm_wait_BndCommsS4D_
 - cpm_ParaManager_frtIF.cpp, 146, 166
- cpm_wait_BndCommsS4DEx
 - cpm_ParaManager, 63
- cpm_wait_BndCommsS4DEx_
 - cpm_ParaManager_frtIF.cpp, 146, 166
- cpm_wait_BndCommV3D
 - cpm_ParaManager, 63
- cpm_wait_BndCommV3D_
 - cpm_ParaManager_frtIF.cpp, 146, 166
- cpm_wait_BndCommV3DEx
 - cpm_ParaManager, 64
- cpm_wait_BndCommV3DEx_
 - cpm_ParaManager_frtIF.cpp, 146, 167
- Create
 - cpm_ObjList, 28
 - CreateLocalDomainInfo
 - cpm_VoxelInfo, 110
 - CreateNeighborRankInfo
 - cpm_VoxelInfo, 110
 - CreateProcessGroup
 - cpm_ParaManager, 65
 - CreateRankMap
 - cpm_VoxelInfo, 110
 - DecideDivPattern
 - cpm_ParaManager, 65
 - DelKeyList
 - cpm_ObjList, 27
 - Delete
 - cpm_ObjList, 28
 - FindVoxelInfo
 - cpm_ParaManager, 65
 - flush
 - cpm_ParaManager, 66
 - Gather
 - cpm_ParaManager, 66
 - Gatherv
 - cpm_ParaManager, 67
 - Get
 - cpm_ObjList, 28
 - get_instance
 - cpm_ParaManager, 68
 - GetBCID
 - cpm_ActiveSubDomainInfo, 9
 - cpm_ParaManager, 69
 - cpm_VoxelInfo, 111
 - GetBndCommBuffer
 - cpm_ParaManager, 69
 - GetBndCommBufferSize
 - cpm_ParaManager, 69
 - getCommNull
 - cpm_Base, 13
 - GetDivNum
 - cpm_GlobalDomainInfo, 22
 - cpm_ParaManager, 70
 - cpm_VoxelInfo, 111
 - GetDivPos
 - cpm_ParaManager, 70
 - cpm_VoxelInfo, 111
 - GetGlobalOrigin
 - cpm_ParaManager, 70
 - cpm_VoxelInfo, 111
 - GetGlobalRegion
 - cpm_ParaManager, 70
 - cpm_VoxelInfo, 112
 - GetGlobalVoxelSize
 - cpm_ParaManager, 71
 - cpm_VoxelInfo, 112
 - GetLocalOrigin
 - cpm_ParaManager, 71
 - cpm_VoxelInfo, 112

- GetLocalRegion
 - cpm_ParaManager, 71
 - cpm_VoxelInfo, 112
- GetLocalVoxelSize
 - cpm_ParaManager, 72
 - cpm_VoxelInfo, 113
- GetMPI_Comm
 - cpm_ParaManager, 72
- GetMPI_Datatype
 - cpm_ParaManager, 72, 73
- GetMPI_Op
 - cpm_ParaManager, 73
- GetMemString
 - cpm_Base, 13
- GetMyRankID
 - cpm_ParaManager, 73
- GetNeighborRankID
 - cpm_ParaManager, 74
 - cpm_VoxelInfo, 113
- GetNumRank
 - cpm_ParaManager, 74
- GetOrigin
 - cpm_DomainInfo, 17
- GetPeriodicRankID
 - cpm_ParaManager, 74
 - cpm_VoxelInfo, 113
- GetPitch
 - cpm_DomainInfo, 18
 - cpm_ParaManager, 75
 - cpm_VoxelInfo, 113
- GetPos
 - cpm_ActiveSubDomainInfo, 9
- getRankNull
 - cpm_Base, 13
- GetRegion
 - cpm_DomainInfo, 18
- GetSpanTime
 - cpm_Base, 14
- GetSubDomainInfo
 - cpm_GlobalDomainInfo, 23
- GetSubDomainNum
 - cpm_GlobalDomainInfo, 23
- GetTime
 - cpm_Base, 14
- GetVoxNum
 - cpm_DomainInfo, 18
- GetVoxelHeadIndex
 - cpm_ParaManager, 75
 - cpm_VoxelInfo, 114
- GetVoxelTailIndex
 - cpm_ParaManager, 75
 - cpm_VoxelInfo, 114
- GetWSpanTime
 - cpm_Base, 14
- GetWTime
 - cpm_Base, 14
- Init
 - cpm_VoxelInfo, 114
- InitArray
 - cpm_ParaManager, 75
- Initialize
 - cpm_ParaManager, 76
- Irecv
 - cpm_ParaManager, 76, 77
- IsCommNull
 - cpm_Base, 14
- IsExistSubDomain
 - cpm_GlobalDomainInfo, 23
- IsParallel
 - cpm_ParaManager, 78
- IsRankNull
 - cpm_Base, 15
- Isend
 - cpm_ParaManager, 77, 78
- m_DelKeyList
 - cpm_ObjList, 29
- m_ObjectMap
 - cpm_ObjList, 29
- m_bcid
 - cpm_ActiveSubDomainInfo, 11
- m_bndCommInfoMap
 - cpm_ParaManager, 100
- m_bufX
 - S_BNDCOMM_BUFFER, 117
- m_bufY
 - S_BNDCOMM_BUFFER, 117
- m_bufZ
 - S_BNDCOMM_BUFFER, 118
- m_comm
 - cpm_VoxelInfo, 115
- m_divNum
 - cpm_GlobalDomainInfo, 24
- m_globalDomainInfo
 - cpm_VoxelInfo, 115
- m_localDomainInfo
 - cpm_VoxelInfo, 115
- m_maxN
 - S_BNDCOMM_BUFFER, 118
- m_maxVC
 - S_BNDCOMM_BUFFER, 118
- m_nRank
 - cpm_ParaManager, 100
 - cpm_VoxelInfo, 115
- m_neighborRankID
 - cpm_VoxelInfo, 115
- m_newKey
 - cpm_ObjList, 29
- m_nwX
 - S_BNDCOMM_BUFFER, 118
- m_nwY
 - S_BNDCOMM_BUFFER, 118
- m_nwZ
 - S_BNDCOMM_BUFFER, 118
- m_origin
 - cpm_DomainInfo, 20
- m_periodicRankID

- cpm_VoxelInfo, 115
- m_pitch
 - cpm_DomainInfo, 20
- m_pos
 - cpm_ActiveSubDomainInfo, 11
- m_procGrpList
 - cpm_ParaManager, 100
- m_rankMap
 - cpm_VoxelInfo, 116
- m_rankNo
 - cpm_ParaManager, 101
 - cpm_VoxelInfo, 116
- m_rankNoMap
 - cpm_ParaManager, 101
- m_region
 - cpm_DomainInfo, 20
- m_reqList
 - cpm_ParaManager, 101
- m_subDomainInfo
 - cpm_GlobalDomainInfo, 24
- m_tp
 - cpm_TextParser, 104
- m_voxNum
 - cpm_DomainInfo, 20
- m_voxelHeadIndex
 - cpm_VoxelInfo, 116
- m_voxelInfoMap
 - cpm_ParaManager, 101
- m_voxelTailIndex
 - cpm_VoxelInfo, 116
- MINUS2PLUS
 - cpm_Define.h, 130
- ObjectMap
 - cpm_ObjList, 27
- operator==
 - cpm_ActiveSubDomainInfo, 9
- PLUS2MINUS
 - cpm_Define.h, 130
- packX
 - cpm_ParaManager, 79
- packXEx
 - cpm_ParaManager, 79
- packY
 - cpm_ParaManager, 80
- packYEx
 - cpm_ParaManager, 80
- packZ
 - cpm_ParaManager, 81
- packZEx
 - cpm_ParaManager, 81
- PeriodicCommS3D
 - cpm_ParaManager, 82
- PeriodicCommS4D
 - cpm_ParaManager, 83
- PeriodicCommS4DEx
 - cpm_ParaManager, 84
- PeriodicCommV3D
 - cpm_ParaManager, 85
- PeriodicCommV3DEx
 - cpm_ParaManager, 86
- REAL_BUF_TYPE
 - cpm_Define.h, 126
- REAL_TYPE
 - cpm_Define.h, 126
- RankNoMap
 - cpm_ObjList.h, 133
 - cpm_ParaManager.h, 135
- Read
 - cpm_TextParser, 103
 - cpm_TextParserDomain, 106
- ReadDomainInfo
 - cpm_TextParserDomain, 106
- ReadMain
 - cpm_TextParserDomain, 107
- ReadSubdomainInfo
 - cpm_TextParserDomain, 107
- readVector
 - cpm_TextParser, 103, 104
- ReallsDouble
 - cpm_Base, 15
- Recv
 - cpm_ParaManager, 87
- S_BNDCOMM_BUFFER, 116
 - ~S_BNDCOMM_BUFFER, 117
 - m_bufX, 117
 - m_bufY, 117
 - m_bufZ, 118
 - m_maxN, 118
 - m_maxVC, 118
 - m_nwX, 118
 - m_nwY, 118
 - m_nwZ, 118
 - S_BNDCOMM_BUFFER, 117
 - S_BNDCOMM_BUFFER, 117
- Send
 - cpm_ParaManager, 88
- sendrecv
 - cpm_ParaManager, 89
- Set
 - cpm_ActiveSubDomainInfo, 10
- SetBCID
 - cpm_ActiveSubDomainInfo, 10
- SetBndCommBuffer
 - cpm_ParaManager, 89
- SetDivNum
 - cpm_GlobalDomainInfo, 24
- SetOrigin
 - cpm_DomainInfo, 18
- SetPitch
 - cpm_DomainInfo, 19
- SetPos
 - cpm_ActiveSubDomainInfo, 10
- SetRegion
 - cpm_DomainInfo, 19

SetVoxNum
 cpm_DomainInfo, [19](#)

unpackX
 cpm_ParaManager, [90](#)

unpackXEx
 cpm_ParaManager, [90](#)

unpackY
 cpm_ParaManager, [91](#)

unpackYEx
 cpm_ParaManager, [91](#)

unpackZ
 cpm_ParaManager, [92](#)

unpackZEx
 cpm_ParaManager, [92](#)

VersionInfo
 cpm_Base, [15](#)

VoxelInfoMap
 cpm_ParaManager.h, [135](#)

VoxelInit
 cpm_ParaManager, [93](#), [94](#)

Wait
 cpm_ParaManager, [94](#)

wait_BndCommS3D
 cpm_ParaManager, [94](#), [95](#)

wait_BndCommS4D
 cpm_ParaManager, [95](#), [96](#)

wait_BndCommS4DEx
 cpm_ParaManager, [97](#)

wait_BndCommV3D
 cpm_ParaManager, [98](#)

wait_BndCommV3DEx
 cpm_ParaManager, [99](#)

Waitall
 cpm_ParaManager, [100](#)

X_DIR
 cpm_Define.h, [127](#)

X_MINUS
 cpm_Define.h, [129](#)

X_PLUS
 cpm_Define.h, [129](#)

Y_DIR
 cpm_Define.h, [127](#)

Y_MINUS
 cpm_Define.h, [129](#)

Y_PLUS
 cpm_Define.h, [129](#)

Z_DIR
 cpm_Define.h, [127](#)

Z_MINUS
 cpm_Define.h, [129](#)

Z_PLUS
 cpm_Define.h, [129](#)