

Cartesian Partition Manager Library  
1.0.2

作成 : Doxygen 1.8.0

Wed Jun 27 2012 18:23:46



# Contents

<b>1</b>	<b>構成索引</b>	<b>1</b>
1.1	クラス階層	1
<b>2</b>	<b>構成索引</b>	<b>3</b>
2.1	構成	3
<b>3</b>	<b>ファイル索引</b>	<b>5</b>
3.1	ファイル一覧	5
<b>4</b>	<b>クラス</b>	<b>7</b>
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>	8
4.1.3.2	<code>GetPos</code>	9
4.1.3.3	<code>operator!=</code>	9
4.1.3.4	<code>operator==</code>	9
4.1.3.5	<code>SetPos</code>	9
4.1.4	変数	10
4.1.4.1	<code>m_pos</code>	10
4.2	クラス <code>cpm_Base</code>	10
4.2.1	説明	11
4.2.2	コンストラクタとデストラクタ	11
4.2.2.1	<code>cpm_Base</code>	11
4.2.2.2	<code>~cpm_Base</code>	11
4.2.3	関数	11
4.2.3.1	<code>cpm_strCompare</code>	11
4.2.3.2	<code>cpm_strCompareN</code>	12

4.2.3.3	getCommNull	12
4.2.3.4	GetMemString	12
4.2.3.5	getRankNull	13
4.2.3.6	GetSpanTime	13
4.2.3.7	GetTime	13
4.2.3.8	GetWSpanTime	13
4.2.3.9	GetWTime	13
4.2.3.10	IsCommNull	14
4.2.3.11	IsRankNull	14
4.2.3.12	ReallIsDouble	14
4.2.3.13	VersionInfo	14
4.2.3.14	VersionInfo	15
4.3	クラス cpm_DomainInfo	15
4.3.1	説明	16
4.3.2	コンストラクタとデストラクタ	16
4.3.2.1	cpm_DomainInfo	16
4.3.2.2	~cpm_DomainInfo	16
4.3.3	関数	16
4.3.3.1	CheckData	16
4.3.3.2	clear	17
4.3.3.3	GetOrigin	17
4.3.3.4	GetPitch	17
4.3.3.5	GetRegion	17
4.3.3.6	GetVoxNum	18
4.3.3.7	SetOrigin	18
4.3.3.8	SetPitch	18
4.3.3.9	SetRegion	18
4.3.3.10	SetVoxNum	19
4.3.4	変数	19
4.3.4.1	m_origin	19
4.3.4.2	m_pitch	19
4.3.4.3	m_region	19
4.3.4.4	m_voxNum	19
4.4	クラス cpm_GlobalDomainInfo	19
4.4.1	説明	21
4.4.2	コンストラクタとデストラクタ	21
4.4.2.1	cpm_GlobalDomainInfo	21
4.4.2.2	~cpm_GlobalDomainInfo	21
4.4.3	関数	21
4.4.3.1	AddSubdomain	21

4.4.3.2	CheckData	21
4.4.3.3	clear	22
4.4.3.4	GetDivNum	22
4.4.3.5	GetSubdomainArraySize	22
4.4.3.6	GetSubdomainInfo	22
4.4.3.7	GetSubdomainNum	23
4.4.3.8	IsExistSubdomain	23
4.4.3.9	SetDivNum	23
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	25
4.5.3	関数	25
4.5.3.1	clear	25
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	27
4.6.4.1	Add	27
4.6.4.2	Create	28
4.6.4.3	Delete	28
4.6.4.4	Get	28
4.6.5	変数	28
4.6.5.1	m_DelKeyList	28
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	35
4.7.2.1	cpm_ParaManager	35
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	36
4.7.3.4	Allgatherv	37
4.7.3.5	Allgatherv	37
4.7.3.6	AllocDoubleS3D	38
4.7.3.7	AllocDoubleS4D	38
4.7.3.8	AllocDoubleS4DEx	38
4.7.3.9	AllocDoubleV3D	39
4.7.3.10	AllocDoubleV3DEx	39
4.7.3.11	AllocFloatS3D	39
4.7.3.12	AllocFloatS4D	40
4.7.3.13	AllocFloatS4DEx	40
4.7.3.14	AllocFloatV3D	40
4.7.3.15	AllocFloatV3DEx	41
4.7.3.16	AllocIntS3D	41
4.7.3.17	AllocIntS4D	41
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	43
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	45
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	50

4.7.3.40	BndCommS4DEx	51
4.7.3.41	BndCommS4DEx_nowait	51
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	53
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	55
4.7.3.50	BndCommV3DEx_nowait	56
4.7.3.51	CalcCommSize	57
4.7.3.52	CopyArray	57
4.7.3.53	cpm_BndCommsS3D_nowait	57
4.7.3.54	cpm_BndCommsS4D_nowait	58
4.7.3.55	cpm_BndCommsS4DEx_nowait	58
4.7.3.56	cpm_BndCommV3D_nowait	59
4.7.3.57	cpm_BndCommV3DEx_nowait	59
4.7.3.58	cpm_Irecv	60
4.7.3.59	cpm_Isend	60
4.7.3.60	cpm_Wait	61
4.7.3.61	cpm_wait_BndCommsS3D	61
4.7.3.62	cpm_wait_BndCommsS4D	62
4.7.3.63	cpm_wait_BndCommsS4DEx	62
4.7.3.64	cpm_wait_BndCommV3D	63
4.7.3.65	cpm_wait_BndCommV3DEx	63
4.7.3.66	cpm_Waitall	64
4.7.3.67	CreateProcessGroup	64
4.7.3.68	DecideDivPattern	65
4.7.3.69	FindVoxelInfo	65
4.7.3.70	flush	65
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	67
4.7.3.77	get_instance	68
4.7.3.78	GetBndCommBuffer	68
4.7.3.79	GetBndCommBufferSize	69

4.7.3.80	GetDivNum	69
4.7.3.81	GetDivPos	69
4.7.3.82	GetGlobalOrigin	69
4.7.3.83	GetGlobalRegion	70
4.7.3.84	GetGlobalVoxelSize	70
4.7.3.85	GetLocalOrigin	70
4.7.3.86	GetLocalRegion	71
4.7.3.87	GetLocalVoxelSize	71
4.7.3.88	GetMPI_Comm	71
4.7.3.89	GetMPI_Datatype	72
4.7.3.90	GetMPI_Datatype	72
4.7.3.91	GetMPI_Op	72
4.7.3.92	GetMyRankID	73
4.7.3.93	GetNeighborRankID	73
4.7.3.94	GetNumRank	73
4.7.3.95	GetPeriodicRankID	74
4.7.3.96	GetPitch	74
4.7.3.97	GetVoxelHeadIndex	74
4.7.3.98	GetVoxelTailIndex	74
4.7.3.99	InitArray	75
4.7.3.100	Initialize	75
4.7.3.101	Initialize	75
4.7.3.102	Irecv	76
4.7.3.103	Irecv	76
4.7.3.104	Isend	76
4.7.3.105	Isend	77
4.7.3.106	IsParallel	77
4.7.3.107	IsParallel	78
4.7.3.108	packX	78
4.7.3.109	packXEx	78
4.7.3.110	packY	79
4.7.3.111	packYEx	79
4.7.3.112	packZ	80
4.7.3.113	packZEx	80
4.7.3.114	PeriodicCommS3D	81
4.7.3.115	PeriodicCommS3D	81
4.7.3.116	PeriodicCommS4D	82
4.7.3.117	PeriodicCommS4D	83
4.7.3.118	PeriodicCommS4DEx	83
4.7.3.119	PeriodicCommS4DEx	84



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

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

4.10.3.14	GetPitch	112
4.10.3.15	GetVoxelHeadIndex	112
4.10.3.16	GetVoxelTailIndex	113
4.10.3.17	Init	113
4.10.4	フレンドと関連する関数	113
4.10.4.1	cpm_ParaManager	113
4.10.5	変数	113
4.10.5.1	m_comm	114
4.10.5.2	m_globalDomainInfo	114
4.10.5.3	m_localDomainInfo	114
4.10.5.4	m_neighborRankID	114
4.10.5.5	m_nRank	114
4.10.5.6	m_periodicRankID	114
4.10.5.7	m_rankMap	114
4.10.5.8	m_rankNo	115
4.10.5.9	m_voxelHeadIndex	115
4.10.5.10	m_voxelTailIndex	115
4.11	構造体 S_BNDCOMM_BUFFER	115
4.11.1	説明	116
4.11.2	コンストラクタとデストラクタ	116
4.11.2.1	S_BNDCOMM_BUFFER	116
4.11.2.2	~S_BNDCOMM_BUFFER	116
4.11.3	変数	116
4.11.3.1	m_bufX	116
4.11.3.2	m_bufY	116
4.11.3.3	m_bufZ	116
4.11.3.4	m_maxN	117
4.11.3.5	m_maxVC	117
4.11.3.6	m_nwX	117
4.11.3.7	m_nwY	117
4.11.3.8	m_nwZ	117
5	ファイル	119
5.1	cpm_Base.h	119
5.1.1	説明	120
5.1.2	マクロ定義	120
5.1.2.1	CPM_INLINE	120
5.2	cpm_Define.h	120
5.2.1	説明	122
5.2.2	マクロ定義	122

5.2.2.1	<code>_IDX_S3D</code>	122
5.2.2.2	<code>_IDX_S4D</code>	123
5.2.2.3	<code>_IDX_S4DEX</code>	123
5.2.2.4	<code>_IDX_V3D</code>	124
5.2.2.5	<code>_IDX_V3DEX</code>	124
5.2.2.6	<code>REAL_BUF_TYPE</code>	124
5.2.2.7	<code>REAL_TYPE</code>	124
5.2.3	列挙型	125
5.2.3.1	<code>CPM_Datatype</code>	125
5.2.3.2	<code>cpm_DirFlag</code>	125
5.2.3.3	<code>cpm_ErrorCode</code>	125
5.2.3.4	<code>cpm_FaceFlag</code>	127
5.2.3.5	<code>CPM_Op</code>	127
5.2.3.6	<code>cpm_PMFlag</code>	128
5.3	<code>cpm_DomainInfo.cpp</code>	128
5.3.1	説明	128
5.4	<code>cpm_DomainInfo.h</code>	129
5.4.1	説明	129
5.5	<code>cpm_ObjList.h</code>	130
5.5.1	説明	130
5.5.2	型定義	131
5.5.2.1	<code>RankNoMap</code>	131
5.6	<code>cpm_ParaManager.cpp</code>	131
5.6.1	説明	131
5.7	<code>cpm_ParaManager.h</code>	132
5.7.1	説明	132
5.7.2	型定義	133
5.7.2.1	<code>BndCommInfoMap</code>	133
5.7.2.2	<code>RankNoMap</code>	133
5.7.2.3	<code>VoxelInfoMap</code>	133
5.8	<code>cpm_ParaManager_Alloc.cpp</code>	133
5.8.1	説明	133
5.9	<code>cpm_ParaManager_BndComm.h</code>	134
5.9.1	説明	134
5.9.2	マクロ定義	134
5.9.2.1	<code>_IDAFX</code>	134
5.9.2.2	<code>_IDXFY</code>	135
5.9.2.3	<code>_IDXFZ</code>	135
5.10	<code>cpm_ParaManager_BndCommEx.h</code>	135
5.10.1	説明	135

5.10.2 マクロ定義	136
5.10.2.1 _IDXFX	136
5.10.2.2 _IDXFY	136
5.10.2.3 _IDXFZ	136
5.11 cpm_ParaManager_frtIF.cpp	137
5.11.1 説明	140
5.11.2 マクロ定義	140
5.11.2.1 cpm_Abort_	140
5.11.2.2 cpm_Allgather_	140
5.11.2.3 cpm_Allgatherv_	140
5.11.2.4 cpm_Allreduce_	140
5.11.2.5 cpm_Barrier_	140
5.11.2.6 cpm_Bcast_	140
5.11.2.7 cpm_BndCommS3D_	140
5.11.2.8 cpm_BndCommS3D_nowait_	140
5.11.2.9 cpm_BndCommS4D_	141
5.11.2.10 cpm_BndCommS4D_nowait_	141
5.11.2.11 cpm_BndCommS4DEx_	141
5.11.2.12 cpm_BndCommS4DEx_nowait_	141
5.11.2.13 cpm_BndCommV3D_	141
5.11.2.14 cpm_BndCommV3D_nowait_	141
5.11.2.15 cpm_BndCommV3DEx_	141
5.11.2.16 cpm_BndCommV3DEx_nowait_	141
5.11.2.17 CPM_EXTERN	141
5.11.2.18 cpm_Gather_	141
5.11.2.19 cpm_Gatherv_	141
5.11.2.20 cpm_GetDivNum_	142
5.11.2.21 cpm_GetDivPos_	142
5.11.2.22 cpm_GetGlobalOrigin_	142
5.11.2.23 cpm_GetGlobalRegion_	142
5.11.2.24 cpm_GetGlobalVoxelSize_	142
5.11.2.25 cpm_GetLocalOrigin_	142
5.11.2.26 cpm_GetLocalRegion_	142
5.11.2.27 cpm_GetLocalVoxelSize_	142
5.11.2.28 cpm_GetMyRankID_	142
5.11.2.29 cpm_GetNeighborRankID_	142
5.11.2.30 cpm_GetNumRank_	142
5.11.2.31 cpm_GetPeriodicRankID_	142
5.11.2.32 cpm_GetPitch_	143
5.11.2.33 cpm_GetVoxelHeadIndex_	143

5.11.2.34	<code>cpm_GetVoxelTailIndex_</code>	143
5.11.2.35	<code>cpm_Initialize_</code>	143
5.11.2.36	<code>cpm_Irecv_</code>	143
5.11.2.37	<code>cpm_Isend_</code>	143
5.11.2.38	<code>cpm_IsParallel_</code>	143
5.11.2.39	<code>cpm_PeriodicCommS3D</code>	143
5.11.2.40	<code>cpm_PeriodicCommS4D</code>	143
5.11.2.41	<code>cpm_PeriodicCommS4DEx</code>	143
5.11.2.42	<code>cpm_PeriodicCommV3D</code>	143
5.11.2.43	<code>cpm_PeriodicCommV3DEx</code>	143
5.11.2.44	<code>cpm_Recv_</code>	144
5.11.2.45	<code>cpm_Send_</code>	144
5.11.2.46	<code>cpm_SetBndCommBuffer_</code>	144
5.11.2.47	<code>cpm_Voxellnit_</code>	144
5.11.2.48	<code>cpm_Voxellnit_nodiv_</code>	144
5.11.2.49	<code>cpm_Wait_</code>	144
5.11.2.50	<code>cpm_wait_BndCommsS3D_</code>	144
5.11.2.51	<code>cpm_wait_BndCommsS4D_</code>	144
5.11.2.52	<code>cpm_wait_BndCommsS4DEx_</code>	144
5.11.2.53	<code>cpm_wait_BndCommV3D_</code>	144
5.11.2.54	<code>cpm_wait_BndCommV3DEx_</code>	144
5.11.2.55	<code>cpm_Waitall_</code>	144
5.11.3	関数	145
5.11.3.1	<code>cpm_Abort_</code>	145
5.11.3.2	<code>cpm_Allgather_</code>	145
5.11.3.3	<code>cpm_Allgatherv_</code>	145
5.11.3.4	<code>cpm_Allreduce_</code>	146
5.11.3.5	<code>cpm_Barrier_</code>	146
5.11.3.6	<code>cpm_Bcast_</code>	146
5.11.3.7	<code>cpm_BndCommS3D_</code>	147
5.11.3.8	<code>cpm_BndCommS3D_nowait_</code>	147
5.11.3.9	<code>cpm_BndCommS4D_</code>	148
5.11.3.10	<code>cpm_BndCommS4D_nowait_</code>	148
5.11.3.11	<code>cpm_BndCommS4DEx_</code>	149
5.11.3.12	<code>cpm_BndCommS4DEx_nowait_</code>	149
5.11.3.13	<code>cpm_BndCommV3D_</code>	150
5.11.3.14	<code>cpm_BndCommV3D_nowait_</code>	150
5.11.3.15	<code>cpm_BndCommV3DEx_</code>	150
5.11.3.16	<code>cpm_BndCommV3DEx_nowait_</code>	151
5.11.3.17	<code>cpm_Gather_</code>	151

5.11.3.18	<code>cpm_Gatherv_</code>	152
5.11.3.19	<code>cpm_GetDivNum_</code>	152
5.11.3.20	<code>cpm_GetDivPos_</code>	153
5.11.3.21	<code>cpm_GetGlobalOrigin_</code>	153
5.11.3.22	<code>cpm_GetGlobalRegion_</code>	153
5.11.3.23	<code>cpm_GetGlobalVoxelSize_</code>	153
5.11.3.24	<code>cpm_GetLocalOrigin_</code>	154
5.11.3.25	<code>cpm_GetLocalRegion_</code>	154
5.11.3.26	<code>cpm_GetLocalVoxelSize_</code>	154
5.11.3.27	<code>cpm_GetMyRankID_</code>	155
5.11.3.28	<code>cpm_GetNeighborRankID_</code>	155
5.11.3.29	<code>cpm_GetNumRank_</code>	155
5.11.3.30	<code>cpm_GetPeriodicRankID_</code>	155
5.11.3.31	<code>cpm_GetPitch_</code>	156
5.11.3.32	<code>cpm_GetVoxelHeadIndex_</code>	156
5.11.3.33	<code>cpm_GetVoxelTailIndex_</code>	156
5.11.3.34	<code>cpm_Initialize_</code>	157
5.11.3.35	<code>cpm_Irecv_</code>	157
5.11.3.36	<code>cpm_Isend_</code>	157
5.11.3.37	<code>cpm_IsParallel_</code>	158
5.11.3.38	<code>cpm_PeriodicComms3D_</code>	158
5.11.3.39	<code>cpm_PeriodicComms4D_</code>	159
5.11.3.40	<code>cpm_PeriodicComms4DEx_</code>	159
5.11.3.41	<code>cpm_PeriodicCommV3D_</code>	160
5.11.3.42	<code>cpm_PeriodicCommV3DEx_</code>	160
5.11.3.43	<code>cpm_Recv_</code>	161
5.11.3.44	<code>cpm_Send_</code>	161
5.11.3.45	<code>cpm_SetBndCommBuffer_</code>	161
5.11.3.46	<code>cpm_Voxellnit_</code>	162
5.11.3.47	<code>cpm_Voxellnit_nodiv_</code>	162
5.11.3.48	<code>cpm_Wait_</code>	163
5.11.3.49	<code>cpm_wait_BndComms3D_</code>	163
5.11.3.50	<code>cpm_wait_BndComms4D_</code>	163
5.11.3.51	<code>cpm_wait_BndComms4DEx_</code>	164
5.11.3.52	<code>cpm_wait_BndCommV3D_</code>	164
5.11.3.53	<code>cpm_wait_BndCommV3DEx_</code>	165
5.11.3.54	<code>cpm_Waitall_</code>	165
5.12	<code>cpm_ParaManager_inline.h</code>	166
5.12.1	説明	166
5.13	<code>cpm_ParaManager_MPI.cpp</code>	166

5.13.1 説明 . . . . .	166
5.14 cpm_TextParser.cpp . . . . .	167
5.14.1 説明 . . . . .	167
5.15 cpm_TextParser.h . . . . .	167
5.15.1 説明 . . . . .	168
5.16 cpm_TextParserDomain.cpp . . . . .	169
5.16.1 説明 . . . . .	169
5.17 cpm_TextParserDomain.h . . . . .	169
5.17.1 説明 . . . . .	170
5.18 cpm_Version.h . . . . .	171
5.18.1 説明 . . . . .	171
5.18.2 マクロ定義 . . . . .	171
5.18.2.1 CPM_REVISION . . . . .	171
5.18.2.2 CPM_VERSION_NO . . . . .	171
5.19 cpm_VoxelInfo.cpp . . . . .	172
5.19.1 説明 . . . . .	172
5.20 cpm_VoxelInfo.h . . . . .	172
5.20.1 説明 . . . . .	173



# 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 . . . . .	10
cpm_ActiveSubdomainInfo . . . . .	7
cpm_LocalDomainInfo . . . . .	24
cpm_DomainInfo . . . . .	15
cpm_GlobalDomainInfo . . . . .	19
cpm_LocalDomainInfo . . . . .	24
cpm_ObjList< T > . . . . .	26
cpm_ParaManager . . . . .	29
cpm_TextParser . . . . .	101
cpm_TextParserDomain . . . . .	104
cpm_VoxelInfo . . . . .	107
S_BNDCOMM_BUFFER . . . . .	115



## 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。

<a href="#">cpm_ActiveSubdomainInfo</a>	7
<a href="#">cpm_Base</a>	10
<a href="#">cpm_DomainInfo</a>	15
<a href="#">cpm_GlobalDomainInfo</a>	19
<a href="#">cpm_LocalDomainInfo</a>	24
<a href="#">cpm_ObjList&lt; T &gt;</a>	26
<a href="#">cpm_ParaManager</a>	29
<a href="#">cpm_TextParser</a>	101
<a href="#">cpm_TextParserDomain</a>	104
<a href="#">cpm_VoxelInfo</a>	107
<a href="#">S_BNDCOMM_BUFFER</a>	115



## 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	119
cpm_Define.h	120
cpm_DomainInfo.cpp	128
cpm_DomainInfo.h	129
cpm_ObjList.h	130
cpm_ParaManager.cpp	131
cpm_ParaManager.h	132
cpm_ParaManager_Alloc.cpp	133
cpm_ParaManager_BndComm.h	134
cpm_ParaManager_BndCommEx.h	135
cpm_ParaManager_frtIF.cpp	137
cpm_ParaManager_inline.h	166
cpm_ParaManager_MPI.cpp	166
cpm_TextParser.cpp	167
cpm_TextParser.h	167
cpm_TextParserDomain.cpp	169
cpm_TextParserDomain.h	169
cpm_Version.h	171
cpm_VoxelInfo.cpp	172
cpm_VoxelInfo.h	172



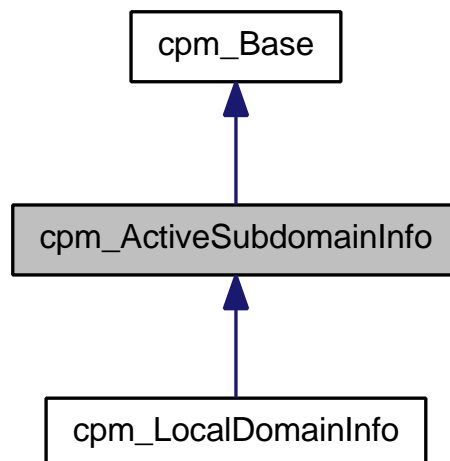
## Chapter 4

# クラス

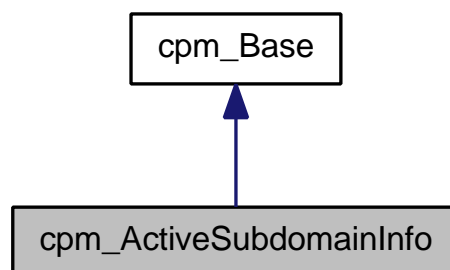
### 4.1 クラス cpm\_ActiveSubdomainInfo

```
#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\[3\]\)](#)

- virtual `~cpm_ActiveSubdomainInfo` ()
- virtual void `clear` ()
- void `SetPos` (int pos[3])
- const int \* `GetPos` () 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

### 4.1.1 説明

CPM のサブ領域情報クラス

`cpm_DomainInfo.h` の 102 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 137 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] )`

コンストラクタ

引数

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

`cpm_DomainInfo.cpp` の 145 行で定義されています。

参照先 `SetPos()`.

#### 4.1.2.3 `cpm_ActiveSubdomainInfo::~cpm_ActiveSubdomainInfo ( )` [virtual]

デストラクタ

`cpm_DomainInfo.cpp` の 153 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 160 行で定義されています。

参照先 `m_pos`.



参照元 `cpm_ActiveSubdomainInfo()`.

#### 4.1.3.2 `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 180 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()`.

#### 4.1.3.3 `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 199 行で定義されています。

参照先 `m_pos`.

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

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

参照先 `m_pos`.

#### 4.1.3.5 `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 170 行で定義されています。

参照先 `m_pos`.

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

#### 4.1.4 変数

##### 4.1.4.1 `int cpm_ActiveSubdomainInfo::m_pos[3]` `[private]`

領域分割内での位置

cpm\_DomainInfo.h の 157 行で定義されています。

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

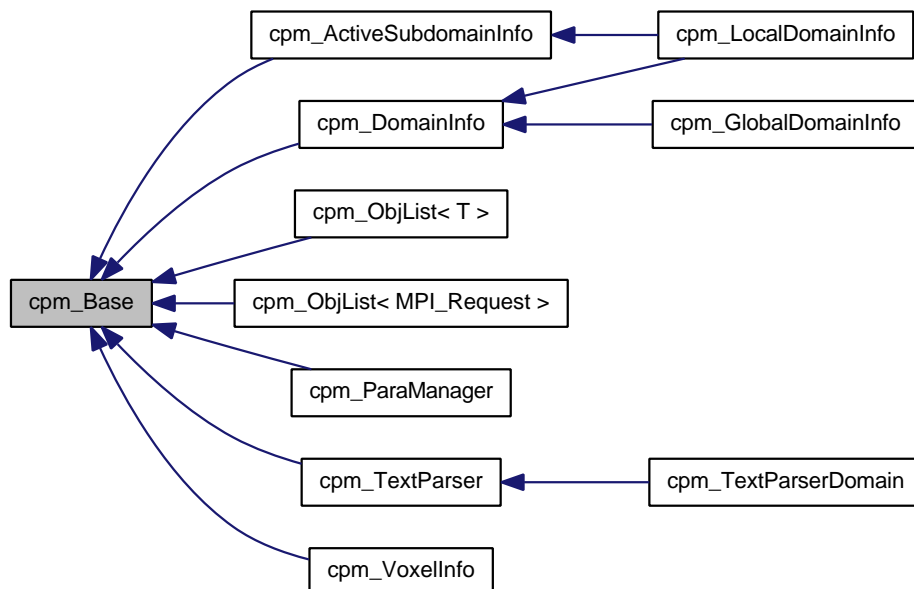
このクラスの説明は次のファイルから生成されました:

- [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 `ReallIsDouble` ()
- 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 241 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 244 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 207 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 229 行で定義されています。

参照先 cpm\_strCompare().

**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 148 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::PeriodicCommsS4D(), と cpm\_ParaManager::PeriodicCommsS4DEx().

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

in	before	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 119 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 104 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

in	before	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 138 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 128 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

true	NULL
false	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\_VoxelInfo::Init(), cpm\_ParaManager::Irecv(), cpm\_ParaManager::Isend(), cpm\_ParaManager::Recv(), cpm\_ParaManager::Send(), と cpm\_ParaManager::VoxelInit().

**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

true	NULL
false	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::RealsIsDouble ( ) [inline, static]

実数型 REAL\_TYPE が倍精度かどうか確認

戻り値

true	倍精度
false	単精度

cpm\_Base.h の 91 行で定義されています。

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

ofs	出力ストリーム
-----	---------

cpm\_Base.h の 182 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

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

cpm\_Base.h の 191 行で定義されています。

参照先 CPM\_VERSION\_NO.

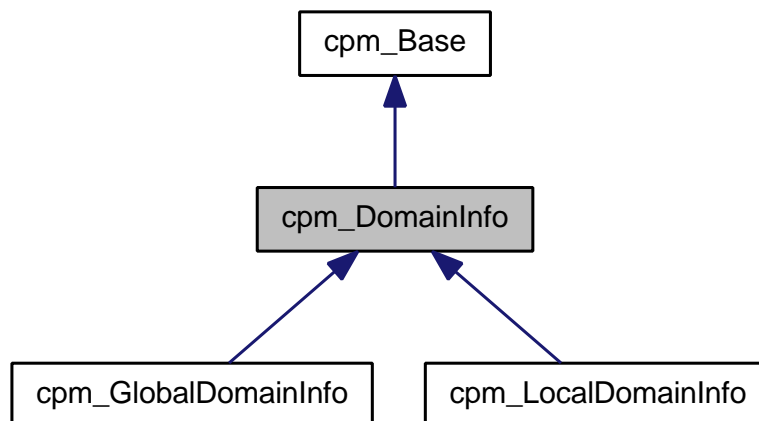
このクラスの説明は次のファイルから生成されました:

- [cpm\\_Base.h](#)

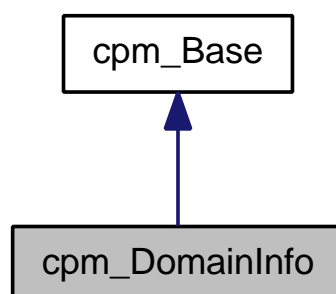
## 4.3 クラス cpm\_DomainInfo

```
#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`
- `cpm_ErrorCode CheckData()`

### Private 変数

- `REAL_TYPE m_origin [3]`  
原点
- `REAL_TYPE m_region [3]`  
空間サイズ
- `REAL_TYPE m_pitch [3]`  
ピッチ
- `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 `cpm_ErrorCode cpm_DomainInfo::CheckData()`

領域情報のチェック `VoxelInit` を実行する上で必要な情報がセットされているかを確認する。



戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_DomainInfo.cpp` の 119 行で定義されています。

参照先 `CPM_ERROR_INVALID_REGION`, `CPM_ERROR_INVALID_VOXELSIZE`, `CPM_SUCCESS`, `m_region`, `m_voxNum`, と `REAL_TYPE`.

参照元 `cpm_GlobalDomainInfo::CheckData()`.

#### 4.3.3.2 `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`, `m_voxNum`, と `REAL_TYPE`.

参照元 `cpm_DomainInfo()`.

#### 4.3.3.3 `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 58 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()`.

#### 4.3.3.4 `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 76 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()`.

#### 4.3.3.5 `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 94 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()`.

#### 4.3.3.6 `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 112 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().

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

原点のセット

引数

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

cpm\_DomainInfo.cpp の 48 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\_ParameterManager::VoxelInit().

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

ピッチのセット

引数

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

cpm\_DomainInfo.cpp の 66 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\_ParameterManager::VoxelInit().

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

空間サイズのセット

引数

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

cpm\_DomainInfo.cpp の 84 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\_ParameterManager::VoxelInit().

## 4.3.3.10 void cpm\_DomainInfo::SetVoxNum ( int vox[3] )

VOXEL 数のセット

引数

in	vox	VOXEL 数情報
----	-----	-----------

cpm\_DomainInfo.cpp の 102 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\_ParameterManager::VoxelInit().

## 4.3.4 変数

## 4.3.4.1 REAL\_TYPE cpm\_DomainInfo::m\_origin[3] [private]

原点

cpm\_DomainInfo.h の 93 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 95 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 94 行で定義されています。

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

## 4.3.4.4 int cpm\_DomainInfo::m\_voxNum[3] [private]

VOXEL 数

cpm\_DomainInfo.h の 96 行で定義されています。

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

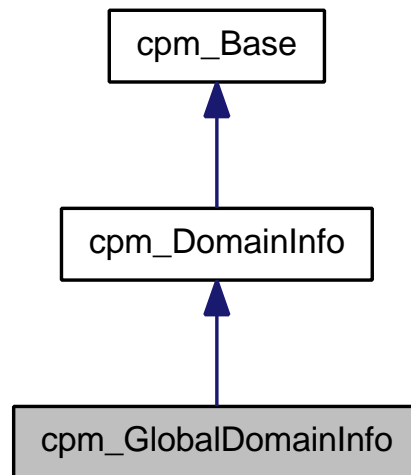
このクラスの説明は次のファイルから生成されました:

- [cpm\\_DomainInfo.h](#)
- [cpm\\_DomainInfo.cpp](#)

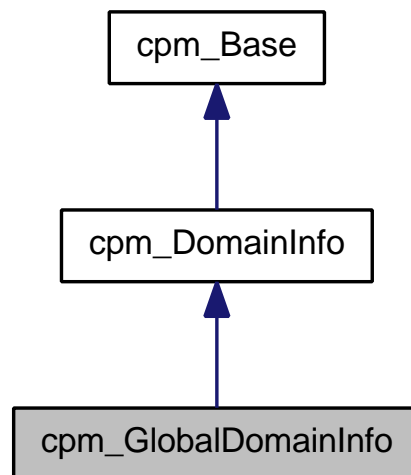
## 4.4 クラス cpm\_GlobalDomainInfo

```
#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
- int [GetSubdomainArraySize](#) () const
- const [cpm\\_ActiveSubdomainInfo](#) \* [GetSubdomainInfo](#) (size\_t idx) const
- [cpm\\_ErrorCode](#) [CheckData](#) (int nRank)

## 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 163 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 209 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 217 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 268 行で定義されています。

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

参照元 `CheckData()`.

##### 4.4.3.2 `cpm_ErrorCode cpm_GlobalDomainInfo::CheckData ( int nRank )`

領域情報のチェック `Voxellnit` を実行する上で必要な情報がセットされているかをチェックする。活性サブドメイン配列が空のとき、全領域が活性サブドメインになるため、このチェック関数内で活性サブドメイン情報を生成する。

引数

<i>in</i>	<i>nRank</i>	並列プロセス数
-----------	--------------	---------

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_DomainInfo.cpp の 313 行で定義されています。

参照先 AddSubdomain(), cpm\_DomainInfo::CheckData(), CPM\_ERROR\_INVALID\_DIVNUM, CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN, CPM\_SUCCESS, m\_divNum, と m\_subDomainInfo.

参照元 cpm\_ParaManager::VoxelInit().

**4.4.3.3 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 224 行で定義されています。

参照先 m\_divNum, と m\_subDomainInfo.

参照元 cpm\_GlobalDomainInfo().

**4.4.3.4 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 247 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().

**4.4.3.5 int cpm\_GlobalDomainInfo::GetSubdomainArraySize ( ) const**

活性サブドメインの数を取得 (情報数) 活性サブドメインの数 = 活性サブドメイン情報配列のサイズ

cpm\_DomainInfo.cpp の 297 行で定義されています。

参照先 m\_subDomainInfo.

**4.4.3.6 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

<i>in</i>	<i>idx</i>	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 305 行で定義されています。

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

参照元 `cpm_VoxelInfo::CreateRankMap()`.

#### 4.4.3.7 `int cpm_GlobalDomainInfo::GetSubdomainNum ( ) const`

活性サブドメインの数を取得 活性サブドメインの数 = 活性サブドメイン情報配列のサイズだが、この配列が空のとき、領域分割数でサブドメイン数を決定して返す

戻り値

活性サブドメインの数

`cpm_DomainInfo.cpp` の 281 行で定義されています。

参照先 `m_divNum`, と `m_subDomainInfo`.

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

#### 4.4.3.8 `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

<code>true</code>	存在する
<code>false</code>	存在しない

`cpm_DomainInfo.cpp` の 255 行で定義されています。

参照先 `m_subDomainInfo`.

参照元 `AddSubdomain()`.

#### 4.4.3.9 `void cpm_GlobalDomainInfo::SetDivNum ( int div[3] )`

領域分割数のセット

引数

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

`cpm_DomainInfo.cpp` の 237 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 239 行で定義されています。

参照元 `CheckData()`, `clear()`, `GetDivNum()`, `GetSubdomainNum()`, 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 240 行で定義されています。

参照元 `AddSubdomain()`, `CheckData()`, `clear()`, `GetSubdomainArraySize()`, `GetSubdomainInfo()`, `GetSubdomainNum()`, と `IsExistSubdomain()`.

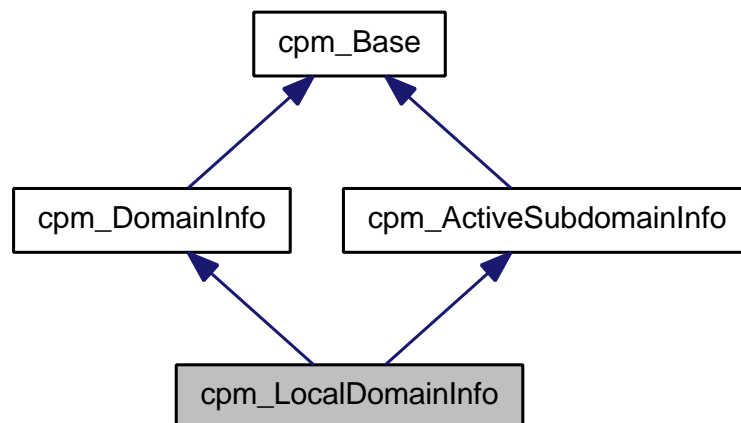
このクラスの説明は次のファイルから生成されました:

- [cpm\\_DomainInfo.h](#)
- [cpm\\_DomainInfo.cpp](#)

## 4.5 クラス `cpm_LocalDomainInfo`

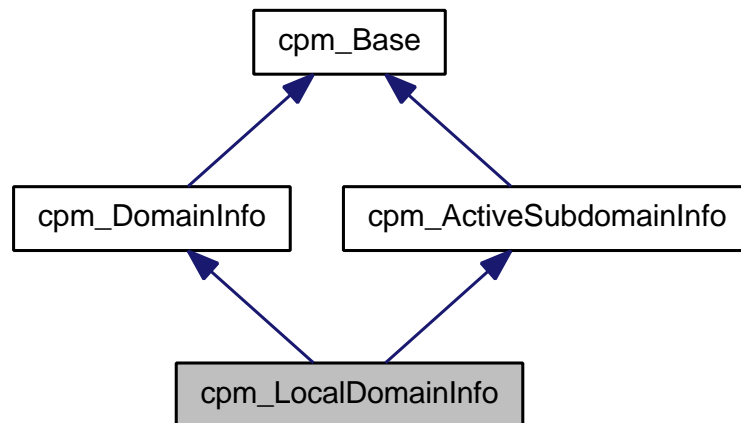
```
#include <cpm_DomainInfo.h>
```

`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 246 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 359 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 366 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 373 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:

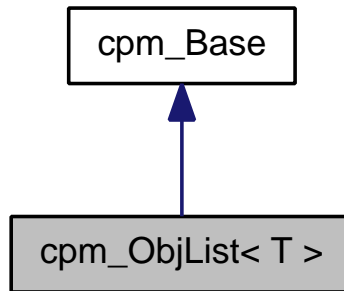
- [cpm\\_DomainInfo.h](#)

- [cpm\\_DomainInfo.cpp](#)

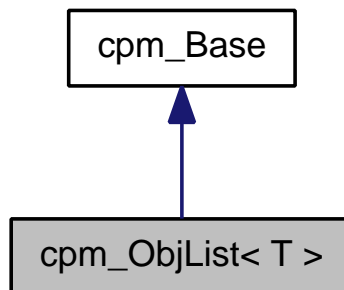
## 4.6 クラス テンプレート `cpm_ObjList< T >`

```
#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

<code>in</code>	<code>obj</code>	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

in	key	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()`.

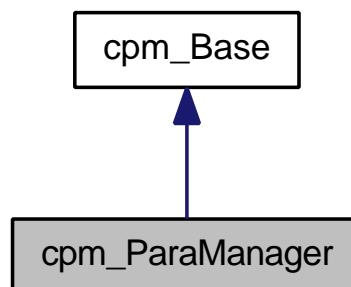
このクラスの説明は次のファイルから生成されました:

- [cpm\\_ObjList.h](#)

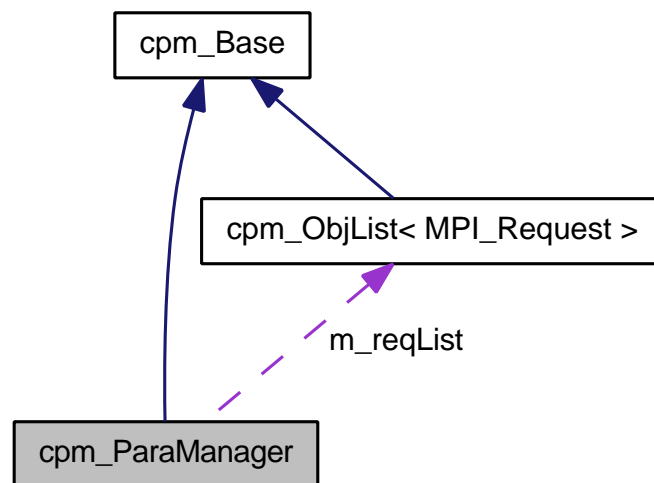
## 4.7 クラス cpm\_ParaManager

```
#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 region[3], size_t maxVC=1, size_t maxN=3, int procGrpNo=0)`
- `cpm_ErrorCode Voxellnit (int vox[3], REAL_TYPE origin[3], REAL_TYPE region[3], 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 * 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_Irecv` (void \*buf, int count, int datatype, int source, int \*reqNo, int procGrpNo=0)
- `cpm_ErrorCode cpm_BndComms3D_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_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_ErrorCode cpm_wait_BndCommS3D` (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_BndCommS4D` (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\\_BndCommsS3D](#) (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\\_BndCommsS4D](#) (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\\_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)
- `template<class T >`  
[CPM\\_INLINE cpm\\_ErrorCode PeriodicCommsS3D](#) (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 divPtn[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\\_BNDCOMM\\_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 \*recvm, T \*recvp, 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 \*recvm, T \*recvp, 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 *recvm`, `T *recvp`, `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 *recvm`, `T *recvp`, `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 *recvm`, `T *recvp`, `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 *recvm`, `T *recvp`, `int nIDm`, `int nIDp`)
- `template<class T >`  
`cpm_ErrorCode 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 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	errorcode	MPI_Abort 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	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	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 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::Allgather ( Ts * sendbuf, int sendcnt, Tr * recvbuf, int * recvcnts, int * displs, 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>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\_Allgather\_().

4.7.3.5 `cpm_ErrorCode cpm_ParaManager::Allgather ( MPI_Datatype stype, void * sendbuf, int sendcnt, MPI_Datatype rtype, void * recvbuf, int * recvcnts, int * displs, 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>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	vc	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	vc	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

<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 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

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 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

<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 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

<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<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 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

<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<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 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

<i>in</i>	<i>vc</i>	仮想セル数
<i>in</i>	<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 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	<i>dtype</i>	送信データのMPI_Datatype
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>op</i>	オペレータ
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 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	<i>procGrpNo</i>	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 Initialize().

#### 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
in	<i>count</i>	送信バッファのサイズ (ワード数)
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

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

in	<i>dtype</i>	送信バッファのMPI_Datatype
	<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>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

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::BndCommsS3D ( 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 BndCommsS4D().

参照元 cpm\_BndCommsS3D\_().

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	dtype	袖通信データのMPI_Datatype
	inout[]	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	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 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	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
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\_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
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>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, pack-



`X()`, `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

in	<i>dtype</i>	袖通信データの <code>MPI_Datatype</code>
	<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>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	dtype	袖通信データのMPI_Datatype
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	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\_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

	inout]	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	<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 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

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>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>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	<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\_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 BndCommS4DEx\_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 BndComms4D().

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 BndCommS4D\_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 BndCommS4DEx().

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 442 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	<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 (cpm_fparam.fi 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 2647 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	<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
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 2731 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 2945 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	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 2689 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	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi 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 2903 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\_().

**4.7.358 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	<i>buf</i>	受信データ
in	<i>count</i>	受信データのサイズ
in	<i>datatype</i>	受信データのデータタイプ (cpm_fparam.fi 参照)
in	<i>source</i>	送信元のランク番号 (procGrpNo 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 2615 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\_().

**4.7.359 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 (cpm_fparam.fi 参照)
in	<i>dest</i>	送信先のランク番号 (procGrpNo 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 2578 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 2514 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	<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 2769 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 2861 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 3029 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 2815 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	<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 (cpm_fparam.fi 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 2983 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	<i>count</i>	リクエストの数
in	<i>reqNoList</i>	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 2537 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 470 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 353 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 `VoxellInit()`.

**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 512 行で定義されています。

参照先 `m_voxellInfoMap`.

参照元 `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

参照元 `Initialize()`.

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

flush

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	<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>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\_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>	受信データのサイズ
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 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 (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 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\_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	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

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 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	procGrpNo	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 1772 行で定義されています。

参照先 m\_bndCommInfoMap.

参照元 BndCommS4D(), BndCommS4D\_nowait(), BndCommS4DEx(), BndCommS4DEx\_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), wait\_BndCommS4D(), と wait\_BndCommS4DEx().

4.7.3.79 `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	procGrpNo	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 745 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`.

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

領域分割数を取得

引数

in	procGrpNo	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 522 行で定義されています。

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

参照元 `cpm_GetDivNum_()`.

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

自ランクの領域分割位置を取得

引数

in	procGrpNo	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 618 行で定義されています。

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

参照元 `cpm_GetDivPos_()`.

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

全体空間の原点を取得

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

全体空間の原点実数配列のポインタ

cpm\_ParaManager.cpp の 558 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetGlobalOrigin().

参照元 cpm\_GetGlobalOrigin\_().

**4.7.3.83** `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 570 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetGlobalRegion().

参照元 cpm\_GetGlobalRegion\_().

**4.7.3.84** `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 546 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetGlobalVoxelSize().

参照元 cpm\_GetGlobalVoxelSize\_().

**4.7.3.85** `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 594 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\_().

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

自ランクの空間サイズを取得

引数

in	procGrpNo	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 606 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetLocalRegion().

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

自ランクのボクセル数を取得

引数

in	procGrpNo	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 582 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().

**4.7.3.88** `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

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

戻り値

MPI コミュニケーター

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 Voxellnit().

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

MPI\_Datatype を取得

引数

in	ptr	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\_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\_Irecv(), cpm\_Isend(), cpm\_PeriodicCommS3D\_(), cpm\_PeriodicCommS4D\_(), cpm\_PeriodicCommS4DEx\_(), cpm\_PeriodicCommV3D\_(), cpm\_PeriodicCommV3DEx\_(), cpm\_Recv\_(), cpm\_Send\_(), cpm\_wait\_BndCommS3D(), cpm\_wait\_BndCommS4D(), cpm\_wait\_BndCommS4DEx(), cpm\_wait\_BndCommV3D(), cpm\_wait\_BndCommV3DEx(), Gather(), Gatherv(), Irecv(), Isend(), Recv(), と Send().

**4.7.3.90** `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

in	datatype	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().

**4.7.3.91** `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

in	op	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\_().

#### 4.7.3.92 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

in	procGrpNo	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\_().

#### 4.7.3.93 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 654 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().

#### 4.7.3.94 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

in	procGrpNo	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 VoxellInit().

**4.7.3.95** `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 666 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetPeriodicRankID().

参照元 cpm\_GetPeriodicRankID\_(), PeriodicCommsS4D(), と PeriodicCommsS4DEx().

**4.7.3.96** `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 534 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetPitch().

参照元 cpm\_GetPitch\_().

**4.7.3.97** `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 630 行で定義されています。

参照先 FindVoxelInfo(), と cpm\_VoxelInfo::GetVoxelHeadIndex().

参照元 cpm\_GetVoxelHeadIndex\_().

**4.7.3.98** `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ックスを 0 とした C 型のインデックス

戻り値

自ランクの終点インデックス整数配列のポインタ

`cpm_ParaManager.cpp` の 642 行で定義されています。

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

参照元 `cpm_GetVoxelTailIndex_()`.

**4.7.3.99** `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	<i>array</i>	初期化する配列のポインタ
in	<i>size</i>	配列サイズ

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

**4.7.3.100** `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 `Barrier()`, `CPM_ERROR_MPI`, `CPM_ERROR_NO_MPI_INIT`, `CPM_SUCCESS`, `flush()`, `IsParallel()`, `m_n-Rank`, と `m_rankNo`.

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

**4.7.3.101** `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	<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

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_ParaManager.cpp` の 167 行で定義されています。

参照先 CPM\_ERROR\_MPI, Initialize(), m\_nRank, と m\_rankNo.

4.7.3.102 `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().

4.7.3.103 `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().

4.7.3.104 `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().

**4.7.3.105** `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().

**4.7.3.106** `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 191 行で定義されています。

参照先 m\_nRank.

参照元 cpm\_IsParallel(), と Initialize().

4.7.3.107 `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 203 行で定義されています。

参照先 m\_nRank.

4.7.3.108 `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().

4.7.3.109 `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`, `_IDXFX`, `CPM_SUCCESS`, と `cpm_Base::IsRankNull()`.

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

4.7.3.110 `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	<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 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()`.

4.7.3.111 `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	<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	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().

4.7.3.112 `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().

4.7.3.113 `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	<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 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()`.

4.7.3.114 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommS3D ( 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 `PeriodicCommS4D()`.

参照元 `cpm_PeriodicCommS3D_()`.

4.7.3.115 `cpm_ErrorCode cpm_ParaManager::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 )`

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

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 706 行で定義されています。

参照先 PeriodicCommS4D().

4.7.3.116 `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
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>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().

4.7.3.117 `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

in	dtype	袖通信データのMPI_Datatype
	inout]	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	dir	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	pm	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
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 724 行で定義されています。

参照先 CPM\_ERROR\_MPI\_INVALID\_DATATYPE, と PeriodicComms4D().

4.7.3.118 `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

	inout]	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	dir	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	pm	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
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 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(), Get-

PeriodicRankID(), 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().

**4.7.3.119** `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

in	dtype	袖通信データのMPI_Datatype
	inout]	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	dir	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	pm	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
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 936 行で定義されています。

参照先 CPM\_ERROR\_MPI\_INVALID\_DATATYPE, と PeriodicCommS4DEx().

**4.7.3.120** `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

	inout]	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	dir	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	pm	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
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\_BndComm.h の 382 行で定義されています。

参照先 PeriodicComms4D().

参照元 cpm\_PeriodicCommV3D\_().

4.7.3.121 **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().

4.7.3.122 **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\_().

**4.7.3.123** `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 PeriodicComms4DEx().

**4.7.3.124** `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().

4.7.3.125 `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().

4.7.3.126 `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().

4.7.3.127 `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().

```
4.7.3.128 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, lrecv(), lsend(), と cpm\_Base::IsRankNull().

参照元 BndCommsS4D(), BndCommsS4D\_nowait(), BndCommsS4DEx(), BndCommsS4DEx\_nowait(), PeriodicCommS4D(), と PeriodicCommS4DEx().

```
4.7.3.129 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 678 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().

4.7.3.130 `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().

4.7.3.131 `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`, `_IDXFx`, `CPM_SUCCESS`, と `cpm_Base::IsRankNull()`.

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

**4.7.3.132** `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()`.

**4.7.3.133** `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()`.

4.7.3.134 `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()`.

4.7.3.135 `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	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	recvm	マイナス方向の受信バッファ
in	recvp	プラス方向の受信バッファ
in	nlDm	マイナス方向の隣接ランク番号
in	nlDp	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().

**4.7.3.136** `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

in	domainInfo	領域分割情報
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 215 行で定義されています。

参照先 Abort(), cpm\_GlobalDomainInfo::CheckData(), 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().

**4.7.3.137** `cpm_ErrorCode cpm_ParaManager::Voxellnit ( int div[3], int vox[3], REAL_TYPE origin[3], REAL_TYPE region[3], 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>region</code>	空間全体のサイズ
<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 291 行で定義されています。

参照先 `CPM_ERROR_INVALID_REGION`, `CPM_ERROR_INVALID_VOXELSIZE`, `CPM_SUCCESS`, `DecideDivPattern()`, `GetNumRank()`, `REAL_TYPE`, `cpm_GlobalDomainInfo::SetDivNum()`, `cpm_DomainInfo::SetOrigin()`, `cpm_DomainInfo::SetPitch()`, `cpm_DomainInfo::SetRegion()`, `cpm_DomainInfo::SetVoxNum()`, と `Voxellnit()`.

**4.7.3.138** `cpm_ErrorCode cpm_ParaManager::Voxellnit ( int vox[3], REAL_TYPE origin[3], REAL_TYPE region[3], 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

<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>region</code>	空間全体のサイズ
<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 342 行で定義されています。

参照先 `Voxellnit()`.

**4.7.3.139** `cpm_ErrorCode cpm_ParaManager::Wait ( MPI_Request * request )`

## Wait

- ・ `MPI_Wait` のインターフェイス

## 引数

<code>in</code>	<code>request</code>	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.

4.7.3.140 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommS3D ( 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
<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 274 行で定義されています。

参照先 wait\_BndCommS4D().

参照元 cpm\_wait\_BndCommS3D().

4.7.3.141 `cpm_ErrorCode cpm_ParaManager::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 )`

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

<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 644 行で定義されています。

参照先 wait\_BndComms4D().

4.7.3.142 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndComms4D ( 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\_BndComms4D(), wait\_BndComms3D(), wait\_BndComms4D(), と wait\_BndCommV3D().

4.7.3.143 `cpm_ErrorCode cpm_ParaManager::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 )`

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().

4.7.3.144 `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
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>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 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().

4.7.3.145 `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

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>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>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 883 行で定義されています。

参照先 CPM\_ERROR\_MPI\_INVALID\_DATATYPE, と wait\_BndCommS4DEx().

**4.7.3.146** `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
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 284 行で定義されています。

参照先 wait\_BndCommS4D().

参照元 cpm\_wait\_BndCommV3D().

**4.7.3.147** `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

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 653 行で定義されています。

参照先 wait\_BndComms4D().

**4.7.3.148** `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
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\_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().

**4.7.3.149** `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

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

<code>in</code>	<code>vc_comm</code>	通信する仮想セル数
<code>in</code>	<code>req</code>	MPI リクエスト
<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 874 行で定義されています。

参照先 `wait_BndComms4DEx()`.

#### 4.7.3.150 `cpm_ErrorCode cpm_ParaManager::Waitall( int count, MPI_Request requests[] )`

Waitall

- MPI\_Waitall のインターフェイス

引数

<code>in</code>	<code>count</code>	リクエストの数
<code>in</code>	<code>requests</code>	リクエストハンドルの配列

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 2050 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 2023 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 2035 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 2026 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 2046 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 2055 行で定義されています。

参照元 cpm\_BndCommS3D\_nowait(), cpm\_BndCommS4D\_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 2041 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:

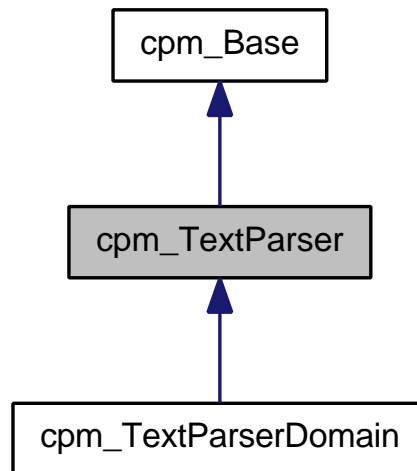
- [cpm\\_ParaManager.h](#)
- [cpm\\_ParaManager.cpp](#)

- [cpm\\_ParaManager\\_Alloc.cpp](#)
- [cpm\\_ParaManager\\_frtIF.cpp](#)
- [cpm\\_ParaManager\\_MPI.cpp](#)
- [cpm\\_ParaManager\\_BndComm.h](#)
- [cpm\\_ParaManager\\_BndCommEx.h](#)
- [cpm\\_ParaManager\\_inline.h](#)

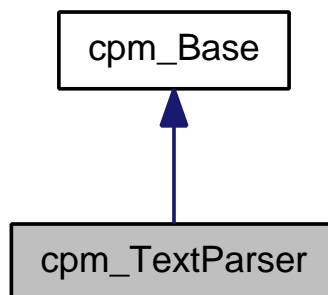
## 4.8 クラス cpm\_TextParser

```
#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 nvec 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 <code>nvec</code> 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()`.

**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	<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 <code>nvec</code> 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 <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	<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 <code>nvec</code> 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 <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::ReadSubdomainInfo()`, `readVector()`, と `~cpm_TextParser()`.

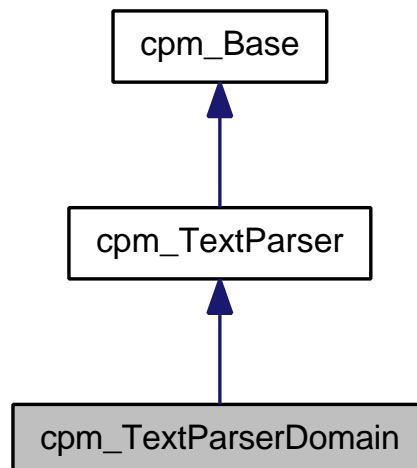
このクラスの説明は次のファイルから生成されました:

- [cpm\\_TextParser.h](#)
- [cpm\\_TextParser.cpp](#)

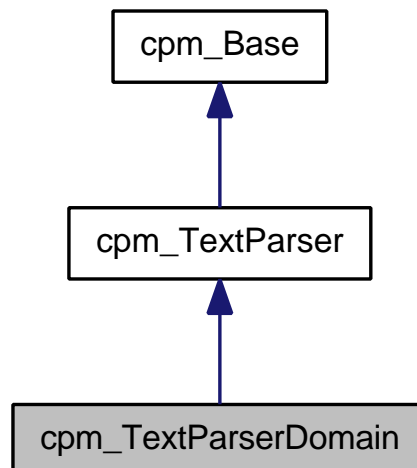
## 4.9 クラス `cpm_TextParserDomain`

```
#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\_RGN, 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

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 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]`

ActiveSubdomainInfo 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 245 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_TP\_INVALID\_G\_DIV, cpm\_Base::cpm\_strCompare(), CPM\_SUCCESS, cpm\_GlobalDomainInfo::GetDivNum(), と cpm\_TextParser::m\_tp.

参照元 ReadMain().

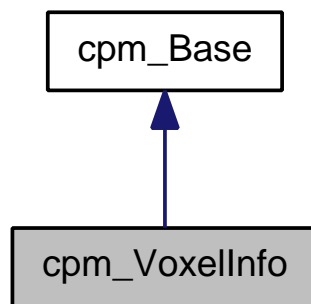
このクラスの説明は次のファイルから生成されました:

- [cpm\\_TextParserDomain.h](#)
- [cpm\\_TextParserDomain.cpp](#)

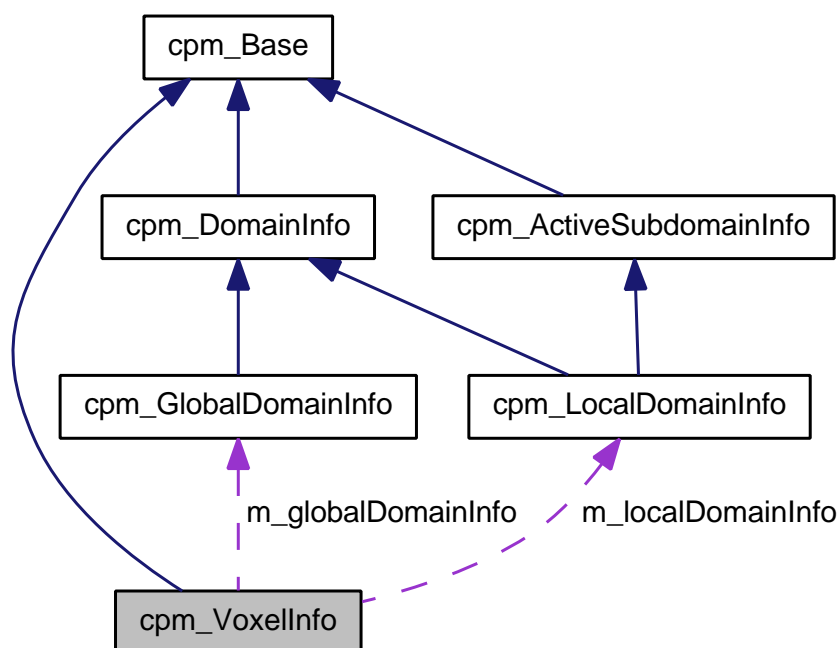
## 4.10 クラス cpm\_VoxelInfo

```
#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 \* [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_GlobalDomainInfo::GetDivNum()`, `cpm_DomainInfo::GetOrigin()`, `cpm_DomainInfo::GetPitch()`, `cpm_DomainInfo::GetVoxNum()`, `m_globalDomainInfo`, `m_localDomainInfo`, `m_rankMap`, `m_rankNo`, `m_voxelHeadIndex`, `m_voxelTailIndex`, `REAL_TYPE`, `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 266 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\_VoxellInfo::CreateRankMap ( ) [private]

ランクマップを生成

戻り値

<i>true</i>	正常終了
<i>false</i>	エラー

cpm\_VoxellInfo.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\_VoxellInfo::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\_VoxellInfo.cpp の 372 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().

#### 4.10.3.5 const int \* cpm\_VoxellInfo::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\_VoxellInfo.cpp の 436 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().

#### 4.10.3.6 const REAL\_TYPE \* cpm\_VoxellInfo::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\_VoxellInfo.cpp の 396 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().

4.10.3.7 `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 404 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()`.

4.10.3.8 `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 388 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()`.

4.10.3.9 `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 420 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()`.

4.10.3.10 `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 428 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()`.

4.10.3.11 `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 412 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().

4.10.3.12 `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 460 行で定義されています。

参照先 m\_neighborRankID.

参照元 cpm\_ParaManager::GetNeighborRankID().

4.10.3.13 `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 468 行で定義されています。

参照先 m\_periodicRankID.

参照元 cpm\_ParaManager::GetPeriodicRankID().

4.10.3.14 `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 380 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().

4.10.3.15 `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 444 行で定義されています。

参照先 `m_voxelHeadIndex`.

参照元 `cpm_ParaManager::GetVoxelHeadIndex()`.

#### 4.10.3.16 `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 452 行で定義されています。

参照先 `m_voxelTailIndex`.

参照元 `cpm_ParaManager::GetVoxelTailIndex()`.

#### 4.10.3.17 `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

<code>in</code>	<code>comm</code>	MPI コミュニケータ
<code>in</code>	<code>dInfo</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)

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`, `CreateLocalDomainInfo()`, `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

cpm\_VoxelInfo.h の 150 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 142 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 145 行で定義されています。

参照元 CreateLocalDomainInfo(), CreateNeighborRankInfo(), 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 153 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]

##### コミュニケータ内のランク数 (=プロセス並列数)

cpm\_VoxelInfo.h の 151 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 154 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 156 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]

コミュニケータ内でのランク番号

cpm\_VoxelInfo.h の 152 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 146 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 147 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](#)



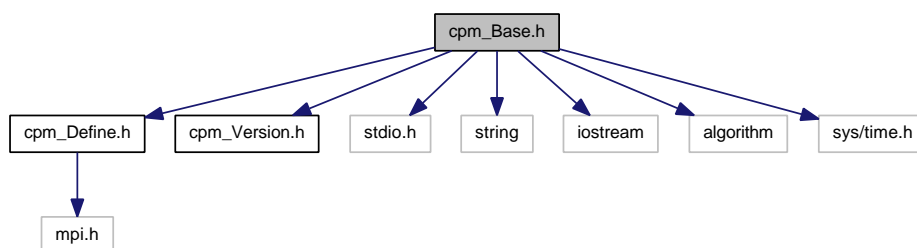
## Chapter 5

# ファイル

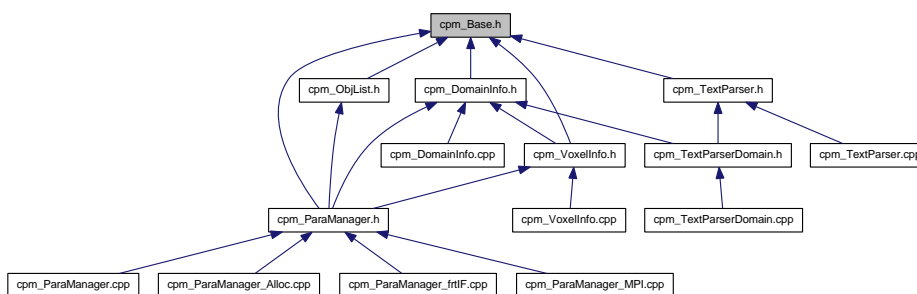
### 5.1 cpm\_Base.h

```
#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

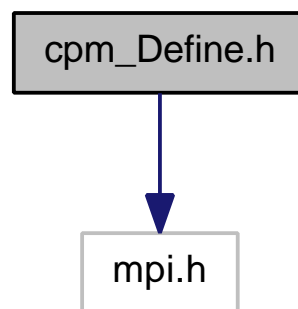
#### 5.1.2.1 `#define CPM_INLINE inline`

`cpm_Base.h` の 38 行で定義されています。

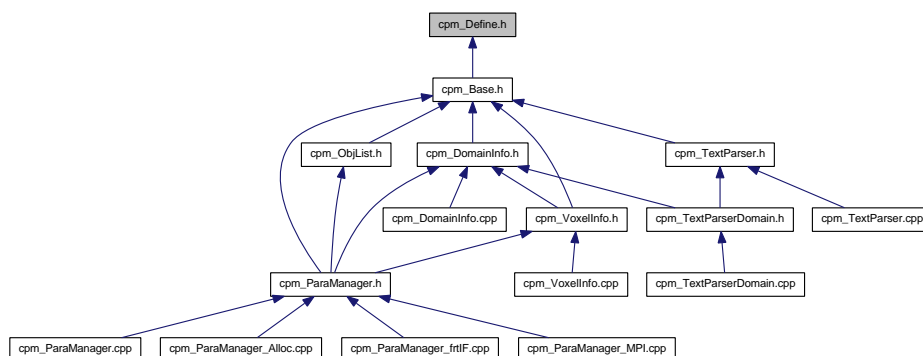
## 5.2 `cpm_Define.h`

```
#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, X\_PLUS = 1, Y\_MINUS = 2, Y\_PLUS = 3,  
Z\_MINUS = 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\_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\_VOXELMAP = 3009, CPM\_ERROR\_INVALID\_REGION = 3010,  
CPM\_ERROR\_INVALID\_DIVNUM = 3011, 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\_LOCALVOXELSIZE = 4006,  
CPM\_ERROR\_GET\_LOCALORIGIN = 4007, CPM\_ERROR\_GET\_LOCALREGION = 4008, CPM\_ERROR\_GET\_DIVPOS = 4009, CPM\_ERROR\_GET\_HEADINDEX = 4011,  
CPM\_ERROR\_GET\_TAILINDEX = 4012, CPM\_ERROR\_GET\_NEIGHBOR\_RANK = 4013, CPM\_ERROR\_GET\_PERIODIC\_FACE = 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\_FACE = 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次元インデックス (i,j,k) -> 1次元インデックス変換マクロ

引数

in	<code>_I</code>	i 方向インデックス
in	<code>_J</code>	j 方向インデックス
in	<code>_K</code>	k 方向インデックス
in	<code>_NI</code>	i 方向インデックスサイズ
in	<code>_NJ</code>	j 方向インデックスサイズ
in	<code>_NK</code>	k 方向インデックスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1次元インデックス

`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) -&gt; 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) -&gt; 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\_DomainInfo::CheckData(), cpm\_DomainInfo::clear(), cpm\_GetGlobalOrigin\_(), cpm\_GetGlobalRegion\_(), cpm\_GetLocalOrigin\_(), cpm\_GetLocalRegion\_(), cpm\_GetPitch\_(), cpm\_VoxelInfo::CreateLocalDomainInfo(), cpm\_TextParserDomain::ReadDomainInfo(), 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\_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\_INVALID\_REGION** 全体空間サイズが不正  
**CPM\_ERROR\_INVALID\_DIVNUM** 領域分割数が不正  
**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\_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  
**X\_PLUS** +X face  
**Y\_MINUS** -Y face  
**Y\_PLUS** +Y face  
**Z\_MINUS** -Z 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:

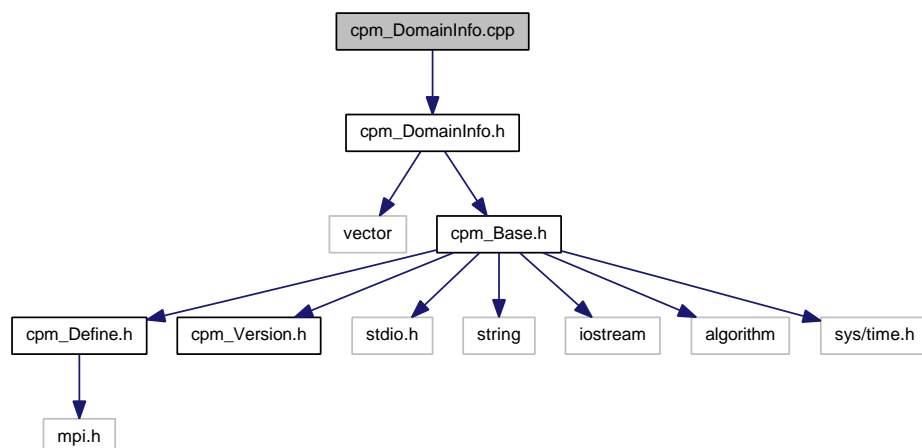
**PLUS2MINUS** plus -> minus direction  
**MINUS2PLUS** minus -> plus direction  
**BOTH** plus <-> minus direction

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

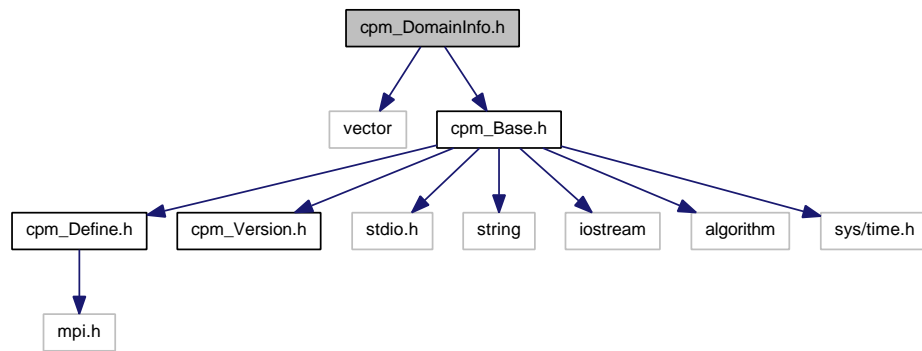
2012/05/31

[cpm\\_DomainInfo.cpp](#) で定義されています。

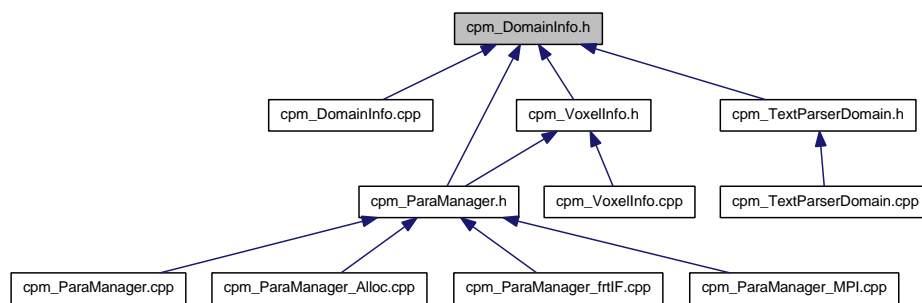
## 5.4 cpm\_DomainInfo.h

```
#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

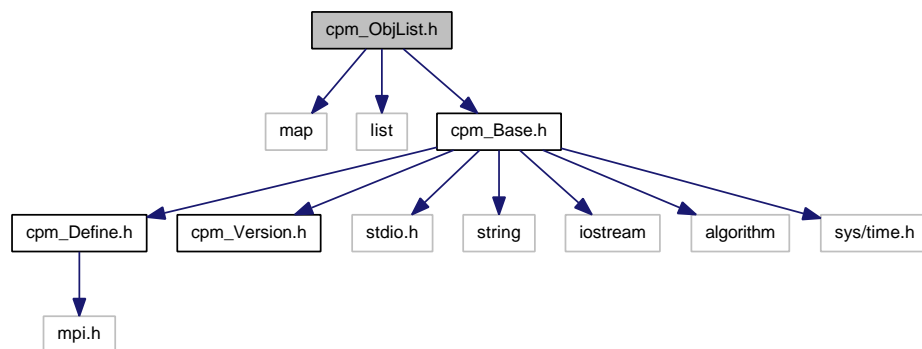
日付

2012/05/31

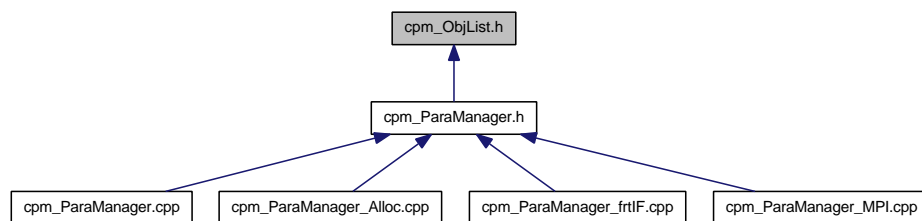
[cpm\\_DomainInfo.h](#) で定義されています。

## 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

作者

University of Tokyo

日付

2012/05/31

[cpm\\_ObjList.h](#) で定義されています。

## 5.5.2 型定義

### 5.5.2.1 typedef std::map<int, int\*> RankNoMap

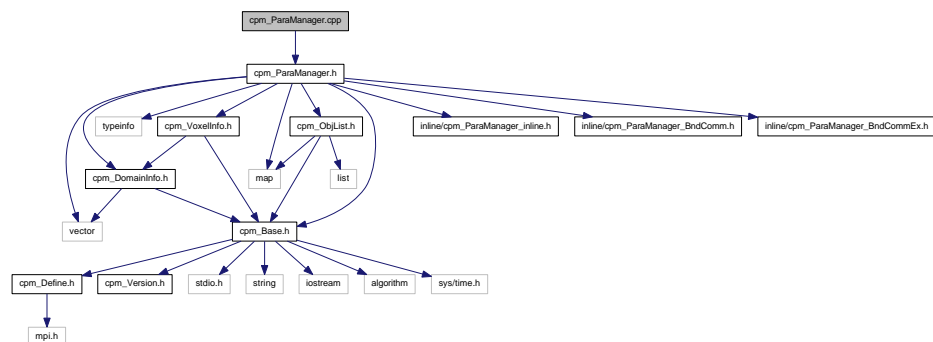
プロセスグループ毎のランク番号マップ

[cpm\\_ObjList.h](#) の 24 行で定義されています。

## 5.6 cpm\_ParaManager.cpp

```
#include "cpm_ParaManager.h"
```

[cpm\\_ParaManager.cpp](#) のインクルード依存関係図



### 5.6.1 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

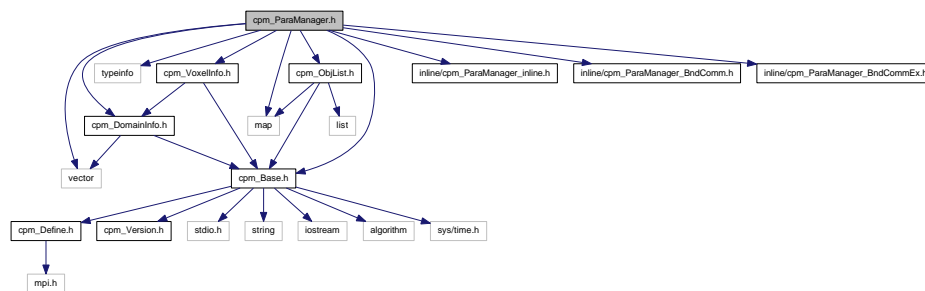
2012/05/31

[cpm\\_ParaManager.cpp](#) で定義されています。

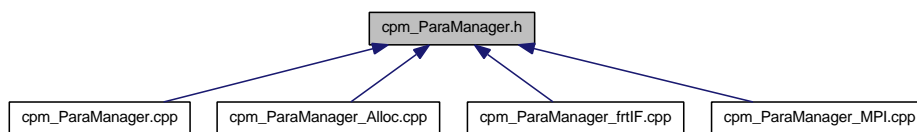
## 5.7 cpm\_ParaManager.h

```
#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。

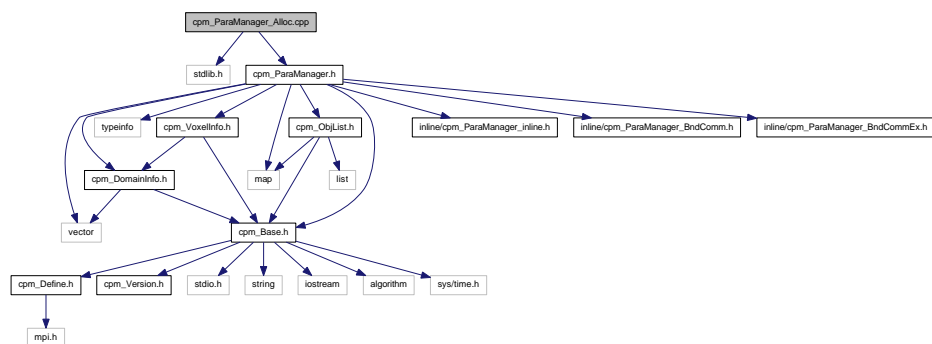
### 5.7.2.3 typedef std::map<int, cpm\_VoxelInfo\*> VoxelInfoMap

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

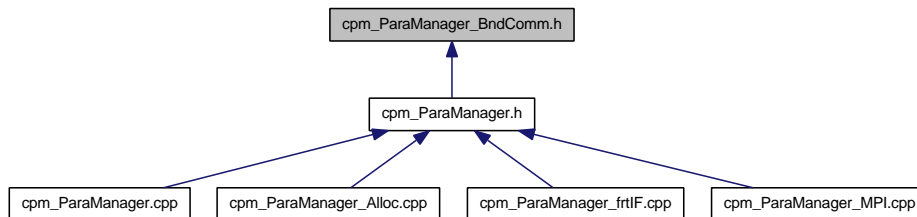
日付

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:

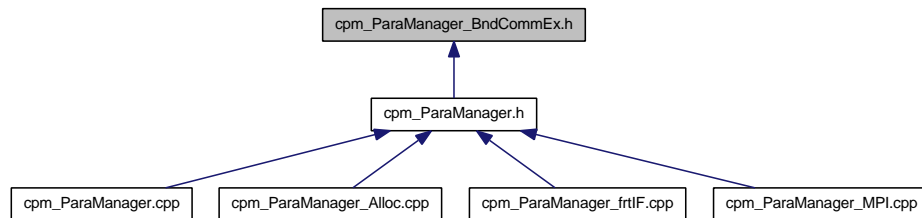
```
( size_t(_N)          * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) * size_t(_VC) \
+ size_t(_K-(_KS))    * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC)      * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

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。

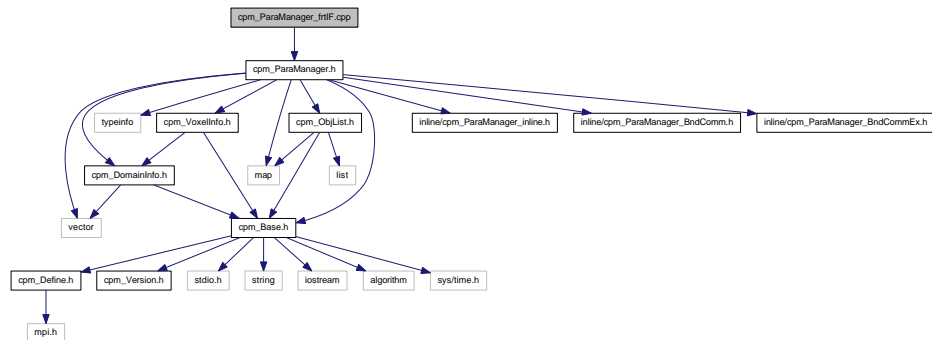
参照元 [cpm\\_ParaManager::packZEx\(\)](#), と [cpm\\_ParaManager::unpackZEx\(\)](#).



## 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\_GetVoxelHeadIndex\_ cpm\_getvoxelheadindex\_
- #define cpm\_GetVoxelTailIndex\_ cpm\_getvoxeltailindex\_
- #define cpm\_GetNeighborRankID\_ cpm\_getneighborrangid\_
- #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_BndCommS3D_ cpm_bndcomms3d_`
- `#define cpm_BndCommV3D_ cpm_bndcommv3d_`
- `#define cpm_BndCommS4D_ cpm_bndcomms4d_`
- `#define cpm_BndCommS3D_nowait_ cpm_bndcomms3d_nowait_`
- `#define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_`
- `#define cpm_BndCommS4D_nowait_ cpm_bndcomms4d_nowait_`
- `#define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_`
- `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`
- `#define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_`
- `#define cpm_BndCommV3DEx_ cpm_bndcommv3dex_`
- `#define cpm_BndCommS4DEx_ cpm_bndcomms4dex_`
- `#define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_`
- `#define cpm_BndCommS4DEx_nowait_ cpm_bndcomms4dex_nowait_`
- `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`
- `#define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_`
- `#define cpm_PeriodicCommsS3D_ cpm_periodiccomms3d_`
- `#define cpm_PeriodicCommV3D_ cpm_periodiccommv3d_`
- `#define cpm_PeriodicCommS4D_ 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 *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Voxellnit_nodiv_ (int *vox, REAL_TYPE *origin, REAL_TYPE *pitch, 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_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_lrecv_ (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_BndCommS4D_ (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_BndCommS3D_ (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_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)`
- `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)`
- `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_BndCommS4D_ (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_BndCommS3D_ (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_BndCommS4DEx_ (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_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)`
- `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_BndCommS4DEx_ (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_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)`
- `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)`
- `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_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)`
- `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 44 行で定義されています。

#### 5.11.2.2 `#define cpm_Allgather_ cpm_allgather_`

`cpm_ParaManager_frtIF.cpp` の 55 行で定義されています。

#### 5.11.2.3 `#define cpm_Allgatherv_ cpm_allgatherv_`

`cpm_ParaManager_frtIF.cpp` の 57 行で定義されています。

#### 5.11.2.4 `#define cpm_Allreduce_ cpm_allreduce_`

`cpm_ParaManager_frtIF.cpp` の 53 行で定義されています。

#### 5.11.2.5 `#define cpm_Barrier_ cpm_barrier_`

`cpm_ParaManager_frtIF.cpp` の 45 行で定義されています。

#### 5.11.2.6 `#define cpm_Bcast_ cpm_bcast_`

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

#### 5.11.2.7 `#define cpm_BndComms3D_ cpm_bndcomms3d_`

`cpm_ParaManager_frtIF.cpp` の 59 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 62 行で定義されています。

#### 5.11.2.9 #define cpm\_BndCommsS4D\_ cpm\_bndcomms4d\_

cpm\_ParaManager\_frtIF.cpp の 61 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 64 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 69 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 71 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 60 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 63 行で定義されています。

#### 5.11.2.15 #define cpm\_BndCommV3DEx\_ cpm\_bndcommv3dex\_

cpm\_ParaManager\_frtIF.cpp の 68 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 70 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 54 行で定義されています。

#### 5.11.2.19 #define cpm\_Gatherv\_ cpm\_gatherv\_

cpm\_ParaManager\_frtIF.cpp の 56 行で定義されています。

5.11.2.20 `#define cpm_GetDivNum_ cpm_getdivnum_`

`cpm_ParaManager_frtIF.cpp` の 29 行で定義されています。

5.11.2.21 `#define cpm_GetDivPos_ cpm_getdivpos_`

`cpm_ParaManager_frtIF.cpp` の 37 行で定義されています。

5.11.2.22 `#define cpm_GetGlobalOrigin_ cpm_getglobalorigin_`

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

5.11.2.23 `#define cpm_GetGlobalRegion_ cpm_getglobalregion_`

`cpm_ParaManager_frtIF.cpp` の 33 行で定義されています。

5.11.2.24 `#define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_`

`cpm_ParaManager_frtIF.cpp` の 31 行で定義されています。

5.11.2.25 `#define cpm_GetLocalOrigin_ cpm_getlocalorigin_`

`cpm_ParaManager_frtIF.cpp` の 35 行で定義されています。

5.11.2.26 `#define cpm_GetLocalRegion_ cpm_getlocalregion_`

`cpm_ParaManager_frtIF.cpp` の 36 行で定義されています。

5.11.2.27 `#define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_`

`cpm_ParaManager_frtIF.cpp` の 34 行で定義されています。

5.11.2.28 `#define cpm_GetMyRankID_ cpm_getmyrankid_`

`cpm_ParaManager_frtIF.cpp` の 42 行で定義されています。

5.11.2.29 `#define cpm_GetNeighborRankID_ cpm_getneighborrankid_`

`cpm_ParaManager_frtIF.cpp` の 40 行で定義されています。

5.11.2.30 `#define cpm_GetNumRank_ cpm_getnumrank_`

`cpm_ParaManager_frtIF.cpp` の 43 行で定義されています。

5.11.2.31 `#define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_`

`cpm_ParaManager_frtIF.cpp` の 41 行で定義されています。

5.11.2.32 `#define cpm_GetPitch_ cpm_getpitch_`

cpm\_ParaManager\_frtIF.cpp の 30 行で定義されています。

5.11.2.33 `#define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_`

cpm\_ParaManager\_frtIF.cpp の 38 行で定義されています。

5.11.2.34 `#define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_`

cpm\_ParaManager\_frtIF.cpp の 39 行で定義されています。

5.11.2.35 `#define cpm_Initialize_ cpm_initialize_`

cpm\_ParaManager\_frtIF.cpp の 25 行で定義されています。

5.11.2.36 `#define cpm_Irecv_ cpm_irecv_`

cpm\_ParaManager\_frtIF.cpp の 52 行で定義されています。

5.11.2.37 `#define cpm_Isend_ cpm_isend_`

cpm\_ParaManager\_frtIF.cpp の 51 行で定義されています。

5.11.2.38 `#define cpm_IsParallel_ cpm_isparallel_`

cpm\_ParaManager\_frtIF.cpp の 28 行で定義されています。

5.11.2.39 `#define cpm_PeriodicCommsS3D cpm_periodiccomms3d_`

cpm\_ParaManager\_frtIF.cpp の 74 行で定義されています。

5.11.2.40 `#define cpm_PeriodicCommsS4D cpm_periodiccomms4d_`

cpm\_ParaManager\_frtIF.cpp の 76 行で定義されています。

5.11.2.41 `#define cpm_PeriodicCommsS4DEx cpm_periodiccomms4dex_`

cpm\_ParaManager\_frtIF.cpp の 78 行で定義されています。

5.11.2.42 `#define cpm_PeriodicCommV3D cpm_periodiccommv3d_`

cpm\_ParaManager\_frtIF.cpp の 75 行で定義されています。

5.11.2.43 `#define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_`

cpm\_ParaManager\_frtIF.cpp の 77 行で定義されています。

#### 5.11.2.44 `#define cpm_Recv_ cpm_recv_`

`cpm_ParaManager_frtIF.cpp` の 50 行で定義されています。

#### 5.11.2.45 `#define cpm_Send_ cpm_send_`

`cpm_ParaManager_frtIF.cpp` の 49 行で定義されています。

#### 5.11.2.46 `#define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_`

`cpm_ParaManager_frtIF.cpp` の 58 行で定義されています。

#### 5.11.2.47 `#define cpm_Voxellnit_ cpm_voxellnit_`

`cpm_ParaManager_frtIF.cpp` の 26 行で定義されています。

#### 5.11.2.48 `#define cpm_Voxellnit_nodiv_ cpm_voxellnit_nodiv_`

`cpm_ParaManager_frtIF.cpp` の 27 行で定義されています。

#### 5.11.2.49 `#define cpm_Wait_ cpm_wait_`

`cpm_ParaManager_frtIF.cpp` の 46 行で定義されています。

#### 5.11.2.50 `#define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_`

`cpm_ParaManager_frtIF.cpp` の 65 行で定義されています。

#### 5.11.2.51 `#define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_`

`cpm_ParaManager_frtIF.cpp` の 67 行で定義されています。

参照元 `cpm_wait_BndCommS3D_()`, と `cpm_wait_BndCommV3D_()`.

#### 5.11.2.52 `#define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_`

`cpm_ParaManager_frtIF.cpp` の 73 行で定義されています。

参照元 `cpm_wait_BndCommV3DEx_()`.

#### 5.11.2.53 `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`

`cpm_ParaManager_frtIF.cpp` の 66 行で定義されています。

#### 5.11.2.54 `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`

`cpm_ParaManager_frtIF.cpp` の 72 行で定義されています。

#### 5.11.2.55 `#define cpm_Waitall_ cpm_waitall_`

`cpm_ParaManager_frtIF.cpp` の 47 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 912 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 1329 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 1431 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 1228 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 940 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 1029 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\_BndCommsS3D\_ ( 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
- BndCommsS3D 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 1559 行で定義されています。

参照先 cpm\_ParaManager::BndCommsS3D(), cpm\_BndCommsS4D\_, 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\_BndCommsS3D\_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
- BndCommsS3D\_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 1701 行で定義されています。

参照先 cpm\_ParaManager::cpm\_BndCommsS3D\_nowait(), cpm\_BndCommsS4D\_nowait\_, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, と cpm\_ParaManager::get\_instance().

**5.11.3.9 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 )**

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
- BndCommsS4D 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 1514 行で定義されています。

参照先 cpm\_ParaManager::BndCommsS4D(), 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\_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 )**

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
- BndCommsS4D\_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 1661 行で定義されています。

参照先 cpm\_ParaManager::cpm\_BndCommsS4D\_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 1924 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 2021 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 1609 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 1747 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 1969 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 2061 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 1279 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 1380 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\_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	<i>div</i>	領域分割数 (3word の整数配列)
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 284 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().



## 5.11.3.20 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 620 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().

## 5.11.3.21 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 410 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.

## 5.11.3.22 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	procGrpNo	プロセスグループ番号
out	wrgn	全体空間サイズ (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 452 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.

## 5.11.3.23 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 368 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().

#### 5.11.3.24 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 536 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.

#### 5.11.3.25 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 578 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.

#### 5.11.3.26 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 494 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().

#### 5.11.3.27 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 838 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().

#### 5.11.3.28 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 748 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().

#### 5.11.3.29 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 876 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().

#### 5.11.3.30 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 793 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().

#### 5.11.3.31 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 326 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.

#### 5.11.3.32 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 663 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().

#### 5.11.3.33 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 706 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().

#### 5.11.3.34 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 144 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().

#### 5.11.3.35 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>	<i>buf</i> 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 1189 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::cpm\_Irecv(), CPM\_SUCCESS, と cpm\_ParaManager::get\_instance().

#### 5.11.3.36 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>	<i>buf</i> 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 1150 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().

#### 5.11.3.37 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 250 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().

#### 5.11.3.38 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 2258 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().

**5.11.3.39 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 2195 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\_().

**5.11.3.40 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 2397 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().

instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::PeriodicComms4DEx().

参照元 cpm\_PeriodicCommV3DEx\_().

**5.11.3.41 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 2327 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().

**5.11.3.42 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 2460 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().

**5.11.3.43 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 1109 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().

**5.11.3.44 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 1069 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().

**5.11.3.45 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	<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
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1477 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().

5.11.3.46 **CPM\_EXTERN void cpm\_Voxellnit\_** ( int \* *div*, int \* *vox*, REAL\_TYPE \* *origin*, REAL\_TYPE \* *pitch*, 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	<i>div</i>	領域分割数 (サイズ 3)
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>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 180 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().

5.11.3.47 **CPM\_EXTERN void cpm\_Voxellnit\_nodiv\_** ( int \* *vox*, REAL\_TYPE \* *origin*, REAL\_TYPE \* *pitch*, 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>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 219 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\_Wait\_ ( int \* reqNo, int \* ierr )

Wait

- Wait のFortran インターフェイス関数

引数

in	reqNo	リクエスト番号 (0 以上の整数)
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 968 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().

## 5.11.3.49 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

	inout]	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 を参照)
out	reqlist	リクエスト番号のリスト (サイズ 12)
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 1834 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().

## 5.11.3.50 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

	inout]	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	<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 1794 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().

**5.11.3.51 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
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 2108 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().

**5.11.3.52 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 1879 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().

**5.11.3.53 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 2148 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().

**5.11.3.54 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

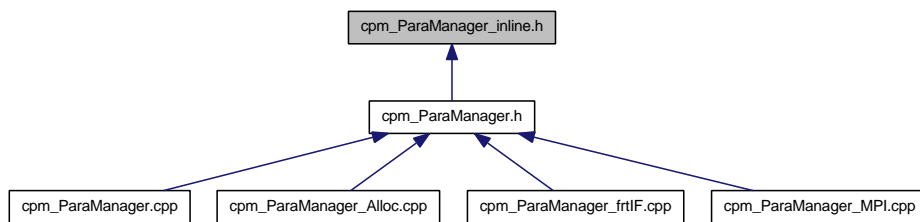
in	<i>count</i>	リクエストの数
in	<i>reqlist</i>	リクエスト番号のリスト (0 以上の整数)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 997 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

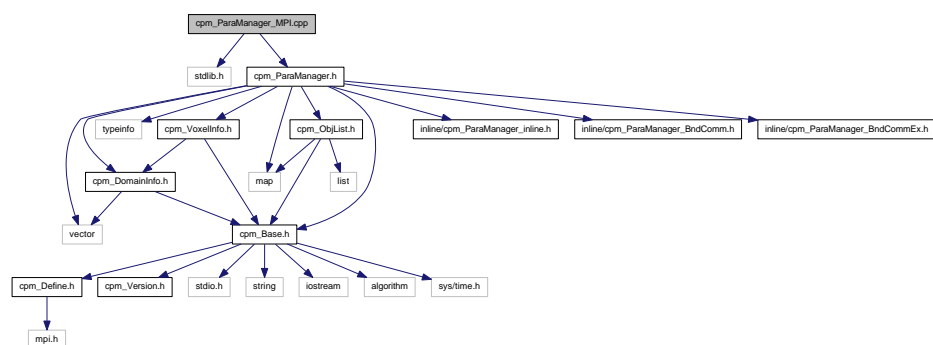
[cpm\\_ParaManager\\_inline.h](#) で定義されています。

## 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

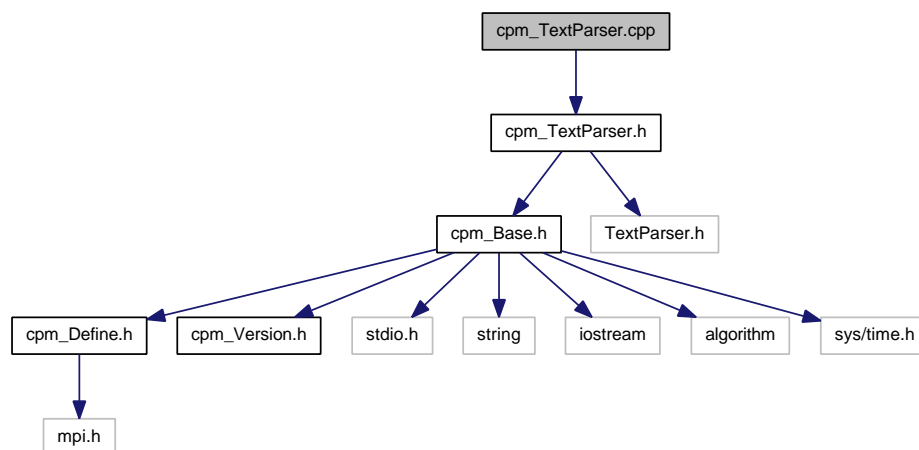
2012/05/31

[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

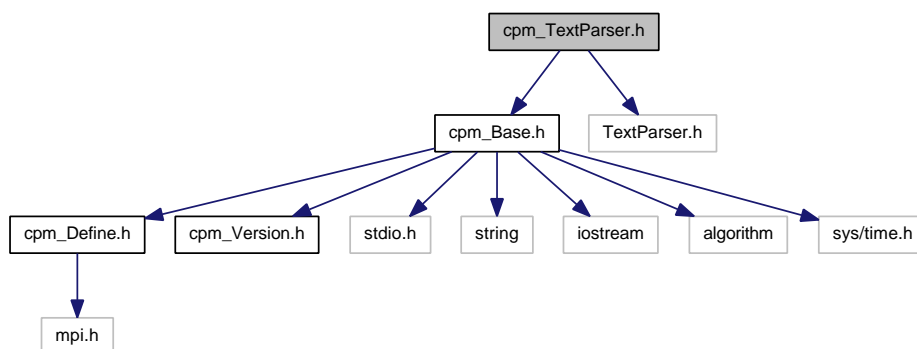
[cpm\\_TextParser.cpp](#) で定義されています。

## 5.15 cpm\_TextParser.h

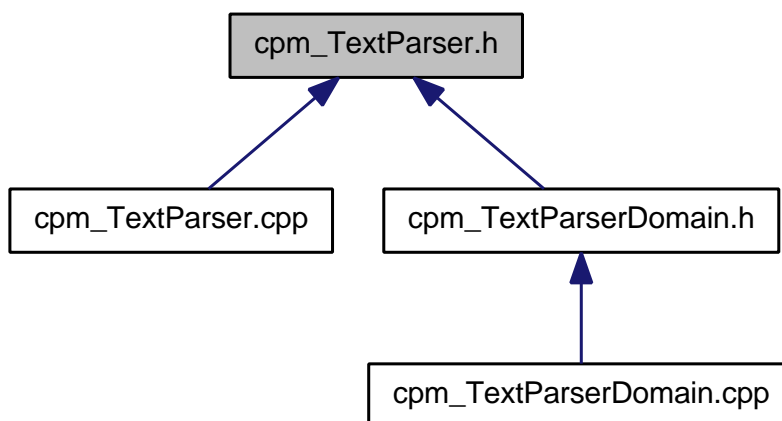
```
#include "cpm_Base.h"
```

```
#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

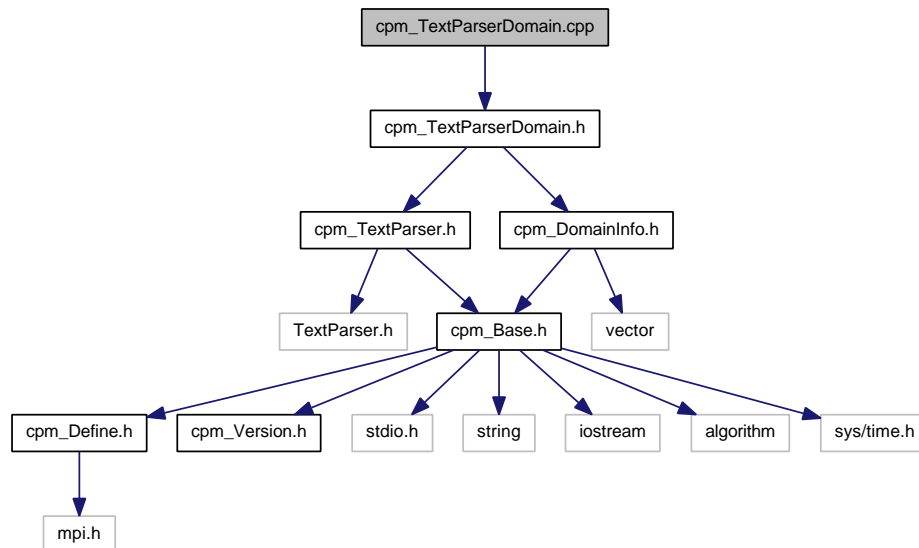
`cpm_TextParser.h` で定義されています。



## 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

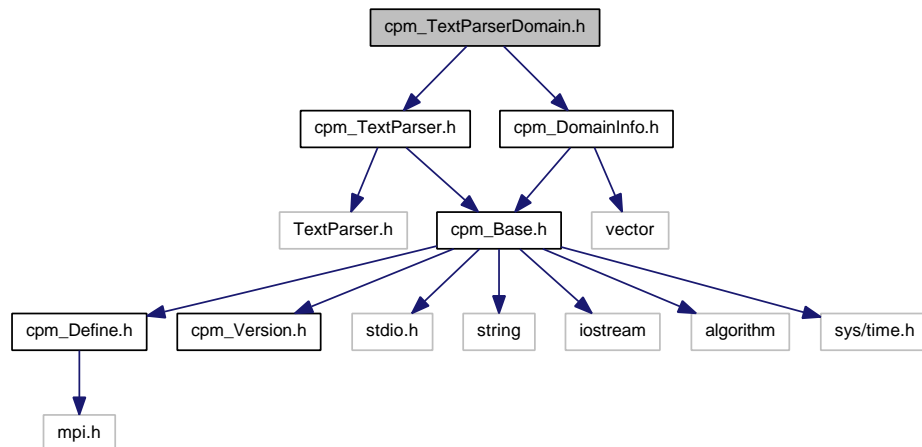
2012/05/31

[cpm\\_TextParserDomain.cpp](#) で定義されています。

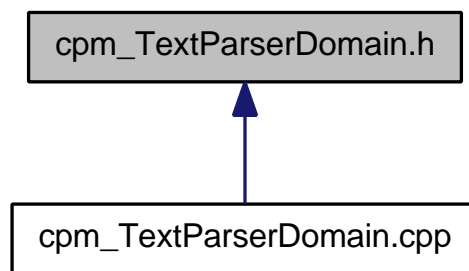
## 5.17 cpm\_TextParserDomain.h

```
#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

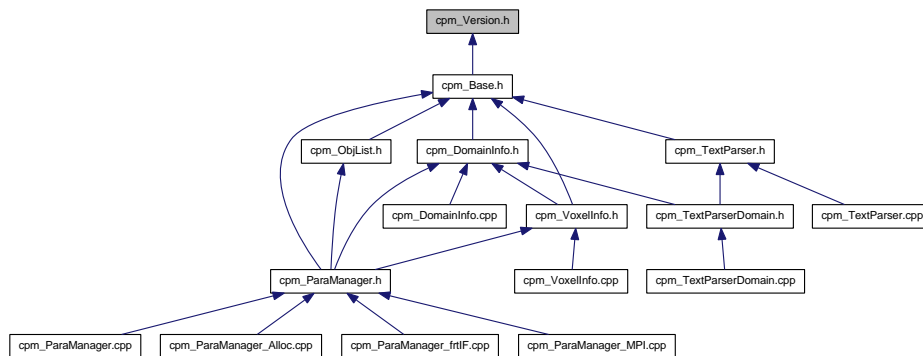
日付

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 "1.0.2"`
- `#define CPM_REVISION "20120627_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 "20120627_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 "1.0.2"`

CPM ライブラリのバージョン

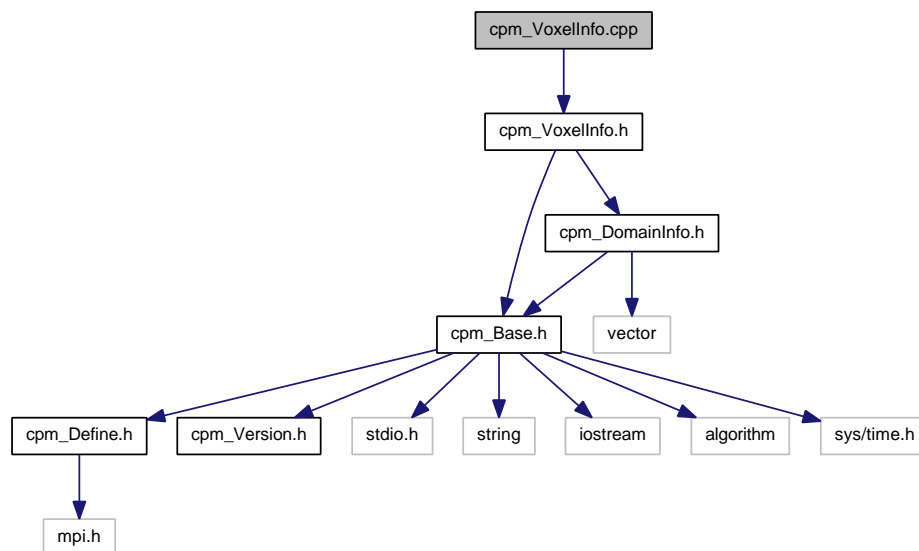
`cpm_Version.h` の 19 行で定義されています。

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

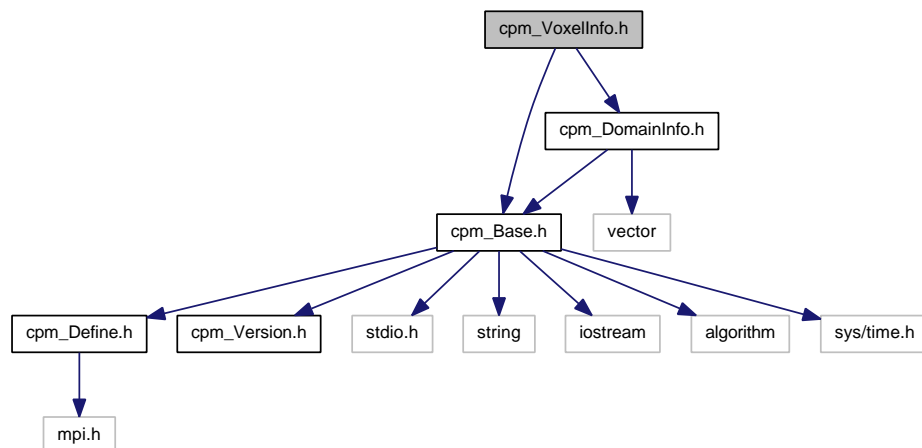
`cpm_VoxelInfo.cpp` で定義されています。

## 5.20 cpm\_VoxelInfo.h

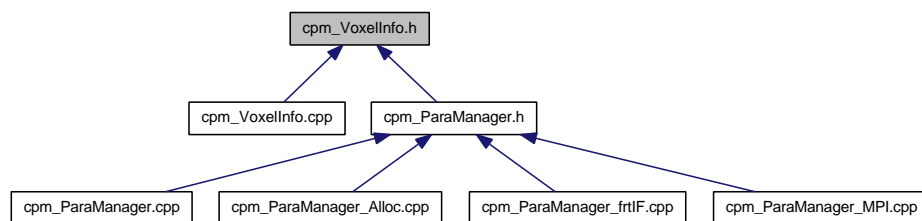
```
#include "cpm_Base.h"
```

```
#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, [116](#)
- ~cpm\_ActiveSubdomainInfo
  - cpm\_ActiveSubdomainInfo, [8](#)
- ~cpm\_Base
  - cpm\_Base, [11](#)
- ~cpm\_DomainInfo
  - cpm\_DomainInfo, [16](#)
- ~cpm\_GlobalDomainInfo
  - cpm\_GlobalDomainInfo, [21](#)
- ~cpm\_LocalDomainInfo
  - cpm\_LocalDomainInfo, [25](#)
- ~cpm\_ObjList
  - cpm\_ObjList, [27](#)
- ~cpm\_ParaManager
  - cpm\_ParaManager, [35](#)
- ~cpm\_TextParser
  - cpm\_TextParser, [102](#)
- ~cpm\_TextParserDomain
  - cpm\_TextParserDomain, [105](#)
- ~cpm\_VoxelInfo
  - cpm\_VoxelInfo, [109](#)
- \_IDXFX
  - cpm\_ParaManager\_BndComm.h, [134](#)
  - cpm\_ParaManager\_BndCommEx.h, [136](#)
- \_IDXFY
  - cpm\_ParaManager\_BndComm.h, [134](#)
  - cpm\_ParaManager\_BndCommEx.h, [136](#)
- \_IDXFZ
  - cpm\_ParaManager\_BndComm.h, [135](#)
  - cpm\_ParaManager\_BndCommEx.h, [136](#)
- \_IDX\_S3D
  - cpm\_Define.h, [122](#)
- \_IDX\_S4D
  - cpm\_Define.h, [122](#)
- \_IDX\_S4DEX
  - cpm\_Define.h, [123](#)
- \_IDX\_V3D
  - cpm\_Define.h, [123](#)
- \_IDX\_V3DEX
  - cpm\_Define.h, [124](#)
- Abort
  - cpm\_ParaManager, [36](#)
- Add
  - cpm\_ObjList, [27](#)
- AddSubdomain
  - cpm\_GlobalDomainInfo, [21](#)
- Allgather
  - cpm\_ParaManager, [36](#)
- Allgatherv
  - cpm\_ParaManager, [37](#)
- AllocDoubleS3D
  - cpm\_ParaManager, [38](#)
- AllocDoubleS4D
  - cpm\_ParaManager, [38](#)
- AllocDoubleS4DEX
  - cpm\_ParaManager, [38](#)
- AllocDoubleV3D
  - cpm\_ParaManager, [39](#)
- AllocDoubleV3DEX
  - cpm\_ParaManager, [39](#)
- AllocFloatS3D
  - cpm\_ParaManager, [39](#)
- AllocFloatS4D
  - cpm\_ParaManager, [40](#)
- AllocFloatS4DEX
  - cpm\_ParaManager, [40](#)
- AllocFloatV3D
  - cpm\_ParaManager, [40](#)
- 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, [42](#)
- AllocRealS4D
  - cpm\_ParaManager, [43](#)
- AllocRealS4DEX
  - cpm\_ParaManager, [43](#)
- AllocRealV3D
  - cpm\_ParaManager, [43](#)
- AllocRealV3DEX
  - cpm\_ParaManager, [44](#)
- Allreduce
  - cpm\_ParaManager, [44](#)
- BOTH
  - cpm\_Define.h, [128](#)
- Barrier
  - cpm\_ParaManager, [45](#)
- Bcast

- cpm\_ParaManager, [45, 46](#)
- BndCommInfoMap
  - cpm\_ParaManager.h, [133](#)
- BndCommS3D
  - cpm\_ParaManager, [46](#)
- BndCommS3D\_nowait
  - cpm\_ParaManager, [47](#)
- BndCommS4D
  - cpm\_ParaManager, [48, 49](#)
- BndCommS4D\_nowait
  - cpm\_ParaManager, [49, 50](#)
- BndCommS4DEx
  - cpm\_ParaManager, [50, 51](#)
- BndCommS4DEx\_nowait
  - cpm\_ParaManager, [51, 52](#)
- BndCommV3D
  - cpm\_ParaManager, [52, 53](#)
- BndCommV3D\_nowait
  - cpm\_ParaManager, [53, 54](#)
- BndCommV3DEx
  - cpm\_ParaManager, [54, 55](#)
- BndCommV3DEx\_nowait
  - cpm\_ParaManager, [55, 56](#)
- CPM\_BAND
  - cpm\_Define.h, [128](#)
- CPM\_BOR
  - cpm\_Define.h, [128](#)
- CPM\_BXOR
  - cpm\_Define.h, [128](#)
- CPM\_BYTE
  - cpm\_Define.h, [125](#)
- CPM\_CHAR
  - cpm\_Define.h, [125](#)
- CPM\_DOUBLE
  - cpm\_Define.h, [125](#)
- CPM\_Datatype
  - cpm\_Define.h, [125](#)
- CPM\_ERROR
  - cpm\_Define.h, [125](#)
- CPM\_ERROR\_ALREADY\_VOXELINIIT
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_BNDCOMM
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_BNDCOMM\_BUFFER
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_BNDCOMM\_VOXELSIZE
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_CREATE\_LOCALDOMAIN
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_CREATE\_NEIGHBOR
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_CREATE\_PROCGROUP
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_CREATE\_RANKMAP
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_DIVNUM
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_DIVPOS
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_GLOBALORIGIN
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_GLOBALREGION
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_GLOBALVOXELSIZE
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_HEADINDEX
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_INFO
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_LOCALORIGIN
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_LOCALREGION
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_LOCALVOXELSIZE
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_MYRANK
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_NEIGHBOR\_RANK
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_NUMRANK
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_PERIODIC\_RANK
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_PITCH
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_GET\_TAILINDEX
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INSERT\_VOXELMAP
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INVALID\_DIVNUM
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INVALID\_DOMAIN\_NO
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INVALID\_OBJKEY
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INVALID\_PTR
  - cpm\_Define.h, [125](#)
- CPM\_ERROR\_INVALID\_REGION
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_INVALID\_VOXELSIZE
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_MPI
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_MPI\_ALLGATHER
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_ALLGATHERV
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_ALLREDUCE
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_BARRIER
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_MPI\_BCAST

- cpm\_Define.h, [126](#)
- CPM\_ERROR\_MPI\_DIMSCREATE
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_GATHER
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_GATHERV
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_INVALID\_COMM
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_INVALID\_DATATYPE
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_INVALID\_OPERATOR
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_INVALID\_REQUEST
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_IRECV
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_ISEND
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_RECV
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_SEND
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_WAIT
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_MPI\_WAITALL
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_NO\_MPI\_INIT
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_NO\_TEXTPARSER
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_NOT\_IN\_PROCGROUP
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_PERIODIC
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_PERIODIC\_INVALID\_DIR
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_PERIODIC\_INVALID\_PM
  - cpm\_Define.h, [127](#)
- CPM\_ERROR\_PM\_INSTANCE
  - cpm\_Define.h, [125](#)
- CPM\_ERROR\_REGIST\_OBJKEY
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TEXTPARSER
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_G\_DIV
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_G\_ORG
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_G\_PITCH
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_G\_RGN
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_G\_VOXEL
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_INVALID\_POS
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_NOVECTOR
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_TP\_VECTOR\_SIZE
  - cpm\_Define.h, [126](#)
- CPM\_ERROR\_VOXELINIT
  - cpm\_Define.h, [126](#)
- CPM\_EXTERN
  - cpm\_ParaManager\_frtIF.cpp, [141](#)
- CPM\_FLOAT
  - cpm\_Define.h, [125](#)
- CPM\_INLINE
  - cpm\_Base.h, [120](#)
- CPM\_INT
  - cpm\_Define.h, [125](#)
- CPM\_LAND
  - cpm\_Define.h, [128](#)
- CPM\_LONG
  - cpm\_Define.h, [125](#)
- CPM\_LONG\_DOUBLE
  - cpm\_Define.h, [125](#)
- CPM\_LOR
  - cpm\_Define.h, [128](#)
- CPM\_LXOR
  - cpm\_Define.h, [128](#)
- CPM\_MAX
  - cpm\_Define.h, [127](#)
- CPM\_MAXLOC
  - cpm\_Define.h, [128](#)
- CPM\_MIN
  - cpm\_Define.h, [127](#)
- CPM\_MINLOC
  - cpm\_Define.h, [128](#)
- CPM\_Op
  - cpm\_Define.h, [127](#)
- CPM\_PROD
  - cpm\_Define.h, [128](#)
- CPM\_REAL
  - cpm\_Define.h, [125](#)
- CPM\_REVISION
  - cpm\_Version.h, [171](#)
- CPM\_SHORT
  - cpm\_Define.h, [125](#)
- CPM\_SUCCESS
  - cpm\_Define.h, [125](#)
- CPM\_SUM
  - cpm\_Define.h, [128](#)
- CPM\_UNSIGNED
  - cpm\_Define.h, [125](#)
- CPM\_UNSIGNED\_CHAR
  - cpm\_Define.h, [125](#)
- CPM\_UNSIGNED\_LONG
  - cpm\_Define.h, [125](#)
- CPM\_UNSIGNED\_SHORT
  - cpm\_Define.h, [125](#)
- CPM\_VERSION\_NO
  - cpm\_Version.h, [171](#)
- CalcCommSize
  - cpm\_ParaManager, [56](#)
- CheckData



- cpm\_DomainInfo, 16
- cpm\_GlobalDomainInfo, 21
- clear
  - cpm\_ActiveSubdomainInfo, 8
  - cpm\_DomainInfo, 17
  - cpm\_GlobalDomainInfo, 22
  - cpm\_LocalDomainInfo, 25
- CopyArray
  - cpm\_ParaManager, 57
- cpm\_Abort\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 145
- cpm\_ActiveSubdomainInfo, 7
  - ~cpm\_ActiveSubdomainInfo, 8
  - clear, 8
  - cpm\_ActiveSubdomainInfo, 8
  - cpm\_ActiveSubdomainInfo, 8
  - GetPos, 9
  - m\_pos, 10
  - operator==, 9
  - SetPos, 9
- cpm\_Allgather\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 145
- cpm\_Allgatherv\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 145
- cpm\_Allreduce\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 146
- cpm\_Barrier\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 146
- cpm\_Base, 10
  - ~cpm\_Base, 11
  - cpm\_Base, 11
  - cpm\_strCompare, 11
  - cpm\_strCompareN, 12
  - cpm\_Base, 11
  - getCommNull, 12
  - GetMemString, 12
  - getRankNull, 12
  - GetSpanTime, 13
  - GetTime, 13
  - GetWSpanTime, 13
  - GetWTime, 13
  - IsCommNull, 14
  - IsRankNull, 14
  - ReallIsDouble, 14
  - VersionInfo, 14, 15
- cpm\_Base.h, 119
  - CPM\_INLINE, 120
- cpm\_Bcast\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 146
- cpm\_BndCommS3D\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 147
- cpm\_BndCommS3D\_nowait
  - cpm\_ParaManager, 57
- cpm\_BndCommS3D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 147
- cpm\_BndCommS4D\_
  - cpm\_ParaManager\_frtIF.cpp, 140, 147
- cpm\_BndCommS4D\_nowait
  - cpm\_ParaManager, 58
- cpm\_BndCommS4D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 148
- cpm\_BndCommS4DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 148
- cpm\_BndCommS4DEx\_nowait
  - cpm\_ParaManager, 58
- cpm\_BndCommS4DEx\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 149
- cpm\_BndCommV3D\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 149
- cpm\_BndCommV3D\_nowait
  - cpm\_ParaManager, 59
- cpm\_BndCommV3D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 150
- cpm\_BndCommV3DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 150
- cpm\_BndCommV3DEx\_nowait
  - cpm\_ParaManager, 59
- cpm\_BndCommV3DEx\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 151
- cpm\_Define.h, 120
  - \_IDX\_S3D, 122
  - \_IDX\_S4D, 122
  - \_IDX\_S4DEX, 123
  - \_IDX\_V3D, 123
  - \_IDX\_V3DEX, 124
  - BOTH, 128
  - CPM\_BAND, 128
  - CPM BOR, 128
  - CPM\_BXOR, 128
  - CPM\_BYTE, 125
  - CPM\_CHAR, 125
  - CPM\_DOUBLE, 125
  - CPM\_Datatype, 125
  - CPM\_ERROR, 125
  - CPM\_ERROR\_ALREADY\_VOXELINIIT, 126
  - CPM\_ERROR\_BNDCOMM, 127
  - CPM\_ERROR\_BNDCOMM\_BUFFER, 127
  - CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, 127
  - CPM\_ERROR\_BNDCOMM\_VOXELSIZE, 127
  - CPM\_ERROR\_CREATE\_LOCALDOMAIN, 126
  - CPM\_ERROR\_CREATE\_NEIGHBOR, 126
  - CPM\_ERROR\_CREATE\_PROCGROUP, 126
  - CPM\_ERROR\_CREATE\_RANKMAP, 126
  - CPM\_ERROR\_GET\_DIVNUM, 126
  - CPM\_ERROR\_GET\_DIVPOS, 126
  - CPM\_ERROR\_GET\_GLOBALORIGIN, 126
  - CPM\_ERROR\_GET\_GLOBALREGION, 126
  - CPM\_ERROR\_GET\_GLOBALVOXELSIZE, 126
  - CPM\_ERROR\_GET\_HEADINDEX, 126
  - CPM\_ERROR\_GET\_INFO, 126
  - CPM\_ERROR\_GET\_LOCALORIGIN, 126
  - CPM\_ERROR\_GET\_LOCALREGION, 126
  - CPM\_ERROR\_GET\_LOCALVOXELSIZE, 126
  - CPM\_ERROR\_GET\_MYRANK, 126
  - CPM\_ERROR\_GET\_NEIGHBOR\_RANK, 126

- CPM\_ERROR\_GET\_NUMRANK, 126
- CPM\_ERROR\_GET\_PERIODIC\_RANK, 126
- CPM\_ERROR\_GET\_PITCH, 126
- CPM\_ERROR\_GET\_TAILINDEX, 126
- CPM\_ERROR\_INSERT\_VOXELMAP, 126
- CPM\_ERROR\_INVALID\_DIVNUM, 126
- CPM\_ERROR\_INVALID\_DOMAIN\_NO, 126
- CPM\_ERROR\_INVALID\_OBJKEY, 126
- CPM\_ERROR\_INVALID\_PTR, 125
- CPM\_ERROR\_INVALID\_REGION, 126
- CPM\_ERROR\_INVALID\_VOXELSIZE, 126
- CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN, 126
- CPM\_ERROR\_MPI, 126
- CPM\_ERROR\_MPI\_ALLGATHER, 127
- CPM\_ERROR\_MPI\_ALLGATHERV, 127
- CPM\_ERROR\_MPI\_ALLREDUCE, 127
- CPM\_ERROR\_MPI\_BARRIER, 126
- CPM\_ERROR\_MPI\_BCAST, 126
- CPM\_ERROR\_MPI\_DIMSCREATE, 127
- CPM\_ERROR\_MPI\_GATHER, 127
- CPM\_ERROR\_MPI\_GATHERV, 127
- CPM\_ERROR\_MPI\_INVALID\_COMM, 127
- CPM\_ERROR\_MPI\_INVALID\_DATATYPE, 127
- CPM\_ERROR\_MPI\_INVALID\_OPERATOR, 127
- CPM\_ERROR\_MPI\_INVALID\_REQUEST, 127
- CPM\_ERROR\_MPI\_Irecv, 127
- CPM\_ERROR\_MPI\_Isend, 127
- CPM\_ERROR\_MPI\_RECV, 127
- CPM\_ERROR\_MPI\_SEND, 127
- CPM\_ERROR\_MPI\_WAIT, 127
- CPM\_ERROR\_MPI\_WAITALL, 127
- CPM\_ERROR\_NO\_MPI\_INIT, 126
- CPM\_ERROR\_NO\_TEXTPARSER, 126
- CPM\_ERROR\_NOT\_IN\_PROCGROUP, 126
- CPM\_ERROR\_PERIODIC, 127
- CPM\_ERROR\_PERIODIC\_INVALID\_DIR, 127
- CPM\_ERROR\_PERIODIC\_INVALID\_PM, 127
- CPM\_ERROR\_PM\_INSTANCE, 125
- CPM\_ERROR\_REGIST\_OBJKEY, 126
- CPM\_ERROR\_TEXTPARSER, 126
- CPM\_ERROR\_TP\_INVALID\_G\_DIV, 126
- CPM\_ERROR\_TP\_INVALID\_G\_ORG, 126
- CPM\_ERROR\_TP\_INVALID\_G\_PITCH, 126
- CPM\_ERROR\_TP\_INVALID\_G\_RGN, 126
- CPM\_ERROR\_TP\_INVALID\_G\_VOXEL, 126
- CPM\_ERROR\_TP\_INVALID\_POS, 126
- CPM\_ERROR\_TP\_NOVECTOR, 126
- CPM\_ERROR\_TP\_VECTOR\_SIZE, 126
- CPM\_ERROR\_VOXELINIT, 126
- CPM\_FLOAT, 125
- CPM\_INT, 125
- CPM\_LAND, 128
- CPM\_LONG, 125
- CPM\_LONG\_DOUBLE, 125
- CPM\_LOR, 128
- CPM\_LXOR, 128
- CPM\_MAX, 127
- CPM\_MAXLOC, 128
- CPM\_MIN, 127
- CPM\_MINLOC, 128
- CPM\_Op, 127
- CPM\_PROD, 128
- CPM\_REAL, 125
- CPM\_SHORT, 125
- CPM\_SUCCESS, 125
- CPM\_SUM, 128
- CPM\_UNSIGNED, 125
- CPM\_UNSIGNED\_CHAR, 125
- CPM\_UNSIGNED\_LONG, 125
- CPM\_UNSIGNED\_SHORT, 125
- cpm\_DirFlag, 125
- cpm\_ErrorCode, 125
- cpm\_FaceFlag, 127
- cpm\_PMFlag, 128
- MINUS2PLUS, 128
- PLUS2MINUS, 128
- REAL\_BUF\_TYPE, 124
- REAL\_TYPE, 124
- X\_DIR, 125
- X\_MINUS, 127
- X\_PLUS, 127
- Y\_DIR, 125
- Y\_MINUS, 127
- Y\_PLUS, 127
- Z\_DIR, 125
- Z\_MINUS, 127
- Z\_PLUS, 127
- cpm\_DirFlag
  - cpm\_DirFlag, 125
- cpm\_DomainInfo, 15
  - ~cpm\_DomainInfo, 16
  - CheckData, 16
  - clear, 17
  - cpm\_DomainInfo, 16
  - cpm\_DomainInfo, 16
  - GetOrigin, 17
  - GetPitch, 17
  - GetRegion, 17
  - GetVoxNum, 17
  - m\_origin, 19
  - m\_pitch, 19
  - m\_region, 19
  - m\_voxNum, 19
  - SetOrigin, 18
  - SetPitch, 18
  - SetRegion, 18
  - SetVoxNum, 18
- cpm\_DomainInfo.cpp, 128
- cpm\_DomainInfo.h, 129
- cpm\_ErrorCode
  - cpm\_ErrorCode, 125
- cpm\_FaceFlag
  - cpm\_FaceFlag, 127
- cpm\_Gather\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 151

- cpm\_Gatherv\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 152
- cpm\_GetDivNum\_
  - cpm\_ParaManager\_frtIF.cpp, 141, 152
- cpm\_GetDivPos\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 152
- cpm\_GetGlobalOrigin\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 153
- cpm\_GetGlobalRegion\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 153
- cpm\_GetGlobalVoxelSize\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 153
- cpm\_GetLocalOrigin\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 154
- cpm\_GetLocalRegion\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 154
- cpm\_GetLocalVoxelSize\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 154
- cpm\_GetMyRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 155
- cpm\_GetNeighborRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 155
- cpm\_GetNumRank\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 155
- cpm\_GetPeriodicRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 155
- cpm\_GetPitch\_
  - cpm\_ParaManager\_frtIF.cpp, 142, 156
- cpm\_GetVoxelHeadIndex\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 156
- cpm\_GetVoxelTailIndex\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 156
- cpm\_GlobalDomainInfo, 19
  - ~cpm\_GlobalDomainInfo, 21
  - AddSubdomain, 21
  - CheckData, 21
  - clear, 22
  - cpm\_GlobalDomainInfo, 21
  - cpm\_GlobalDomainInfo, 21
  - GetDivNum, 22
  - GetSubdomainArraySize, 22
  - GetSubdomainInfo, 22
  - GetSubdomainNum, 23
  - IsExistSubdomain, 23
  - m\_divNum, 24
  - m\_subDomainInfo, 24
  - SetDivNum, 23
- cpm\_Initialize\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 157
- cpm\_Irecv
  - cpm\_ParaManager, 60
- cpm\_Irecv\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 157
- cpm\_IsParallel\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 158
- cpm\_Isend
  - cpm\_ParaManager, 60
- cpm\_Isend\_
  - cpm\_ParaManager\_frtIF.cpp, 143, 157
- cpm\_LocalDomainInfo, 24
  - ~cpm\_LocalDomainInfo, 25
  - clear, 25
  - cpm\_LocalDomainInfo, 25
  - cpm\_LocalDomainInfo, 25
- cpm\_ObjList
  - ~cpm\_ObjList, 27
  - Add, 27
  - cpm\_ObjList, 27
  - cpm\_ObjList, 27
  - Create, 27
  - DelKeyList, 27
  - Delete, 28
  - Get, 28
  - m\_DelKeyList, 28
  - m\_ObjectMap, 29
  - m\_newKey, 29
  - ObjectMap, 27
- cpm\_ObjList< T >, 26
- cpm\_ObjList.h, 130
  - RankNoMap, 131
- cpm\_PMFlag
  - cpm\_Define.h, 128
- cpm\_ParaManager, 29
  - ~cpm\_ParaManager, 35
  - Abort, 36
  - Allgather, 36
  - Allgatherv, 37
  - AllocDoubleS3D, 38
  - AllocDoubleS4D, 38
  - AllocDoubleS4DEx, 38
  - AllocDoubleV3D, 39
  - AllocDoubleV3DEx, 39
  - AllocFloatS3D, 39
  - AllocFloatS4D, 40
  - AllocFloatS4DEx, 40
  - AllocFloatV3D, 40
  - AllocFloatV3DEx, 41
  - AllocIntS3D, 41
  - AllocIntS4D, 41
  - AllocIntS4DEx, 42
  - AllocIntV3D, 42
  - AllocIntV3DEx, 42
  - AllocRealS3D, 42
  - AllocRealS4D, 43
  - AllocRealS4DEx, 43
  - AllocRealV3D, 43
  - AllocRealV3DEx, 44
  - Allreduce, 44
  - Barrier, 45
  - Bcast, 45, 46
  - BndCommS3D, 46
  - BndCommS3D\_nowait, 47
  - BndCommS4D, 48, 49
  - BndCommS4D\_nowait, 49, 50
  - BndCommS4DEx, 50, 51
  - BndCommS4DEx\_nowait, 51, 52

- BndCommV3D, [52, 53](#)
- BndCommV3D\_nowait, [53, 54](#)
- BndCommV3DEx, [54, 55](#)
- BndCommV3DEx\_nowait, [55, 56](#)
- CalcCommSize, [56](#)
- CopyArray, [57](#)
- cpm\_BndCommsS3D\_nowait, [57](#)
- cpm\_BndCommsS4D\_nowait, [58](#)
- cpm\_BndCommsS4DEx\_nowait, [58](#)
- cpm\_BndCommV3D\_nowait, [59](#)
- cpm\_BndCommV3DEx\_nowait, [59](#)
- cpm\_Irecv, [60](#)
- cpm\_Isend, [60](#)
- cpm\_ParaManager, [35](#)
- cpm\_Wait, [61](#)
- cpm\_Waitall, [64](#)
- cpm\_wait\_BndCommS3D, [61](#)
- cpm\_wait\_BndCommS4D, [62](#)
- cpm\_wait\_BndCommS4DEx, [62](#)
- cpm\_wait\_BndCommV3D, [63](#)
- cpm\_wait\_BndCommV3DEx, [63](#)
- cpm\_ParaManager, [35](#)
- cpm\_VoxelInfo, [113](#)
- CreateProcessGroup, [64](#)
- DecideDivPattern, [65](#)
- FindVoxelInfo, [65](#)
- flush, [65](#)
- Gather, [66](#)
- Gatherv, [66, 67](#)
- get\_instance, [67, 68](#)
- GetBndCommBuffer, [68](#)
- GetBndCommBufferSize, [68](#)
- GetDivNum, [69](#)
- GetDivPos, [69](#)
- GetGlobalOrigin, [69](#)
- GetGlobalRegion, [70](#)
- GetGlobalVoxelSize, [70](#)
- GetLocalOrigin, [70](#)
- GetLocalRegion, [71](#)
- GetLocalVoxelSize, [71](#)
- GetMPI\_Comm, [71](#)
- GetMPI\_Datatype, [72](#)
- GetMPI\_Op, [72](#)
- GetMyRankID, [73](#)
- GetNeighborRankID, [73](#)
- GetNumRank, [73](#)
- GetPeriodicRankID, [74](#)
- GetPitch, [74](#)
- GetVoxelHeadIndex, [74](#)
- GetVoxelTailIndex, [74](#)
- InitArray, [75](#)
- Initialize, [75](#)
- Irecv, [76](#)
- IsParallel, [77](#)
- Isend, [76, 77](#)
- m\_bndCommInfoMap, [99](#)
- m\_nRank, [99](#)
- m\_procGrpList, [99](#)
- m\_rankNo, [100](#)
- m\_rankNoMap, [100](#)
- m\_reqList, [100](#)
- m\_voxelInfoMap, [100](#)
- packX, [78](#)
- packXEx, [78](#)
- packY, [79](#)
- packYEx, [79](#)
- packZ, [80](#)
- packZEx, [80](#)
- PeriodicCommsS3D, [81](#)
- PeriodicCommsS4D, [82](#)
- PeriodicCommsS4DEx, [83, 84](#)
- PeriodicCommV3D, [84, 85](#)
- PeriodicCommV3DEx, [85, 86](#)
- Recv, [86, 87](#)
- Send, [87](#)
- sendrecv, [88](#)
- SetBndCommBuffer, [88](#)
- unpackX, [89](#)
- unpackXEx, [89](#)
- unpackY, [90](#)
- unpackYEx, [90](#)
- unpackZ, [91](#)
- unpackZEx, [91](#)
- VoxelInit, [92, 93](#)
- Wait, [93](#)
- wait\_BndCommsS3D, [94](#)
- wait\_BndCommsS4D, [95](#)
- wait\_BndCommsS4DEx, [96](#)
- wait\_BndCommV3D, [97](#)
- wait\_BndCommV3DEx, [98](#)
- Waitall, [99](#)
- cpm\_ParaManager.cpp, [131](#)
- cpm\_ParaManager.h, [132](#)
  - BndCommInfoMap, [133](#)
  - RankNoMap, [133](#)
  - VoxelInfoMap, [133](#)
- cpm\_ParaManager\_Alloc.cpp, [133](#)
- cpm\_ParaManager\_BndComm.h, [134](#)
  - \_IDXFX, [134](#)
  - \_IDXFY, [134](#)
  - \_IDXFZ, [135](#)
- cpm\_ParaManager\_BndCommEx.h, [135](#)
  - \_IDXFX, [136](#)
  - \_IDXFY, [136](#)
  - \_IDXFZ, [136](#)
- cpm\_ParaManager\_MPI.cpp, [166](#)
- cpm\_ParaManager\_frtIF.cpp, [137](#)
  - CPM\_EXTERN, [141](#)
  - cpm\_Abort\_, [140, 145](#)
  - cpm\_Allgather\_, [140, 145](#)
  - cpm\_Allgatherv\_, [140, 145](#)
  - cpm\_Allreduce\_, [140, 146](#)
  - cpm\_Barrier\_, [140, 146](#)
  - cpm\_Bcast\_, [140, 146](#)
  - cpm\_BndCommS3D\_, [140, 147](#)
  - cpm\_BndCommS3D\_nowait\_, [140, 147](#)

- cpm\_BndCommsS4D\_, 140, 147
- cpm\_BndCommsS4D\_nowait\_, 141, 148
- cpm\_BndCommsS4DEx\_, 141, 148
- cpm\_BndCommsS4DEx\_nowait\_, 141, 149
- cpm\_BndCommV3D\_, 141, 149
- cpm\_BndCommV3D\_nowait\_, 141, 150
- cpm\_BndCommV3DEx\_, 141, 150
- cpm\_BndCommV3DEx\_nowait\_, 141, 151
- cpm\_Gather\_, 141, 151
- cpm\_Gatherv\_, 141, 152
- cpm\_GetDivNum\_, 141, 152
- cpm\_GetDivPos\_, 142, 152
- cpm\_GetGlobalOrigin\_, 142, 153
- cpm\_GetGlobalRegion\_, 142, 153
- cpm\_GetGlobalVoxelSize\_, 142, 153
- cpm\_GetLocalOrigin\_, 142, 154
- cpm\_GetLocalRegion\_, 142, 154
- cpm\_GetLocalVoxelSize\_, 142, 154
- cpm\_GetMyRankID\_, 142, 155
- cpm\_GetNeighborRankID\_, 142, 155
- cpm\_GetNumRank\_, 142, 155
- cpm\_GetPeriodicRankID\_, 142, 155
- cpm\_GetPitch\_, 142, 156
- cpm\_GetVoxelHeadIndex\_, 143, 156
- cpm\_GetVoxelTailIndex\_, 143, 156
- cpm\_Initialize\_, 143, 157
- cpm\_Irecv\_, 143, 157
- cpm\_IsParallel\_, 143, 158
- cpm\_Isend\_, 143, 157
- cpm\_PeriodicCommsS3D\_, 143
- cpm\_PeriodicCommsS3D\_, 158
- cpm\_PeriodicCommsS4D\_, 143
- cpm\_PeriodicCommsS4D\_, 158
- cpm\_PeriodicCommsS4DEx\_, 143
- cpm\_PeriodicCommsS4DEx\_, 159
- cpm\_PeriodicCommV3D\_, 143
- cpm\_PeriodicCommV3D\_, 160
- cpm\_PeriodicCommV3DEx\_, 143
- cpm\_PeriodicCommV3DEx\_, 160
- cpm\_Recv\_, 143, 161
- cpm\_Send\_, 144, 161
- cpm\_SetBndCommBuffer\_, 144, 161
- cpm\_Voxellnit\_, 144, 162
- cpm\_Voxellnit\_nodiv\_, 144, 162
- cpm\_Wait\_, 144, 162
- cpm\_Waitall\_, 144, 165
- cpm\_wait\_BndCommsS3D\_, 144, 163
- cpm\_wait\_BndCommsS4D\_, 144, 163
- cpm\_wait\_BndCommsS4DEx\_, 144, 164
- cpm\_wait\_BndCommV3D\_, 144, 164
- cpm\_wait\_BndCommV3DEx\_, 144, 165
- cpm\_ParaManager\_inline.h, 166
- cpm\_PeriodicCommS3D
  - cpm\_ParaManager\_frtlF.cpp, 143
- cpm\_PeriodicCommS3D\_
  - cpm\_ParaManager\_frtlF.cpp, 158
- cpm\_PeriodicCommS4D
  - cpm\_ParaManager\_frtlF.cpp, 143
- cpm\_PeriodicCommS4D\_
  - cpm\_ParaManager\_frtlF.cpp, 158
- cpm\_PeriodicCommS4DEx\_
  - cpm\_ParaManager\_frtlF.cpp, 143
- cpm\_PeriodicCommS4DEx\_
  - cpm\_ParaManager\_frtlF.cpp, 159
- cpm\_PeriodicCommV3D
  - cpm\_ParaManager\_frtlF.cpp, 143
- cpm\_PeriodicCommV3D\_
  - cpm\_ParaManager\_frtlF.cpp, 160
- cpm\_PeriodicCommV3DEx
  - cpm\_ParaManager\_frtlF.cpp, 143
- cpm\_PeriodicCommV3DEx\_
  - cpm\_ParaManager\_frtlF.cpp, 160
- cpm\_Recv\_
  - cpm\_ParaManager\_frtlF.cpp, 143, 161
- cpm\_Send\_
  - cpm\_ParaManager\_frtlF.cpp, 144, 161
- cpm\_SetBndCommBuffer\_
  - cpm\_ParaManager\_frtlF.cpp, 144, 161
- cpm\_TextParser, 101
  - ~cpm\_TextParser, 102
  - cpm\_TextParser, 102
  - cpm\_TextParser, 102
  - m\_tp, 104
  - Read, 102
  - readVector, 102, 103
- cpm\_TextParser.cpp, 167
- cpm\_TextParser.h, 167
- cpm\_TextParserDomain, 104
  - ~cpm\_TextParserDomain, 105
  - cpm\_TextParserDomain, 105
  - cpm\_TextParserDomain, 105
  - Read, 105
  - ReadDomainInfo, 106
  - ReadMain, 106
  - ReadSubdomainInfo, 106
- cpm\_TextParserDomain.cpp, 169
- cpm\_TextParserDomain.h, 169
- cpm\_Version.h, 171
  - CPM\_REVISION, 171
  - CPM\_VERSION\_NO, 171
- cpm\_VoxellInfo, 107
  - ~cpm\_VoxellInfo, 109
  - cpm\_ParaManager, 113
  - cpm\_VoxellInfo, 109
  - cpm\_VoxellInfo, 109
  - CreateLocalDomainInfo, 109
  - CreateNeighborRankInfo, 109
  - CreateRankMap, 110
  - GetDivNum, 110
  - GetDivPos, 110
  - GetGlobalOrigin, 110
  - GetGlobalRegion, 110
  - GetGlobalVoxelSize, 111
  - GetLocalOrigin, 111
  - GetLocalRegion, 111
  - GetLocalVoxelSize, 111

- GetNeighborRankID, 112
- GetPeriodicRankID, 112
- GetPitch, 112
- GetVoxelHeadIndex, 112
- GetVoxelTailIndex, 113
- Init, 113
- m\_comm, 113
- m\_globalDomainInfo, 114
- m\_localDomainInfo, 114
- m\_nRank, 114
- m\_neighborRankID, 114
- m\_periodicRankID, 114
- m\_rankMap, 114
- m\_rankNo, 114
- m\_voxelHeadIndex, 115
- m\_voxelTailIndex, 115
- cpm\_VoxelInfo.cpp, 172
- cpm\_VoxelInfo.h, 172
- cpm\_VoxelInit\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 162
- cpm\_VoxelInit\_nodiv\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 162
- cpm\_Wait
  - cpm\_ParaManager, 61
- cpm\_Wait\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 162
- cpm\_Waitall
  - cpm\_ParaManager, 64
- cpm\_Waitall\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 165
- cpm\_strCompare
  - cpm\_Base, 11
- cpm\_strCompareN
  - cpm\_Base, 12
- cpm\_wait\_BndCommsS3D
  - cpm\_ParaManager, 61
- cpm\_wait\_BndCommsS3D\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 163
- cpm\_wait\_BndCommsS4D
  - cpm\_ParaManager, 62
- cpm\_wait\_BndCommsS4D\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 163
- cpm\_wait\_BndCommsS4DEx
  - cpm\_ParaManager, 62
- cpm\_wait\_BndCommsS4DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 164
- cpm\_wait\_BndCommV3D
  - cpm\_ParaManager, 63
- cpm\_wait\_BndCommV3D\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 164
- cpm\_wait\_BndCommV3DEx
  - cpm\_ParaManager, 63
- cpm\_wait\_BndCommV3DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 144, 165
- Create
  - cpm\_ObjList, 27
- CreateLocalDomainInfo
  - cpm\_VoxelInfo, 109
- CreateNeighborRankInfo
  - cpm\_VoxelInfo, 109
- CreateProcessGroup
  - cpm\_ParaManager, 64
- CreateRankMap
  - cpm\_VoxelInfo, 110
- DecideDivPattern
  - cpm\_ParaManager, 65
- DelKeyList
  - cpm\_ObjList, 27
- Delete
  - cpm\_ObjList, 28
- FindVoxelInfo
  - cpm\_ParaManager, 65
- flush
  - cpm\_ParaManager, 65
- Gather
  - cpm\_ParaManager, 66
- Gatherv
  - cpm\_ParaManager, 66, 67
- Get
  - cpm\_ObjList, 28
- get\_instance
  - cpm\_ParaManager, 67, 68
- GetBndCommBuffer
  - cpm\_ParaManager, 68
- GetBndCommBufferSize
  - cpm\_ParaManager, 68
- getCommNull
  - cpm\_Base, 12
- GetDivNum
  - cpm\_GlobalDomainInfo, 22
  - cpm\_ParaManager, 69
  - cpm\_VoxelInfo, 110
- GetDivPos
  - cpm\_ParaManager, 69
  - cpm\_VoxelInfo, 110
- GetGlobalOrigin
  - cpm\_ParaManager, 69
  - cpm\_VoxelInfo, 110
- GetGlobalRegion
  - cpm\_ParaManager, 70
  - cpm\_VoxelInfo, 110
- GetGlobalVoxelSize
  - cpm\_ParaManager, 70
  - cpm\_VoxelInfo, 111
- GetLocalOrigin
  - cpm\_ParaManager, 70
  - cpm\_VoxelInfo, 111
- GetLocalRegion
  - cpm\_ParaManager, 71
  - cpm\_VoxelInfo, 111
- GetLocalVoxelSize
  - cpm\_ParaManager, 71
  - cpm\_VoxelInfo, 111
- GetMPI\_Comm

- cpm\_ParaManager, 71
- GetMPI\_Datatype
  - cpm\_ParaManager, 72
- GetMPI\_Op
  - cpm\_ParaManager, 72
- GetMemString
  - cpm\_Base, 12
- GetMyRankID
  - cpm\_ParaManager, 73
- GetNeighborRankID
  - cpm\_ParaManager, 73
  - cpm\_VoxelInfo, 112
- GetNumRank
  - cpm\_ParaManager, 73
- GetOrigin
  - cpm\_DomainInfo, 17
- GetPeriodicRankID
  - cpm\_ParaManager, 74
  - cpm\_VoxelInfo, 112
- GetPitch
  - cpm\_DomainInfo, 17
  - cpm\_ParaManager, 74
  - cpm\_VoxelInfo, 112
- GetPos
  - cpm\_ActiveSubdomainInfo, 9
- getRankNull
  - cpm\_Base, 12
- GetRegion
  - cpm\_DomainInfo, 17
- GetSpanTime
  - cpm\_Base, 13
- GetSubdomainArraySize
  - cpm\_GlobalDomainInfo, 22
- GetSubdomainInfo
  - cpm\_GlobalDomainInfo, 22
- GetSubdomainNum
  - cpm\_GlobalDomainInfo, 23
- GetTime
  - cpm\_Base, 13
- GetVoxNum
  - cpm\_DomainInfo, 17
- GetVoxelHeadIndex
  - cpm\_ParaManager, 74
  - cpm\_VoxelInfo, 112
- GetVoxelTailIndex
  - cpm\_ParaManager, 74
  - cpm\_VoxelInfo, 113
- GetWSpanTime
  - cpm\_Base, 13
- GetWTime
  - cpm\_Base, 13
- Init
  - cpm\_VoxelInfo, 113
- InitArray
  - cpm\_ParaManager, 75
- Initialize
  - cpm\_ParaManager, 75
- Irecv
  - cpm\_ParaManager, 76
- IsCommNull
  - cpm\_Base, 14
- IsExistSubdomain
  - cpm\_GlobalDomainInfo, 23
- IsParallel
  - cpm\_ParaManager, 77
- IsRankNull
  - cpm\_Base, 14
- Isend
  - cpm\_ParaManager, 76, 77
- m\_DelKeyList
  - cpm\_ObjList, 28
- m\_ObjectMap
  - cpm\_ObjList, 29
- m\_bndCommInfoMap
  - cpm\_ParaManager, 99
- m\_bufX
  - S\_BNDCOMM\_BUFFER, 116
- m\_bufY
  - S\_BNDCOMM\_BUFFER, 116
- m\_bufZ
  - S\_BNDCOMM\_BUFFER, 116
- m\_comm
  - cpm\_VoxelInfo, 113
- m\_divNum
  - cpm\_GlobalDomainInfo, 24
- m\_globalDomainInfo
  - cpm\_VoxelInfo, 114
- m\_localDomainInfo
  - cpm\_VoxelInfo, 114
- m\_maxN
  - S\_BNDCOMM\_BUFFER, 116
- m\_maxVC
  - S\_BNDCOMM\_BUFFER, 117
- m\_nRank
  - cpm\_ParaManager, 99
  - cpm\_VoxelInfo, 114
- m\_neighborRankID
  - cpm\_VoxelInfo, 114
- m\_newKey
  - cpm\_ObjList, 29
- m\_nwX
  - S\_BNDCOMM\_BUFFER, 117
- m\_nwY
  - S\_BNDCOMM\_BUFFER, 117
- m\_nwZ
  - S\_BNDCOMM\_BUFFER, 117
- m\_origin
  - cpm\_DomainInfo, 19
- m\_periodicRankID
  - cpm\_VoxelInfo, 114
- m\_pitch
  - cpm\_DomainInfo, 19
- m\_pos
  - cpm\_ActiveSubdomainInfo, 10
- m\_procGrpList
  - cpm\_ParaManager, 99



- m\_rankMap
  - cpm\_VoxelInfo, 114
- m\_rankNo
  - cpm\_ParaManager, 100
  - cpm\_VoxelInfo, 114
- m\_rankNoMap
  - cpm\_ParaManager, 100
- m\_region
  - cpm\_DomainInfo, 19
- m\_reqList
  - cpm\_ParaManager, 100
- m\_subDomainInfo
  - cpm\_GlobalDomainInfo, 24
- m\_tp
  - cpm\_TextParser, 104
- m\_voxNum
  - cpm\_DomainInfo, 19
- m\_voxelHeadIndex
  - cpm\_VoxelInfo, 115
- m\_voxelInfoMap
  - cpm\_ParaManager, 100
- m\_voxelTailIndex
  - cpm\_VoxelInfo, 115
- MINUS2PLUS
  - cpm\_Define.h, 128
- ObjectMap
  - cpm\_ObjList, 27
- operator==
  - cpm\_ActiveSubdomainInfo, 9
- PLUS2MINUS
  - cpm\_Define.h, 128
- packX
  - cpm\_ParaManager, 78
- packXEx
  - cpm\_ParaManager, 78
- packY
  - cpm\_ParaManager, 79
- packYEx
  - cpm\_ParaManager, 79
- packZ
  - cpm\_ParaManager, 80
- packZEx
  - cpm\_ParaManager, 80
- PeriodicCommS3D
  - cpm\_ParaManager, 81
- PeriodicCommS4D
  - cpm\_ParaManager, 82
- PeriodicCommS4DEx
  - cpm\_ParaManager, 83, 84
- PeriodicCommV3D
  - cpm\_ParaManager, 84, 85
- PeriodicCommV3DEx
  - cpm\_ParaManager, 85, 86
- REAL\_BUF\_TYPE
  - cpm\_Define.h, 124
- REAL\_TYPE
  - cpm\_Define.h, 124
- RankNoMap
  - cpm\_ObjList.h, 131
  - cpm\_ParaManager.h, 133
- Read
  - cpm\_TextParser, 102
  - cpm\_TextParserDomain, 105
- ReadDomainInfo
  - cpm\_TextParserDomain, 106
- ReadMain
  - cpm\_TextParserDomain, 106
- ReadSubdomainInfo
  - cpm\_TextParserDomain, 106
- readVector
  - cpm\_TextParser, 102, 103
- ReallsDouble
  - cpm\_Base, 14
- Recv
  - cpm\_ParaManager, 86, 87
- S\_BNDCOMM\_BUFFER, 115
  - ~S\_BNDCOMM\_BUFFER, 116
  - m\_bufX, 116
  - m\_bufY, 116
  - m\_bufZ, 116
  - m\_maxN, 116
  - m\_maxVC, 117
  - m\_nwX, 117
  - m\_nwY, 117
  - m\_nwZ, 117
  - S\_BNDCOMM\_BUFFER, 116
  - S\_BNDCOMM\_BUFFER, 116
- Send
  - cpm\_ParaManager, 87
- sendrecv
  - cpm\_ParaManager, 88
- SetBndCommBuffer
  - cpm\_ParaManager, 88
- SetDivNum
  - cpm\_GlobalDomainInfo, 23
- SetOrigin
  - cpm\_DomainInfo, 18
- SetPitch
  - cpm\_DomainInfo, 18
- SetPos
  - cpm\_ActiveSubdomainInfo, 9
- SetRegion
  - cpm\_DomainInfo, 18
- SetVoxNum
  - cpm\_DomainInfo, 18
- unpackX
  - cpm\_ParaManager, 89
- unpackXEx
  - cpm\_ParaManager, 89
- unpackY
  - cpm\_ParaManager, 90
- unpackYEx
  - cpm\_ParaManager, 90



- unpackZ
  - cpm\_ParaManager, [91](#)
- unpackZEx
  - cpm\_ParaManager, [91](#)
- VersionInfo
  - cpm\_Base, [14](#), [15](#)
- VoxelInfoMap
  - cpm\_ParaManager.h, [133](#)
- VoxelInit
  - cpm\_ParaManager, [92](#), [93](#)
- Wait
  - cpm\_ParaManager, [93](#)
- wait\_BndCommS3D
  - cpm\_ParaManager, [94](#)
- wait\_BndCommS4D
  - cpm\_ParaManager, [95](#)
- wait\_BndCommS4DEx
  - cpm\_ParaManager, [96](#)
- wait\_BndCommV3D
  - cpm\_ParaManager, [97](#)
- wait\_BndCommV3DEx
  - cpm\_ParaManager, [98](#)
- Waitall
  - cpm\_ParaManager, [99](#)
- X\_DIR
  - cpm\_Define.h, [125](#)
- X\_MINUS
  - cpm\_Define.h, [127](#)
- X\_PLUS
  - cpm\_Define.h, [127](#)
- Y\_DIR
  - cpm\_Define.h, [125](#)
- Y\_MINUS
  - cpm\_Define.h, [127](#)
- Y\_PLUS
  - cpm\_Define.h, [127](#)
- Z\_DIR
  - cpm\_Define.h, [125](#)
- Z\_MINUS
  - cpm\_Define.h, [127](#)
- Z\_PLUS
  - cpm\_Define.h, [127](#)