

# Cartesian Partition Manager

## 0.1

作成 : Doxygen 1.8.0

Fri Jun 1 2012 13:33:39



# 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>GetBCID</code>	9
4.1.3.3	<code>GetPos</code>	9
4.1.3.4	<code>operator!=</code>	9
4.1.3.5	<code>operator==</code>	9
4.1.3.6	<code>Set</code>	9
4.1.3.7	<code>SetBCID</code>	10
4.1.3.8	<code>SetPos</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	12
4.2.2.1	<code>cpm_Base</code>	12
4.2.2.2	<code>~cpm_Base</code>	12
4.2.3	関数	12
4.2.3.1	<code>cpm_strCompare</code>	12

4.2.3.2	cpm_strCompareN	12
4.2.3.3	getCommNull	13
4.2.3.4	GetMemString	13
4.2.3.5	getRankNull	13
4.2.3.6	GetSpanTime	13
4.2.3.7	GetTime	14
4.2.3.8	GetWSpanTime	14
4.2.3.9	GetWTime	14
4.2.3.10	IsCommNull	14
4.2.3.11	IsRankNull	14
4.2.3.12	ReallIsDouble	15
4.2.3.13	VersionInfo	15
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	17
4.3.2.2	~cpm_DomainInfo	17
4.3.3	関数	17
4.3.3.1	clear	17
4.3.3.2	GetOrigin	17
4.3.3.3	GetPitch	17
4.3.3.4	GetRegion	17
4.3.3.5	GetVoxNum	18
4.3.3.6	SetOrigin	18
4.3.3.7	SetPitch	18
4.3.3.8	SetRegion	18
4.3.3.9	SetVoxNum	18
4.4	クラス cpm_GlobalDomainInfo	19
4.4.1	説明	20
4.4.2	コンストラクタとデストラクタ	20
4.4.2.1	cpm_GlobalDomainInfo	20
4.4.2.2	~cpm_GlobalDomainInfo	20
4.4.3	関数	20
4.4.3.1	AddSubDomain	20
4.4.3.2	clear	20
4.4.3.3	GetDivNum	21
4.4.3.4	GetSubDomainInfo	21
4.4.3.5	GetSubDomainNum	21
4.4.3.6	IsExistSubDomain	21

4.4.3.7	SetDivNum	22
4.5	クラス cpm_LocalDomainInfo	22
4.5.1	説明	23
4.5.2	コンストラクタとデストラクタ	23
4.5.2.1	cpm_LocalDomainInfo	23
4.5.2.2	~cpm_LocalDomainInfo	23
4.5.3	関数	23
4.5.3.1	clear	23
4.6	クラス テンプレート cpm_ObjList< T >	23
4.6.1	説明	24
4.6.2	コンストラクタとデストラクタ	24
4.6.2.1	cpm_ObjList	24
4.6.2.2	~cpm_ObjList	25
4.6.3	関数	25
4.6.3.1	Add	25
4.6.3.2	Create	25
4.6.3.3	Delete	25
4.6.3.4	Get	26
4.7	クラス cpm_ParaManager	26
4.7.1	説明	31
4.7.2	関数	31
4.7.2.1	Abort	31
4.7.2.2	Allgather	31
4.7.2.3	Allgather	32
4.7.2.4	Allgatherv	32
4.7.2.5	Allgatherv	33
4.7.2.6	AllocDoubleS3D	33
4.7.2.7	AllocDoubleS4D	33
4.7.2.8	AllocDoubleS4DEx	34
4.7.2.9	AllocDoubleV3D	34
4.7.2.10	AllocDoubleV3DEx	34
4.7.2.11	AllocFloatS3D	35
4.7.2.12	AllocFloatS4D	35
4.7.2.13	AllocFloatS4DEx	35
4.7.2.14	AllocFloatV3D	36
4.7.2.15	AllocFloatV3DEx	36
4.7.2.16	AllocIntS3D	36
4.7.2.17	AllocIntS4D	37
4.7.2.18	AllocIntS4DEx	37
4.7.2.19	AllocIntV3D	37

4.7.2.20	<a href="#">AllocIntV3DEx</a>	37
4.7.2.21	<a href="#">AllocRealS3D</a>	38
4.7.2.22	<a href="#">AllocRealS4D</a>	38
4.7.2.23	<a href="#">AllocRealS4DEx</a>	38
4.7.2.24	<a href="#">AllocRealV3D</a>	39
4.7.2.25	<a href="#">AllocRealV3DEx</a>	39
4.7.2.26	<a href="#">Allreduce</a>	39
4.7.2.27	<a href="#">Allreduce</a>	40
4.7.2.28	<a href="#">Barrier</a>	40
4.7.2.29	<a href="#">Bcast</a>	41
4.7.2.30	<a href="#">Bcast</a>	41
4.7.2.31	<a href="#">BndCommS3D</a>	41
4.7.2.32	<a href="#">BndCommS3D</a>	42
4.7.2.33	<a href="#">BndCommS3D_nowait</a>	42
4.7.2.34	<a href="#">BndCommS3D_nowait</a>	43
4.7.2.35	<a href="#">BndCommS4D</a>	43
4.7.2.36	<a href="#">BndCommS4D</a>	44
4.7.2.37	<a href="#">BndCommS4D_nowait</a>	44
4.7.2.38	<a href="#">BndCommS4D_nowait</a>	45
4.7.2.39	<a href="#">BndCommS4DEx</a>	46
4.7.2.40	<a href="#">BndCommS4DEx</a>	46
4.7.2.41	<a href="#">BndCommS4DEx_nowait</a>	47
4.7.2.42	<a href="#">BndCommS4DEx_nowait</a>	47
4.7.2.43	<a href="#">BndCommV3D</a>	48
4.7.2.44	<a href="#">BndCommV3D</a>	48
4.7.2.45	<a href="#">BndCommV3D_nowait</a>	49
4.7.2.46	<a href="#">BndCommV3D_nowait</a>	49
4.7.2.47	<a href="#">BndCommV3DEx</a>	50
4.7.2.48	<a href="#">BndCommV3DEx</a>	50
4.7.2.49	<a href="#">BndCommV3DEx_nowait</a>	51
4.7.2.50	<a href="#">BndCommV3DEx_nowait</a>	51
4.7.2.51	<a href="#">CopyArray</a>	52
4.7.2.52	<a href="#">cpm_BndCommS3D_nowait</a>	52
4.7.2.53	<a href="#">cpm_BndCommS4D_nowait</a>	53
4.7.2.54	<a href="#">cpm_BndCommS4DEx_nowait</a>	53
4.7.2.55	<a href="#">cpm_BndCommV3D_nowait</a>	54
4.7.2.56	<a href="#">cpm_BndCommV3DEx_nowait</a>	54
4.7.2.57	<a href="#">cpm_Irecv</a>	55
4.7.2.58	<a href="#">cpm_Isend</a>	55
4.7.2.59	<a href="#">cpm_Wait</a>	56

4.7.2.60	<a href="#">cpm_wait_BndCommsS3D</a>	56
4.7.2.61	<a href="#">cpm_wait_BndCommsS4D</a>	57
4.7.2.62	<a href="#">cpm_wait_BndCommsS4DEx</a>	57
4.7.2.63	<a href="#">cpm_wait_BndCommV3D</a>	58
4.7.2.64	<a href="#">cpm_wait_BndCommV3DEx</a>	58
4.7.2.65	<a href="#">cpm_Waitall</a>	59
4.7.2.66	<a href="#">CreateProcessGroup</a>	59
4.7.2.67	<a href="#">FindVoxelInfo</a>	60
4.7.2.68	<a href="#">flush</a>	60
4.7.2.69	<a href="#">flush</a>	60
4.7.2.70	<a href="#">Gather</a>	60
4.7.2.71	<a href="#">Gather</a>	61
4.7.2.72	<a href="#">Gatherv</a>	61
4.7.2.73	<a href="#">Gatherv</a>	62
4.7.2.74	<a href="#">get_instance</a>	62
4.7.2.75	<a href="#">get_instance</a>	63
4.7.2.76	<a href="#">GetBCID</a>	63
4.7.2.77	<a href="#">GetBndCommBufferSize</a>	63
4.7.2.78	<a href="#">GetDivNum</a>	64
4.7.2.79	<a href="#">GetDivPos</a>	64
4.7.2.80	<a href="#">GetGlobalOrigin</a>	64
4.7.2.81	<a href="#">GetGlobalRegion</a>	64
4.7.2.82	<a href="#">GetGlobalVoxelSize</a>	65
4.7.2.83	<a href="#">GetLocalOrigin</a>	65
4.7.2.84	<a href="#">GetLocalRegion</a>	65
4.7.2.85	<a href="#">GetLocalVoxelSize</a>	66
4.7.2.86	<a href="#">GetMPI_Comm</a>	66
4.7.2.87	<a href="#">GetMPI_Datatype</a>	66
4.7.2.88	<a href="#">GetMPI_Datatype</a>	67
4.7.2.89	<a href="#">GetMPI_Op</a>	67
4.7.2.90	<a href="#">GetMyRankID</a>	67
4.7.2.91	<a href="#">GetNeighborRankID</a>	68
4.7.2.92	<a href="#">GetNumRank</a>	68
4.7.2.93	<a href="#">GetPeriodicRankID</a>	68
4.7.2.94	<a href="#">GetPitch</a>	69
4.7.2.95	<a href="#">GetVoxelHeadIndex</a>	69
4.7.2.96	<a href="#">GetVoxelTailIndex</a>	69
4.7.2.97	<a href="#">InitArray</a>	69
4.7.2.98	<a href="#">Initialize</a>	70
4.7.2.99	<a href="#">Initialize</a>	70

4.7.2.100	irecv	70
4.7.2.101	irecv	71
4.7.2.102	Isend	71
4.7.2.103	Isend	72
4.7.2.104	IsParallel	72
4.7.2.105	IsParallel	72
4.7.2.106	PeriodicCommS3D	72
4.7.2.107	PeriodicCommS3D	73
4.7.2.108	PeriodicCommS4D	73
4.7.2.109	PeriodicCommS4D	74
4.7.2.110	PeriodicCommS4DEx	75
4.7.2.111	PeriodicCommS4DEx	75
4.7.2.112	PeriodicCommV3D	76
4.7.2.113	PeriodicCommV3D	76
4.7.2.114	PeriodicCommV3DEx	77
4.7.2.115	PeriodicCommV3DEx	77
4.7.2.116	Recv	78
4.7.2.117	Recv	78
4.7.2.118	Send	79
4.7.2.119	Send	79
4.7.2.120	SetBndCommBuffer	79
4.7.2.121	VoxelInit	80
4.7.2.122	VoxelInit	80
4.7.2.123	VoxelInit	81
4.7.2.124	Wait	81
4.7.2.125	wait_BndCommS3D	82
4.7.2.126	wait_BndCommS3D	82
4.7.2.127	wait_BndCommS4D	83
4.7.2.128	wait_BndCommS4D	83
4.7.2.129	wait_BndCommS4DEx	84
4.7.2.130	wait_BndCommS4DEx	84
4.7.2.131	wait_BndCommV3D	85
4.7.2.132	wait_BndCommV3D	85
4.7.2.133	wait_BndCommV3DEx	86
4.7.2.134	wait_BndCommV3DEx	86
4.7.2.135	Waitall	87
4.8	クラス cpm_TextParser	87
4.8.1	説明	88
4.8.2	コンストラクタとデストラクタ	89
4.8.2.1	cpm_TextParser	89



4.8.2.2	~cpm_TextParser	89
4.8.3	関数	89
4.8.3.1	Read	89
4.8.3.2	readVector	89
4.8.3.3	readVector	90
4.8.3.4	readVector	90
4.8.4	変数	90
4.8.4.1	m_tp	90
4.9	クラス cpm_TextParserDomain	91
4.9.1	説明	91
4.9.2	コンストラクタとデストラクタ	92
4.9.2.1	cpm_TextParserDomain	92
4.9.2.2	~cpm_TextParserDomain	92
4.9.3	関数	92
4.9.3.1	Read	92
4.10	クラス cpm_VoxellInfo	92
4.10.1	説明	93
4.10.2	フレンドと関連する関数	93
4.10.2.1	cpm_ParaManager	93
4.11	構造体 S_BNDCOMM_BUFFER	93
4.11.1	説明	94
4.11.2	コンストラクタとデストラクタ	94
4.11.2.1	S_BNDCOMM_BUFFER	94
4.11.2.2	~S_BNDCOMM_BUFFER	94
4.11.3	変数	94
4.11.3.1	m_bufX	94
4.11.3.2	m_bufY	95
4.11.3.3	m_bufZ	95
4.11.3.4	m_maxN	95
4.11.3.5	m_maxVC	95
4.11.3.6	m_nwX	95
4.11.3.7	m_nwY	96
4.11.3.8	m_nwZ	96
5	ファイル	97
5.1	cpm_Base.h	97
5.1.1	説明	98
5.1.2	マクロ定義	98
5.1.2.1	CPM_INLINE	98
5.2	cpm_Define.h	98

5.2.1	説明	100
5.2.2	マクロ定義	100
5.2.2.1	_IDX_S3D	100
5.2.2.2	_IDX_S4D	100
5.2.2.3	_IDX_S4DEX	101
5.2.2.4	_IDX_V3D	101
5.2.2.5	_IDX_V3DEX	102
5.2.2.6	REAL_BUF_TYPE	102
5.2.2.7	REAL_TYPE	102
5.2.3	列挙型	102
5.2.3.1	CPM_Datatype	102
5.2.3.2	cpm_DirFlag	103
5.2.3.3	cpm_ErrorCode	103
5.2.3.4	cpm_FaceFlag	105
5.2.3.5	CPM_Op	105
5.2.3.6	cpm_PMFlag	106
5.3	cpm_DomainInfo.cpp	106
5.3.1	説明	106
5.4	cpm_DomainInfo.h	107
5.4.1	説明	107
5.5	cpm_ObjList.h	108
5.5.1	説明	108
5.5.2	型定義	109
5.5.2.1	RankNoMap	109
5.6	cpm_ParaManager.cpp	109
5.6.1	説明	109
5.7	cpm_ParaManager.h	110
5.7.1	説明	110
5.7.2	型定義	111
5.7.2.1	BndCommInfoMap	111
5.7.2.2	RankNoMap	111
5.7.2.3	VoxelInfoMap	111
5.8	cpm_ParaManager_Alloc.cpp	111
5.8.1	説明	111
5.9	cpm_ParaManager_BndComm.h	112
5.9.1	説明	112
5.9.2	マクロ定義	112
5.9.2.1	_IDAFX	112
5.9.2.2	_IDXFY	113
5.9.2.3	_IDXFZ	113

5.10 cpm_ParaManager_BndCommEx.h . . . . .	113
5.10.1 説明 . . . . .	113
5.10.2 マクロ定義 . . . . .	114
5.10.2.1 _IDXFX . . . . .	114
5.10.2.2 _IDXFY . . . . .	114
5.10.2.3 _IDXFZ . . . . .	114
5.11 cpm_ParaManager_frtIF.cpp . . . . .	114
5.11.1 説明 . . . . .	117
5.11.2 マクロ定義 . . . . .	117
5.11.2.1 cpm_Abort_ . . . . .	117
5.11.2.2 cpm_Allgather_ . . . . .	117
5.11.2.3 cpm_Allgatherv_ . . . . .	118
5.11.2.4 cpm_Allreduce_ . . . . .	118
5.11.2.5 cpm_Barrier_ . . . . .	118
5.11.2.6 cpm_Bcast_ . . . . .	118
5.11.2.7 cpm_BndCommS3D_ . . . . .	118
5.11.2.8 cpm_BndCommS3D_nowait_ . . . . .	118
5.11.2.9 cpm_BndCommS4D_ . . . . .	118
5.11.2.10 cpm_BndCommS4D_nowait_ . . . . .	118
5.11.2.11 cpm_BndCommS4DEx_ . . . . .	118
5.11.2.12 cpm_BndCommS4DEx_nowait_ . . . . .	118
5.11.2.13 cpm_BndCommV3D_ . . . . .	118
5.11.2.14 cpm_BndCommV3D_nowait_ . . . . .	119
5.11.2.15 cpm_BndCommV3DEx_ . . . . .	119
5.11.2.16 cpm_BndCommV3DEx_nowait_ . . . . .	119
5.11.2.17 CPM_EXTERN . . . . .	119
5.11.2.18 cpm_Gather_ . . . . .	119
5.11.2.19 cpm_Gatherv_ . . . . .	119
5.11.2.20 cpm_GetBCID_ . . . . .	119
5.11.2.21 cpm_GetDivNum_ . . . . .	119
5.11.2.22 cpm_GetDivPos_ . . . . .	119
5.11.2.23 cpm_GetGlobalOrigin_ . . . . .	119
5.11.2.24 cpm_GetGlobalRegion_ . . . . .	119
5.11.2.25 cpm_GetGlobalVoxelSize_ . . . . .	119
5.11.2.26 cpm_GetLocalOrigin_ . . . . .	120
5.11.2.27 cpm_GetLocalRegion_ . . . . .	120
5.11.2.28 cpm_GetLocalVoxelSize_ . . . . .	120
5.11.2.29 cpm_GetMyRankID_ . . . . .	120
5.11.2.30 cpm_GetNeighborRankID_ . . . . .	120
5.11.2.31 cpm_GetNumRank_ . . . . .	120

5.11.2.32	<a href="#">cpm_GetPeriodicRankID_</a>	120
5.11.2.33	<a href="#">cpm_GetPitch_</a>	120
5.11.2.34	<a href="#">cpm_GetVoxelHeadIndex_</a>	120
5.11.2.35	<a href="#">cpm_GetVoxelTailIndex_</a>	120
5.11.2.36	<a href="#">cpm_Initialize_</a>	120
5.11.2.37	<a href="#">cpm_Irecv_</a>	120
5.11.2.38	<a href="#">cpm_Isend_</a>	121
5.11.2.39	<a href="#">cpm_IsParallel_</a>	121
5.11.2.40	<a href="#">cpm_PeriodicCommS3D</a>	121
5.11.2.41	<a href="#">cpm_PeriodicCommS4D</a>	121
5.11.2.42	<a href="#">cpm_PeriodicCommS4DEx</a>	121
5.11.2.43	<a href="#">cpm_PeriodicCommV3D</a>	121
5.11.2.44	<a href="#">cpm_PeriodicCommV3DEx</a>	121
5.11.2.45	<a href="#">cpm_Recv_</a>	121
5.11.2.46	<a href="#">cpm_Send_</a>	121
5.11.2.47	<a href="#">cpm_SetBndCommBuffer_</a>	121
5.11.2.48	<a href="#">cpm_Voxellnit_</a>	121
5.11.2.49	<a href="#">cpm_Voxellnit_nodiv_</a>	121
5.11.2.50	<a href="#">cpm_Wait_</a>	122
5.11.2.51	<a href="#">cpm_wait_BndCommsS3D_</a>	122
5.11.2.52	<a href="#">cpm_wait_BndCommsS4D_</a>	122
5.11.2.53	<a href="#">cpm_wait_BndCommsS4DEx_</a>	122
5.11.2.54	<a href="#">cpm_wait_BndCommV3D_</a>	122
5.11.2.55	<a href="#">cpm_wait_BndCommV3DEx_</a>	122
5.11.2.56	<a href="#">cpm_Waitall_</a>	122
5.11.3	<a href="#">関数</a>	122
5.11.3.1	<a href="#">cpm_Abort_</a>	122
5.11.3.2	<a href="#">cpm_Allgather_</a>	122
5.11.3.3	<a href="#">cpm_Allgatherv_</a>	123
5.11.3.4	<a href="#">cpm_Allreduce_</a>	123
5.11.3.5	<a href="#">cpm_Barrier_</a>	124
5.11.3.6	<a href="#">cpm_Bcast_</a>	124
5.11.3.7	<a href="#">cpm_BndCommsS3D_</a>	124
5.11.3.8	<a href="#">cpm_BndCommsS3D_nowait_</a>	125
5.11.3.9	<a href="#">cpm_BndCommsS4D_</a>	125
5.11.3.10	<a href="#">cpm_BndCommsS4D_nowait_</a>	126
5.11.3.11	<a href="#">cpm_BndCommS4DEx_</a>	126
5.11.3.12	<a href="#">cpm_BndCommS4DEx_nowait_</a>	127
5.11.3.13	<a href="#">cpm_BndCommV3D_</a>	127
5.11.3.14	<a href="#">cpm_BndCommV3D_nowait_</a>	128

5.11.3.15 cpm_BndCommV3DEx_ . . . . .	128
5.11.3.16 cpm_BndCommV3DEx_nowait_ . . . . .	128
5.11.3.17 cpm_Gather_ . . . . .	129
5.11.3.18 cpm_Gatherv_ . . . . .	129
5.11.3.19 cpm_GetBCID_ . . . . .	130
5.11.3.20 cpm_GetDivNum_ . . . . .	130
5.11.3.21 cpm_GetDivPos_ . . . . .	130
5.11.3.22 cpm_GetGlobalOrigin_ . . . . .	131
5.11.3.23 cpm_GetGlobalRegion_ . . . . .	131
5.11.3.24 cpm_GetGlobalVoxelSize_ . . . . .	131
5.11.3.25 cpm_GetLocalOrigin_ . . . . .	131
5.11.3.26 cpm_GetLocalRegion_ . . . . .	132
5.11.3.27 cpm_GetLocalVoxelSize_ . . . . .	132
5.11.3.28 cpm_GetMyRankID_ . . . . .	132
5.11.3.29 cpm_GetNeighborRankID_ . . . . .	133
5.11.3.30 cpm_GetNumRank_ . . . . .	133
5.11.3.31 cpm_GetPeriodicRankID_ . . . . .	133
5.11.3.32 cpm_GetPitch_ . . . . .	133
5.11.3.33 cpm_GetVoxelHeadIndex_ . . . . .	134
5.11.3.34 cpm_GetVoxelTailIndex_ . . . . .	134
5.11.3.35 cpm_Initialize_ . . . . .	134
5.11.3.36 cpm_Irecv_ . . . . .	135
5.11.3.37 cpm_Isend_ . . . . .	135
5.11.3.38 cpm_IsParallel_ . . . . .	135
5.11.3.39 cpm_PeriodicComms3D_ . . . . .	136
5.11.3.40 cpm_PeriodicComms4D_ . . . . .	136
5.11.3.41 cpm_PeriodicComms4DEx_ . . . . .	137
5.11.3.42 cpm_PeriodicCommV3D_ . . . . .	137
5.11.3.43 cpm_PeriodicCommV3DEx_ . . . . .	138
5.11.3.44 cpm_Recv_ . . . . .	138
5.11.3.45 cpm_Send_ . . . . .	139
5.11.3.46 cpm_SetBndCommBuffer_ . . . . .	139
5.11.3.47 cpm_Voxellnit_ . . . . .	139
5.11.3.48 cpm_Voxellnit_nodiv_ . . . . .	140
5.11.3.49 cpm_Wait_ . . . . .	140
5.11.3.50 cpm_wait_BndComms3D_ . . . . .	141
5.11.3.51 cpm_wait_BndComms4D_ . . . . .	141
5.11.3.52 cpm_wait_BndComms4DEx_ . . . . .	142
5.11.3.53 cpm_wait_BndCommV3D_ . . . . .	142
5.11.3.54 cpm_wait_BndCommV3DEx_ . . . . .	142

5.11.3.55 cpm_Waitall_ . . . . .	143
5.12 cpm_ParaManager_inline.h . . . . .	143
5.12.1 説明 . . . . .	143
5.13 cpm_ParaManager_MPI.cpp . . . . .	144
5.13.1 説明 . . . . .	144
5.14 cpm_TextParser.cpp . . . . .	145
5.14.1 説明 . . . . .	145
5.15 cpm_TextParser.h . . . . .	145
5.15.1 説明 . . . . .	146
5.16 cpm_TextParserDomain.cpp . . . . .	147
5.16.1 説明 . . . . .	147
5.17 cpm_TextParserDomain.h . . . . .	147
5.17.1 説明 . . . . .	148
5.18 cpm_Version.h . . . . .	149
5.18.1 説明 . . . . .	149
5.18.2 マクロ定義 . . . . .	149
5.18.2.1 CPM_REVISION . . . . .	149
5.18.2.2 CPM_VERSION_NO . . . . .	149
5.19 cpm_VoxelInfo.cpp . . . . .	150
5.19.1 説明 . . . . .	150
5.20 cpm_VoxelInfo.h . . . . .	150
5.20.1 説明 . . . . .	151

# 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 . . . . .	22
cpm_DomainInfo . . . . .	15
cpm_GlobalDomainInfo . . . . .	19
cpm_LocalDomainInfo . . . . .	22
cpm_ObjList< T > . . . . .	23
cpm_ParaManager . . . . .	26
cpm_TextParser . . . . .	87
cpm_TextParserDomain . . . . .	91
cpm_VoxelInfo . . . . .	92
S_BNDCOMM_BUFFER . . . . .	93





## 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>	22
<a href="#">cpm_ObjList&lt; T &gt;</a>	23
<a href="#">cpm_ParaManager</a>	26
<a href="#">cpm_TextParser</a>	87
<a href="#">cpm_TextParserDomain</a>	91
<a href="#">cpm_VoxelInfo</a>	92
<a href="#">S_BNDCOMM_BUFFER</a>	93



## 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 . . . . .	97
cpm_Define.h . . . . .	98
cpm_DomainInfo.cpp . . . . .	106
cpm_DomainInfo.h . . . . .	107
cpm_ObjList.h . . . . .	108
cpm_ParaManager.cpp . . . . .	109
cpm_ParaManager.h . . . . .	110
cpm_ParaManager_Alloc.cpp . . . . .	111
cpm_ParaManager_BndComm.h . . . . .	112
cpm_ParaManager_BndCommEx.h . . . . .	113
cpm_ParaManager_frtIF.cpp . . . . .	114
cpm_ParaManager_inline.h . . . . .	143
cpm_ParaManager_MPI.cpp . . . . .	144
cpm_TextParser.cpp . . . . .	145
cpm_TextParser.h . . . . .	145
cpm_TextParserDomain.cpp . . . . .	147
cpm_TextParserDomain.h . . . . .	147
cpm_Version.h . . . . .	149
cpm_VoxelInfo.cpp . . . . .	150
cpm_VoxelInfo.h . . . . .	150



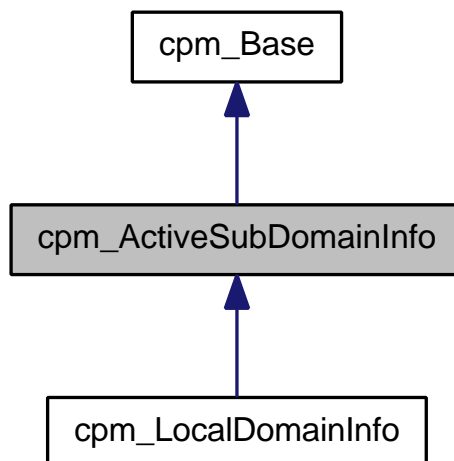
## Chapter 4

# クラス

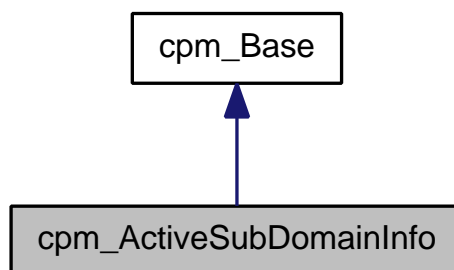
### 4.1 クラス cpm\_ActiveSubDomainInfo

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

cpm\_ActiveSubDomainInfo に対する継承グラフ



cpm\_ActiveSubDomainInfo のコラボレーション図



#### Public メソッド

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

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

#### 4.1.1 説明

CPM のサブ領域情報クラス

cpm\_DomainInfo.h の 97 行で定義されています。

#### 4.1.2 コンストラクタとデストラクタ

##### 4.1.2.1 cpm\_ActiveSubDomainInfo::cpm\_ActiveSubDomainInfo ( )

デフォルトコンストラクタ

cpm\_DomainInfo.cpp の 118 行で定義されています。

参照先 `clear()`.

##### 4.1.2.2 cpm\_ActiveSubDomainInfo::cpm\_ActiveSubDomainInfo ( int pos[3], int bcid[6] )

コンストラクタ

引数

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

cpm\_DomainInfo.cpp の 126 行で定義されています。

参照先 `Set()`.

##### 4.1.2.3 cpm\_ActiveSubDomainInfo::~cpm\_ActiveSubDomainInfo ( ) [virtual]

デストラクタ

cpm\_DomainInfo.cpp の 134 行で定義されています。

#### 4.1.3 関数

##### 4.1.3.1 void cpm\_ActiveSubDomainInfo::clear ( ) [virtual]

情報のクリア

`cpm_LocalDomainInfo` で再定義されています。

cpm\_DomainInfo.cpp の 141 行で定義されています。

参照先 X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 `cpm_ActiveSubDomainInfo()`.

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

BCID の取得

戻り値

境界条件ID 情報整数配列のポインタ

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

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

位置の取得

戻り値

位置情報整数配列のポインタ

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

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

比較演算子

引数

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

戻り値

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

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

#### 4.1.3.5 `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 205 行で定義されています。

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

値のセット

引数

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

cpm\_DomainInfo.cpp の 157 行で定義されています。

参照先 SetBCID(), と SetPos().

参照元 cpm\_ActiveSubDomainInfo().

#### 4.1.3.7 void cpm\_ActiveSubDomainInfo::SetBCID ( int *bcid*[6] )

境界条件ID のセット

引数

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

cpm\_DomainInfo.cpp の 184 行で定義されています。

参照先 X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 Set().

#### 4.1.3.8 void cpm\_ActiveSubDomainInfo::SetPos ( int *pos*[3] )

位置のセット

引数

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

cpm\_DomainInfo.cpp の 166 行で定義されています。

参照元 Set().

このクラスの説明は次のファイルから生成されました:

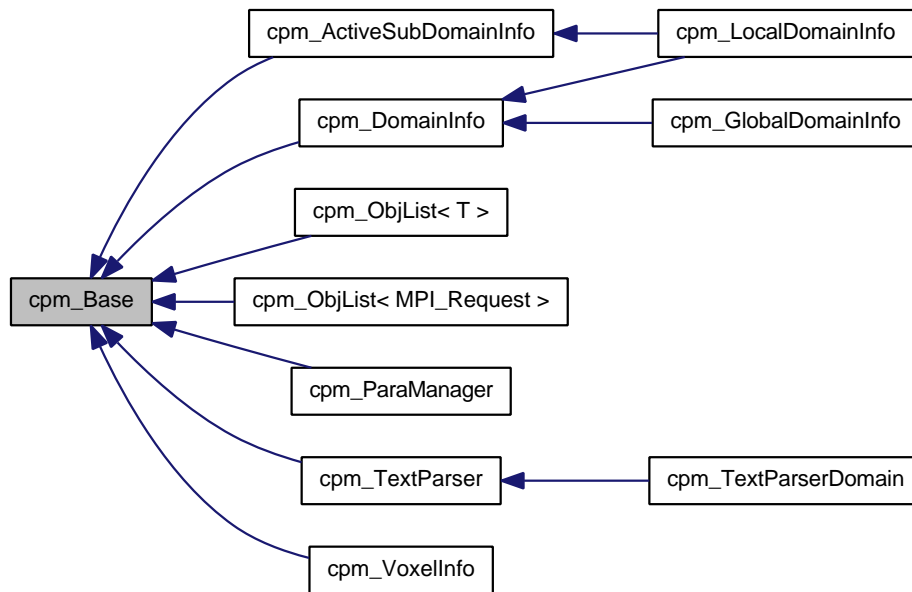
- [cpm\\_DomainInfo.h](#)
- [cpm\\_DomainInfo.cpp](#)

## 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 41 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 235 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 238 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

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

##### 戻り値

<code>0</code>	一致する
<code>0</code> 以外	一致しない

`cpm_Base.h` の 201 行で定義されています。

参照元 `cpm_strCompareN()`.

### 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

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

##### 戻り値

<code>0</code>	一致する
<code>0</code> 以外	一致しない

`cpm_Base.h` の 223 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 68 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

<code>in</code>	<code>mem</code>	メモリ量 (byte)
-----------------	------------------	-------------

戻り値

メモリ量の文字列

`cpm_Base.h` の 142 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 50 行で定義されています。

参照元 `cpm_ParaManager::GetMyRankID()`, `cpm_ParaManager::PeriodicCommS4D()`, と `cpm_ParaManager::PeriodicCommS4DEx()`.

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

経過時刻の取得 (gettimeofday 版)

引数

<code>in</code>	<code>before</code>	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 113 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 98 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 132 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 122 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 77 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\_ParaManager::Irecv(), cpm\_ParaManager::Isend(), cpm\_ParaManager::Recv(), cpm\_ParaManager::Send(), と cpm\_ParaManager::Voxellnit().

#### 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 59 行で定義されています。

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

実数型 `REAL_TYPE` が倍精度かどうか確認

戻り値

<i>true</i>	倍精度
<i>false</i>	単精度

`cpm_Base.h` の 87 行で定義されています。

参照先 `REAL_TYPE`.

参照元 `cpm_ParaManager::GetMPI_Datatype()`.

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

バージョンを出力する

引数

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

`cpm_Base.h` の 176 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 185 行で定義されています。

参照先 `CPM_VERSION_NO`.

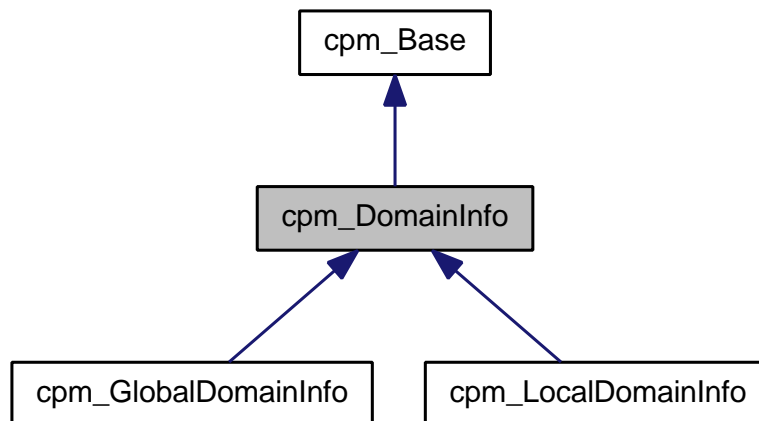
このクラスの説明は次のファイルから生成されました:

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

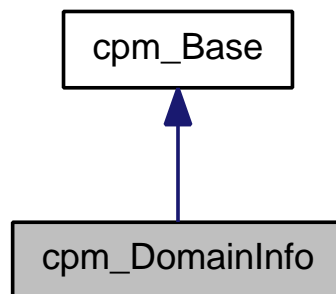
## 4.3 クラス `cpm_DomainInfo`

```
#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

### 4.3.1 説明

CPM の領域情報クラス

cpm\_DomainInfo.h の 22 行で定義されています。

### 4.3.2 コンストラクタとデストラクタ

#### 4.3.2.1 `cpm_DomainInfo::cpm_DomainInfo ( )`

コンストラクタ

`cpm_DomainInfo.cpp` の 18 行で定義されています。

参照先 `clear()`.

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

デストラクタ

`cpm_DomainInfo.cpp` の 26 行で定義されています。

### 4.3.3 関数

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

情報のクリア

[cpm\\_LocalDomainInfo](#), と [cpm\\_GlobalDomainInfo](#) で再定義されています。

`cpm_DomainInfo.cpp` の 33 行で定義されています。

参照元 `cpm_DomainInfo()`.

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

原点の取得

戻り値

原点情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 57 行で定義されています。

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

ピッチの取得

戻り値

ピッチ情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 75 行で定義されています。

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

空間サイズの取得

戻り値

空間サイズ情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 93 行で定義されています。

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

VOXEL 数の取得

戻り値

VOXEL 数情報実数配列のポインタ

cpm\_DomainInfo.cpp の 111 行で定義されています。

#### 4.3.3.6 `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 47 行で定義されています。

参照元 cpm\_ParaManager::Voxellnit().

#### 4.3.3.7 `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 65 行で定義されています。

参照元 cpm\_ParaManager::Voxellnit().

#### 4.3.3.8 `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 83 行で定義されています。

参照元 cpm\_ParaManager::Voxellnit().

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

VOXEL 数のセット

引数

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

cpm\_DomainInfo.cpp の 101 行で定義されています。

参照元 cpm\_ParaManager::Voxellnit().



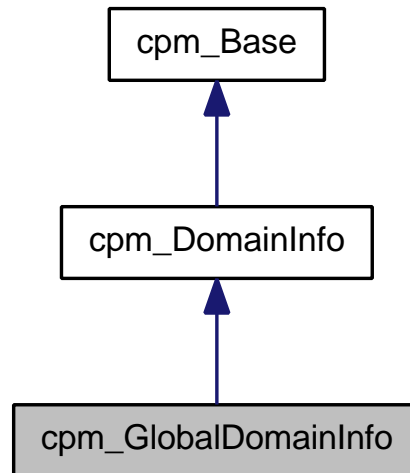
このクラスの説明は次のファイルから生成されました:

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

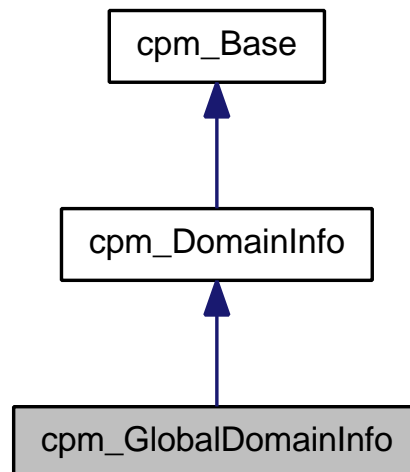
## 4.4 クラス cpm\_GlobalDomainInfo

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

cpm\_GlobalDomainInfo に対する継承グラフ



cpm\_GlobalDomainInfo のコラボレーション図



### Public メソッド

- [cpm\\_GlobalDomainInfo \(\)](#)
- [virtual ~cpm\\_GlobalDomainInfo \(\)](#)
- [virtual void clear \(\)](#)
- [void SetDivNum \(int div\[3\]\)](#)
- [const int \\* GetDivNum \(\) const](#)

- bool `IsExistSubDomain` (`cpm_ActiveSubDomainInfo` subDomain)
- bool `AddSubDomain` (`cpm_ActiveSubDomainInfo` subDomain)
- int `GetSubDomainNum` () const
- const `cpm_ActiveSubDomainInfo` \* `GetSubDomainInfo` (size\_t idx) const

#### 4.4.1 説明

CPM の全体領域情報クラス

`cpm_DomainInfo.h` の 176 行で定義されています。

#### 4.4.2 コンストラクタとデストラクタ

##### 4.4.2.1 `cpm_GlobalDomainInfo::cpm_GlobalDomainInfo ( )`

コンストラクタ

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

参照先 `clear()`.

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

デストラクタ

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

#### 4.4.3 関数

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

活性サブドメイン情報の追加

引数

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

戻り値

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

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

参照先 `IsExistSubDomain()`.

参照元 `cpm_ParaManager::VoxellInit()`.

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

情報のクリア

`cpm_DomainInfo` を再定義しています。

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

参照元 `cpm_GlobalDomainInfo()`.

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

領域分割数の取得

戻り値

領域分割数整数配列のポインタ

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

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

活性サブドメイン情報を取得

引数

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

戻り値

活性サブドメイン情報ポインタ

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

参照先 `GetSubDomainNum()`.

#### 4.4.3.5 `int cpm_GlobalDomainInfo::GetSubDomainNum ( ) const`

活性サブドメインの数を取得

戻り値

活性サブドメインの数

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

参照元 `GetSubDomainInfo()`, と `cpm_ParaManager::VoxellInit()`.

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

活性サブドメイン情報の存在チェック

引数

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

戻り値

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

`cpm_DomainInfo.cpp` の 272 行で定義されています。

参照元 `AddSubDomain()`.

#### 4.4.3.7 void cpm\_GlobalDomainInfo::SetDivNum ( int div[3] )

領域分割数のセット

引数

in	div	領域分割数
----	-----	-------

cpm\_DomainInfo.cpp の 254 行で定義されています。

参照元 cpm\_ParaManager::Voxellnit().

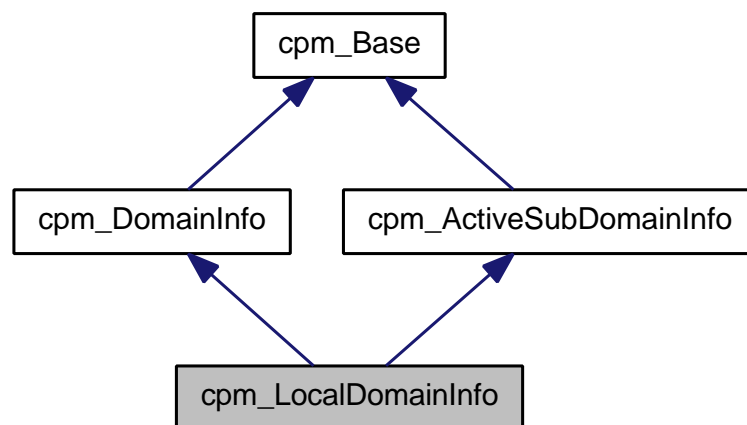
このクラスの説明は次のファイルから生成されました:

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

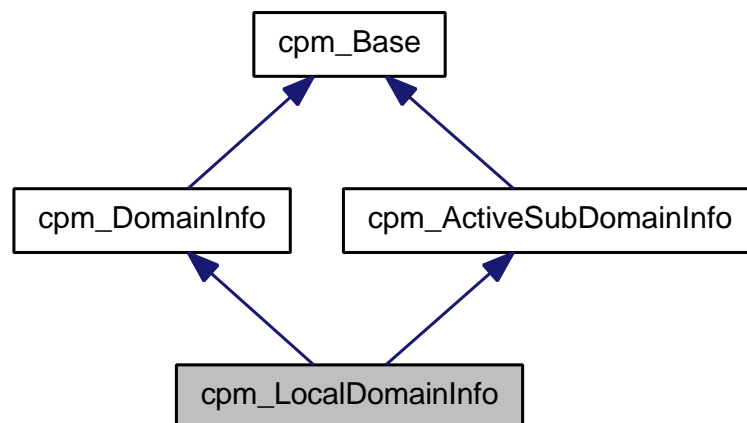
## 4.5 クラス cpm\_LocalDomainInfo

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

cpm\_LocalDomainInfo に対する継承グラフ



cpm\_LocalDomainInfo のコラボレーション図



## Public メソッド

- `cpm_LocalDomainInfo()`
- `virtual ~cpm_LocalDomainInfo()`
- `virtual void clear()`

### 4.5.1 説明

CPM のローカル領域情報クラス

`cpm_DomainInfo.h` の 244 行で定義されています。

### 4.5.2 コンストラクタとデストラクタ

#### 4.5.2.1 `cpm_LocalDomainInfo::cpm_LocalDomainInfo()`

コンストラクタ

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

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

デストラクタ

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

### 4.5.3 関数

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

情報のクリア

`cpm_DomainInfo` を再定義しています。

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

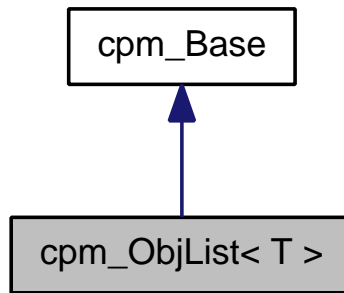
このクラスの説明は次のファイルから生成されました:

- `cpm_DomainInfo.h`
- `cpm_DomainInfo.cpp`

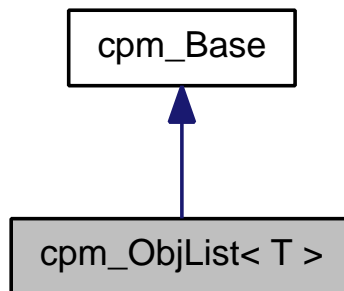
## 4.6 クラス テンプレート `cpm_ObjList< T >`

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

cpm\_ObjList< T > に対する継承グラフ



cpm\_ObjList< T > のコラボレーション図



## Public メソッド

- [cpm\\_ObjList](#) ()
- [~cpm\\_ObjList](#) ()
- T \* [Create](#) ()
- int [Add](#) (T \*obj)
- [cpm\\_StatusCode Delete](#) (int key)
- T \* [Get](#) (int key)

### 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 コンストラクタとデストラクタ

```
4.6.2.1 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。

4.6.2.2 `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。

### 4.6.3 関数

4.6.3.1 `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()`.

4.6.3.2 `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()`.

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

オブジェクトの削除

引数

<code>in</code>	<code>key</code>	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().

4.6.3.4 `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().

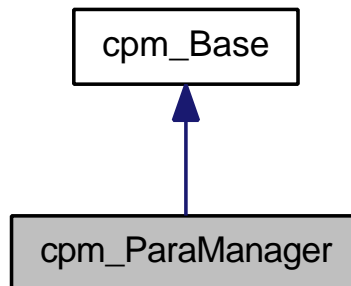
このクラスの説明は次のファイルから生成されました:

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

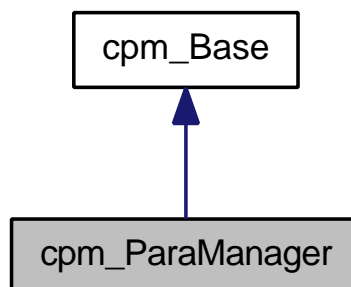
## 4.7 クラス cpm\_ParaManager

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

cpm\_ParaManager に対する継承グラフ



cpm\_ParaManager のコラボレーション図





## Public メソッド

- `cpm_ErrorCode Initialize ()`
- `cpm_ErrorCode Initialize (int &argc, char **&argv)`
- `bool IsParallel ()`
- `bool IsParallel () const`
- `cpm_ErrorCode Voxellnit (cpm_GlobalDomainInfo *domainInfo, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)`
- `cpm_ErrorCode Voxellnit (int div[3], int vox[3], REAL_TYPE origin[3], REAL_TYPE pitch[3], int obcid[6], size_t maxVC=1, size_t maxN=3, int procGrpNo=0)`
- `cpm_ErrorCode Voxellnit (int vox[3], REAL_TYPE origin[3], REAL_TYPE pitch[3], int obcid[6], size_t maxVC=1, size_t maxN=3, int procGrpNo=0)`
- `int CreateProcessGroup (int nproc, int *proclist, int parentProcGrpNo=0)`
- `const cpm_VoxelInfo * FindVoxelInfo (int procGrpNo=0)`
- `const int * GetDivNum (int procGrpNo=0)`
- `const REAL_TYPE * GetPitch (int procGrpNo=0)`
- `const int * GetGlobalVoxelSize (int procGrpNo=0)`
- `const REAL_TYPE * GetGlobalOrigin (int procGrpNo=0)`
- `const REAL_TYPE * GetGlobalRegion (int procGrpNo=0)`
- `const int * GetLocalVoxelSize (int procGrpNo=0)`
- `const REAL_TYPE * GetLocalOrigin (int procGrpNo=0)`
- `const REAL_TYPE * GetLocalRegion (int procGrpNo=0)`
- `const int * GetDivPos (int procGrpNo=0)`
- `const int * GetBCID (int procGrpNo=0)`
- `const int * GetVoxelHeadIndex (int procGrpNo=0)`
- `const int * GetVoxelTailIndex (int procGrpNo=0)`
- `const int * GetNeighborRankID (int procGrpNo=0)`
- `const int * GetPeriodicRankID (int procGrpNo=0)`
- `int GetMyRankID (int procGrpNo=0)`
- `int GetNumRank (int procGrpNo=0)`
- `MPI_Comm GetMPI_Comm (int procGrpNo=0)`
- `void Abort (int errorcode)`
- `cpm_ErrorCode Barrier (int procGrpNo=0)`
- `cpm_ErrorCode Wait (MPI_Request *request)`
- `cpm_ErrorCode Waitall (int count, MPI_Request requests[])`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Bcast (T *buf, int count, int root, int procGrpNo=0)`
- `cpm_ErrorCode Bcast (MPI_Datatype dtype, void *buf, int count, int root, int procGrpNo=0)`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Send (T *buf, int count, int dest, int procGrpNo=0)`
- `cpm_ErrorCode Send (MPI_Datatype dtype, void *buf, int count, int dest, int procGrpNo=0)`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Recv (T *buf, int count, int source, int procGrpNo=0)`
- `cpm_ErrorCode Recv (MPI_Datatype dtype, void *buf, int count, int source, int procGrpNo=0)`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Isend (T *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)`
- `cpm_ErrorCode Isend (MPI_Datatype dtype, void *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Irecv (T *buf, int count, int source, MPI_Request *request, int procGrpNo=0)`
- `cpm_ErrorCode Irecv (MPI_Datatype dtype, void *buf, int count, int source, MPI_Request *request, int procGrpNo=0)`
- `template<class T >`  
`CPM_INLINE cpm_ErrorCode Allreduce (T *sendbuf, T *recvbuf, int count, MPI_Op op, int procGrpNo=0)`
- `cpm_ErrorCode Allreduce (MPI_Datatype dtype, void *sendbuf, void *recvbuf, int count, MPI_Op op, int procGrpNo=0)`

- `template<class Ts , class Tr >`  
`CPM_INLINE cpm_ErrorCode Gather` (Ts \*sendbuf, int sendcnt, Tr \*recvbuf, int recvcnt, int root, int procGrpNo=0)
- `cpm_ErrorCode Gather` (MPI\_Datatype stype, void \*sendbuf, int sendcnt, MPI\_Datatype rtype, void \*recvbuf, int recvcnt, int root, int procGrpNo=0)
- `template<class Ts , class Tr >`  
`CPM_INLINE cpm_ErrorCode Allgather` (Ts \*sendbuf, int sendcnt, Tr \*recvbuf, int recvcnt, int procGrpNo=0)
- `cpm_ErrorCode Allgather` (MPI\_Datatype stype, void \*sendbuf, int sendcnt, MPI\_Datatype rtype, void \*recvbuf, int recvcnt, int procGrpNo=0)
- `template<class Ts , class Tr >`  
`CPM_INLINE cpm_ErrorCode Gatherv` (Ts \*sendbuf, int sendcnt, Tr \*recvbuf, int \*recvcnts, int \*displs, int root, int procGrpNo=0)
- `cpm_ErrorCode Gatherv` (MPI\_Datatype stype, void \*sendbuf, int sendcnt, MPI\_Datatype rtype, void \*recvbuf, int \*recvcnts, int \*displs, int root, int procGrpNo=0)
- `template<class Ts , class Tr >`  
`CPM_INLINE cpm_ErrorCode Allgatherv` (Ts \*sendbuf, int sendcnt, Tr \*recvbuf, int \*recvcnts, int \*displs, int procGrpNo=0)
- `cpm_ErrorCode Allgatherv` (MPI\_Datatype stype, void \*sendbuf, int sendcnt, MPI\_Datatype rtype, void \*recvbuf, int \*recvcnts, int \*displs, int procGrpNo=0)
- `cpm_ErrorCode cpm_Wait` (int reqNo)
- `cpm_ErrorCode cpm_Waitall` (int count, int reqNoList[])
- `cpm_ErrorCode cpm_Isend` (void \*buf, int count, int datatype, int dest, int \*reqNo, int procGrpNo=0)
- `cpm_ErrorCode cpm_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_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_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_BndCommS4DEx` (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 BndComms3D` (T \*array, int imax, int jmax, int kmax, int vc, int vc\_comm, int procGrpNo=0)
- `cpm_ErrorCode BndComms3D` (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 PeriodicCommsS3D` (`MPI_Datatype dtype`, `void *array`, `int imax`, `int jmax`, `int kmax`, `int vc`, `int vc_comm`, `cpm_DirFlag dir`, `cpm_PMFlag pm`, `int procGrpNo=0`)
- `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 PeriodicCommsS4D` (`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 PeriodicCommsS4D` (`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 BndCommsS4DEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)`
- `cpm_ErrorCode BndCommsS4DEx (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 BndCommsS4DEx_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 BndCommsS4DEx_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_BndCommsS4DEx (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_BndCommsS4DEx (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 PeriodicCommsS4DEx (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 PeriodicCommsS4DEx (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)`

#### 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 関数

##### 4.7.2.1 `void cpm_ParaManager::Abort ( int errorcode )`

Abort

- `MPI_Abort` のインターフェイス

引数

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

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

参照元 `cpm_Abort()`, と `Voxellnit()`.

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

Allgather

- `MPI_Allgather` のインターフェイス

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	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\_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\_().

**4.7.2.3 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	stype	送信データのMPI_Datatype
in	sendbuf	送信データ
in	sendcnt	送信データのサイズ
in	rtype	受信データのMPI_Datatype
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\_MPI.cpp の 433 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().

**4.7.2.4 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	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	recvcnts	各ランクからの受信データサイズ
in	displs	各ランクからの受信データ配置位置
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 246 行で定義されています。

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

参照元 cpm\_Allgatherv\_().

**4.7.2.5 cpm\_ErrorCode cpm\_ParaManager::Allgatherv ( MPI\_Datatype stype, void \* sendbuf, int sendcnt, MPI\_Datatype rtype, void \* recvbuf, int \* recvcnts, int \* displs, int procGrpNo = 0 )**

Allgatherv

- MPI\_Allgatherv のインターフェイス
- MPI\_Datatype を指定するバージョン

引数

in	stype	送信データのMPI_Datatype
in	sendbuf	送信データ
in	sendcnt	送信データのサイズ
in	rtype	受信データのMPI_Datatype
out	recvbuf	受信データ
in	recvcnts	各ランクからの受信データサイズ
in	displs	各ランクからの受信データ配置位置
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 492 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().

**4.7.2.6 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	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 76 行で定義されています。

参照先 AllocDoubleS4D().

**4.7.2.7 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	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 32 行で定義されています。

参照先 GetLocalVoxelSize().

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

**4.7.2.8 double \* cpm\_ParaManager::AllocDoubleS4DEx ( int *nmax*, int *vc*, int *procGrpNo* = 0 )**

配列確保 double(*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 172 行で定義されています。

参照先 AllocDoubleS4D().

**4.7.2.9 double \* cpm\_ParaManager::AllocDoubleV3D ( int *vc*, int *procGrpNo* = 0 )**

配列確保 double(*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 108 行で定義されています。

参照先 AllocDoubleS4D().

**4.7.2.10 double \* cpm\_ParaManager::AllocDoubleV3DEx ( int *vc*, int *procGrpNo* = 0 )**

配列確保 double(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 140 行で定義されています。

参照先 `AllocDoubleS4D()`.

**4.7.2.11** `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()`.

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

配列確保 `float(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 44 行で定義されています。

参照先 `GetLocalVoxelSize()`.

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

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

配列確保 `float(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 180 行で定義されています。

参照先 AllocFloatS4D().

**4.7.2.14** 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().

**4.7.2.15** float \* cpm\_ParaManager::AllocFloatV3DEx ( int vc, int procGrpNo = 0 )

配列確保 float(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 148 行で定義されています。

参照先 AllocFloatS4D().

**4.7.2.16** int \* cpm\_ParaManager::AllocIntS3D ( int vc, int procGrpNo = 0 )

配列確保 int(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 92 行で定義されています。

参照先 AllocIntS4D().

4.7.2.17 `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()`.

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

配列確保 `int(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 188 行で定義されています。

参照先 `AllocIntS4D()`.

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

配列確保 `int(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 124 行で定義されています。

参照先 `AllocIntS4D()`.

4.7.2.20 `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().

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

配列確保 `REAL_TYPE(imax,jmax,kmax)`

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocRealS4D().

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

配列確保 `REAL_TYPE(imax,jmax,kmax,nmax)`

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm\_ParaManager\_Alloc.cpp の 20 行で定義されています。

参照先 GetLocalVoxelSize(), と REAL\_TYPE.

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

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

配列確保 `REAL_TYPE(nmax,imax,jmax,kmax)`

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

cpm\_ParaManager\_Alloc.cpp の 164 行で定義されています。

参照先 AllocRealS4D().

#### 4.7.2.24 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().

#### 4.7.2.25 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().

#### 4.7.2.26 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\_().

**4.7.2.27** `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().

**4.7.2.28** `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 flush().

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

Bcast

- MPI\_Bcast のインターフェイス

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

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

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

参照元 cpm\_Bcast\_().

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

Bcast

- MPI\_Bcast のインターフェイス
- MPI\_Datatype を指定するバージョン

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 234 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_BCAST, CPM\_ERROR\_NOT\_IN\_PROCGROUP, CPM\_SUCCESS, GetMPI\_Comm(), と cpm\_Base::IsCommNull().

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

袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の袖通信を行う

## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 43 行で定義されています。

参照先 BndCommS4D().

参照元 cpm\_BndCommS3D\_().

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

袖通信 (Scalar3D 版, MPI\_Datatype 指定)

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

## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 521 行で定義されています。

参照先 BndCommS4D().

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

非同期版袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait\_BndCommS3D をコールする



## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 165 行で定義されています。

参照先 BndCommS4D\_nowait().

参照元 cpm\_BndCommS3D\_nowait().

**4.7.2.34 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 583 行で定義されています。

参照先 BndCommS4D\_nowait().

**4.7.2.35 template<class T > CPM\_INLINE cpm\_ErrorCode cpm\_ParaManager::BndCommS4D ( T \* array, int imax, int jmax, int kmax, int nmax, int vc, int vc\_comm, int procGrpNo = 0 )**

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 61 行で定義されています。

参照先 CPM\_ERROR\_BNDCOMM\_BUFFER, CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, CPM\_ERROR\_GET\_NEIGHBOR\_RANK, CPM\_ERROR\_INVALID\_PTR, CPM\_SUCCESS, 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, Waitall(), X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 BndCommS3D(), BndCommS4D(), BndCommV3D(), と cpm\_BndCommS4D\_().

**4.7.2.36 cpm\_ErrorCode cpm\_ParaManager::BndCommS4D ( MPI\_Datatype dtype, void \* array, int imax, int jmax, int kmax, int nmax, int vc, int vc\_comm, int procGrpNo = 0 )**

袖通信 (Scalar4D 版, MPI\_Datatype 指定)

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

## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 539 行で定義されています。

参照先 BndCommS4D(), と CPM\_ERROR\_MPI\_INVALID\_DATATYPE.

**4.7.2.37 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, 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, 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()`.

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

非同期版袖通信 (Scalar4D 版, MPI\_Datatype 指定)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 601 行で定義されています。

参照先 BndCommS4D\_nowait(), と CPM\_ERROR\_MPI\_INVALID\_DATATYPE.

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

袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
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\_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, 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, Waitall(), X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 BndCommS4DEx(), BndCommV3DEx(), と cpm\_BndCommS4DEx\_().

**4.7.2.40** `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 778 行で定義されています。

参照先 BndCommS4DEx(), と CPM\_ERROR\_MPI\_INVALID\_DATATYPE.

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

非同期版袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の非同期袖通信を行う
- wait と展開は行わず、request を返す
- wait、展開は wait\_BndCommS4DEx をコールする

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndCommEx.h の 171 行で定義されています。

参照先 CPM\_ERROR\_BNDCOMM\_BUFFER, CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, CPM\_ERROR\_GET\_NEIGHBOR\_RANK, CPM\_ERROR\_INVALID\_PTR, CPM\_SUCCESS, 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, X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 BndCommS4DEx\_nowait(), BndCommV3DEx\_nowait(), と cpm\_BndCommS4DEx\_nowait().

4.7.2.42 `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 831 行で定義されています。

参照先 BndCommsS4DEx\_nowait(), と CPM\_ERROR\_MPI\_INVALID\_DATATYPE.

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

## 袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う

## 引数

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

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

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

参照先 BndCommsS4D().

参照元 cpm\_BndCommV3D\_().

4.7.2.44 `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 530 行で定義されています。

参照先 BndCommS4D().

4.7.2.45 `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().

4.7.2.46 `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 592 行で定義されています。

参照先 BndCommsS4D\_nowait().

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

袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndCommEx.h の 49 行で定義されています。

参照先 BndCommsS4DEx().

参照元 cpm\_BndCommV3DEx\_().

```
4.7.2.48 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 769 行で定義されています。

参照先 BndCommS4DEx().

4.7.2.49 `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().

4.7.2.50 `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 822 行で定義されています。

参照先 BndCommS4DEx\_nowait().

4.7.2.51 `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。

4.7.2.52 `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 2695 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(), と GetMPI\_Datatype().

参照元 cpm\_BndCommS3D\_nowait().

**4.7.2.53 cpm\_ErrorCode cpm\_ParaManager::cpm\_BndCommS4D\_nowait ( void \* array, int imax, int jmax, int kmax, int nmax, int vc, int vc\_comm, int datatype, int \* reqNo, int procGrpNo = 0 )**

cpm\_BndCommS4D\_nowait

- BndCommS4D\_nowait のインターフェイス
- Fortran インターフェイス用

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 2779 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(), と GetMPI\_Datatype().

参照元 cpm\_BndCommS3D\_nowait(), cpm\_BndCommS4D\_nowait(), と cpm\_BndCommV3D\_nowait().

**4.7.2.54 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	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 2993 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(), と GetMPI\_Datatype().

参照元 cpm\_BndComms4DEx\_nowait(), と cpm\_BndCommV3DEx\_nowait().

**4.7.2.55 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 2737 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(), と GetMPI\_Datatype().

参照元 cpm\_BndCommV3D\_nowait().

**4.7.2.56 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	<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 2951 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()`, と `GetMPI_Datatype()`.

参照元 `cpm_BndCommV3DEx_nowait_()`.

**4.7.2.57** `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 ( <code>cpm_fparam.fi</code> 参照)
in	<i>source</i>	送信元のランク番号 ( <code>procGrpNo</code> 内でのランク番号)
out	<i>reqNo</i>	リクエスト番号 (Fortran 用)
in	<i>procGrpNo</i>	プロセスグループ番号

## 戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2663 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()`, と `Irecv()`.

参照元 `cpm_Irecv_()`.

**4.7.2.58** `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 2626 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(), と Isend().

参照元 cpm\_Isend().

## 4.7.2.59 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 2562 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(), と cpm\_ObjList< T >::Get().

参照元 cpm\_Wait().

## 4.7.2.60 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 (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 2817 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(), と wait\_BndCommS3D().

参照元 cpm\_wait\_BndCommS3D\_().

**4.7.2.61 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 2909 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(), と wait\_BndCommS4D().

参照元 cpm\_wait\_BndCommS3D(), cpm\_wait\_BndCommS4D\_(), と cpm\_wait\_BndCommV3D().

**4.7.2.62 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\_frtlF.cpp の 3077 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(), と wait\_BndCommS4DEx().

参照元 cpm\_wait\_BndCommS4DEx(), と cpm\_wait\_BndCommV3DEx().

**4.7.2.63 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\_frtlF.cpp の 2863 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(), と wait\_BndCommV3D().

参照元 cpm\_wait\_BndCommV3D\_().

**4.7.2.64 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 ( <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 3031 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()`, と `wait_BndCommV3DEx()`.

参照元 `cpm_wait_BndCommV3DEx()`.

4.7.2.65 `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 2585 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()`, と `cpm_ObjList< T >::Get()`.

参照元 `cpm_Waitall()`.

4.7.2.66 `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 339 行で定義されています。

参照先 GetMPI\_Comm(), と cpm\_Base::IsCommNull().

**4.7.2.67** `const cpm_VoxelInfo * cpm_ParaManager::FindVoxelInfo ( int procGrpNo = 0 )`

VOXEL 空間マップを検索

引数

in	procGrpNo	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 381 行で定義されています。

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

**4.7.2.68** `void cpm_ParaManager::flush ( std::ostream & out, int procGrpNo = 0 )`

flush

cpm\_ParaManager.cpp の 669 行で定義されています。

参照先 Barrier().

参照元 Initialize().

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

flush

cpm\_ParaManager.cpp の 682 行で定義されています。

参照先 Barrier().

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

Gather

- MPI\_Gather のインターフェイス

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

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

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

参照元 `cpm_Gather_()`.

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

Gather

- MPI\_Gather のインターフェイス
- MPI\_Datatype を指定するバージョン

引数

in	<i>stype</i>	送信データのMPI_Datatype
in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>rtype</i>	受信データのMPI_Datatype
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	送信データのサイズ
in	<i>root</i>	受信するランク番号 (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 404 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()`.

**4.7.2.72** `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\_().

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 462 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().

**4.7.2.74 cpm\_ParaManager \* cpm\_ParaManager::get\_instance ( ) [static]**

唯一のインスタンスの取得

戻り値

インスタンスのポインタ

cpm\_ParaManager.cpp の 19 行で定義されています。

参照元 cpm\_Abort\_(), cpm\_Allgather\_(), cpm\_Allgatherv\_(), cpm\_Allreduce\_(), cpm\_Barrier\_(), cpm\_Bcast\_(), cpm\_BndCommS3D\_(), cpm\_BndCommS3D\_nowait\_(), cpm\_BndCommS4D\_(), cpm\_BndCommS4D\_nowait\_(), cpm\_BndCommS4DEx\_(), cpm\_BndCommS4DEx\_nowait\_(), cpm\_BndCommV3D\_(), cpm\_BndCommV3D\_nowait\_(), cpm\_BndCommV3DEx\_(), cpm\_BndCommV3DEx\_nowait\_(), cpm\_Gather\_(), cpm\_Gatherv\_(), cpm\_GetBCID\_(), cpm\_GetDivNum\_(), cpm\_GetDivPos\_(), cpm\_GetGlobalOrigin\_(), cpm\_GetGlobalRegion\_(), cpm\_GetGlobalVoxelSize\_(), cpm\_GetLocalOrigin\_(), cpm\_GetLocalRegion\_(), cpm\_GetLocalVoxelSize\_(), cpm\_GetMyRankID\_(), cpm\_GetNeighborRankID\_(), cpm\_GetNumRank\_(), cpm\_GetPeriodicRankID\_(), cpm\_GetPitch\_(), cpm\_GetVoxelHeadIndex\_(), cpm\_GetVoxelTailIndex\_(), cpm\_Initialize\_(), cpm\_Irecv\_(), cpm\_Isend\_(), cpm\_IsParallel\_(), cpm\_PeriodicCommS3D\_(), cpm\_PeriodicCommS4D\_(), cpm\_PeriodicCommS4DEx\_(), cpm\_PeriodicCommV3D\_(), cpm\_PeriodicCommV3DEx\_(), cpm\_Recv\_(), cpm\_Send\_(), cpm\_SetBndCommBuffer\_(), cpm-

\_Voxellnit\_(), cpm\_Voxellnit\_nodiv\_(), cpm\_Wait\_(), cpm\_wait\_BndCommS3D\_(), cpm\_wait\_BndCommS4D\_(), cpm\_wait\_BndCommS4DEx\_(), cpm\_wait\_BndCommV3D\_(), cpm\_wait\_BndCommV3DEx\_(), と cpm\_Waitall\_().

#### 4.7.2.75 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().

#### 4.7.2.76 const int \* cpm\_ParaManager::GetBCID ( int procGrpNo = 0 )

自ランクのBCID 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BCID 整数配列のポインタ

cpm\_ParaManager.cpp の 499 行で定義されています。

参照先 FindVoxellInfo().

参照元 cpm\_GetBCID\_().

#### 4.7.2.77 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 626 行で定義されています。

参照先 S\_BNDCOMM\_BUFFER::m\_nwX, S\_BNDCOMM\_BUFFER::m\_nwY, S\_BNDCOMM\_BUFFER::m\_nwZ, と REAL\_BUF\_TYPE.

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

領域分割数を取得

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

領域分割数整数配列のポインタ

cpm\_ParaManager.cpp の 391 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetDivNum\_().

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

自ランクの領域分割位置を取得

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

自ランクの領域分割位置整数配列のポインタ

cpm\_ParaManager.cpp の 487 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetDivPos\_().

**4.7.2.80** `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 427 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetGlobalOrigin\_().

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

全体空間サイズを取得

引数

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

戻り値

全体空間サイズ実数配列のポインタ

cpm\_ParaManager.cpp の 439 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetGlobalRegion\_().

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

全体ボクセル数を取得

引数

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

戻り値

全体ボクセル数の整数配列ポインタ (3word)

cpm\_ParaManager.cpp の 415 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetGlobalVoxelSize\_().

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

自ランクの空間原点を取得

引数

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

戻り値

自ランクの空間原点実数配列のポインタ

cpm\_ParaManager.cpp の 463 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetLocalOrigin\_(), と cpm\_GetLocalRegion\_().

**4.7.2.84** `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 475 行で定義されています。

参照先 FindVoxelInfo().

**4.7.2.85** `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 451 行で定義されています。

参照先 FindVoxelInfo().

参照元 AllocDoubleS4D(), AllocFloatS4D(), AllocIntS4D(), AllocRealS4D(), cpm\_GetLocalVoxelSize(), と SetBndCommBuffer().

**4.7.2.86** `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

cpm\_ParaManager\_MPI.cpp の 131 行で定義されています。

参照先 cpm\_Base::getCommNull().

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

**4.7.2.87** `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()`.

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

`MPI_Datatype` を取得

- Fortran データタイプから `MPI_Datatype` を取得

引数

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

戻り値

`MPI_Datatype`

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

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

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

`MPI_Op` を取得

- Fortran オペレータタイプから `MPI_Op` を取得

引数

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

戻り値

`MPI_Op`

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

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

参照元 `cpm_Allreduce_()`.

#### 4.7.2.90 `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	<i>procGrpNo</i>	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(), と cpm\_Base::IsCommNull().

参照元 cpm\_GetMyRankID\_().

#### 4.7.2.91 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 535 行で定義されています。

参照先 FindVoxellInfo().

参照元 BndCommS4D(), BndCommS4D\_nowait(), BndCommS4DEx(), BndCommS4DEx\_nowait(), cpm\_GetNeighborRankID\_(), wait\_BndCommS4D(), と wait\_BndCommS4DEx().

#### 4.7.2.92 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	<i>procGrpNo</i>	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().

参照元 cpm\_GetNumRank\_(), と Voxellnit().

#### 4.7.2.93 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 547 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetPeriodicRankID\_(), PeriodicComms4D(), と PeriodicComms4DEx().

#### 4.7.2.94 `const REAL_TYPE * cpm_ParaManager::GetPitch ( int procGrpNo = 0 )`

ピッチを取得

引数

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

戻り値

ピッチ実数配列のポインタ

cpm\_ParaManager.cpp の 403 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetPitch\_().

#### 4.7.2.95 `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 511 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetVoxelHeadIndex\_().

#### 4.7.2.96 `const int * cpm_ParaManager::GetVoxelTailIndex ( int procGrpNo = 0 )`

自ランクの終点VOXEL の全体空間でのインデクスを取得

- ・ 全体空間の先頭インデクスを 0 としたC 型のインデクス

戻り値

自ランクの終点インデクス整数配列のポインタ

cpm\_ParaManager.cpp の 523 行で定義されています。

参照先 FindVoxelInfo().

参照元 cpm\_GetVoxelTailIndex\_().

#### 4.7.2.97 `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。

#### 4.7.2.98 cpm\_ErrorCode cpm\_ParaManager::Initialize ( )

初期化処理 (MPI\_Init は実行済みの場合)

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager.cpp の 117 行で定義されています。

参照先 CPM\_ERROR\_MPI, CPM\_ERROR\_NO\_MPI\_INIT, CPM\_SUCCESS, flush(), と IsParallel().

参照元 cpm\_Initialize(), get\_instance(), と Initialize().

#### 4.7.2.99 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 163 行で定義されています。

参照先 CPM\_ERROR\_MPI, と Initialize().

#### 4.7.2.100 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().

4.7.2.101 **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().

4.7.2.102 **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().

**4.7.2.103** `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().

**4.7.2.104** `bool cpm_ParaManager::IsParallel ( )`

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

戻り値

<i>true</i>	並列実行
<i>false</i>	逐次実行

cpm\_ParaManager.cpp の 187 行で定義されています。

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

**4.7.2.105** `bool cpm_ParaManager::IsParallel ( ) const`

並列実行であるかチェックする (const)

- 並列実行であっても、並列数が 1 のときは false となる

戻り値

<i>true</i>	並列実行
<i>false</i>	逐次実行

cpm\_ParaManager.cpp の 199 行で定義されています。

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

周期境界袖通信 (Scalar3D 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 372 行で定義されています。

参照先 PeriodicCommS4D().

参照元 cpm\_PeriodicCommS3D().

**4.7.2.107** `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 707 行で定義されています。

参照先 PeriodicCommS4D().

**4.7.2.108** `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, 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, PLUS2MINUS, 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().

**4.7.2.109 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	<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>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 725 行で定義されています。

参照先 CPM\_ERROR\_MPI\_INVALID\_DATATYPE, と PeriodicCommS4D().



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

周期境界袖通信 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndCommEx.h の 358 行で定義されています。

参照先 BOTH, CPM\_ERROR\_BNDCOMM\_BUFFER, CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, CPM\_ERROR\_GET\_PERIODIC\_RANK, CPM\_ERROR\_INVALID\_PTR, CPM\_SUCCESS, 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, PLUS2MINUS, 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().

4.7.2.111 `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	<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>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 937 行で定義されています。

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

4.7.2.112 `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

	<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 382 行で定義されています。

参照先 PeriodicCommsS4D().

参照元 cpm\_PeriodicCommV3D\_().

4.7.2.113 `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 716 行で定義されています。

参照先 PeriodicComms4D().

4.7.2.114 `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\_().

4.7.2.115 `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 928 行で定義されています。

参照先 PeriodicComms4DEx().

4.7.2.116 `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\_().

4.7.2.117 `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().

4.7.2.118 `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().

4.7.2.119 `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().

4.7.2.120 `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

Cartesian Partition Manager に対してFri Jun 1 2012 13:33:39 に生成されました。 Doxygen

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager.cpp の 559 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(), 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 Voxellnit().

4.7.2.121 **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 211 行で定義されています。

参照先 Abort(), CPM\_ERROR\_ALREADY\_VOXELINIIT, CPM\_ERROR\_INSERT\_VOXELMAP, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN, CPM\_ERROR\_MPI\_INVALID\_COMM, CPM\_SUCCESS, GetMPI\_Comm(), cpm\_GlobalDomainInfo::GetSubDomainNum(), cpm\_Base::IsCommNull(), と SetBndCommBuffer().

参照元 cpm\_Voxellnit(), cpm\_Voxellnit\_nodiv(), と Voxellnit().

4.7.2.122 **cpm\_ErrorCode cpm\_ParaManager::Voxellnit ( int div[3], int vox[3], REAL\_TYPE origin[3], REAL\_TYPE pitch[3], int obcid[6], size\_t maxVC = 1, size\_t maxN = 3, int procGrpNo = 0 )**

領域分割

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 279 行で定義されています。

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

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

領域分割

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

引数

in	<i>vox</i>	空間全体のボクセル数
in	<i>origin</i>	空間全体の原点
in	<i>pitch</i>	ボクセルピッチ
in	<i>obcid</i>	全体空間の外部境界条件ID
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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 319 行で定義されています。

参照先 `CPM_ERROR`, `GetNumRank()`, と `Voxellnit()`.

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

Wait

- ・ `MPI_Wait` のインターフェイス

引数

in	<i>request</i>	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`.

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

非同期版袖通信の wait、展開 (Scalar3D 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 274 行で定義されています。

参照先 wait\_BndCommsS4D().

参照元 cpm\_wait\_BndCommsS3D().

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

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

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 645 行で定義されています。

参照先 wait\_BndCommsS4D().



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

非同期版袖通信の wait、展開 (Scalar4D 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 294 行で定義されています。

参照先 CPM\_ERROR\_BNDCOMM\_BUFFER, CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, CPM\_ERROR\_GET\_NEIGHBOR\_RANK, CPM\_ERROR\_INVALID\_PTR, CPM\_SUCCESS, 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, Waitall(), X\_MINUS, X\_PLUS, Y\_MINUS, Y\_PLUS, Z\_MINUS, と Z\_PLUS.

参照元 cpm\_wait\_BndCommsS4D(), wait\_BndCommsS3D(), wait\_BndCommsS4D(), と wait\_BndCommV3D().

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

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

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 663 行で定義されています。

参照先 CPM\_ERROR\_MPI\_INVALID\_DATATYPE, と wait\_BndCommS4D().

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

非同期版袖通信の wait、展開 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndCommEx.h の 270 行で定義されています。

参照先 CPM\_ERROR\_BNDCOMM\_BUFFER, CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, CPM\_ERROR\_GET\_NEIGHBOR\_RANK, CPM\_ERROR\_INVALID\_PTR, CPM\_SUCCESS, 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, 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().

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

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

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 884 行で定義されています。

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

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

非同期版袖通信の wait、展開 (Vector3D 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndComm.h の 284 行で定義されています。

参照先 wait\_BndCommS4D().

参照元 cpm\_wait\_BndCommV3D().

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

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

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_MPI.cpp の 654 行で定義されています。

参照先 wait\_BndComms4D().

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

非同期版袖通信の wait、展開 (Vector3DEx 版)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

cpm\_ParaManager\_BndCommEx.h の 260 行で定義されています。

参照先 wait\_BndCommS4DEx().

参照元 cpm\_wait\_BndCommV3DEx().

4.7.2.134 `cpm_ErrorCode cpm_ParaManager::wait_BndCommV3DEx ( MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0 )`

非同期版袖通信の wait、展開 (Vector3DEx 版, MPI\_Datatype 指定)

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

引数

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

戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 875 行で定義されています。

参照先 `wait_BndComms4DEx()`.

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

Waitall

- `MPI_Waitall` のインターフェイス

引数

in	<i>count</i>	リクエストの数
in	<i>requests</i>	リクエストハンドル配列

戻り値

終了コード (CPM\_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 201 行で定義されています。

参照先 `CPM_ERROR_MPI_WAITALL`, と `CPM_SUCCESS`.

参照元 `BndComms4D()`, `BndComms4DEx()`, `PeriodicComms4D()`, `PeriodicComms4DEx()`, `wait_BndComms4D()`, と `wait_BndComms4DEx()`.

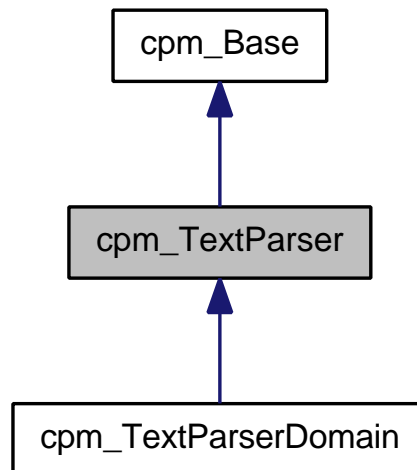
このクラスの説明は次のファイルから生成されました:

- [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](#)

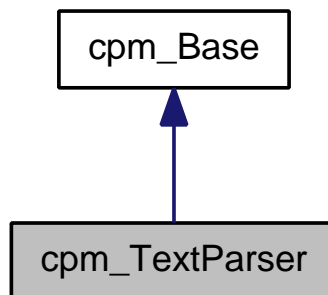
## 4.8 クラス `cpm_TextParser`

```
#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	filename	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.

### 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	label	ベクトルデータのテキストラベル
out	vec	読み込んだベクトルデータ (サイズは nvec 確保されている必要がある)
in	nvec	読み込んだベクトルデータの数

#### 戻り値

1000 未満	テキストパーサのエラーコード
CPM_ERROR_TP_NOVECTOR(2001)	指定ラベルがベクトルデータではない
CPM_ERROR_TP_VECTOR_SIZE(2002)	ベクトルデータのサイズが nvec 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.

#### 4.8.3.3 `int cpm_TextParser::readVector ( std::string label, double * vec, const int nvec )` [protected]

ベクトルデータの読み込み (倍精度実数版)

引数

in	label	ベクトルデータのテキストラベル
out	vec	読み込んだベクトルデータ (サイズは nvec 確保されている必要がある)
in	nvec	読み込んだベクトルデータの数

戻り値

1000 未満	テキストパーサのエラーコード
CPM_ERROR_TP_NOVECTOR(2001)	指定ラベルがベクトルデータではない
CPM_ERROR_TP_VECTOR_SIZE(2002)	ベクトルデータのサイズが nvec と一致しない

cpm\_TextParser.cpp の 88 行で定義されています。

参照先 CPM\_ERROR\_TP\_NOVECTOR, CPM\_ERROR\_TP\_VECTOR\_SIZE, と m\_tp.

#### 4.8.3.4 `int cpm_TextParser::readVector ( std::string label, int * vec, const int nvec )` [protected]

ベクトルデータの読み込み (整数版)

引数

in	label	ベクトルデータのテキストラベル
out	vec	読み込んだベクトルデータ (サイズは nvec 確保されている必要がある)
in	nvec	読み込んだベクトルデータの数

戻り値

1000 未満	テキストパーサのエラーコード
CPM_ERROR_TP_NOVECTOR(2001)	指定ラベルがベクトルデータではない
CPM_ERROR_TP_VECTOR_SIZE(2002)	ベクトルデータのサイズが nvec 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(), 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:

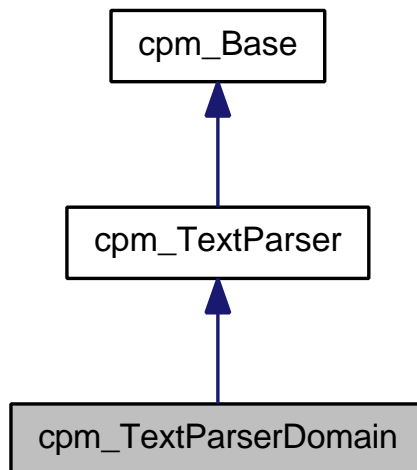
- [cpm\\_TextParser.h](#)
- [cpm\\_TextParser.cpp](#)



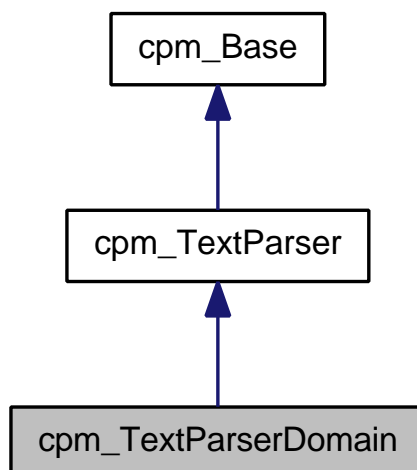
## 4.9 クラス `cpm_TextParserDomain`

```
#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)`

#### 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	filename	読み込むファイル名
out	errorcode	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。

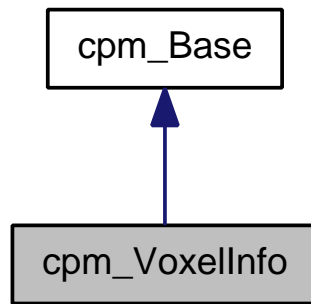
このクラスの説明は次のファイルから生成されました:

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

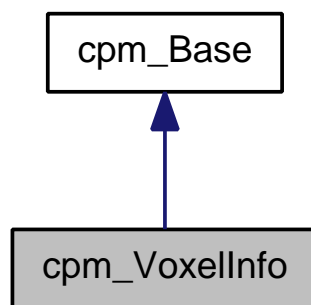
## 4.10 クラス cpm\_VoxelInfo

```
#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



## 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 フレンドと関連する関数

4.10.2.1 `friend class cpm_ParaManager [friend]`

cpm\_VoxelInfo.h の 25 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\\_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](#)

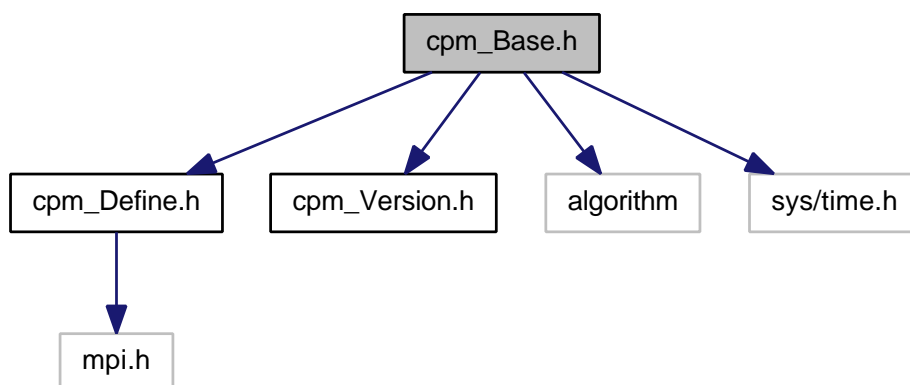
## Chapter 5

# ファイル

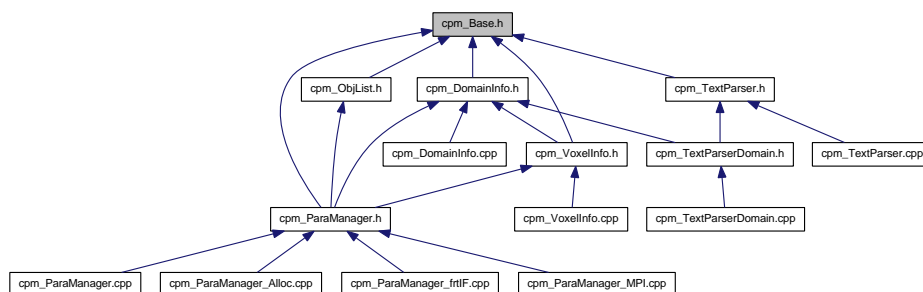
### 5.1 cpm\_Base.h

```
#include "cpm_Define.h"
#include "cpm_Version.h"
#include <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

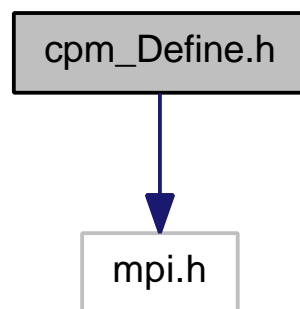
#### 5.1.2.1 `#define CPM_INLINE inline`

`cpm_Base.h` の 34 行で定義されています。

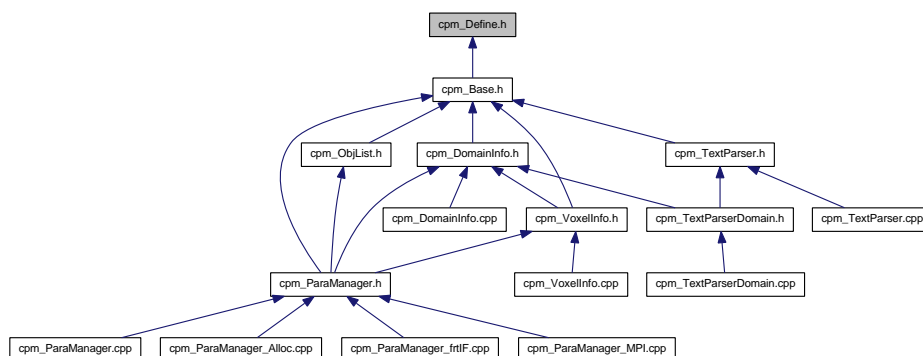
## 5.2 `cpm_Define.h`

```
#include "mpi.h"
```

`cpm_Define.h` のインクルード依存関係図



このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。





## マクロ定義

- #define REAL\_TYPE float
- #define REAL\_BUF\_TYPE REAL\_TYPE
- #define \_IDX\_S3D(\_I, \_J, \_K, \_NI, \_NJ, \_NK, \_VC)
- #define \_IDX\_S4D(\_I, \_J, \_K, \_N, \_NI, \_NJ, \_NK, \_VC)
- #define \_IDX\_V3D(\_I, \_J, \_K, \_N, \_NI, \_NJ, \_NK, \_VC) (\_IDX\_S4D(\_I, \_J, \_K, \_N, \_NI, \_NJ, \_NK, \_VC))
- #define \_IDX\_S4DEX(\_N, \_I, \_J, \_K, \_NN, \_NI, \_NJ, \_NK, \_VC)
- #define \_IDX\_V3DEX(\_N, \_I, \_J, \_K, \_NI, \_NJ, \_NK, \_VC) (\_IDX\_S4DEX(\_N, \_I, \_J, \_K, \_NI, \_NJ, \_NK, \_VC))

## 列挙型

- enum cpm\_FaceFlag {  
X\_MINUS = 0, Y\_MINUS = 1, Z\_MINUS = 2, X\_PLUS = 3,  
Y\_PLUS = 4, Z\_PLUS = 5 }
- enum cpm\_DirFlag { X\_DIR = 0, Y\_DIR = 1, Z\_DIR = 2 }
- enum cpm\_PMFlag { PLUS2MINUS = 0, MINUS2PLUS = 1, BOTH = 2 }
- enum cpm\_ErrorCode {  
CPM\_SUCCESS = 0, CPM\_ERROR = 1000, CPM\_ERROR\_PM\_INSTANCE = 1001, CPM\_ERROR\_INVALID\_PTR = 1002,  
CPM\_ERROR\_INVALID\_DOMAIN\_NO = 1003, CPM\_ERROR\_INVALID\_OBJKEY = 1004, CPM\_ERROR\_REGIST\_OBJKEY = 1005, CPM\_ERROR\_TEXTPARSER = 2000,  
CPM\_ERROR\_NO\_TEXTPARSER = 2001, CPM\_ERROR\_TP\_NOVECTOR = 2002, CPM\_ERROR\_TP\_VECTOR\_SIZE = 2003, CPM\_ERROR\_TP\_INVALID\_G\_ORG = 2004,  
CPM\_ERROR\_TP\_INVALID\_G\_VOXEL = 2005, CPM\_ERROR\_TP\_INVALID\_G\_PITCH = 2006,  
CPM\_ERROR\_TP\_INVALID\_G\_RGN = 2007, CPM\_ERROR\_TP\_INVALID\_G\_DIV = 2008,  
CPM\_ERROR\_TP\_INVALID\_POS = 2009, CPM\_ERROR\_TP\_INVALID\_BCID = 2010, CPM\_ERROR\_VOXELINIT = 3000, CPM\_ERROR\_NOT\_IN\_PROCGROUP = 3001,  
CPM\_ERROR\_ALREADY\_VOXELINIT = 3002, CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN = 3003, CPM\_ERROR\_CREATE\_RANKMAP = 3004, CPM\_ERROR\_CREATE\_NEIGHBOR = 3005,  
CPM\_ERROR\_CREATE\_LOCALDOMAIN = 3006, CPM\_ERROR\_INSERT\_VOXELMAP = 3007, CPM\_ERROR\_CREATE\_PROCGROUP = 3008, CPM\_ERROR\_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\_BCID = 4010, CPM\_ERROR\_GET\_HEADINDEX = 4011, CPM\_ERROR\_GET\_TAILINDEX = 4012,  
CPM\_ERROR\_GET\_NEIGHBOR\_RANK = 4013, CPM\_ERROR\_GET\_PERIODIC\_RANK = 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\_BNDCOMM = 9500,  
CPM\_ERROR\_BNDCOMM\_VOXELSIZE = 9501, CPM\_ERROR\_BNDCOMM\_BUFFER = 9502, CPM\_ERROR\_BNDCOMM\_E = 9503, CPM\_ERROR\_PERIODIC = 9600,  
CPM\_ERROR\_PERIODIC\_INVALID\_DIR = 9601, CPM\_ERROR\_PERIODIC\_INVALID\_PM = 9602, CPM\_ERROR\_MPI\_INVALID\_COMM = 9100, CPM\_ERROR\_MPI\_INVALID\_DATATYPE = 9101,  
CPM\_ERROR\_MPI\_INVALID\_OPERATOR = 9102, CPM\_ERROR\_MPI\_INVALID\_REQUEST = 9103 }
- enum CPM\_Datatype {  
CPM\_CHAR = 1, CPM\_UNSIGNED\_CHAR = 2, CPM\_BYTE = 3, CPM\_SHORT = 4,  
CPM\_UNSIGNED\_SHORT = 5, CPM\_INT = 6, CPM\_UNSIGNED = 7, CPM\_LONG = 8,  
CPM\_UNSIGNED\_LONG = 9, CPM\_FLOAT = 10, CPM\_DOUBLE = 11, CPM\_LONG\_DOUBLE = 12,  
CPM\_REAL = 52 }

```

• enum CPM_Op {
    CPM_MAX = 100, CPM_MIN = 101, CPM_SUM = 102, CPM_PROD = 103,
    CPM_LAND = 104, CPM_BAND = 105, CPM_LOR = 106, CPM_BOR = 107,
    CPM_LXOR = 108, CPM_BXOR = 109, CPM_MINLOC = 110, CPM_MAXLOC = 111 }

```

## 5.2.1 説明

CPM の定義マクロ記述ヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm\\_Define.h](#) で定義されています。

## 5.2.2 マクロ定義

### 5.2.2.1 #define \_IDX\_S3D( \_I, \_J, \_K, \_NI, \_NJ, \_NK, \_VC )

値:

```

( size_t(_K+_VC) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)

```

3次元インデクス (i,j,k) -> 1次元インデクス変換マクロ

引数

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

戻り値

1次元インデクス

[cpm\\_Define.h](#) の 56 行で定義されています。

### 5.2.2.2 #define \_IDX\_S4D( \_I, \_J, \_K, \_N, \_NI, \_NJ, \_NK, \_VC )

値:

```

( size_t(_N) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) * size_t(_NK+2*_VC) \
+ _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC) \
)

```

4次元インデクス (i,j,k,n) -> 1次元インデクス変換マクロ

引数

in	<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

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。

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	<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>_NN</code>	成分数
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cpm\_Define.h の 102 行で定義されています。

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) -&gt; 1 次元インデクス変換マクロ

引数

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

cpm\_Define.h の 88 行で定義されています。

5.2.2.5 `#define _IDX_V3DEX( _N, _I, _J, _K, _NI, _NJ, _NK, _VC ) ( _IDX_S4DEX( _N, _I, _J, _K, 3, _NI, _NJ, _NK, _VC ) )`

3次元インデクス (3,i,j,k) -> 1次元インデクス変換マクロ

引数

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

cpm\_Define.h の 116 行で定義されています。

5.2.2.6 `#define REAL_BUF_TYPE REAL_TYPE`

袖通信バッファの型指定

- デフォルトでは、`REAL_BUF_TYPE=REAL_TYPE`
- コンパイル時オプション-D\_BUFSIZE\_DOUBLE\_を付与することで `REAL_BUF_TYPE=double` になる
- コンパイル時オプション-D\_BUFSIZE\_LONG\_DOUBLE\_を付与することで `REAL_BUF_TYPE=long double` になる

cpm\_Define.h の 42 行で定義されています。

参照元 `cpm_ParaManager::GetBndCommBufferSize()`, と `cpm_ParaManager::SetBndCommBuffer()`.

5.2.2.7 `#define REAL_TYPE float`

実数型の指定

- デフォルトでは、`REAL_TYPE=float`
- コンパイル時オプション-D\_REAL\_IS\_DOUBLE\_を付与することで `REAL_TYPE=double` になる

cpm\_Define.h の 27 行で定義されています。

参照元 `cpm_ParaManager::AllocRealS4D()`, `cpm_GetGlobalOrigin_()`, `cpm_GetGlobalRegion_()`, `cpm_GetLocalOrigin_()`, `cpm_GetLocalRegion_()`, `cpm_GetPitch_()`, `cpm_Base::ReallsDouble()`, と `cpm_ParaManager::VoxelInit()`.

## 5.2.3 列挙型

### 5.2.3.1 enum CPM\_Datatype

fortran 用のデータタイプ

列挙型の値:

**CPM\_CHAR** char

**CPM\_UNSIGNED\_CHAR** unsigned char  
**CPM\_BYTE** byte(not support)  
**CPM\_SHORT** short  
**CPM\_UNSIGNED\_SHORT** unsigned short  
**CPM\_INT** int  
**CPM\_UNSIGNED** unsigned  
**CPM\_LONG** long  
**CPM\_UNSIGNED\_LONG** unsigned long  
**CPM\_FLOAT** float  
**CPM\_DOUBLE** double  
**CPM\_LONG\_DOUBLE** long double  
**CPM\_REAL** REAL\_TYPE.

cpm\_Define.h の 230 行で定義されています。

### 5.2.3.2 enum cpm\_DirFlag

軸方向フラグ

列挙型の値:

**X\_DIR** X direction.  
**Y\_DIR** Y direction.  
**Z\_DIR** Z direction.

cpm\_Define.h の 130 行で定義されています。

### 5.2.3.3 enum cpm\_ErrorCode

CPM のエラーコード

列挙型の値:

**CPM\_SUCCESS** 正常終了  
**CPM\_ERROR** その他のエラー  
**CPM\_ERROR\_PM\_INSTANCE** 並列管理クラス cpm\_ParaManager のインスタンス失敗  
**CPM\_ERROR\_INVALID\_PTR** ポインタのエラー  
**CPM\_ERROR\_INVALID\_DOMAIN\_NO** 領域番号が不正  
**CPM\_ERROR\_INVALID\_OBJKEY** 指定登録番号のオブジェクトが存在しない  
**CPM\_ERROR\_REGIST\_OBJKEY** オブジェクト登録に失敗:  
**CPM\_ERROR\_TEXTPARSER** テキストパーサーに関するエラー  
**CPM\_ERROR\_NO\_TEXTPARSER** テキストパーサーを組み込んでいない  
**CPM\_ERROR\_TP\_NOVECTOR** 領域分割情報ファイルのベクトルデータ読み込みエラー  
**CPM\_ERROR\_TP\_VECTOR\_SIZE** 領域分割情報ファイルのベクトルデータのサイズが不正  
**CPM\_ERROR\_TP\_INVALID\_G\_ORG** 領域分割情報ファイルのドメイン原点情報が不正  
**CPM\_ERROR\_TP\_INVALID\_G\_VOXEL** 領域分割情報ファイルのドメインVOXEL 数情報が不正  
**CPM\_ERROR\_TP\_INVALID\_G\_PITCH** 領域分割情報ファイルのドメインピッチ情報が不正  
**CPM\_ERROR\_TP\_INVALID\_G\_RGN** 領域分割情報ファイルのドメイン空間サイズ情報が不正  
**CPM\_ERROR\_TP\_INVALID\_G\_DIV** 領域分割情報ファイルのドメイン領域分割数情報が不正

**CPM\_ERROR\_TP\_INVALID\_POS** 領域分割情報ファイルのサブドメイン位置情報が不正  
**CPM\_ERROR\_TP\_INVALID\_BCID** 領域分割情報ファイルのサブドメインBCID 情報が不正  
**CPM\_ERROR\_VOXELINIT** Voxellnit でエラー  
**CPM\_ERROR\_NOT\_IN\_PROCGROUP** 自ランクがプロセスグループに含まれていない  
**CPM\_ERROR\_ALREADY\_VOXELINIT** 指定されたプロセスグループが既に領域分割済み:  
**CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN** 並列数とサブドメイン数が一致していない  
**CPM\_ERROR\_CREATE\_RANKMAP** ランクマップ生成に失敗  
**CPM\_ERROR\_CREATE\_NEIGHBOR** 隣接ランク情報生成に失敗  
**CPM\_ERROR\_CREATE\_LOCALDOMAIN** ローカル領域情報生成に失敗  
**CPM\_ERROR\_INSERT\_VOXELMAP** 領域情報のマップへの登録失敗  
**CPM\_ERROR\_CREATE\_PROCGROUP** プロセスグループ生成に失敗  
**CPM\_ERROR\_GET\_INFO** 情報取得系関数でエラー  
**CPM\_ERROR\_GET\_DIVNUM** 領域分割数の取得エラー  
**CPM\_ERROR\_GET\_PITCH** ピッチの取得エラー  
**CPM\_ERROR\_GET\_GLOBALVOXELSIZE** 全体ボクセル数の取得エラー  
**CPM\_ERROR\_GET\_GLOBALORIGIN** 全体空間の原点の取得エラー  
**CPM\_ERROR\_GET\_GLOBALREGION** 全体空間サイズの取得エラー  
**CPM\_ERROR\_GET\_LOCALVOXELSIZE** 自ランクのボクセル数の取得エラー  
**CPM\_ERROR\_GET\_LOCALORIGIN** 自ランクの空間原点の取得エラー  
**CPM\_ERROR\_GET\_LOCALREGION** 自ランクの空間サイズの取得エラー  
**CPM\_ERROR\_GET\_DIVPOS** 自ランクの領域分割位置の取得エラー  
**CPM\_ERROR\_GET\_BCID** 自ランクのBCID の取得エラー  
**CPM\_ERROR\_GET\_HEADINDEX** 始点インデックスの取得エラー  
**CPM\_ERROR\_GET\_TAILINDEX** 終点インデックスの取得エラー  
**CPM\_ERROR\_GET\_NEIGHBOR\_RANK** 隣接ランク番号の取得エラー  
**CPM\_ERROR\_GET\_PERIODIC\_RANK** 周期境界位置の隣接ランク番号の取得エラー  
**CPM\_ERROR\_GET\_MYRANK** ランク番号の取得エラー  
**CPM\_ERROR\_GET\_NUMRANK** ランク数の取得エラー  
**CPM\_ERROR\_MPI** MPI のエラー  
**CPM\_ERROR\_NO\_MPI\_INIT** MPI\_Init がコールされていない  
**CPM\_ERROR\_MPI\_BARRIER** MPI\_Barrier でエラー  
**CPM\_ERROR\_MPI\_BCAST** MPI\_Bcast でエラー  
**CPM\_ERROR\_MPI\_SEND** MPI\_Send でエラー  
**CPM\_ERROR\_MPI\_RECV** MPI\_Recv でエラー  
**CPM\_ERROR\_MPI\_ISEND** MPI\_Isend でエラー  
**CPM\_ERROR\_MPI\_Irecv** MPI\_Irecv でエラー  
**CPM\_ERROR\_MPI\_WAIT** MPI\_Wait でエラー  
**CPM\_ERROR\_MPI\_WAITALL** MPI\_Waitall でエラー  
**CPM\_ERROR\_MPI\_ALLREDUCE** MPI\_Allreduce でエラー  
**CPM\_ERROR\_MPI\_GATHER** MPI\_Gather でエラー  
**CPM\_ERROR\_MPI\_ALLGATHER** MPI\_Allgather でエラー  
**CPM\_ERROR\_MPI\_GATHERV** MPI\_Gatherv でエラー  
**CPM\_ERROR\_MPI\_ALLGATHERV** MPI\_Allgatherv でエラー  
**CPM\_ERROR\_BNDCOMM** BndComm でエラー  
**CPM\_ERROR\_BNDCOMM\_VOXELSIZE** VoxelSize 取得でエラー

**CPM\_ERROR\_BNDCOMM\_BUFFER** 袖通信バッファ取得でエラー  
**CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH** 袖通信バッファサイズが足りない  
**CPM\_ERROR\_PERIODIC** PeriodicComm でエラー  
**CPM\_ERROR\_PERIODIC\_INVALID\_DIR** 不正な軸方向フラグが指定された  
**CPM\_ERROR\_PERIODIC\_INVALID\_PM** 不正な正負方向フラグが指定された  
**CPM\_ERROR\_MPI\_INVALID\_COMM** MPI コミュニケータが不正  
**CPM\_ERROR\_MPI\_INVALID\_DATATYPE** 対応しない型が指定された  
**CPM\_ERROR\_MPI\_INVALID\_OPERATOR** 対応しないオペレータが指定された  
**CPM\_ERROR\_MPI\_INVALID\_REQUEST** 不正なリクエストが指定された

cpm\_Define.h の 146 行で定義されています。

#### 5.2.3.4 enum cpm\_FaceFlag

面フラグ

列挙型の値:

**X\_MINUS** -X face  
**Y\_MINUS** -Y face  
**Z\_MINUS** -Z face  
**X\_PLUS** +X face  
**Y\_PLUS** +Y face  
**Z\_PLUS** +Z face

cpm\_Define.h の 119 行で定義されています。

#### 5.2.3.5 enum CPM\_Op

fortran 用のオペレータ

列挙型の値:

**CPM\_MAX** 最大値  
**CPM\_MIN** 最小値  
**CPM\_SUM** 和  
**CPM\_PROD** 積  
**CPM\_LAND** 論理積  
**CPM\_BAND** ビット演算の積  
**CPM\_LOR** 論理和  
**CPM BOR** ビット演算の和  
**CPM\_LXOR** 排他的論理和  
**CPM\_BXOR** ビット演算の排他的論理和  
**CPM\_MINLOC** 最大値と位置 (not support)  
**CPM\_MAXLOC** 最小値と位置 (not support)

cpm\_Define.h の 257 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

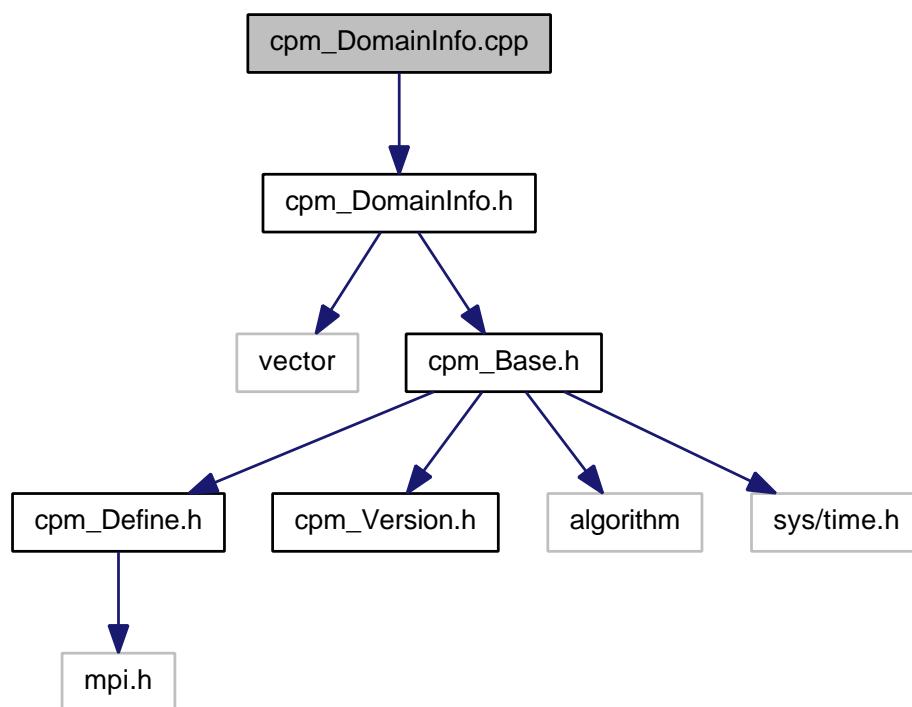
**BOTH** plus <-> minus direction

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。

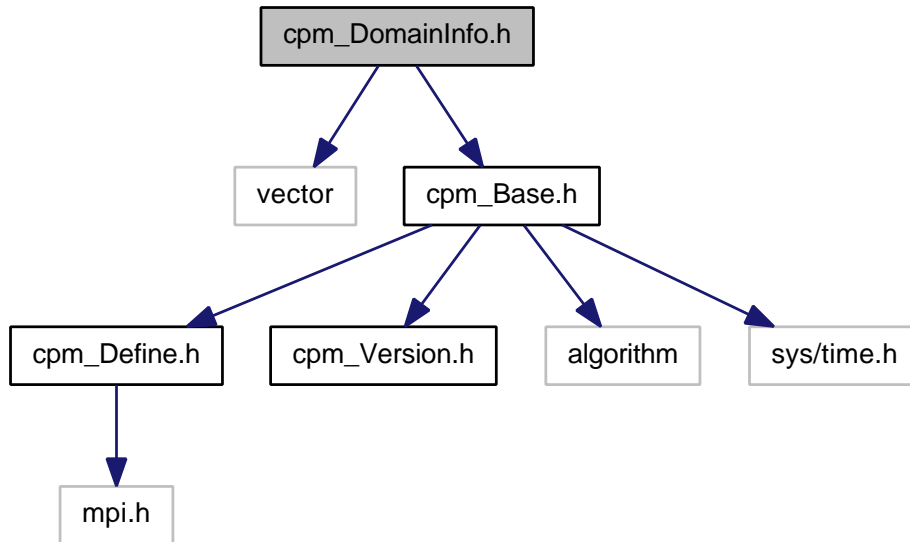


## 5.4 cpm\_DomainInfo.h

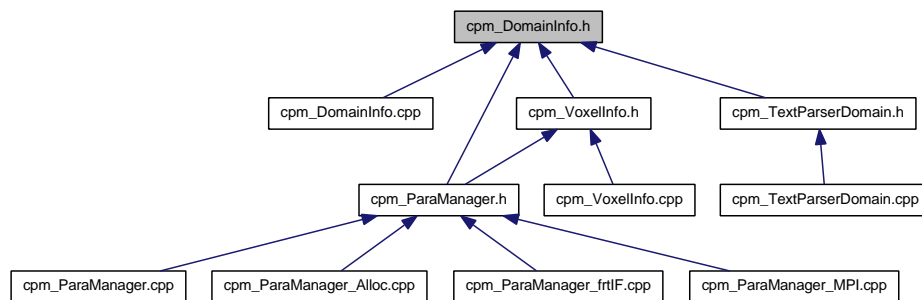
```
#include <vector>
```

```
#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

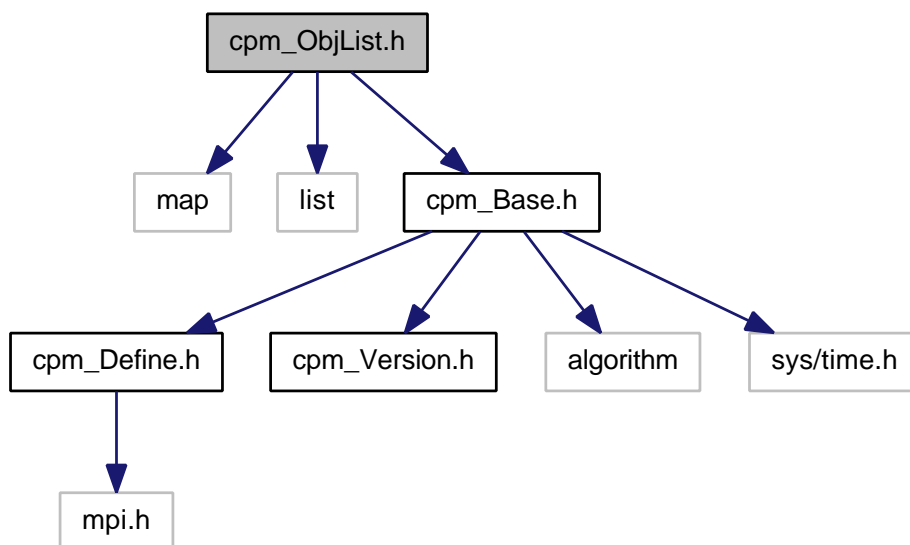
2012/05/31

[cpm\\_DomainInfo.h](#) で定義されています。

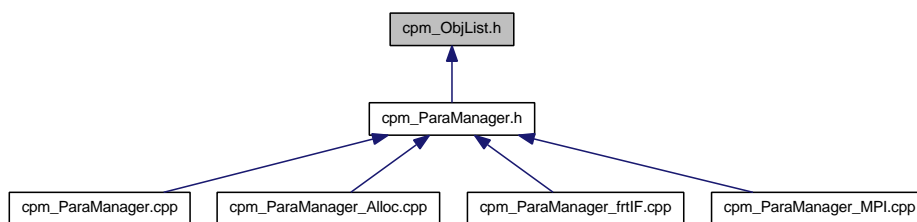
## 5.5 cpm\_ObjList.h

```
#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

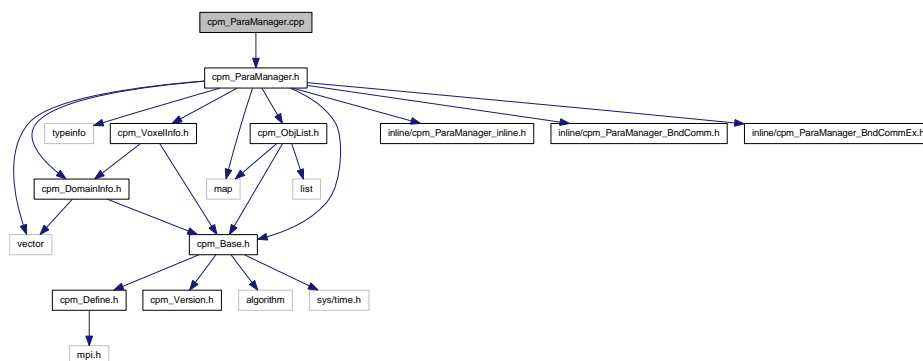
プロセスグループ毎のランク番号マップ

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

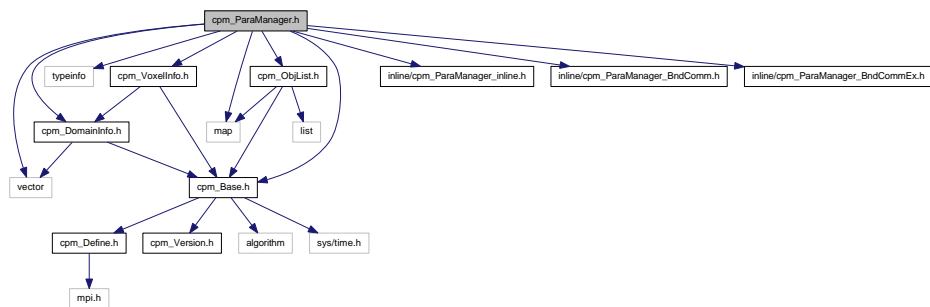
2012/05/31

[cpm\\_ParaManager.cpp](#) で定義されています。

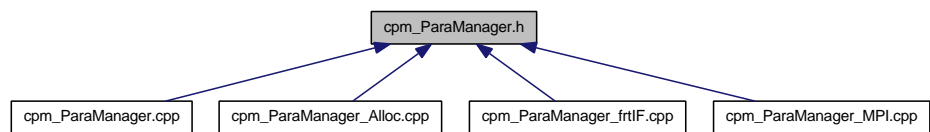
## 5.7 cpm\_ParaManager.h

```
#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



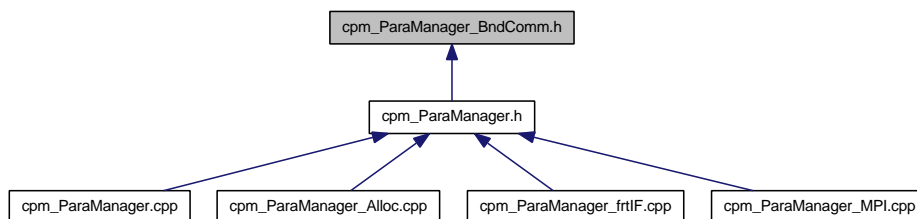
日付

2012/05/31

[cpm\\_ParaManager\\_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。

## 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。

## 5.9.2.3 #define \_IDXFZ( \_I, \_J, \_K, \_N, \_NI, \_NJ, \_KS, \_VC )

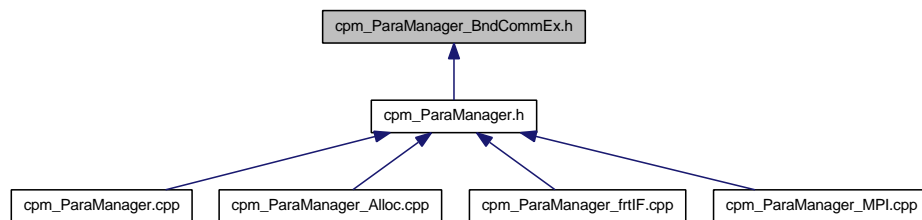
値:

```
( 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。

## 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 [\\_IDXFX](#)(\_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 _IDXFX( _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。

### 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。

### 5.10.2.3 `#define _IDXFZ( _N, _I, _J, _K, _NN, _NI, _NJ, _KS, _VC )`

値:

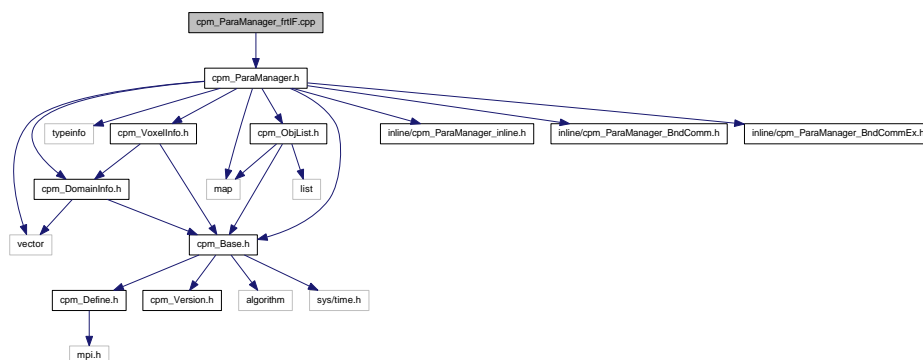
```
( size_t(_NN) \
* ( size_t(_K-(_KS)) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
) \
+ size_t(_N) \
)
```

`cpm_ParaManager_BndCommEx.h` の 36 行で定義されています。

## 5.11 `cpm_ParaManager_frtIF.cpp`

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

`cpm_ParaManager_frtIF.cpp` のインクルード依存関係図





## マクロ定義

- #define CPM\_EXTERN extern "C"
- #define cpm\_Initialize\_ cpm\_initialize\_
- #define cpm\_Voxellnit\_ cpm\_voxelinit\_
- #define cpm\_Voxellnit\_nodiv\_ cpm\_voxelinit\_nodiv\_
- #define cpm\_IsParallel\_ cpm\_isparallel\_
- #define cpm\_GetDivNum\_ cpm\_getdivnum\_
- #define cpm\_GetPitch\_ cpm\_getpitch\_
- #define cpm\_GetGlobalVoxelSize\_ cpm\_getglobalvoxelsize\_
- #define cpm\_GetGlobalOrigin\_ cpm\_getglobalorigin\_
- #define cpm\_GetGlobalRegion\_ cpm\_getglobalregion\_
- #define cpm\_GetLocalVoxelSize\_ cpm\_getlocalvoxelsize\_
- #define cpm\_GetLocalOrigin\_ cpm\_getlocalorigin\_
- #define cpm\_GetLocalRegion\_ cpm\_getlocalregion\_
- #define cpm\_GetDivPos\_ cpm\_getdivpos\_
- #define cpm\_GetBCID\_ cpm\_getbcid\_
- #define cpm\_GetVoxelHeadIndex\_ cpm\_getvoxelheadindex\_
- #define cpm\_GetVoxelTailIndex\_ cpm\_getvoxeltailindex\_
- #define cpm\_GetNeighborRankID\_ cpm\_getneighborrankid\_
- #define cpm\_GetPeriodicRankID\_ cpm\_getperiodicrankid\_
- #define cpm\_GetMyRankID\_ cpm\_getmyrankid\_
- #define cpm\_GetNumRank\_ cpm\_getnumrank\_
- #define cpm\_Abort\_ cpm\_abort\_
- #define cpm\_Barrier\_ cpm\_barrier\_
- #define cpm\_Wait\_ cpm\_wait\_
- #define cpm\_Waitall\_ cpm\_waitall\_
- #define cpm\_Bcast\_ cpm\_bcast\_
- #define cpm\_Send\_ cpm\_send\_
- #define cpm\_Recv\_ cpm\_recv\_
- #define cpm\_Isend\_ cpm\_isend\_
- #define cpm\_Irecv\_ cpm\_irecv\_
- #define cpm\_Allreduce\_ cpm\_allreduce\_
- #define cpm\_Gather\_ cpm\_gather\_
- #define cpm\_Allgather\_ cpm\_allgather\_
- #define cpm\_Gatherv\_ cpm\_gatherv\_
- #define cpm\_Allgatherv\_ cpm\_allgatherv\_
- #define cpm\_SetBndCommBuffer\_ cpm\_setbndcommbuffer\_
- #define cpm\_BndCommsS3D\_ cpm\_bndcomms3d\_
- #define cpm\_BndCommV3D\_ cpm\_bndcommv3d\_
- #define cpm\_BndCommsS4D\_ cpm\_bndcomms4d\_
- #define cpm\_BndCommsS3D\_nowait\_ cpm\_bndcomms3d\_nowait\_
- #define cpm\_BndCommV3D\_nowait\_ cpm\_bndcommv3d\_nowait\_
- #define cpm\_BndCommsS4D\_nowait\_ cpm\_bndcomms4d\_nowait\_
- #define cpm\_wait\_BndCommsS3D\_ cpm\_wait\_bndcomms3d\_
- #define cpm\_wait\_BndCommV3D\_ cpm\_wait\_bndcommv3d\_
- #define cpm\_wait\_BndCommsS4D\_ cpm\_wait\_bndcomms4d\_
- #define cpm\_BndCommV3DEx\_ cpm\_bndcommv3dex\_
- #define cpm\_BndCommsS4DEx\_ cpm\_bndcomms4dex\_
- #define cpm\_BndCommV3DEx\_nowait\_ cpm\_bndcommv3dex\_nowait\_
- #define cpm\_BndCommsS4DEx\_nowait\_ cpm\_bndcomms4dex\_nowait\_
- #define cpm\_wait\_BndCommV3DEx\_ cpm\_wait\_bndcommv3dex\_
- #define cpm\_wait\_BndCommsS4DEx\_ cpm\_wait\_bndcomms4dex\_
- #define cpm\_PeriodicCommS3D\_ cpm\_periodiccomms3d\_
- #define cpm\_PeriodicCommV3D\_ cpm\_periodiccommv3d\_
- #define cpm\_PeriodicCommS4D\_ cpm\_periodiccomms4d\_
- #define cpm\_PeriodicCommV3DEx\_ cpm\_periodiccommv3dex\_
- #define cpm\_PeriodicCommS4DEx\_ cpm\_periodiccomms4dex\_

## 関数

- CPM\_EXTERN void `cpm_Initialize_` (int \*ierr)
- CPM\_EXTERN void `cpm_Voxellnit_` (int \*div, int \*vox, REAL\_TYPE \*origin, REAL\_TYPE \*pitch, int \*obcid, int \*maxVC, int \*maxN, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Voxellnit_nodiv_` (int \*vox, REAL\_TYPE \*origin, REAL\_TYPE \*pitch, int \*obcid, int \*maxVC, int \*maxN, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_IsParallel_` (int \*ipara, int \*ierr)
- CPM\_EXTERN void `cpm_GetDivNum_` (int \*div, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetPitch_` (REAL\_TYPE \*pch, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetGlobalVoxelSize_` (int \*wsz, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetGlobalOrigin_` (REAL\_TYPE \*worg, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetGlobalRegion_` (REAL\_TYPE \*wrgn, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetLocalVoxelSize_` (int \*lsz, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetLocalOrigin_` (REAL\_TYPE \*lorg, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetLocalRegion_` (REAL\_TYPE \*lrgn, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetDivPos_` (int \*pos, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetBCID_` (int \*bcid, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetVoxelHeadIndex_` (int \*idx, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetVoxelTailIndex_` (int \*idx, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetNeighborRankID_` (int \*nID, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetPeriodicRankID_` (int \*nID, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetMyRankID_` (int \*id, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_GetNumRank_` (int \*nrank, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Abort_` (int \*errorcode)
- CPM\_EXTERN void `cpm_Barrier_` (int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Wait_` (int \*reqNo, int \*ierr)
- CPM\_EXTERN void `cpm_Waitall_` (int \*count, int \*reqlist, int \*ierr)
- CPM\_EXTERN void `cpm_Bcast_` (void \*buf, int \*count, int \*datatype, int \*root, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Send_` (void \*buf, int \*count, int \*datatype, int \*dest, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Recv_` (void \*buf, int \*count, int \*datatype, int \*source, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Isend_` (void \*buf, int \*count, int \*datatype, int \*dest, int \*procGrpNo, int \*reqNo, int \*ierr)
- CPM\_EXTERN void `cpm_Irecv_` (void \*buf, int \*count, int \*datatype, int \*source, int \*procGrpNo, int \*reqNo, int \*ierr)
- CPM\_EXTERN void `cpm_Allreduce_` (void \*sendbuf, void \*recvbuf, int \*count, int \*datatype, int \*op, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Gather_` (void \*sendbuf, int \*sendcnt, int \*sendtype, void \*recvbuf, int \*recvcnt, int \*recvtype, int \*root, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Allgather_` (void \*sendbuf, int \*sendcnt, int \*sendtype, void \*recvbuf, int \*recvcnt, int \*recvtype, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Gatherv_` (void \*sendbuf, int \*sendcnt, int \*sendtype, void \*recvbuf, int \*recvcnts, int \*displs, int \*recvtype, int \*root, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_Allgatherv_` (void \*sendbuf, int \*sendcnt, int \*sendtype, void \*recvbuf, int \*recvcnts, int \*displs, int \*recvtype, int \*root, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_SetBndCommBuffer_` (int \*maxVC, int \*maxN, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_BndCommsS4D_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*nmax, int \*vc, int \*vc\_comm, int \*datatype, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_BndCommsS3D_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*vc, int \*vc\_comm, int \*datatype, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_BndCommV3D_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*vc, int \*vc\_comm, int \*datatype, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_BndCommsS4D_nowait_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*nmax, int \*vc, int \*vc\_comm, int \*datatype, int \*reqlist, int \*procGrpNo, int \*ierr)
- CPM\_EXTERN void `cpm_BndCommsS3D_nowait_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*vc, int \*vc\_comm, int \*datatype, int \*reqlist, int \*procGrpNo, int \*ierr)

- `CPM_EXTERN void cpm_BndCommV3D_nowait_` (void \*array, int \*imax, int \*jmax, int \*kmax, int \*vc, int \*vc\_comm, int \*datatype, int \*reqlist, int \*procGrpNo, int \*ierr)
- `CPM_EXTERN void cpm_wait_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 45 行で定義されています。

#### 5.11.2.2 `#define cpm_Allgather_ cpm_allgather_`

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

#### 5.11.2.3 `#define cpm_Allgatherv_cpm_allgatherv_`

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

#### 5.11.2.4 `#define cpm_Allreduce_cpm_allreduce_`

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

#### 5.11.2.5 `#define cpm_Barrier_cpm_barrier_`

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

#### 5.11.2.6 `#define cpm_Bcast_cpm_bcast_`

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

#### 5.11.2.7 `#define cpm_BndComms3D_cpm_bndcomms3d_`

`cpm_ParaManager_frtIF.cpp` の 60 行で定義されています。

#### 5.11.2.8 `#define cpm_BndComms3D_nowait_cpm_bndcomms3d_nowait_`

`cpm_ParaManager_frtIF.cpp` の 63 行で定義されています。

#### 5.11.2.9 `#define cpm_BndComms4D_cpm_bndcomms4d_`

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

参照元 `cpm_BndComms3D_()`, と `cpm_BndCommV3D_()`.

#### 5.11.2.10 `#define cpm_BndComms4D_nowait_cpm_bndcomms4d_nowait_`

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

参照元 `cpm_BndComms3D_nowait_()`, と `cpm_BndCommV3D_nowait_()`.

#### 5.11.2.11 `#define cpm_BndComms4DEx_cpm_bndcomms4dex_`

`cpm_ParaManager_frtIF.cpp` の 70 行で定義されています。

参照元 `cpm_BndCommV3DEx_()`.

#### 5.11.2.12 `#define cpm_BndComms4DEx_nowait_cpm_bndcomms4dex_nowait_`

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

参照元 `cpm_BndCommV3DEx_nowait_()`.

#### 5.11.2.13 `#define cpm_BndCommV3D_cpm_bndcommv3d_`

`cpm_ParaManager_frtIF.cpp` の 61 行で定義されています。

5.11.2.14 `#define cpm_BndCommV3D_nowait_ cpm.bndcommv3d_nowait_`

cpm\_ParaManager\_frtIF.cpp の 64 行で定義されています。

5.11.2.15 `#define cpm_BndCommV3DEx_ cpm.bndcommv3dex_`

cpm\_ParaManager\_frtIF.cpp の 69 行で定義されています。

5.11.2.16 `#define cpm_BndCommV3DEx_nowait_ cpm.bndcommv3dex_nowait_`

cpm\_ParaManager\_frtIF.cpp の 71 行で定義されています。

5.11.2.17 `#define CPM_EXTERN extern "C"`

extern 宣言

cpm\_ParaManager\_frtIF.cpp の 17 行で定義されています。

5.11.2.18 `#define cpm_Gather_ cpm.gather_`

cpm\_ParaManager\_frtIF.cpp の 55 行で定義されています。

5.11.2.19 `#define cpm_Gatherv_ cpm.gatherv_`

cpm\_ParaManager\_frtIF.cpp の 57 行で定義されています。

5.11.2.20 `#define cpm_GetBCID_ cpm.getbcid_`

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

5.11.2.21 `#define cpm_GetDivNum_ cpm.getdivnum_`

cpm\_ParaManager\_frtIF.cpp の 29 行で定義されています。

5.11.2.22 `#define cpm_GetDivPos_ cpm.getdivpos_`

cpm\_ParaManager\_frtIF.cpp の 37 行で定義されています。

5.11.2.23 `#define cpm_GetGlobalOrigin_ cpm.getglobalorigin_`

cpm\_ParaManager\_frtIF.cpp の 32 行で定義されています。

5.11.2.24 `#define cpm_GetGlobalRegion_ cpm.getglobalregion_`

cpm\_ParaManager\_frtIF.cpp の 33 行で定義されています。

5.11.2.25 `#define cpm_GetGlobalVoxelSize_ cpm.getglobalvoxelsize_`

cpm\_ParaManager\_frtIF.cpp の 31 行で定義されています。

5.11.2.26 `#define cpm_GetLocalOrigin_ cpm_getlocalorigin_`

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

5.11.2.27 `#define cpm_GetLocalRegion_ cpm_getlocalregion_`

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

5.11.2.28 `#define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_`

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

5.11.2.29 `#define cpm_GetMyRankID_ cpm_getmyrankid_`

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

5.11.2.30 `#define cpm_GetNeighborRankID_ cpm_getneighborrandid_`

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

5.11.2.31 `#define cpm_GetNumRank_ cpm_getnumrank_`

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

5.11.2.32 `#define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_`

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

5.11.2.33 `#define cpm_GetPitch_ cpm_getpitch_`

`cpm_ParaManager_frtrIF.cpp` の 30 行で定義されています。

5.11.2.34 `#define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_`

`cpm_ParaManager_frtrIF.cpp` の 39 行で定義されています。

5.11.2.35 `#define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_`

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

5.11.2.36 `#define cpm_Initialize_ cpm_initialize_`

`cpm_ParaManager_frtrIF.cpp` の 25 行で定義されています。

5.11.2.37 `#define cpm_Irecv_ cpm_irecv_`

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

5.11.2.38 `#define cpm_Isend_ cpm.isend_`

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

5.11.2.39 `#define cpm_IsParallel_ cpm.isparallel_`

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

5.11.2.40 `#define cpm_PeriodicCommsS3D cpm.periodiccomms3d_`

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

5.11.2.41 `#define cpm_PeriodicCommsS4D cpm.periodiccomms4d_`

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

5.11.2.42 `#define cpm_PeriodicCommsS4DEx cpm.periodiccomms4dex_`

cpm\_ParaManager\_frtIF.cpp の 79 行で定義されています。

5.11.2.43 `#define cpm_PeriodicCommV3D cpm.periodiccommv3d_`

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

5.11.2.44 `#define cpm_PeriodicCommV3DEx cpm.periodiccommv3dex_`

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

5.11.2.45 `#define cpm_Recv_ cpm.recv_`

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

5.11.2.46 `#define cpm_Send_ cpm.send_`

cpm\_ParaManager\_frtIF.cpp の 50 行で定義されています。

5.11.2.47 `#define cpm_SetBndCommBuffer_ cpm.setbndcommbuffer_`

cpm\_ParaManager\_frtIF.cpp の 59 行で定義されています。

5.11.2.48 `#define cpm_Voxellnit_ cpm.voxellnit_`

cpm\_ParaManager\_frtIF.cpp の 26 行で定義されています。

5.11.2.49 `#define cpm_Voxellnit_nodiv_ cpm.voxellnit_nodiv_`

cpm\_ParaManager\_frtIF.cpp の 27 行で定義されています。

#### 5.11.2.50 `#define cpm_Wait_ cpm_wait_`

`cpm_ParaManager_frtIF.cpp` の 47 行で定義されています。

#### 5.11.2.51 `#define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_`

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

#### 5.11.2.52 `#define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_`

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

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

#### 5.11.2.53 `#define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_`

`cpm_ParaManager_frtIF.cpp` の 74 行で定義されています。

参照元 `cpm_wait_BndCommV3DEx_()`.

#### 5.11.2.54 `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`

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

#### 5.11.2.55 `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`

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

#### 5.11.2.56 `#define cpm_Waitall_ cpm_waitall_`

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

### 5.11.3 関数

#### 5.11.3.1 `CPM_EXTERN void cpm_Abort_ ( int * errorcode )`

Abort

- Abort のFortran インターフェイス関数

引数

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

`cpm_ParaManager_frtIF.cpp` の 960 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 1377 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 \* *root*, 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 1479 行で定義されています。

参照先 cpm\_ParaManager::Allgatherv(), 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 1276 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 988 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 1077 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, C-  
PM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::get\_instance(), と cpm\_ParaManager::GetMPI\_Datatype().

#### 5.11.3.7 CPM\_EXTERN void cpm\_BndComms3D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* procGrpNo, int \* ierr )

袖通信 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の袖通信を行う
- BndComms3D のFortran インターフェイス関数

引数

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

in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1607 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 1749 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 1562 行で定義されています。

参照先 cpm\_ParaManager::BndCommS4D(), CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::get\_instance(), と cpm\_ParaManager::GetMPI\_Datatype().

**5.11.3.10 CPM\_EXTERN void cpm\_BndCommS4D\_nowait\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* nmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommS4D\_nowait のFortran インターフェイス関数

引数

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

cpm\_ParaManager\_frtIF.cpp の 1709 行で定義されています。

参照先 cpm\_ParaManager::cpm\_BndCommS4D\_nowait(), CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, と cpm\_ParaManager::get\_instance().

**5.11.3.11 CPM\_EXTERN void cpm\_BndCommS4DEx\_ ( void \* array, int \* nmax, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* procGrpNo, int \* ierr )**

袖通信 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommS4DEx のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1972 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 2069 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 1657 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 1795 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 2017 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 2109 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 1327 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 1428 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::Gatherv(), cpm\_ParaManager::get\_instance(), と cpm\_ParaManager::GetMPI\_Datatype().

#### 5.11.3.19 CPM\_EXTERN void cpm\_GetBCID\_ ( int \* *bcid*, int \* *procGrpNo*, int \* *ierr* )

自ランクのBCID を取得

- GetBCID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>bcid</i>	自ランクのBCID(6word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 665 行で定義されています。

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

#### 5.11.3.20 CPM\_EXTERN void cpm\_GetDivNum\_ ( int \* *div*, int \* *procGrpNo*, int \* *ierr* )

領域分割数を取得

- GetDivNum のFortran インターフェイス関数

引数

out	<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 288 行で定義されています。

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

#### 5.11.3.21 CPM\_EXTERN void cpm\_GetDivPos\_ ( int \* *pos*, int \* *procGrpNo*, int \* *ierr* )

自ランクの領域分割位置を取得

- GetDivPos のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>pos</i>	自ランクの領域分割位置 (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 623 行で定義されています。

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

#### 5.11.3.22 CPM\_EXTERN void cpm\_GetGlobalOrigin\_ ( REAL\_TYPE \* *worg*, int \* *procGrpNo*, int \* *ierr* )

全体空間の原点を取得

- GetGlobalOrigin のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>worg</i>	全体空間の原点 (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 414 行で定義されています。

参照先 CPM\_ERROR\_GET\_GLOBALORIGIN, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, CPM\_SUCCESS, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetGlobalOrigin(), と REAL\_TYPE.

#### 5.11.3.23 CPM\_EXTERN void cpm\_GetGlobalRegion\_ ( REAL\_TYPE \* *wrgn*, int \* *procGrpNo*, int \* *ierr* )

全体空間サイズを取得

- GetGlobalRegion のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>wrgn</i>	全体空間サイズ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 456 行で定義されています。

参照先 CPM\_ERROR\_GET\_GLOBALREGION, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, CPM\_SUCCESS, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetGlobalRegion(), と REAL\_TYPE.

#### 5.11.3.24 CPM\_EXTERN void cpm\_GetGlobalVoxelSize\_ ( int \* *wsz*, int \* *procGrpNo*, int \* *ierr* )

全体ボクセル数を取得

- GetGlobalVoxelSize のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>wsz</i>	全体ボクセル数 (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 372 行で定義されています。

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

#### 5.11.3.25 CPM\_EXTERN void cpm\_GetLocalOrigin\_ ( REAL\_TYPE \* *lorg*, int \* *procGrpNo*, int \* *ierr* )

自ランクの空間原点を取得

- GetLocalOrigin のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lorg</i>	自ランクの空間原点 (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 540 行で定義されています。

参照先 CPM\_ERROR\_GET\_LOCALORIGIN, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, CPM\_SUCCESS, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetLocalOrigin(), と REAL\_TYPE.

#### 5.11.3.26 CPM\_EXTERN void cpm\_GetLocalRegion\_ ( REAL\_TYPE \* *lrgn*, int \* *procGrpNo*, int \* *ierr* )

自ランクの空間サイズを取得

- GetLocalRegion のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lrgn</i>	自ランクの空間サイズ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 581 行で定義されています。

参照先 CPM\_ERROR\_GET\_LOCALREGION, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, CPM\_SUCCESS, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetLocalOrigin(), と REAL\_TYPE.

#### 5.11.3.27 CPM\_EXTERN void cpm\_GetLocalVoxelSize\_ ( int \* *lsz*, int \* *procGrpNo*, int \* *ierr* )

自ランクのボクセル数を取得

- GetLocalVoxelSize のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>lsz</i>	自ランクのボクセル数 (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 498 行で定義されています。

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

#### 5.11.3.28 CPM\_EXTERN void cpm\_GetMyRankID\_ ( int \* *id*, int \* *procGrpNo*, int \* *ierr* )

ランク番号の取得

- GetMyRankID のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>id</i>	ランク番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 886 行で定義されています。

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

#### 5.11.3.29 CPM\_EXTERN void cpm\_GetNeighborRankID\_( int \* nID, int \* procGrpNo, int \* ierr )

自ランクの隣接ランク番号を取得

- GetNeighborRankID のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	nID	自ランクの隣接ランク番号 (6word の整数配列)
out	ier	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 796 行で定義されています。

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

#### 5.11.3.30 CPM\_EXTERN void cpm\_GetNumRank\_( int \* nrank, int \* procGrpNo, int \* ierr )

ランク数の取得

- GetNumRank のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	nrank	ランク数
out	ier	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 924 行で定義されています。

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

#### 5.11.3.31 CPM\_EXTERN void cpm\_GetPeriodicRankID\_( int \* nID, int \* procGrpNo, int \* ierr )

自ランクの周期境界の隣接ランク番号を取得

- GetPeriodicRankID のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	nID	自ランクの周期境界の隣接ランク番号 6word の整数配列)
out	ier	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 841 行で定義されています。

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

#### 5.11.3.32 CPM\_EXTERN void cpm\_GetPitch\_( REAL\_TYPE \* pch, int \* procGrpNo, int \* ierr )

ピッチを取得

- GetPitch のFortran インターフェイス関数

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>pch</i>	ピッチ (3word の実数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 330 行で定義されています。

参照先 CPM\_ERROR\_GET\_PITCH, CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, CPM\_SUCCESS, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetPitch(), と REAL\_TYPE.

#### 5.11.3.33 CPM\_EXTERN void cpm\_GetVoxelHeadIndex\_ ( int \* *idx*, int \* *procGrpNo*, int \* *ierr* )

自ランクの始点VOXEL の全体空間でのインデクスを取得

- GetVoxelHeadIndex のFortran インターフェイス関数
- 全体空間の先頭インデクスを 0 としたC 型のインデクス

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>idx</i>	自ランクの始点VOXEL インデクス (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 711 行で定義されています。

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

#### 5.11.3.34 CPM\_EXTERN void cpm\_GetVoxelTailIndex\_ ( int \* *idx*, int \* *procGrpNo*, int \* *ierr* )

自ランクの終点VOXEL の全体空間でのインデクスを取得

- GetVoxelTailIndex のFortran インターフェイス関数
- 全体空間の先頭インデクスを 0 としたC 型のインデクス

引数

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>idx</i>	自ランクの終点VOXEL インデクス (3word の整数配列)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 754 行で定義されています。

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

#### 5.11.3.35 CPM\_EXTERN void cpm\_Initialize\_ ( int \* *ierr* )

初期化処理 (MPI\_Init は実行済みの場合)

- Initialize のFortran インターフェイス関数
- Fortran でMPI\_Init がコールされている必要がある

## 引数

out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)
-----	------	--------------------------------------

cpm\_ParaManager\_frtIF.cpp の 146 行で定義されています。

参照先 CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::get\_instance(), と cpm\_ParaManager::Initialize().

**5.11.3.36 CPM\_EXTERN void cpm\_lrecv\_ ( void \* buf, int \* count, int \* datatype, int \* source, int \* procGrpNo, int \* reqNo, int \* ierr )**

## lrecv

- lrecv のFortran インターフェイス関数

## 引数

	inout]	buf 受信バッファ
in	count	受信バッファのサイズ (ワード数)
in	datatype	データタイプ (fparam.fi を参照)
in	source	送信元先のランク番号 (procGrpNo 内でのランク番号)
in	procGrpNo	プロセスグループ番号
in	reqNo	リクエスト番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 1237 行で定義されています。

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

**5.11.3.37 CPM\_EXTERN void cpm\_lsend\_ ( void \* buf, int \* count, int \* datatype, int \* dest, int \* procGrpNo, int \* reqNo, int \* ierr )**

## lsend

- lsend のFortran インターフェイス関数

## 引数

	inout]	buf 送信バッファ
in	count	送信バッファのサイズ (ワード数)
in	datatype	データタイプ (fparam.fi を参照)
in	dest	送信先のランク番号 (procGrpNo 内でのランク番号)
in	procGrpNo	プロセスグループ番号
out	reqNo	リクエスト番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 1198 行で定義されています。

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

**5.11.3.38 CPM\_EXTERN void cpm\_IsParallel\_ ( int \* ipara, int \* ierr )**

並列実行であるかチェックする

- IsParallel のFortran インターフェイス関数

## 引数

out	<i>ipara</i>	並列実行フラグ (1=並列実行、1 以外=逐次実行)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 254 行で定義されています。

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

**5.11.3.39 CPM\_EXTERN void cpm\_PeriodicCommS3D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* dir, int \* pm, int \* datatype, int \* procGrpNo, int \* ierr )**

周期境界袖通信 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS3D のFortran インターフェイス関数

## 引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 2306 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PERIODIC\_INVALID\_DIR, CPM\_ERROR\_PERIODIC\_INVALID\_PM, CPM\_ERROR\_PM\_INSTANCE, cpm\_PeriodicCommS4D\_(), cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::PeriodicCommS3D().

**5.11.3.40 CPM\_EXTERN void cpm\_PeriodicCommS4D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* nmax, int \* vc, int \* vc\_comm, int \* dir, int \* pm, int \* datatype, int \* procGrpNo, int \* ierr )**

周期境界袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS4D のFortran インターフェイス関数

## 引数

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

in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 2243 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::PeriodicCommsS4D().

参照元 cpm\_PeriodicCommS3D\_(), と cpm\_PeriodicCommV3D\_().

**5.11.3.41 CPM\_EXTERN void cpm\_PeriodicCommS4DEx\_ ( void \* array, int \* nmax, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* dir, int \* pm, int \* datatype, int \* procGrpNo, int \* ierr )**

周期境界袖通信 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommS4DEx のFortran インターフェイス関数

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>dir</i>	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	<i>pm</i>	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 2445 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::PeriodicCommsS4DEx().

参照元 cpm\_PeriodicCommV3DEx\_().

**5.11.3.42 CPM\_EXTERN void cpm\_PeriodicCommV3D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* dir, int \* pm, int \* datatype, int \* procGrpNo, int \* ierr )**

周期境界袖通信 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommV3D のFortran インターフェイス関数

引数

	<i>inout[]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)

in	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	datatype	袖通信データのデータタイプ (cpm_fparam.fi を参照)
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 2375 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PERIODIC\_INVALID\_DIR, CPM\_ERROR\_PERIODIC\_INVALID\_PM, CPM\_ERROR\_PM\_INSTANCE, cpm\_PeriodicCommS4D\_(), cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::PeriodicCommV3D().

**5.11.3.43 CPM\_EXTERN void cpm\_PeriodicCommV3DEx\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* dir, int \* pm, int \* datatype, int \* procGrpNo, int \* ierr )**

周期境界袖通信 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicCommV3DEx のFortran インターフェイス関数

引数

	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	datatype	袖通信データのデータタイプ (cpm_fparam.fi を参照)
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 2508 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PERIODIC\_INVALID\_DIR, CPM\_ERROR\_PERIODIC\_INVALID\_PM, CPM\_ERROR\_PM\_INSTANCE, cpm\_PeriodicCommS4DEx\_(), cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::PeriodicCommV3DEx().

**5.11.3.44 CPM\_EXTERN void cpm\_Recv\_ ( void \* buf, int \* count, int \* datatype, int \* source, int \* procGrpNo, int \* ierr )**

Recv

- Recv のFortran インターフェイス関数

引数

	inout]	buf 受信バッファ
in	count	受信バッファのサイズ (ワード数)
in	datatype	データタイプ (fparam.fi を参照)
in	source	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 1157 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::Recv().

#### 5.11.3.45 CPM\_EXTERN void cpm\_Send\_ ( void \* buf, int \* count, int \* datatype, int \* dest, int \* procGrpNo, int \* ierr )

Send

- Send のFortran インターフェイス関数

引数

	<i>inout]</i>	buf 送信バッファ
in	<i>count</i>	送信バッファのサイズ (ワード数)
in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 1117 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_MPI\_INVALID\_DATATYPE, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::get\_instance(), cpm\_ParaManager::GetMPI\_Datatype(), と cpm\_ParaManager::Send().

#### 5.11.3.46 CPM\_EXTERN void cpm\_SetBndCommBuffer\_ ( int \* maxVC, int \* maxN, int \* procGrpNo, int \* ierr )

袖通信バッファのセット (Fortran インターフェイス)

- 袖通信バッファ確保処理のFortran インターフェイス関数

引数

in	<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 1525 行で定義されています。

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

#### 5.11.3.47 CPM\_EXTERN void cpm\_Voxellnit\_ ( int \* div, int \* vox, REAL\_TYPE \* origin, REAL\_TYPE \* pitch, int \* obcid, int \* maxVC, int \* maxN, int \* procGrpNo, int \* ierr )

領域分割

- Voxellnit のFortran インターフェイス関数
- 領域分割の各種情報を引数で渡して領域分割を行う
- プロセスグループの全てのランクが活性ドメインになる
- 領域分割数を指定する

## 引数

in	<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>obcid</i>	全体空間の外部境界条件ID(サイズ 6)
in	<i>maxVC</i>	最大の袖数 (袖通信用)
in	<i>maxN</i>	最大の成分数 (袖通信用)
in	<i>procGrpNo</i>	領域分割を行うプロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 183 行で定義されています。

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

**5.11.3.48 CPM\_EXTERN void cpm\_Voxellnit\_nodiv\_ ( int \* vox, REAL\_TYPE \* origin, REAL\_TYPE \* pitch, int \* obcid, int \* maxVC, int \* maxN, int \* procGrpNo, int \* ierr )**

## 領域分割

- Voxellnit のFortran インターフェイス関数
- 領域分割の各種情報を引数で渡して領域分割を行う
- プロセスグループの全てのランクが活性ドメインになる
- プロセスグループのランク数で自動領域分割

## 引数

in	<i>vox</i>	空間全体のボクセル数 (サイズ 3)
in	<i>origin</i>	空間全体の原点 (サイズ 3)
in	<i>pitch</i>	ボクセルピッチ (サイズ 3)
in	<i>obcid</i>	全体空間の外部境界条件ID(サイズ 6)
in	<i>maxVC</i>	最大の袖数 (袖通信用)
in	<i>maxN</i>	最大の成分数 (袖通信用)
in	<i>procGrpNo</i>	領域分割を行うプロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 223 行で定義されています。

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

**5.11.3.49 CPM\_EXTERN void cpm\_Wait\_ ( int \* reqNo, int \* ierr )**

## Wait

- Wait のFortran インターフェイス関数

## 引数

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

cpm\_ParaManager\_frtIF.cpp の 1016 行で定義されています。

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

**5.11.3.50 CPM\_EXTERN void cpm\_wait\_BndCommS3D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信の wait、展開 (Scalar3D 版) のFortran インターフェイス

- (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait\_BndCommS3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1882 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::cpm\_wait\_BndCommS3D(), cpm\_wait\_BndCommS4D\_, と cpm\_ParaManager::get\_instance().

**5.11.3.51 CPM\_EXTERN void cpm\_wait\_BndCommS4D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* nmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信の wait、展開 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait\_BndCommS4D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1842 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::cpm\_wait\_BndCommS4D(), と cpm\_ParaManager::get\_instance().

**5.11.3.52 CPM\_EXTERN void cpm\_wait\_BndCommS4DEx\_ ( void \* array, int \* nmax, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信の wait、展開 (Scalar4DEx 版) のFortran インターフェイス

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait\_BndCommS4DEx のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
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 2156 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_ParaManager::cpm\_wait\_BndCommS4DEx(), と cpm\_ParaManager::get\_instance().

**5.11.3.53 CPM\_EXTERN void cpm\_wait\_BndCommV3D\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信の wait、展開 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う
- wait\_BndCommV3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 1927 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_wait\_BndCommS4D\_, cpm\_ParaManager::cpm\_wait\_BndCommV3D(), と cpm\_ParaManager::get\_instance().

**5.11.3.54 CPM\_EXTERN void cpm\_wait\_BndCommV3DEx\_ ( void \* array, int \* imax, int \* jmax, int \* kmax, int \* vc, int \* vc\_comm, int \* datatype, int \* reqlist, int \* procGrpNo, int \* ierr )**

非同期版袖通信の wait、展開 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait\_BndCommV3DEx のFortran インターフェイス関数

引数

	<i>inout</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm\_ParaManager\_frtIF.cpp の 2196 行で定義されています。

参照先 CPM\_ERROR\_INVALID\_PTR, CPM\_ERROR\_PM\_INSTANCE, cpm\_wait\_BndCommS4DEx\_, cpm\_ParaManager::cpm\_wait\_BndCommV3DEx(), と cpm\_ParaManager::get\_instance().

#### 5.11.3.55 CPM\_EXTERN void cpm\_Waitall\_ ( int \* count, int \* reqlist, int \* ierr )

Waitall

- Waitall のFortran インターフェイス関数

引数

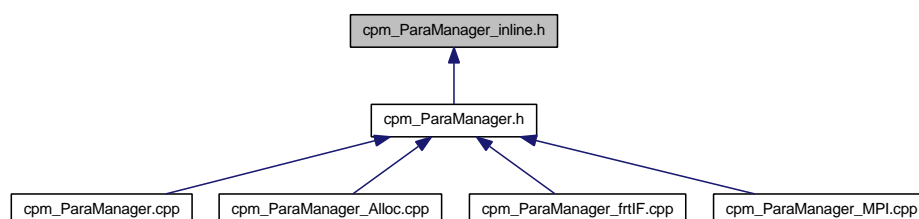
in	<i>count</i>	リクエストの数
in	<i>reqlist</i>	リクエスト番号のリスト (0 以上の整数)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm\_ParaManager\_frtIF.cpp の 1045 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

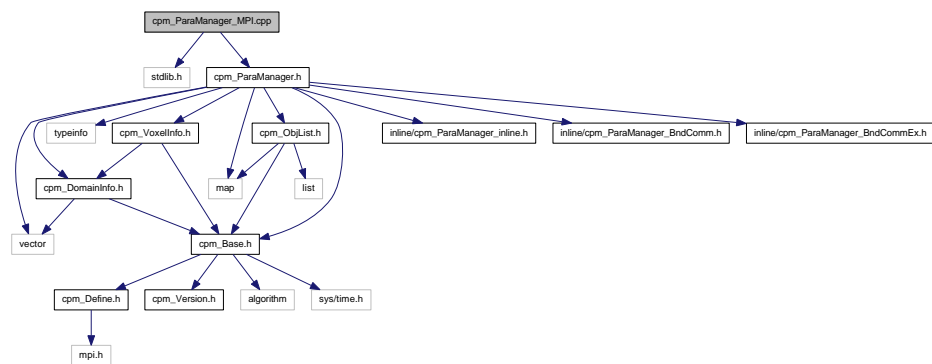
[cpm\\_ParaManager\\_inline.h](#) で定義されています。

## 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

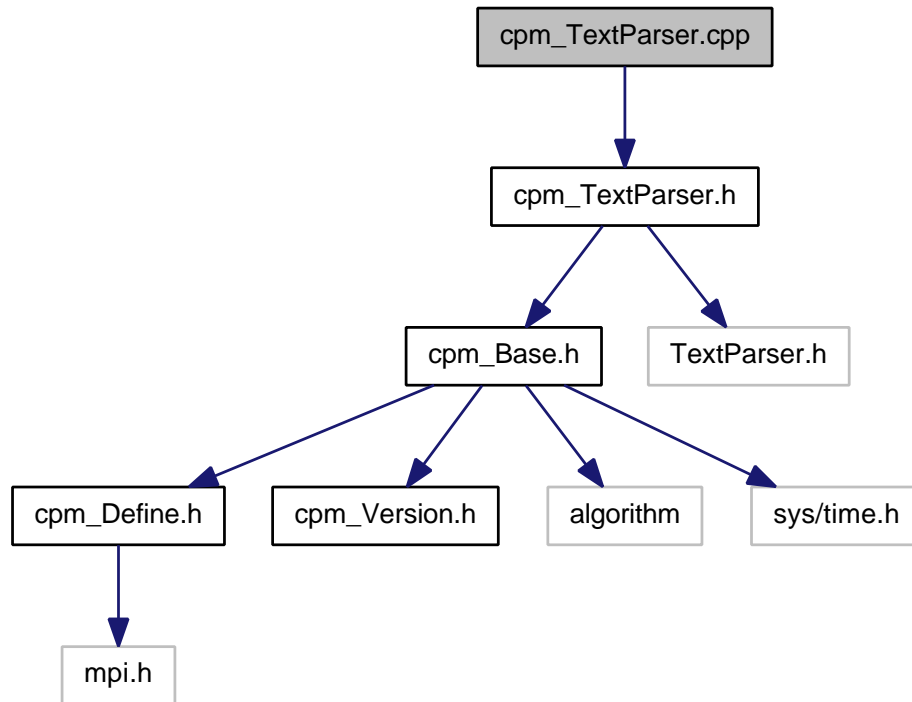
日付

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

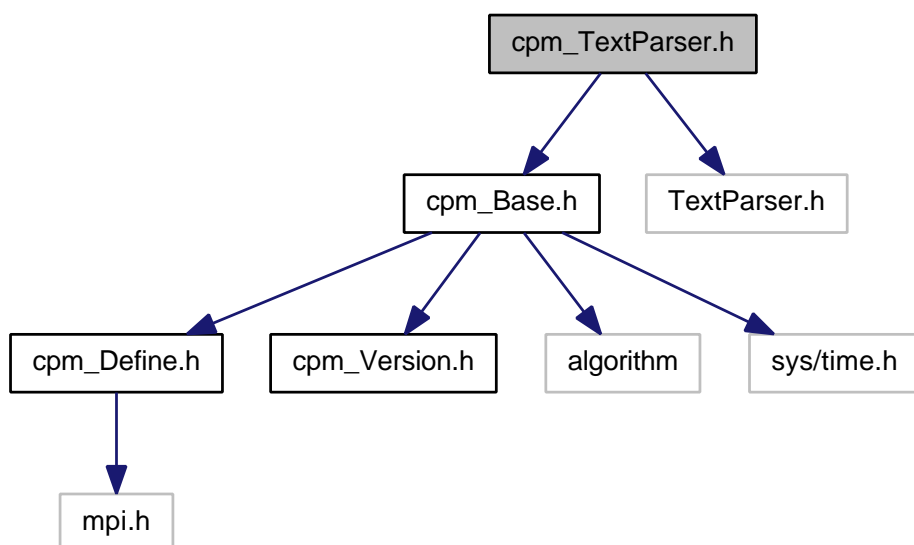
2012/05/31

[cpm\\_TextParser.cpp](#) で定義されています。

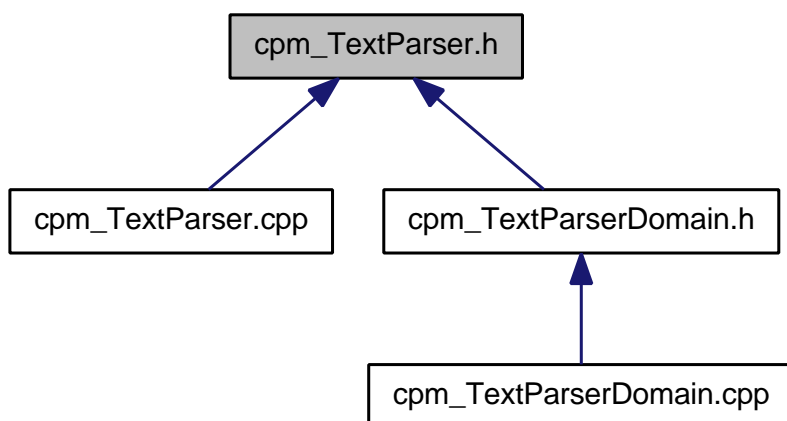
## 5.15 cpm\_TextParser.h

```
#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

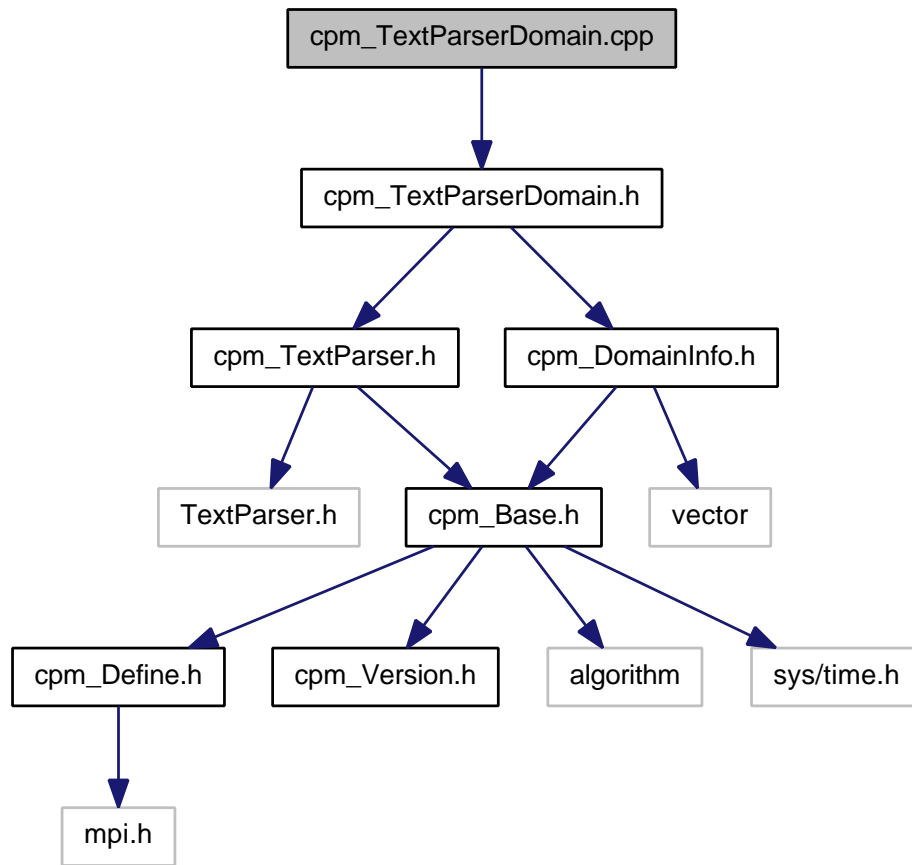
[cpm\\_TextParser.h](#) で定義されています。



## 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

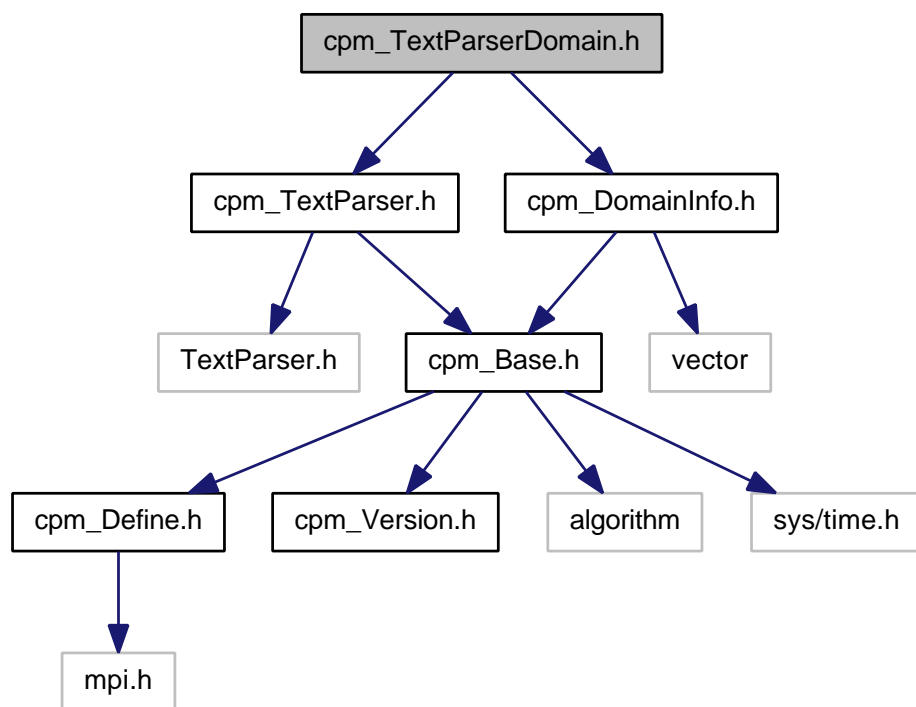
[cpm\\_TextParserDomain.cpp](#) で定義されています。

## 5.17 cpm\_TextParserDomain.h

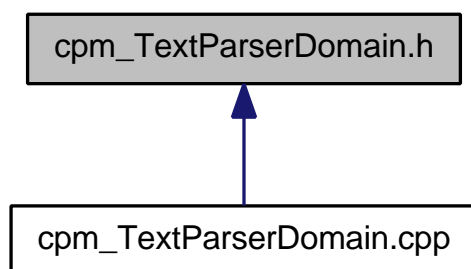
```
#include "cpm_TextParser.h"
```

```
#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

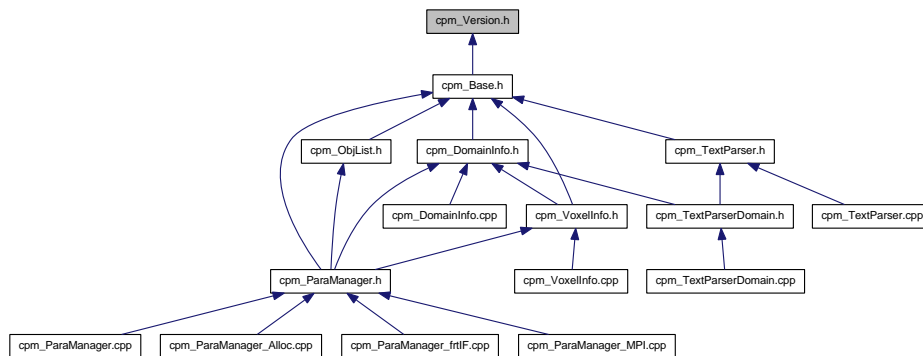
日付

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 "0.1.1"`
- `#define CPM_REVISION "20120601_1200"`

#### 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 "20120601_1200"`

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 "0.1.1"`

CPM ライブラリのバージョン

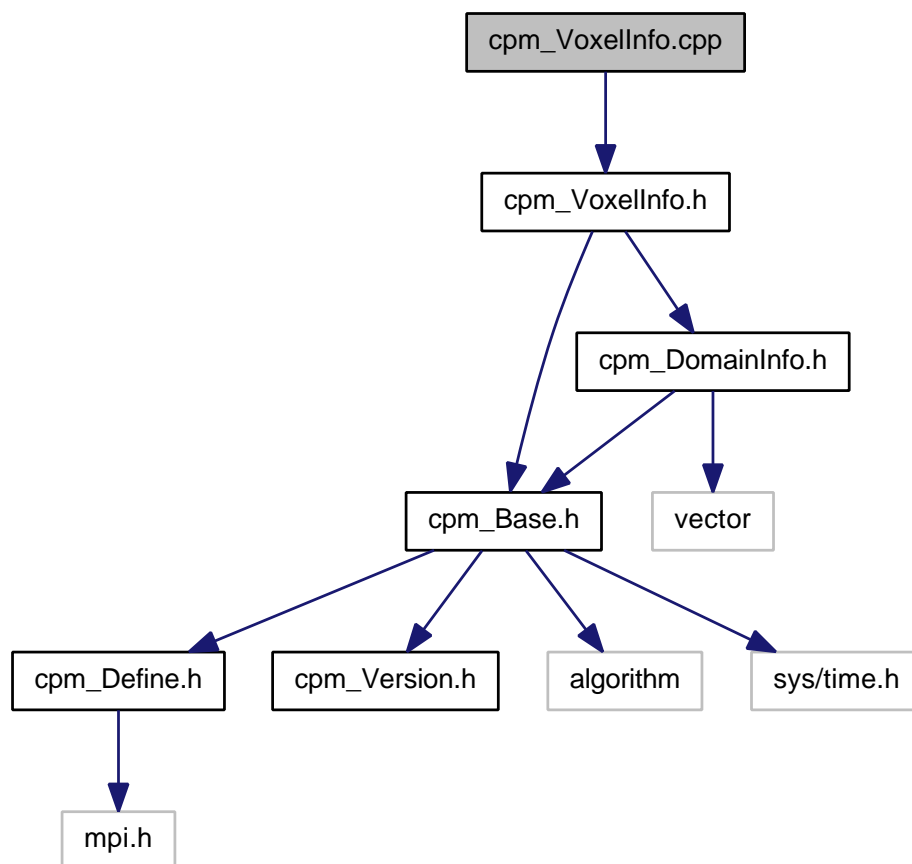
`cpm_Version.h` の 19 行で定義されています。

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

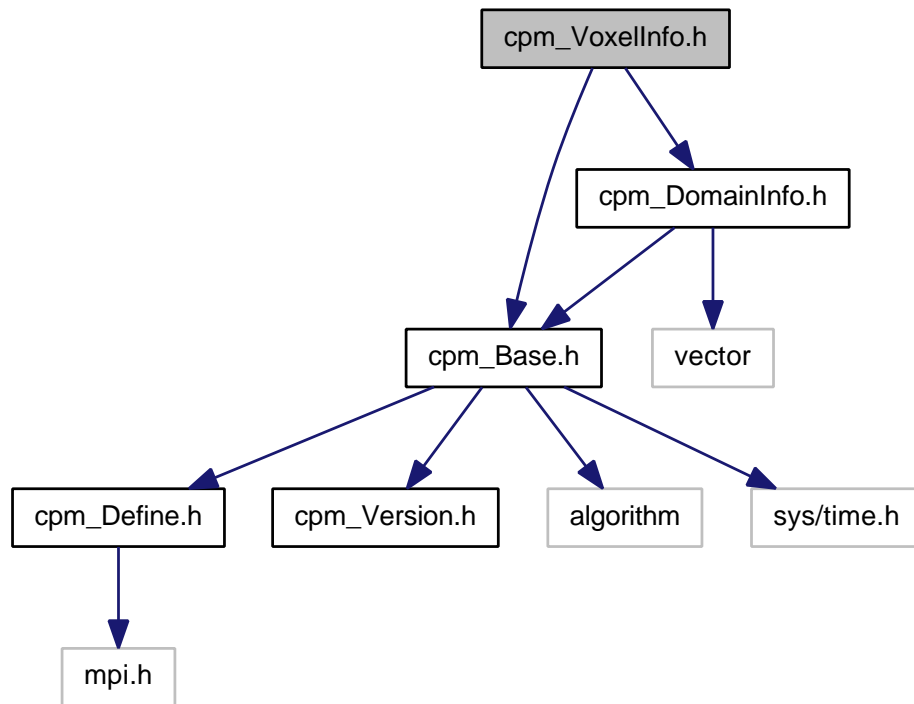
[cpm\\_VoxelInfo.cpp](#) で定義されています。

## 5.20 cpm\_VoxelInfo.h

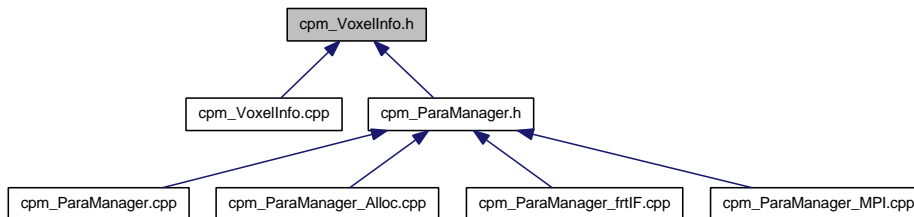
```
#include "cpm_Base.h"
```

```
#include "cpm_DomainInfo.h"
```

cpm\_VoxellInfo.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_VoxellInfo`

### 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_VoxellInfo.h` で定義されています。

# Index

- ~S\_BNDCOMM\_BUFFER
  - S\_BNDCOMM\_BUFFER, [94](#)
- ~cpm\_ActiveSubDomainInfo
  - cpm\_ActiveSubDomainInfo, [8](#)
- ~cpm\_Base
  - cpm\_Base, [12](#)
- ~cpm\_DomainInfo
  - cpm\_DomainInfo, [17](#)
- ~cpm\_GlobalDomainInfo
  - cpm\_GlobalDomainInfo, [20](#)
- ~cpm\_LocalDomainInfo
  - cpm\_LocalDomainInfo, [23](#)
- ~cpm\_ObjList
  - cpm\_ObjList, [24](#)
- ~cpm\_TextParser
  - cpm\_TextParser, [89](#)
- ~cpm\_TextParserDomain
  - cpm\_TextParserDomain, [92](#)
- \_IDXXFX
  - cpm\_ParaManager\_BndComm.h, [112](#)
  - cpm\_ParaManager\_BndCommEx.h, [114](#)
- \_IDXFY
  - cpm\_ParaManager\_BndComm.h, [112](#)
  - cpm\_ParaManager\_BndCommEx.h, [114](#)
- \_IDXFZ
  - cpm\_ParaManager\_BndComm.h, [113](#)
  - cpm\_ParaManager\_BndCommEx.h, [114](#)
- \_IDX\_S3D
  - cpm\_Define.h, [100](#)
- \_IDX\_S4D
  - cpm\_Define.h, [100](#)
- \_IDX\_S4DEX
  - cpm\_Define.h, [101](#)
- \_IDX\_V3D
  - cpm\_Define.h, [101](#)
- \_IDX\_V3DEX
  - cpm\_Define.h, [102](#)
- Abort
  - cpm\_ParaManager, [31](#)
- Add
  - cpm\_ObjList, [25](#)
- AddSubDomain
  - cpm\_GlobalDomainInfo, [20](#)
- Allgather
  - cpm\_ParaManager, [31](#), [32](#)
- Allgatherv
  - cpm\_ParaManager, [32](#), [33](#)
- AllocDoubleS3D
  - cpm\_ParaManager, [33](#)
- AllocDoubleS4D
  - cpm\_ParaManager, [33](#)
- AllocDoubleS4DEX
  - cpm\_ParaManager, [34](#)
- AllocDoubleV3D
  - cpm\_ParaManager, [34](#)
- AllocDoubleV3DEX
  - cpm\_ParaManager, [34](#)
- AllocFloatS3D
  - cpm\_ParaManager, [35](#)
- AllocFloatS4D
  - cpm\_ParaManager, [35](#)
- AllocFloatS4DEX
  - cpm\_ParaManager, [35](#)
- AllocFloatV3D
  - cpm\_ParaManager, [36](#)
- AllocFloatV3DEX
  - cpm\_ParaManager, [36](#)
- AllocIntS3D
  - cpm\_ParaManager, [36](#)
- AllocIntS4D
  - cpm\_ParaManager, [36](#)
- AllocIntS4DEX
  - cpm\_ParaManager, [37](#)
- AllocIntV3D
  - cpm\_ParaManager, [37](#)
- AllocIntV3DEX
  - cpm\_ParaManager, [37](#)
- AllocRealS3D
  - cpm\_ParaManager, [38](#)
- AllocRealS4D
  - cpm\_ParaManager, [38](#)
- AllocRealS4DEX
  - cpm\_ParaManager, [38](#)
- AllocRealV3D
  - cpm\_ParaManager, [39](#)
- AllocRealV3DEX
  - cpm\_ParaManager, [39](#)
- Allreduce
  - cpm\_ParaManager, [39](#), [40](#)
- BOTH
  - cpm\_Define.h, [106](#)
- Barrier
  - cpm\_ParaManager, [40](#)
- Bcast
  - cpm\_ParaManager, [40](#), [41](#)
- BndCommInfoMap
  - cpm\_ParaManager.h, [111](#)
- BndCommS3D

- cpm\_ParaManager, [41](#), [42](#)
- BndCommS3D\_nowait
  - cpm\_ParaManager, [42](#), [43](#)
- BndCommS4D
  - cpm\_ParaManager, [43](#), [44](#)
- BndCommS4D\_nowait
  - cpm\_ParaManager, [44](#), [45](#)
- BndCommS4DEx
  - cpm\_ParaManager, [46](#)
- BndCommS4DEx\_nowait
  - cpm\_ParaManager, [47](#)
- BndCommV3D
  - cpm\_ParaManager, [48](#)
- BndCommV3D\_nowait
  - cpm\_ParaManager, [49](#)
- BndCommV3DEx
  - cpm\_ParaManager, [50](#)
- BndCommV3DEx\_nowait
  - cpm\_ParaManager, [51](#)
- CPM\_BAND
  - cpm\_Define.h, [105](#)
- CPM\_BOR
  - cpm\_Define.h, [105](#)
- CPM\_BXOR
  - cpm\_Define.h, [105](#)
- CPM\_BYTE
  - cpm\_Define.h, [103](#)
- CPM\_CHAR
  - cpm\_Define.h, [102](#)
- CPM\_DOUBLE
  - cpm\_Define.h, [103](#)
- CPM\_Datatype
  - cpm\_Define.h, [102](#)
- CPM\_ERROR
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_ALREADY\_VOXELINIIT
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_BNDCOMM
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_BNDCOMM\_BUFFER
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_BNDCOMM\_VOXELSIZE
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_CREATE\_LOCALDOMAIN
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_CREATE\_NEIGHBOR
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_CREATE\_PROCGROUP
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_CREATE\_RANKMAP
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_BCID
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_DIVNUM
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_DIVPOS
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_GLOBALORIGIN
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_GLOBALREGION
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_GLOBALVOXELSIZE
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_HEADINDEX
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_INFO
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_LOCALORIGIN
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_LOCALREGION
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_LOCALVOXELSIZE
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_MYRANK
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_NEIGHBOR\_RANK
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_NUMRANK
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_PERIODIC\_RANK
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_PITCH
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_GET\_TAILINDEX
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_INSERT\_VOXELMAP
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_INVALID\_DOMAIN\_NO
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_INVALID\_OBJKEY
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_INVALID\_PTR
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_ALLGATHER
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_ALLGATHERV
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_ALLREDUCE
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_BARRIER
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_BCAST
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_GATHER
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_GATHERV
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_INVALID\_COMM
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_MPI\_INVALID\_DATATYPE

- cpm\_Define.h, [105](#)
- CPM\_ERROR\_MPI\_INVALID\_OPERATOR
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_MPI\_INVALID\_REQUEST
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_MPI\_IRecv
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_Isend
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_Recv
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_Send
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_WAIT
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_MPI\_WAITALL
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_NO\_MPI\_INIT
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_NO\_TEXTPARSER
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_NOT\_IN\_PROCGROUP
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_PERIODIC
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_PERIODIC\_INVALID\_DIR
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_PERIODIC\_INVALID\_PM
  - cpm\_Define.h, [105](#)
- CPM\_ERROR\_PM\_INSTANCE
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_REGIST\_OBJKEY
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TEXTPARSER
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_BCID
  - cpm\_Define.h, [104](#)
- CPM\_ERROR\_TP\_INVALID\_G\_DIV
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_G\_ORG
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_G\_PITCH
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_G\_RGN
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_G\_VOXEL
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_INVALID\_POS
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_NOVECTOR
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_TP\_VECTOR\_SIZE
  - cpm\_Define.h, [103](#)
- CPM\_ERROR\_VOXELINIT
  - cpm\_Define.h, [104](#)
- CPM\_EXTERN
  - cpm\_ParaManager\_frtIF.cpp, [119](#)
- CPM\_FLOAT
  - cpm\_Define.h, [103](#)
- CPM\_INLINE
  - cpm\_Base.h, [98](#)
- CPM\_INT
  - cpm\_Define.h, [103](#)
- CPM\_LAND
  - cpm\_Define.h, [105](#)
- CPM\_LONG
  - cpm\_Define.h, [103](#)
- CPM\_LONG\_DOUBLE
  - cpm\_Define.h, [103](#)
- CPM\_LOR
  - cpm\_Define.h, [105](#)
- CPM\_LXOR
  - cpm\_Define.h, [105](#)
- CPM\_MAX
  - cpm\_Define.h, [105](#)
- CPM\_MAXLOC
  - cpm\_Define.h, [105](#)
- CPM\_MIN
  - cpm\_Define.h, [105](#)
- CPM\_MINLOC
  - cpm\_Define.h, [105](#)
- CPM\_Op
  - cpm\_Define.h, [105](#)
- CPM\_PROD
  - cpm\_Define.h, [105](#)
- CPM\_REAL
  - cpm\_Define.h, [103](#)
- CPM\_REVISION
  - cpm\_Version.h, [149](#)
- CPM\_SHORT
  - cpm\_Define.h, [103](#)
- CPM\_SUCCESS
  - cpm\_Define.h, [103](#)
- CPM\_SUM
  - cpm\_Define.h, [105](#)
- CPM\_UNSIGNED
  - cpm\_Define.h, [103](#)
- CPM\_UNSIGNED\_CHAR
  - cpm\_Define.h, [102](#)
- CPM\_UNSIGNED\_LONG
  - cpm\_Define.h, [103](#)
- CPM\_UNSIGNED\_SHORT
  - cpm\_Define.h, [103](#)
- CPM\_VERSION\_NO
  - cpm\_Version.h, [149](#)
- clear
  - cpm\_ActiveSubDomainInfo, [8](#)
  - cpm\_DomainInfo, [17](#)
  - cpm\_GlobalDomainInfo, [20](#)
  - cpm\_LocalDomainInfo, [23](#)
- CopyArray
  - cpm\_ParaManager, [52](#)
- cpm\_Abort\_
  - cpm\_ParaManager\_frtIF.cpp, [117](#), [122](#)
- cpm\_ActiveSubDomainInfo, [7](#)
  - ~cpm\_ActiveSubDomainInfo, [8](#)



- clear, [8](#)
- cpm\_ActiveSubDomainInfo, [8](#)
- cpm\_ActiveSubDomainInfo, [8](#)
- GetBCID, [8](#)
- GetPos, [9](#)
- operator==, [9](#)
- Set, [9](#)
- SetBCID, [10](#)
- SetPos, [10](#)
- cpm\_Allgather\_
  - cpm\_ParaManager\_frtIF.cpp, [117](#), [122](#)
- cpm\_Allgatherv\_
  - cpm\_ParaManager\_frtIF.cpp, [117](#), [123](#)
- cpm\_Allreduce\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [123](#)
- cpm\_Barrier\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [124](#)
- cpm\_Base, [10](#)
  - ~cpm\_Base, [12](#)
  - cpm\_Base, [12](#)
  - cpm\_strCompare, [12](#)
  - cpm\_strCompareN, [12](#)
  - cpm\_Base, [12](#)
  - getCommNull, [12](#)
  - GetMemString, [13](#)
  - getRankNull, [13](#)
  - GetSpanTime, [13](#)
  - GetTime, [13](#)
  - GetWSpanTime, [14](#)
  - GetWTime, [14](#)
  - IsCommNull, [14](#)
  - IsRankNull, [14](#)
  - ReallIsDouble, [15](#)
  - VersionInfo, [15](#)
- cpm\_Base.h, [97](#)
  - CPM\_INLINE, [98](#)
- cpm\_Bcast\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [124](#)
- cpm\_BndCommS3D\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [124](#)
- cpm\_BndCommS3D\_nowait
  - cpm\_ParaManager, [52](#)
- cpm\_BndCommS3D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [125](#)
- cpm\_BndCommS4D\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [125](#)
- cpm\_BndCommS4D\_nowait
  - cpm\_ParaManager, [53](#)
- cpm\_BndCommS4D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [126](#)
- cpm\_BndCommS4DEx\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [126](#)
- cpm\_BndCommS4DEx\_nowait
  - cpm\_ParaManager, [53](#)
- cpm\_BndCommS4DEx\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [127](#)
- cpm\_BndCommV3D\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [127](#)
- cpm\_BndCommV3D\_nowait
  - cpm\_ParaManager, [54](#)
- cpm\_BndCommV3D\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, [118](#), [127](#)
- cpm\_BndCommV3DEx\_
  - cpm\_ParaManager\_frtIF.cpp, [119](#), [128](#)
- cpm\_BndCommV3DEx\_nowait
  - cpm\_ParaManager, [54](#)
- cpm\_BndCommV3DEx\_nowait\_
  - cpm\_ParaManager\_frtIF.cpp, [119](#), [128](#)
- cpm\_Define.h, [98](#)
  - \_IDX\_S3D, [100](#)
  - \_IDX\_S4D, [100](#)
  - \_IDX\_S4DEX, [101](#)
  - \_IDX\_V3D, [101](#)
  - \_IDX\_V3DEX, [102](#)
  - BOTH, [106](#)
  - CPM\_BAND, [105](#)
  - CPM\_BOR, [105](#)
  - CPM\_BXOR, [105](#)
  - CPM\_BYTE, [103](#)
  - CPM\_CHAR, [102](#)
  - CPM\_DOUBLE, [103](#)
  - CPM\_Datatype, [102](#)
  - CPM\_ERROR, [103](#)
  - CPM\_ERROR\_ALREADY\_VOXELINIIT, [104](#)
  - CPM\_ERROR\_BNDCOMM, [104](#)
  - CPM\_ERROR\_BNDCOMM\_BUFFER, [104](#)
  - CPM\_ERROR\_BNDCOMM\_BUFFERLENGTH, [105](#)
  - CPM\_ERROR\_BNDCOMM\_VOXELSIZE, [104](#)
  - CPM\_ERROR\_CREATE\_LOCALDOMAIN, [104](#)
  - CPM\_ERROR\_CREATE\_NEIGHBOR, [104](#)
  - CPM\_ERROR\_CREATE\_PROCGROUP, [104](#)
  - CPM\_ERROR\_CREATE\_RANKMAP, [104](#)
  - CPM\_ERROR\_GET\_BCID, [104](#)
  - CPM\_ERROR\_GET\_DIVNUM, [104](#)
  - CPM\_ERROR\_GET\_DIVPOS, [104](#)
  - CPM\_ERROR\_GET\_GLOBALORIGIN, [104](#)
  - CPM\_ERROR\_GET\_GLOBALREGION, [104](#)
  - CPM\_ERROR\_GET\_GLOBALVOXELSIZE, [104](#)
  - CPM\_ERROR\_GET\_HEADINDEX, [104](#)
  - CPM\_ERROR\_GET\_INFO, [104](#)
  - CPM\_ERROR\_GET\_LOCALORIGIN, [104](#)
  - CPM\_ERROR\_GET\_LOCALREGION, [104](#)
  - CPM\_ERROR\_GET\_LOCALVOXELSIZE, [104](#)
  - CPM\_ERROR\_GET\_MYRANK, [104](#)
  - CPM\_ERROR\_GET\_NEIGHBOR\_RANK, [104](#)
  - CPM\_ERROR\_GET\_NUMRANK, [104](#)
  - CPM\_ERROR\_GET\_PERIODIC\_RANK, [104](#)
  - CPM\_ERROR\_GET\_PITCH, [104](#)
  - CPM\_ERROR\_GET\_TAILINDEX, [104](#)
  - CPM\_ERROR\_INSERT\_VOXELMAP, [104](#)
  - CPM\_ERROR\_INVALID\_DOMAIN\_NO, [103](#)
  - CPM\_ERROR\_INVALID\_OBJKEY, [103](#)
  - CPM\_ERROR\_INVALID\_PTR, [103](#)
  - CPM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN, [104](#)

- CPM\_ERROR\_MPI, 104
- CPM\_ERROR\_MPI\_ALLGATHER, 104
- CPM\_ERROR\_MPI\_ALLGATHERV, 104
- CPM\_ERROR\_MPI\_ALLREDUCE, 104
- CPM\_ERROR\_MPI\_BARRIER, 104
- CPM\_ERROR\_MPI\_BCAST, 104
- CPM\_ERROR\_MPI\_GATHER, 104
- CPM\_ERROR\_MPI\_GATHERV, 104
- CPM\_ERROR\_MPI\_INVALID\_COMM, 105
- CPM\_ERROR\_MPI\_INVALID\_DATATYPE, 105
- CPM\_ERROR\_MPI\_INVALID\_OPERATOR, 105
- CPM\_ERROR\_MPI\_INVALID\_REQUEST, 105
- CPM\_ERROR\_MPI\_Irecv, 104
- CPM\_ERROR\_MPI\_Isend, 104
- CPM\_ERROR\_MPI\_RECV, 104
- CPM\_ERROR\_MPI\_SEND, 104
- CPM\_ERROR\_MPI\_WAIT, 104
- CPM\_ERROR\_MPI\_WAITALL, 104
- CPM\_ERROR\_NO\_MPI\_INIT, 104
- CPM\_ERROR\_NO\_TEXTPARSER, 103
- CPM\_ERROR\_NOT\_IN\_PROCGROUP, 104
- CPM\_ERROR\_PERIODIC, 105
- CPM\_ERROR\_PERIODIC\_INVALID\_DIR, 105
- CPM\_ERROR\_PERIODIC\_INVALID\_PM, 105
- CPM\_ERROR\_PM\_INSTANCE, 103
- CPM\_ERROR\_REGIST\_OBJKEY, 103
- CPM\_ERROR\_TEXTPARSER, 103
- CPM\_ERROR\_TP\_INVALID\_BCID, 104
- CPM\_ERROR\_TP\_INVALID\_G\_DIV, 103
- CPM\_ERROR\_TP\_INVALID\_G\_ORG, 103
- CPM\_ERROR\_TP\_INVALID\_G\_PITCH, 103
- CPM\_ERROR\_TP\_INVALID\_G\_RGN, 103
- CPM\_ERROR\_TP\_INVALID\_G\_VOXEL, 103
- CPM\_ERROR\_TP\_INVALID\_POS, 103
- CPM\_ERROR\_TP\_NOVECTOR, 103
- CPM\_ERROR\_TP\_VECTOR\_SIZE, 103
- CPM\_ERROR\_VOXELINIT, 104
- CPM\_FLOAT, 103
- CPM\_INT, 103
- CPM\_LAND, 105
- CPM\_LONG, 103
- CPM\_LONG\_DOUBLE, 103
- CPM\_LOR, 105
- CPM\_LXOR, 105
- CPM\_MAX, 105
- CPM\_MAXLOC, 105
- CPM\_MIN, 105
- CPM\_MINLOC, 105
- CPM\_Op, 105
- CPM\_PROD, 105
- CPM\_REAL, 103
- CPM\_SHORT, 103
- CPM\_SUCCESS, 103
- CPM\_SUM, 105
- CPM\_UNSIGNED, 103
- CPM\_UNSIGNED\_CHAR, 102
- CPM\_UNSIGNED\_LONG, 103
- CPM\_UNSIGNED\_SHORT, 103
- cpm\_DirFlag, 103
- cpm\_ErrorCode, 103
- cpm\_FaceFlag, 105
- cpm\_PMFlag, 105
- MINUS2PLUS, 106
- PLUS2MINUS, 106
- REAL\_BUF\_TYPE, 102
- REAL\_TYPE, 102
- X\_DIR, 103
- X\_MINUS, 105
- X\_PLUS, 105
- Y\_DIR, 103
- Y\_MINUS, 105
- Y\_PLUS, 105
- Z\_DIR, 103
- Z\_MINUS, 105
- Z\_PLUS, 105
- cpm\_DirFlag
  - cpm\_Define.h, 103
- cpm\_DomainInfo, 15
  - ~cpm\_DomainInfo, 17
  - clear, 17
  - cpm\_DomainInfo, 16
  - cpm\_DomainInfo, 16
  - GetOrigin, 17
  - GetPitch, 17
  - GetRegion, 17
  - GetVoxNum, 17
  - SetOrigin, 18
  - SetPitch, 18
  - SetRegion, 18
  - SetVoxNum, 18
- cpm\_DomainInfo.cpp, 106
- cpm\_DomainInfo.h, 107
- cpm\_ErrorCode
  - cpm\_Define.h, 103
- cpm\_FaceFlag
  - cpm\_Define.h, 105
- cpm\_Gather\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 129
- cpm\_Gatherv\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 129
- cpm\_GetBCID\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 130
- cpm\_GetDivNum\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 130
- cpm\_GetDivPos\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 130
- cpm\_GetGlobalOrigin\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 131
- cpm\_GetGlobalRegion\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 131
- cpm\_GetGlobalVoxelSize\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 131
- cpm\_GetLocalOrigin\_
  - cpm\_ParaManager\_frtIF.cpp, 119, 131
- cpm\_GetLocalRegion\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 132

- cpm\_GetLocalVoxelSize\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 132
- cpm\_GetMyRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 132
- cpm\_GetNeighborRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 133
- cpm\_GetNumRank\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 133
- cpm\_GetPeriodicRankID\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 133
- cpm\_GetPitch\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 133
- cpm\_GetVoxelHeadIndex\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 134
- cpm\_GetVoxelTailIndex\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 134
- cpm\_GlobalDomainInfo, 19
  - ~cpm\_GlobalDomainInfo, 20
  - AddSubDomain, 20
  - clear, 20
  - cpm\_GlobalDomainInfo, 20
  - cpm\_GlobalDomainInfo, 20
  - GetDivNum, 20
  - GetSubDomainInfo, 21
  - GetSubDomainNum, 21
  - IsExistSubDomain, 21
  - SetDivNum, 21
- cpm\_Initialize\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 134
- cpm\_Irecv
  - cpm\_ParaManager, 55
- cpm\_Irecv\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 135
- cpm\_IsParallel\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 135
- cpm\_Isend
  - cpm\_ParaManager, 55
- cpm\_Isend\_
  - cpm\_ParaManager\_frtIF.cpp, 120, 135
- cpm\_LocalDomainInfo, 22
  - ~cpm\_LocalDomainInfo, 23
  - clear, 23
  - cpm\_LocalDomainInfo, 23
  - cpm\_LocalDomainInfo, 23
- cpm\_ObjList
  - ~cpm\_ObjList, 24
  - Add, 25
  - cpm\_ObjList, 24
  - cpm\_ObjList, 24
  - Create, 25
  - Delete, 25
  - Get, 26
- cpm\_ObjList< T >, 23
- cpm\_ObjList.h, 108
  - RankNoMap, 109
- cpm\_PMFlag
  - cpm\_Define.h, 105
- cpm\_ParaManager, 26
  - Abort, 31
  - Allgather, 31, 32
  - Allgatherv, 32, 33
  - AllocDoubleS3D, 33
  - AllocDoubleS4D, 33
  - AllocDoubleS4DEx, 34
  - AllocDoubleV3D, 34
  - AllocDoubleV3DEx, 34
  - AllocFloatS3D, 35
  - AllocFloatS4D, 35
  - AllocFloatS4DEx, 35
  - AllocFloatV3D, 36
  - AllocFloatV3DEx, 36
  - AllocIntS3D, 36
  - AllocIntS4D, 36
  - AllocIntS4DEx, 37
  - AllocIntV3D, 37
  - AllocIntV3DEx, 37
  - AllocRealS3D, 38
  - AllocRealS4D, 38
  - AllocRealS4DEx, 38
  - AllocRealV3D, 39
  - AllocRealV3DEx, 39
  - Allreduce, 39, 40
  - Barrier, 40
  - Bcast, 40, 41
  - BndCommS3D, 41, 42
  - BndCommS3D\_nowait, 42, 43
  - BndCommS4D, 43, 44
  - BndCommS4D\_nowait, 44, 45
  - BndCommS4DEx, 46
  - BndCommS4DEx\_nowait, 47
  - BndCommV3D, 48
  - BndCommV3D\_nowait, 49
  - BndCommV3DEx, 50
  - BndCommV3DEx\_nowait, 51
  - CopyArray, 52
  - cpm\_BndCommS3D\_nowait, 52
  - cpm\_BndCommS4D\_nowait, 53
  - cpm\_BndCommS4DEx\_nowait, 53
  - cpm\_BndCommV3D\_nowait, 54
  - cpm\_BndCommV3DEx\_nowait, 54
  - cpm\_Irecv, 55
  - cpm\_Isend, 55
  - cpm\_Wait, 56
  - cpm\_Waitall, 59
  - cpm\_wait\_BndCommS3D, 56
  - cpm\_wait\_BndCommS4D, 57
  - cpm\_wait\_BndCommS4DEx, 57
  - cpm\_wait\_BndCommV3D, 58
  - cpm\_wait\_BndCommV3DEx, 58
  - cpm\_VoxelInfo, 93
  - CreateProcessGroup, 59
  - FindVoxelInfo, 60
  - flush, 60
  - Gather, 60, 61
  - Gatherv, 61, 62
  - get\_instance, 62, 63

- GetBCID, 63
- GetBndCommBufferSize, 63
- GetDivNum, 63
- GetDivPos, 64
- GetGlobalOrigin, 64
- GetGlobalRegion, 64
- GetGlobalVoxelSize, 65
- GetLocalOrigin, 65
- GetLocalRegion, 65
- GetLocalVoxelSize, 66
- GetMPI\_Comm, 66
- GetMPI\_Datatype, 66, 67
- GetMPI\_Op, 67
- GetMyRankID, 67
- GetNeighborRankID, 68
- GetNumRank, 68
- GetPeriodicRankID, 68
- GetPitch, 69
- GetVoxelHeadIndex, 69
- GetVoxelTailIndex, 69
- InitArray, 69
- Initialize, 70
- Irecv, 70, 71
- IsParallel, 72
- Isend, 71
- PeriodicCommS3D, 72, 73
- PeriodicCommS4D, 73, 74
- PeriodicCommS4DEx, 74, 75
- PeriodicCommV3D, 76
- PeriodicCommV3DEx, 77
- Recv, 78
- Send, 78, 79
- SetBndCommBuffer, 79
- VoxelInit, 80, 81
- Wait, 81
- wait\_BndCommsS3D, 81, 82
- wait\_BndCommsS4D, 82, 83
- wait\_BndCommsS4DEx, 84
- wait\_BndCommV3D, 85
- wait\_BndCommV3DEx, 86
- Waitall, 87
- cpm\_ParaManager.cpp, 109
- cpm\_ParaManager.h, 110
  - BndCommInfoMap, 111
  - RankNoMap, 111
  - VoxelInfoMap, 111
- cpm\_ParaManager\_Alloc.cpp, 111
- cpm\_ParaManager\_BndComm.h, 112
  - \_IDXX, 112
  - \_IDXFY, 112
  - \_IDXFZ, 113
- cpm\_ParaManager\_BndCommEx.h, 113
  - \_IDXX, 114
  - \_IDXFY, 114
  - \_IDXFZ, 114
- cpm\_ParaManager\_MPI.cpp, 144
- cpm\_ParaManager\_frtrIF.cpp, 114
  - CPM\_EXTERN, 119
- cpm\_Abort\_, 117, 122
- cpm\_Allgather\_, 117, 122
- cpm\_Allgatherv\_, 117, 123
- cpm\_Allreduce\_, 118, 123
- cpm\_Barrier\_, 118, 124
- cpm\_Bcast\_, 118, 124
- cpm\_BndCommsS3D\_, 118, 124
- cpm\_BndCommsS3D\_nowait\_, 118, 125
- cpm\_BndCommsS4D\_, 118, 125
- cpm\_BndCommsS4D\_nowait\_, 118, 126
- cpm\_BndCommsS4DEx\_, 118, 126
- cpm\_BndCommsS4DEx\_nowait\_, 118, 127
- cpm\_BndCommV3D\_, 118, 127
- cpm\_BndCommV3D\_nowait\_, 118, 127
- cpm\_BndCommV3DEx\_, 119, 128
- cpm\_BndCommV3DEx\_nowait\_, 119, 128
- cpm\_Gather\_, 119, 129
- cpm\_Gatherv\_, 119, 129
- cpm\_GetBCID\_, 119, 130
- cpm\_GetDivNum\_, 119, 130
- cpm\_GetDivPos\_, 119, 130
- cpm\_GetGlobalOrigin\_, 119, 131
- cpm\_GetGlobalRegion\_, 119, 131
- cpm\_GetGlobalVoxelSize\_, 119, 131
- cpm\_GetLocalOrigin\_, 119, 131
- cpm\_GetLocalRegion\_, 120, 132
- cpm\_GetLocalVoxelSize\_, 120, 132
- cpm\_GetMyRankID\_, 120, 132
- cpm\_GetNeighborRankID\_, 120, 133
- cpm\_GetNumRank\_, 120, 133
- cpm\_GetPeriodicRankID\_, 120, 133
- cpm\_GetPitch\_, 120, 133
- cpm\_GetVoxelHeadIndex\_, 120, 134
- cpm\_GetVoxelTailIndex\_, 120, 134
- cpm\_Initialize\_, 120, 134
- cpm\_Irecv\_, 120, 135
- cpm\_IsParallel\_, 121, 135
- cpm\_Isend\_, 120, 135
- cpm\_PeriodicCommS3D, 121
- cpm\_PeriodicCommS3D\_, 136
- cpm\_PeriodicCommS4D, 121
- cpm\_PeriodicCommS4D\_, 136
- cpm\_PeriodicCommS4DEx, 121
- cpm\_PeriodicCommS4DEx\_, 137
- cpm\_PeriodicCommV3D, 121
- cpm\_PeriodicCommV3D\_, 137
- cpm\_PeriodicCommV3DEx, 121
- cpm\_PeriodicCommV3DEx\_, 138
- cpm\_Recv\_, 121, 138
- cpm\_Send\_, 121, 139
- cpm\_SetBndCommBuffer\_, 121, 139
- cpm\_VoxelInit\_, 121, 139
- cpm\_VoxelInit\_nodiv\_, 121, 140
- cpm\_Wait\_, 121, 140
- cpm\_Waitall\_, 122, 143
- cpm\_wait\_BndCommsS3D\_, 122, 141
- cpm\_wait\_BndCommsS4D\_, 122, 141
- cpm\_wait\_BndCommsS4DEx\_, 122, 141

- cpm\_wait\_BndCommV3D\_, 122, 142
  - cpm\_wait\_BndCommV3DEx\_, 122, 142
- cpm\_ParaManager\_inline.h, 143
- cpm\_PeriodicCommS3D
  - cpm\_ParaManager\_frtIF.cpp, 121
- cpm\_PeriodicCommS3D\_
  - cpm\_ParaManager\_frtIF.cpp, 136
- cpm\_PeriodicCommS4D
  - cpm\_ParaManager\_frtIF.cpp, 121
- cpm\_PeriodicCommS4D\_
  - cpm\_ParaManager\_frtIF.cpp, 136
- cpm\_PeriodicCommS4DEx
  - cpm\_ParaManager\_frtIF.cpp, 121
- cpm\_PeriodicCommS4DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 137
- cpm\_PeriodicCommV3D
  - cpm\_ParaManager\_frtIF.cpp, 121
- cpm\_PeriodicCommV3D\_
  - cpm\_ParaManager\_frtIF.cpp, 137
- cpm\_PeriodicCommV3DEx
  - cpm\_ParaManager\_frtIF.cpp, 121
- cpm\_PeriodicCommV3DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 138
- cpm\_Recv\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 138
- cpm\_Send\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 139
- cpm\_SetBndCommBuffer\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 139
- cpm\_TextParser, 87
  - ~cpm\_TextParser, 89
  - cpm\_TextParser, 89
  - cpm\_TextParser, 89
  - m\_tp, 90
  - Read, 89
  - readVector, 89, 90
- cpm\_TextParser.cpp, 145
- cpm\_TextParser.h, 145
- cpm\_TextParserDomain, 91
  - ~cpm\_TextParserDomain, 92
  - cpm\_TextParserDomain, 92
  - cpm\_TextParserDomain, 92
  - Read, 92
- cpm\_TextParserDomain.cpp, 147
- cpm\_TextParserDomain.h, 147
- cpm\_Version.h, 149
  - CPM\_REVISION, 149
  - CPM\_VERSION\_NO, 149
- cpm\_VoxelInfo, 92
  - cpm\_ParaManager, 93
- cpm\_VoxelInfo.cpp, 150
- cpm\_VoxelInfo.h, 150
- cpm\_VoxelInit\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 139
- cpm\_VoxelInit\_nodiv\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 140
- cpm\_Wait
  - cpm\_ParaManager, 56
- cpm\_Wait\_
  - cpm\_ParaManager\_frtIF.cpp, 121, 140
- cpm\_Waitall
  - cpm\_ParaManager, 59
- cpm\_Waitall\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 143
- cpm\_strCompare
  - cpm\_Base, 12
- cpm\_strCompareN
  - cpm\_Base, 12
- cpm\_wait\_BndCommS3D
  - cpm\_ParaManager, 56
- cpm\_wait\_BndCommS3D\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 141
- cpm\_wait\_BndCommS4D
  - cpm\_ParaManager, 57
- cpm\_wait\_BndCommS4D\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 141
- cpm\_wait\_BndCommS4DEx
  - cpm\_ParaManager, 57
- cpm\_wait\_BndCommS4DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 141
- cpm\_wait\_BndCommV3D
  - cpm\_ParaManager, 58
- cpm\_wait\_BndCommV3D\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 142
- cpm\_wait\_BndCommV3DEx
  - cpm\_ParaManager, 58
- cpm\_wait\_BndCommV3DEx\_
  - cpm\_ParaManager\_frtIF.cpp, 122, 142
- Create
  - cpm\_ObjList, 25
- CreateProcessGroup
  - cpm\_ParaManager, 59
- Delete
  - cpm\_ObjList, 25
- FindVoxelInfo
  - cpm\_ParaManager, 60
- flush
  - cpm\_ParaManager, 60
- Gather
  - cpm\_ParaManager, 60, 61
- Gatherv
  - cpm\_ParaManager, 61, 62
- Get
  - cpm\_ObjList, 26
- get\_instance
  - cpm\_ParaManager, 62, 63
- GetBCID
  - cpm\_ActiveSubDomainInfo, 8
  - cpm\_ParaManager, 63
- GetBndCommBufferSize
  - cpm\_ParaManager, 63
- getCommNull
  - cpm\_Base, 12
- GetDivNum

- cpm\_GlobalDomainInfo, 20
- cpm\_ParaManager, 63
- GetDivPos
  - cpm\_ParaManager, 64
- GetGlobalOrigin
  - cpm\_ParaManager, 64
- GetGlobalRegion
  - cpm\_ParaManager, 64
- GetGlobalVoxelSize
  - cpm\_ParaManager, 65
- GetLocalOrigin
  - cpm\_ParaManager, 65
- GetLocalRegion
  - cpm\_ParaManager, 65
- GetLocalVoxelSize
  - cpm\_ParaManager, 66
- GetMPI\_Comm
  - cpm\_ParaManager, 66
- GetMPI\_Datatype
  - cpm\_ParaManager, 66, 67
- GetMPI\_Op
  - cpm\_ParaManager, 67
- GetMemString
  - cpm\_Base, 13
- GetMyRankID
  - cpm\_ParaManager, 67
- GetNeighborRankID
  - cpm\_ParaManager, 68
- GetNumRank
  - cpm\_ParaManager, 68
- GetOrigin
  - cpm\_DomainInfo, 17
- GetPeriodicRankID
  - cpm\_ParaManager, 68
- GetPitch
  - cpm\_DomainInfo, 17
  - cpm\_ParaManager, 69
- GetPos
  - cpm\_ActiveSubDomainInfo, 9
- getRankNull
  - cpm\_Base, 13
- GetRegion
  - cpm\_DomainInfo, 17
- GetSpanTime
  - cpm\_Base, 13
- GetSubDomainInfo
  - cpm\_GlobalDomainInfo, 21
- GetSubDomainNum
  - cpm\_GlobalDomainInfo, 21
- GetTime
  - cpm\_Base, 13
- GetVoxNum
  - cpm\_DomainInfo, 17
- GetVoxelHeadIndex
  - cpm\_ParaManager, 69
- GetVoxelTailIndex
  - cpm\_ParaManager, 69
- GetWSpanTime
  - cpm\_Base, 14
- GetWTime
  - cpm\_Base, 14
- InitArray
  - cpm\_ParaManager, 69
- Initialize
  - cpm\_ParaManager, 70
- Irecv
  - cpm\_ParaManager, 70, 71
- IsCommNull
  - cpm\_Base, 14
- IsExistSubDomain
  - cpm\_GlobalDomainInfo, 21
- IsParallel
  - cpm\_ParaManager, 72
- IsRankNull
  - cpm\_Base, 14
- Isend
  - cpm\_ParaManager, 71
- m\_bufX
  - S\_BNDCOMM\_BUFFER, 94
- m\_bufY
  - S\_BNDCOMM\_BUFFER, 95
- m\_bufZ
  - S\_BNDCOMM\_BUFFER, 95
- m\_maxN
  - S\_BNDCOMM\_BUFFER, 95
- m\_maxVC
  - S\_BNDCOMM\_BUFFER, 95
- m\_nwX
  - S\_BNDCOMM\_BUFFER, 95
- m\_nwY
  - S\_BNDCOMM\_BUFFER, 95
- m\_nwZ
  - S\_BNDCOMM\_BUFFER, 96
- m\_tp
  - cpm\_TextParser, 90
- MINUS2PLUS
  - cpm\_Define.h, 106
- operator==
  - cpm\_ActiveSubDomainInfo, 9
- PLUS2MINUS
  - cpm\_Define.h, 106
- PeriodicCommS3D
  - cpm\_ParaManager, 72, 73
- PeriodicCommS4D
  - cpm\_ParaManager, 73, 74
- PeriodicCommS4DEx
  - cpm\_ParaManager, 74, 75
- PeriodicCommV3D
  - cpm\_ParaManager, 76
- PeriodicCommV3DEx
  - cpm\_ParaManager, 77
- REAL\_BUF\_TYPE

- cpm\_Define.h, 102
- REAL\_TYPE
  - cpm\_Define.h, 102
- RankNoMap
  - cpm\_ObjList.h, 109
  - cpm\_ParaManager.h, 111
- Read
  - cpm\_TextParser, 89
  - cpm\_TextParserDomain, 92
- readVector
  - cpm\_TextParser, 89, 90
- ReallsDouble
  - cpm\_Base, 15
- Recv
  - cpm\_ParaManager, 78
- S\_BNDCOMM\_BUFFER, 93
  - ~S\_BNDCOMM\_BUFFER, 94
  - m\_bufX, 94
  - m\_bufY, 95
  - m\_bufZ, 95
  - m\_maxN, 95
  - m\_maxVC, 95
  - m\_nwX, 95
  - m\_nwY, 95
  - m\_nwZ, 96
  - S\_BNDCOMM\_BUFFER, 94
  - S\_BNDCOMM\_BUFFER, 94
- Send
  - cpm\_ParaManager, 78, 79
- Set
  - cpm\_ActiveSubDomainInfo, 9
- SetBCID
  - cpm\_ActiveSubDomainInfo, 10
- SetBndCommBuffer
  - cpm\_ParaManager, 79
- SetDivNum
  - cpm\_GlobalDomainInfo, 21
- SetOrigin
  - cpm\_DomainInfo, 18
- SetPitch
  - cpm\_DomainInfo, 18
- SetPos
  - cpm\_ActiveSubDomainInfo, 10
- SetRegion
  - cpm\_DomainInfo, 18
- SetVoxNum
  - cpm\_DomainInfo, 18
- VersionInfo
  - cpm\_Base, 15
- VoxelInfoMap
  - cpm\_ParaManager.h, 111
- VoxelInit
  - cpm\_ParaManager, 80, 81
- Wait
  - cpm\_ParaManager, 81
- wait\_BndCommS3D
  - cpm\_ParaManager, 81, 82
- wait\_BndCommS4D
  - cpm\_ParaManager, 82, 83
- wait\_BndCommS4DEx
  - cpm\_ParaManager, 84
- wait\_BndCommV3D
  - cpm\_ParaManager, 85
- wait\_BndCommV3DEx
  - cpm\_ParaManager, 86
- Waitall
  - cpm\_ParaManager, 87
- X\_DIR
  - cpm\_Define.h, 103
- X\_MINUS
  - cpm\_Define.h, 105
- X\_PLUS
  - cpm\_Define.h, 105
- Y\_DIR
  - cpm\_Define.h, 103
- Y\_MINUS
  - cpm\_Define.h, 105
- Y\_PLUS
  - cpm\_Define.h, 105
- Z\_DIR
  - cpm\_Define.h, 103
- Z\_MINUS
  - cpm\_Define.h, 105
- Z\_PLUS
  - cpm\_Define.h, 105